T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Doktora Tezi

SAHTE İNTERNET SİTELERİNİN URL ÖZELLİKLERİ TEMELİNDE TESPİT EDİLMESİ AMACIYLA ÖZELLİK SEÇME METOTLARININ ve ÖĞRENME ALGORİTMALARININ ANALİZİ

Mustafa AYDIN

Enformatik Anabilim Dalı

Enformatik Programı

DANIŞMAN
Prof. Dr. Sevinç GÜLSEÇEN

II. DANIŞMAN
Prof. Dr. Kutluk Kağan SÜMER

Haziran, 2022

İSTANBUL

Bu çalışma, 1.06.2022 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Sevinç GÜLSEÇEN(Danışman)
İstanbul Üniversitesi
Fen Fakültesi

Prof. Dr. Ünal Halit ÖZDEN
İstanbul Ticaret Üniversitesi
İnsan ve Toplum Bilimleri Fakültesi

Doç. Dr. Çiğdem EROL
İstanbul Üniversitesi
Fen Fakültesi

Doç. Dr. Aycan HEPSAĞ
İstanbul Üniversitesi
İktisat Fakültesi

Dr. Öğr. Üyesi Funda Hatice SEZGİN
İstanbul Üniversitesi Cerrahpaşa
Mühendislik Fakültesi

İntihal Programı Beyanı

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Hayatımın her aşamasında koşulsuz yanımda olan canım aileme doktora çalışmalarımda da vermiş olduğu destek ve çalışma boyunca göstermiş olduğu sabır için sonsuz teşekkür ediyor bu tezi başta rahmetli babam Mehmet AYDIN olmak üzere tüm aileme ithaf ediyorum.

Bu uzun yolda hiç bıkmadan yanımda olan ve her ihtiyaç duyduğumda yardımıma koşan çok değerli kardeşlerim Murat YÖNAÇ ve Muhammet ERİŞEN'e, doktora çalışmalarım boyunca bana desteğini esirgemeyen çok kıymetli arkadaşlarım Arif YILMAZ ve İsmail BÜTÜN'e, tez çalışmasına katkı sağlayan yazılım geliştirme fikirleriyle bana destek olan çok sevgili arkadaşlarım Alper TAŞ ve Abdullah Emir ÇİL'e ve bu yolda bir şekilde yanımda olmuş tüm dostlarıma teşekkür ediyorum.

Haziran 2022 Mustafa AYDIN

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v**
ŞEKİL LİSTESİ ..... **vii**
TABLO LİSTESİ ..... **viii**
SİMGE VE KISALTMA LİSTESİ ..... **ix**
ÖZET ..... **x**
SUMMARY ..... **xii**
1. GİRİŞ **1**
1.1 ÇALIŞMANIN AMACI VE KAPSAMI 5
1.2 ÇALIŞMANIN SINIRLILIKLARI 6
2. GENEL KISIMLAR **8**
2.1 MAKİNE ÖĞRENMESİ 8
2.1.1 Naive Bayes (NB) 9
2.1.2 J48 Karar Ağacı (DT) 10
2.1.3 Sınıflandırma ve Regresyon Ağacı (CART) 13
2.1.4 Rastgele Orman (RF) 14
2.1.5 Sıralı Minimal Optimizasyon (SMO) 16
2.1.6 Sinir Ağları (NNet) 17
2.1.7 Derin Öğrenme (DL) 19
2.2 ÖZELLİK SEÇME METOTLARI 22
2.3 PERFORMANS METRİKLERİ 23
2.4 LİTERATÜR TARAMASI 26
3. MALZEME VE YÖNTEM **41**
3.1 VERİ SETİ 41
3.1.1 Yurtdışı Kaynaklı Veri Seti 41
3.1.2 Yurtiçi Kaynaklı Veri Seti 43
3.2 VERİ ANALİZİ 44
3.2.1 URL Gelişmiş Özellikleri 45
3.2.2 Özellik Çıkartma 45
3.2.3 Özellik Seçimi 61

3.3 YÖNTEMLER....62
3.3.1 Özellik Seçimli Çoklu Makine Öğrenmesi Modeli....63
3.3.2 Özellik Seçimli Derin Öğrenme Modeli....64
4. BULGULAR....66
5. TARTIŞMA VE SONUÇ....74
KAYNAKLAR....79
EKLER....87
ÖZGEÇMİŞ....101

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1: Farklı Endüstrilerde Oltalama Saldırılarına Eğilim Kıyaslaması..... 1
Şekil 1.2: Yaş Gruplarının Oltalama Terimini Doğru Bilme Oranları..... 2
Şekil 1.3: 2017-2020 Yılları Dünya Genelinde Gerçekleşen Oltalama Vaka Sayıları..... 3
Şekil 1.4: 2018 - 2021 Yılları Oltalama Vaka Tutarlarının Oranı..... 4
Şekil 2.1: Karar Ağacı Algoritmasının Genel Yapısı..... 11
Şekil 2.2: Yapay Sinir Ağının Temel Yapısı..... 18
Şekil 2.3: Bir Nöronun İçyapısı..... 18
Şekil 2.4: Çapraz Doğrulama..... 25
Şekil 3.1: En Çok Hedeflenen Markalar İçin Veri Toplama Şeması..... 43
Şekil 3.2: Özellik Çıkarma Amaçlı Kullanılan Yazılım Kod Örneği..... 44
Şekil 3.3: URL Özellik Çıkartma Şeması..... 46
Şekil 3.4: Özelliklere İlişkin Veri İşleme Şeması..... 46
Şekil 3.5: Kelime Modülünün İçyapısı ve Bölümleri..... 51
Şekil 3.6: Veri Önişleme Adımları..... 60
Şekil 3.7: Özellik Seçme Metotları Kullanım Mimarisi..... 62
Şekil 3.8: Özellik Seçimli Çoklu Makine Öğrenmesi Modeli Sınıflandırma Şeması..... 64
Şekil 3.9: Özellik Seçimli Derin Öğrenme Örnek Model Mimarisi..... 65
Şekil 4.1: İlk 5 Sınıflandırma Algoritmasının Karşılaştırmalı Grafiği..... 69
Şekil 4.2: İlk 4 Sınıflandırma Algoritmasının Karşılaştırmalı Grafiği..... 70
Şekil 4.3: a) Mimari 1 (b) Mimari 2 (c) Mimari 3 (d) Mimari 4 (e) Mimari 5 (f) Mimari 6..... 71

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Karışıklık Matrisi 24

Tablo 3.1: Oltalama Saldırılarında En Fazla Hedefte Olan Marka İsimleri 41

Tablo 3.2: Yurtiçi Veri Setinden Elde Edilen Göz Alıcı Kelimelerin Frekansı 56

Tablo 3.3: Eşik Değerinin Üstünde Kalan Göz Alıcı Kelimeler 58

Tablo 3.4: Veri Setinin Bölünmesi 60

Tablo 4.1: Özellik Seçme Metotları Bazında Seçilen Özellik Sayıları 66

Tablo 4.2: Seçilen Ortak Özellikler 66

Tablo 4.3: Sınıflandırma Algoritmalarına Dayalı Analiz Sonuçları 67

Tablo 4.4: Sınıflandırma Algoritması ve Özellik Seçme Metotu Uyumluluk Tablosu 67

Tablo 4.5: Doğruluk Metriğine Dayalı Analiz Sonuçları 68

Tablo 4.6: Derin Öğrenme Modelinin Yurtdışı Veri Seti Test Sonuçları 70

Tablo 4.7: Derin Öğrenme Modelinin Yurtiçi Veri Seti Test Sonuçları 72

Tablo 4.8: Derin Öğrenme Modelinin Yurtiçi Veri Seti Test Sonuçları (Dengeli Veri) 73

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
ADAM	: Adaptif Momentum Tahmini *(Adaptive Moment Estimation)*
APWG	: Oltalama Saldırısı Önleme Çalışma Grubu *(Anti-Phishing Working Group)*
BDDK	: Bankacılık Düzenleme ve Denetleme Kurumu
CART	: Sınıflandırma ve Regresyon Ağacı *(Classification and Regression Trees)*
CNN	: Konvolüsyonel/Evrişimsel Sinir Ağları *(Convolutional Neural Network)*
DNN	: Derin Sinir Ağları *(Deep Neural Network)*
DNS:	: Alan Adı Sistemi *(Domain Name System)*
HTML	: Hiper Metin İşaretleme Dili *(Hyper Text Markup Language)*
HTTP	: Bağlantılı Metin Aktarım Protokolü *(Hyper Text Transfer Protocol)*
HTTPS	: Güvenli Bağlantılı Metin Aktarım Protokolü *(HTTP Secure)*
IP	: İnternet Protokolü *(Internet Protocol)*
MLP	: Çok Katmanlı Algılayıcı *(Multi Layer Perceptron)*
NLP	: Doğal Dil İşleme *(Natural Language Processing)*
SMOTE	:Sentetik Azınlık Aşırı Örnekleme Tekniği *(Synthetic Minority Oversampling Technique)*
SOME	: Siber Olaylara Müdahale Ekibi
SSL	: Güvenli Soket Katmanı *(Secure Socket Layer)*
SVM	: Destek Vektör Makineleri *(Support Vector Machine)*
URL	: Tekdüzen Kaynak Bulucu *(Uniform Resource Locator)*
USOM	: Ulusal Siber Olaylara Müdahale Merkezi
TLD	: Üst Seviye Alan Adı *(Top Level Domain)*
TLS	: Taşıma Katmanı Güvenliği *(Transport Layer Security)*
www	: Dünya Çapında Ağ *(World Wide Web)*

ÖZET

DOKTORA TEZİ

SAHTE İNTERNET SİTELERİNİN URL ÖZELLİKLERİ TEMELİNDE TESPİT EDİLMESİ AMACIYLA ÖZELLİK SEÇME METOTLARININ ve ÖĞRENME ALGORİTMALARININ ANALİZİ

Mustafa AYDIN

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman : Prof. Dr. Sevinç GÜLSEÇEN

II. Danışman : Prof. Dr. Kutluk Kağan SÜMER

Oltalama saldırıları, kimlik avcılarının sahte bir internet sitesini yasal bir site gibi göstererek internet kullanıcılarını inandırdığı saldırılardır. Özellikle finansal hassas bilgileri çalmak için kullanılan oltalama saldırıları kullanıcılar için kritik bir tehdit oluşturmakta ve oltalama saldırılarından kaynaklanan kayıplar artmaya devam etmektedir. Yapılan çalışmalar ve elde edilen istatistikler genel olarak değerlendirildiğinde oltalama saldırıları gerek küresel çapta gerek Türkiye çapında mücadele edilmesi gereken kritik siber güvenlik konularından birisi olmaya devam etmektedir.

Oltalama sitelerinin engellenmesine yönelik çalışmalara başlamadan önce tespit başarısını arttırmak amacıyla bu sitelerin belirgin ve ortak özellikleri tespit edilmelidir. Bu çalışmada oltalama sitelerinin en belirgin tespit edilebilir özelliklerinden birisi olan URL içeriği üzerinde durulmuştur. Bu amaç doğrultusunda literatürde kabul edilen performans metriklerine bağlı olarak yüksek başarı oranına sahip bir sınıflandırıcı iş akışı modeli önerisi hedeflenmiştir.

Oltalama amaçlı kullanılan URL adreslerinin tespiti için bu çalışmada 2 farklı model kullanılmıştır. İlk modelde oltalama saldırısı tespiti amaçlı oluşturulan veri kümesi üzerinde bazı özellik seçme yöntemlerinin ve sınıflandırma algoritmalarının performansı analiz edilmiştir. Araştırmadaki temel amaç, farklı sınıflandırma algoritmalarının ve farklı özellik seçme yöntemlerinin birbirleriyle olan en iyi uyumluluğunu bularak sahte internet sitesi tespit

doğruluğunu maksimize etmektir. Bu çalışmada, Korelasyon alt küme tabanlı, Tutarlılık alt küme tabanlı, Kazanç Oranı nitelik tabanlı ve Relief-F nitelik tabanlı özellik seçme yöntemleri ve Naïve Bayes, SMO (Sıralı Minimal Optimizasyon), CART (Sınıflandırma ve Regresyon Ağacı), J48 (Karar Ağacı) ve Rastgele Orman olmak üzere beş tür sınıflandırma algoritması üzerine çalışılmıştır. Bu algoritmalar WEKA yazılımı kullanılarak incelenmiştir. Rastgele Orman algoritması, tüm özellik seçme yöntemlerinde en iyi performansı göstermiştir. İlave olarak, J48 algoritması ikinci, CART algoritması ise üçüncü en iyi sınıflandırma algoritması olarak öne çıkmıştır.

Çalışmanın diğer modelinde oltalama sitelerinin tespiti için derin öğrenme modeli olarak ileri beslemeli derin sinir ağlarının kullanımı tercih edilmiştir. İleri beslemeli derin sinir ağları temelde çok katmanlı algılayıcıların altyapısına dayanmaktadır. Bu modelin başarısına katman ve düğümlerin etkisini araştırmak için 6 adet farklı deneysel mimari hazırlanmıştır. Toplamda bu 6 adet farklı mimariye sahip derin öğrenme modelleri oltalama veri setiyle eğitilmiş ve en optimum çözüm tespit edilmiştir.

Bu çalışmada derin öğrenme kullanılarak hızlı bir algılama yöntemine dayalı çok boyutlu bir oltalama tespiti yaklaşımı önerilmektedir. Oltalama URL'si ve meşru URL içeren bir veri kümesi üzerinde yapılan testler sonucunda, doğruluk parametresi için %99,46 oranı elde edilmiştir. Literatürde yer alan çalışmalar göz önünde bulundurulduğunda oltalama URL adreslerinin tespiti için derin öğrenme modeli kullanımının doğru bir yaklaşım olduğu kanıtlanmaktadır.

Haziran 2022, 115 sayfa.

Anahtar kelimeler: Oltalama Saldırıları, Makine Öğrenmesi, Derin Öğrenme Yöntemi

SUMMARY

Ph.D. THESIS

ANALYSIS of FEATURE SELECTION METHODS and LEARNING ALGORITHMS for PHISHING WEBSITES DETECTION BASED on URL

Mustafa AYDIN

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Prof. Dr. Sevinç GÜLSEÇEN

Co-Supervisor : Prof. Dr. Kutluk Kağan SÜMER

Phishing attacks are attacks in which phishers deceive internet users by making a fake website look like a legitimate one. Phishing attacks are especially used to capture financially sensitive information, hence pose a critical threat to users and the losses from phishing attacks continue to increase. When the studies and the gathered statistics are evaluated in general, phishing attacks continue to be one of the critical cyber security issues that need to be tackled both globally and throughout Turkey.

Before starting the studies on the blocking of phishing sites, the distinctive and common features of these sites should be determined in order to increase success rate of the detection. In this study, URL content, which is one of the most distinctive detectable features of phishing sites, is emphasized. For this purpose, it is aimed to propose a classifier workflow model with a high success rate, depending on the performance metrics accepted in the literature.

Two different models were used in this study to detect URL addresses used for phishing purposes. In the first model of the study, the performance of some special feature selection and classification algorithms on the dataset created for the detection of the phishing attack websites was analyzed. The main purpose of the research is to maximize the fake website detection

accuracy by finding the best compatibility between different classification algorithms and different feature selection methods. In this study, four types of feature selection methods and five types of classification algorithms were studied, namely CFS (Correlation-based Feature Selection) subset based, Consistency subset based, Gain Ratio attribute based, Relief-F attribute based feature selection methods and Naïve Bayes, SMO (Sequential Minimal Optimization), CART (Classification and Regression Tree), J48 (Decision Tree) and Random Forest classification algorithms. These algorithms were analyzed using WEKA software. The Random Forest algorithm showed the best performance in all feature selection methods. In addition, the J48 algorithm stood out as the second-best classification algorithm and the CART algorithm as the third-best classification algorithm.

In the other model of the study, the use of feedforward deep neural networks was preferred as a deep learning model for detecting phishing sites. Feedforward deep neural networks are basically based on the infrastructure of multilayer neurons. In order to investigate the effect of layers and nodes on the success of this model, 6 different experimental architectures were prepared. In total, these 6 deep learning models with different architectures were trained with the phishing dataset and the most optimum solution was determined.

In this study, a multidimensional phishing detection approach based on a rapid detection method using deep learning is proposed. As a result of tests on a dataset containing phishing and legitimate URLs, an accuracy rate of 99.46% was obtained. Considering the studies in the literature, it is proven that the use of a deep learning model is an appropriate approach to detect phishing URL addresses.

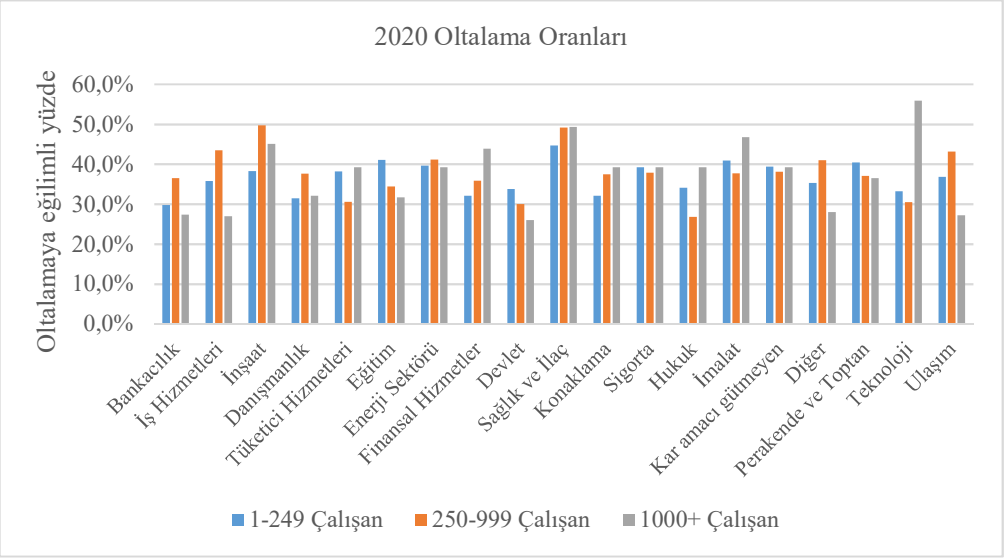
June 2022, 115 pages.

Keywords: Phishing Websites, Machine Learning, Deep Learning

1. GİRİŞ

Geçtiğimiz yıllar içinde, insanlardan ve şirketlerden hassas ve değerli bilgiler elde etmek amacıyla, gerçek web sitelerini taklit etmek için internette pek çok sahte site oluşturulmuştur. Çevrimiçi olarak gerçekleştirilen bu tür saldırılar, oltalama olarak bilinmekte ve tüketiciler, işletmeler ve birçok şirket ortağının maddi varlığı için ciddi tehditler taşımaya devam etmektedir. Aynı zamanda birçok siber saldırı da oltalama yoluyla başarılı olmaktadır. Araştırmacılar tarafından bilinen ve kabul edilmiş APWG (Anti-Phishing Working Group) ve PhishTank organizasyonları tarafından sunulan istatistikler bugüne kadar oltalama sitelerinin sayısının sürekli artma eğiliminde olduğunu göstermektedir (Faris, 2021).

Phishing By Industry Benchmarking Raporunda (2020) yer verilen çalışmada farklı sektör çalışanlarının oltalama saldırılarına eğilimli olma riskini tespit etmek için 19 endüstri, 17.000 şirket, 4 milyon kullanıcı ve 9,5 milyon oltalama test saldırısı ile deney ortamı kurulmuştur. Şekil 1.1'de gösterilen grafikte oltalamalara karşı eğilimli çalışanların yüzdeleri, her bir endüstri içindeki çalışan sayılarına göre 3 segmentte ölçeklendirilmiştir.

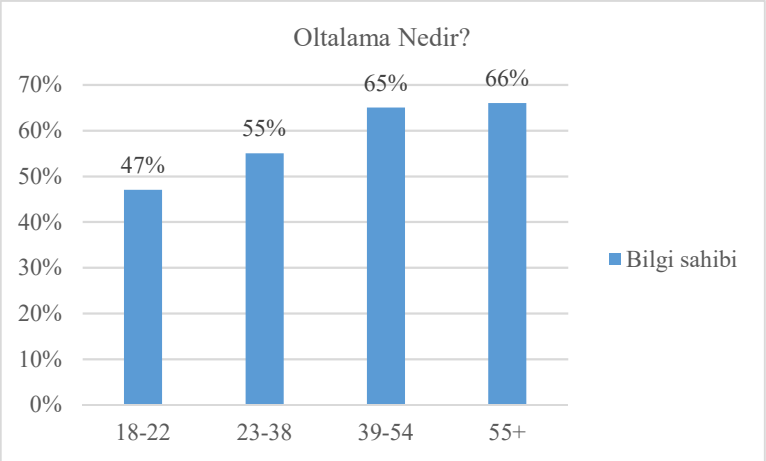


Şekil 1.1: Farklı Endüstrilerde Oltalama Saldırılarına Eğilim Kıyaslaması (Phishing By Industry Benchmarking Report, 2020)

Tüm endüstriler ve kuruluşlar genelinde oltalamaya eğilim yüzdelerine ait ortalamanın %37,9 olması sektörler arasında farklılık gösterse de, kullanıcıların bir kuruluşun oltalama saldırılarına karşı son savunma hattı olarak başarısız olduğu gerçeğini göstermektedir. Bu ortalama değer her 3 çalışandan 1'inin şüpheli bir bağlantıya veya e-postaya tıklamasının veya dolandırıcılığa yol açan talimatları yerine getirmesinin muhtemel olduğu anlamına gelmektedir.

Şekil 1.1'deki grafik incelendiğinde 1000 üzeri çalışan sayısı olan kuruluşların oltalama saldırılarına diğer ölçekteki kuruluşlardan daha fazla maruz kalabileceği gözlemlenmektedir. Bu durumun ortaya çıkmasında en büyük pay sahipliği teknoloji sektörüne aittir. Orta ölçekli firmalarda ise sağlık ve ilaç ile inşaat sektörü oltalama saldırılarına maruz kalması daha muhtemel sektörlerdir. Oltalama saldırılarına karşı küçük ölçekli firmalar arasında sağlık ve ilaç sektöründeki firmalar daha savunmasız durmakla birlikte eğitim sektöründeki küçük ölçekli firmaların diğer ölçekteki firmalardan daha savunmasız olduğu görülmektedir. Bu raporda yapılan çalışma genel olarak değerlendirildiğinde şirketlerin çalışan bazında ölçekleri büyüdükçe oltalama saldırılarına karşı savunmasız kalma durumunun arttığı görülmektedir.

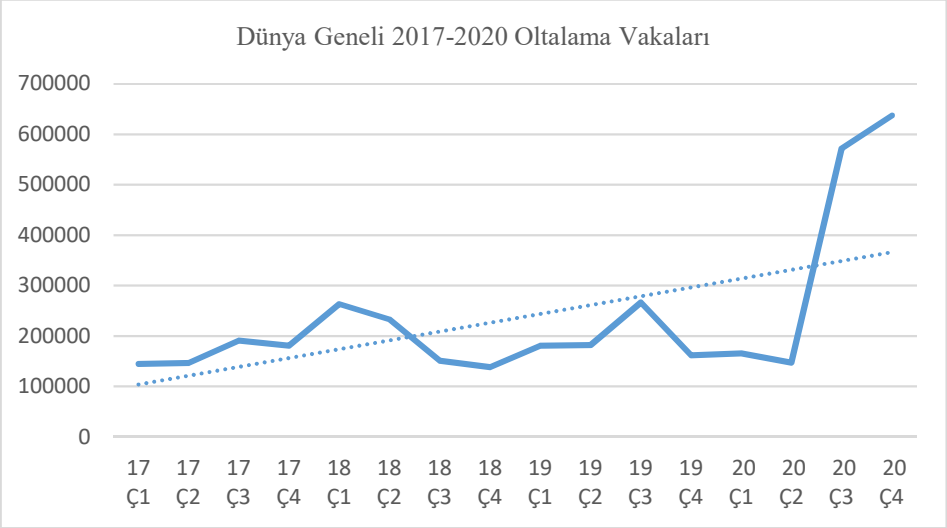
State Of The Phish Raporu'nda (2020) yer alan bir başka çalışmada ise oltalama hakkında 7 farklı ülkeden (Amerika Birleşik Devletleri, Avustralya, Fransa, Almanya, Japonya, İspanya ve Birleşik Krallık) 3500'ün üzerinde çalışanla anket yapılmıştır. "Oltalama" terimi hakkında doğru bilgi sahibi olanlar %61, yanlış bilgi sahibi olanlar %24 ve herhangi bir fikri olmayanlar %15'lik bir oranı temsil etmektedir. Şekil 1.2'de yer verilen grafikte ise farklı yaş gruplarının oltalama konusunda bilgi sahibi olup olmadığı incelenmiştir.



Şekil 1.2: Yaş Gruplarının Oltalama Terimini Doğru Bilme Oranları
(State of the Phish Report, 2020)

Şekil 1.2'de elde edilen grafikte 39 yaş ve üstünün oltalama hakkında daha fazla bilgi sahibi olduğu ortaya çıkmaktadır. Bu da nispeten gençlerin oltalama saldırılarına karşı daha savunmasız olduğunu göstermektedir.

Statista veri platformundan elde edilen Şekil 1.3'deki dünya genelinde yaşanan toplam oltalama vakaları incelendiğinde ise 2017'den 2020'nin son çeyreğine doğru ciddi bir artış olduğu dikkat çekmektedir.



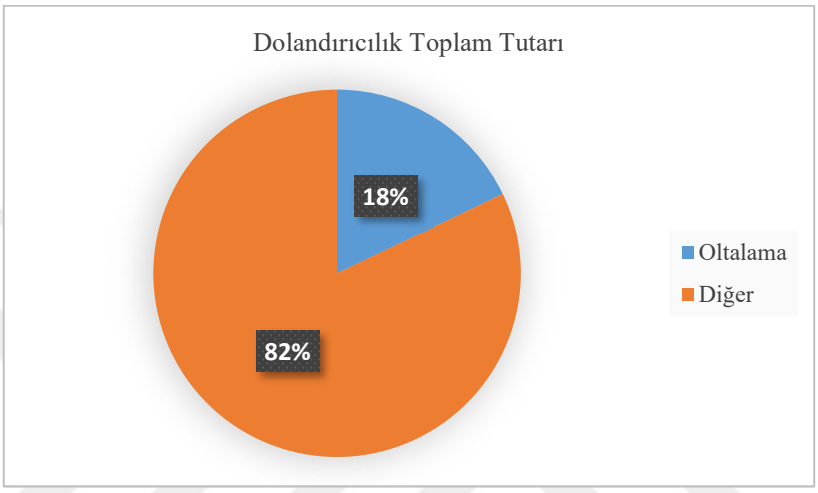
Şekil 1.3: 2017-2020 Yılları Dünya Genelinde Gerçekleşen Oltalama Vaka Sayıları (www.statista.com, 2020)

Şekil 1.3'deki grafik incelendiğinde oltalama saldırılarının hala trend saldırılardan biri olarak yerini koruduğu gözlemlenebilmektedir. Kaspersky'in 2020 spam ve oltalama raporuna göre kurumsal sektöre yönelik hedefli saldırıların genel olarak artan eğiliminin sonraki yıllarda da devam edeceği belirtilmiştir. Bunun sebebi ise popüler hale gelen uzaktan çalışma modelinin çalışanları daha savunmasız hale getirmesidir.

Oltalama bakımından Türkiye değerlendirmesi yapıldığında saldırıların finans sektörüne daha çok yoğunlaştığı Ulusal Siber Olaylara Müdahale Merkezi(USOM) tarafından elde edilen verilerden çıkarılmaktadır. USOM'un sitesinde yer alan tespit edilmiş oltalama URL adresleri incelendiğinde bankacılık sektörünü hedef alacak anahtar kelimelerin sıkça kullanıldığı görülmektedir. Bankacılık sektöründe yaşanan oltalama vakaları hakkında değerlendirme yapmak için bankaların SOME'lerinden (Siber Olaylara Müdahale Ekibi) yaşanan dolandırıcılık vakalarına ilişkin veriler elde edilmiştir. Oltalama vakaları sebebiyle yaşanan

maddi kayba gelindiğinde bu oranın da göz ardı edilemeyecek kadar büyük olduğu bankaların dolandırıcılık verilerinden ortaya çıkmaktadır.

Şekil 1.4'deki grafikte ise 2018 yılının son çeyreğinden 2021 yılının ilk çeyreğine kadar yaşanan oltalama vakalarından kaynaklı maddi kayıp tutarının tüm dolandırıcılık vakalarından kaynaklı maddi kayıp tutarına oranı yansıtılmaktadır.



Şekil 1.4: 2018 - 2021 Yılları Oltalama Vaka Tutarlarının Oranı (USOM, 2021)

Şekil 1.4'de yer alan grafikte oltalama saldırısı kaynaklı tutar oranının tüm vakalar kaynaklı tutarın %18'ini oluşturduğu görülmektedir. Bu oran diğer dolandırıcılık kaynaklarının çeşitliliği düşünüldüğünde oltalamanın verdiği maddi hasarın ne kadar büyük olduğunu göstermektedir.

Oltalama vakalarına ilişkin verilerden elde edilen istatistikler ve paylaşılan bilgiler ışığında, genel olarak değerlendirildiğinde oltalama saldırıları gerek küresel çapta gerek Türkiye çapında mücadele edilmesi gereken kritik siber güvenlik konularından birisi olmaya devam etmektedir.

Oltalama saldırılarının tespiti kullanıcıların farkındalığını artırarak veya bu tip saldırıları algılayan yazılımlar kullanılarak gerçekleştirilebilir. Oltalama saldırıları, günümüzde birçok araştırmacının dikkatini çekerek çeşitli araştırmaların yapılmasına ve aldatma kastı olan girişimleri tespit edebilecek yazılımların kurumların internete açık sunucularına yerleştirilmesine sebep olmuştur. Bununla birlikte oltalama saldırıları giderek daha da karmaşık, anlaşılması güç ve ileri derece geliştirilmiş hale gelmiş ve oltalama tekniklerine karşı

oluşturulan filtreleri aşma başarısı gösterebilmişlerdir. Başarılı olmuş oltalama siteleri ise genellikle bireysel verileri ele geçirmekte ve kişileri maddi zararlara uğratmaktadır. Bu yüzden bu sitelerin kişisel kullanım seviyesine indirgenerek engellenmesi çok önemlidir.

Bu tez çalışmasının GENEL KISIMLAR başlığı dört alt başlıktan oluşmaktadır. Makine Öğrenmesi başlığı altında çalışmanın yöntemini oluşturan makine öğrenmesi yöntemleri ile ilgili genel açıklamalara yer verilmiş, Özellik Seçme Metotları başlığında ise tez çalışması kapsamında ele alınan özellik seçme algoritmalarından bahsedilmiştir. Ayrıca Performans Metrikleri başlığında sonuçların değerlendirilmesi için dikkate alınmış metrikler hakkında bilgi verilmiş olup Literatür Taraması başlığında ise literatürden seçilip incelenen çalışmaların özetlerine yer verilmiştir. MALZEME VE YÖNTEM bölümünde veri seti ile ilgili detaylar, veri setine uygulanan işlemler ve uygulanan yöntemlerin açıklamaları yer almaktadır. Modellerden elde edilen sonuçlar BULGULAR bölümünde, yapılan çalışmaların sonuçları ile karşılaştırma ve bu çalışmanın literatüre katkısı TARTIŞMA VE SONUÇ bölümünde sunulmuştur.

1.1 ÇALIŞMANIN AMACI VE KAPSAMI

Bu çalışmanın amacı, bir web sitesinin URL özelliklerini kullanarak, oltalama amacıyla oluşturulmuş sahte internet sayfalarının gerçek sayfalardan ayırt edilerek tespit edilmesidir. Bu amaç doğrultusunda literatürde kabul edilen performans metriklerine bağlı olarak yüksek başarı oranına sahip bir sınıflandırıcı iş akışı modeli önerisi hedeflenmiştir. Ayrıca bu çalışmada özellik tabanlı yaklaşım kullanılarak önerilen oltalama saldırısı tespit tekniğini olabildiğince yalın ve modüler hale getirmek de amaçlanmıştır.

Bu çalışmanın bir diğer amacı da, karşılaşılan ve yeni ortaya çıkan oltalama saldırısı taktiklerini dikkate alacak bir sistem tasarlamak ve bu sistemi özellik kümesinin istenildiği gibi değiştirilmesine imkân verecek şekilde esnek ve modüler hale getirmektir. Modüler bir yapının sistemin gerçek hayatta kullanımını kolaylaştıracağı düşünülmektedir.

Bu çalışmada, oltalama saldırısı amacıyla oluşturulmuş sahte internet sitelerinin tespiti amacıyla URL yapısına bağlı olarak en etkili özellikleri tespit edecek özellik seçme yöntemleri ve buna bağlı olarak çalışacak sınıflandırma algoritmaları analiz edilmiştir. Belirlenmiş özellik setlerini kullanarak oltalama saldırısı amacıyla oluşturulmuş sahte internet sitelerini tespit etmek için yüksek performanslı bir sınıflandırıcı sistemi önerilmiştir.

Oltalama saldırılarının tespiti veya önlenmesi, kullanıcı farkındalığını artırarak veya yazılım tabanlı saldırı tespit/algılama teknikleri kullanılarak gerçekleştirilebilir. Oltalama saldırısı sorununu ele alan mevcut yazılım teknikleri olmasına rağmen, bu saldırılar gün geçtikçe çok yönlü hale gelerek saldırı önleme teknikleriyle ayarlanan filtreleri de atlatabilecek duruma gelmiştir. Statik kurallarla çalışan filtrelerin düşük başarı oranı problemine karşı, bu çalışmada URL özellikleri kullanılarak oltalama saldırısı amacıyla oluşturulmuş sahte internet sitelerini tespit etmek için bir sınıflandırıcı oluşturmak üzere makine öğrenmesi algoritmaları analiz edilmiştir.

Yüksek başarı oranı elde etmek için bir web sitesinin her türlü özelliğinin kullanılması işlem hızını düşürmektedir. Söz konusu araştırma probleminde işlem hızı çok önem arz etmektedir. Bir web sitesinin dolandırıcılık amaçlı oluşturulduğunun tespitinin geç yapılması olası zararı önlemede yetersiz kalmaktadır. Bu çalışmada çok çeşitli oltalama saldırısı sayfalarını başarılı bir şekilde tanımlamak için bu sayfalarla ilgili bir dizi özellik çıkarılmış ve analiz edilmiştir. Oluşturulan özellik kümesi, yapılan analizlere, uzman görüşlerine ve oltalama saldırısı tespitine ilişkin çeşitli mevcut akademik çalışmalara dayanarak oluşturulmuştur. Özellik tabanlı yaklaşımın arkasındaki amaç, oltalama saldırısı algılama tekniğini olabildiğince yalın hale getirmektir. Bu yaklaşımımızın ana hedeflerinden birisi de karşılaşılacak yeni oltalama saldırısı stratejilerini de her zaman dikkate alacak şekilde özellik kümesinin güncellenmesine imkân verecek esnek ve modüler bir model önermektir.

Oltalama saldırısında kullanılan parametrelerin belirlenerek güncel sahte URL adreslerinin tespit edilmesi, tespit işleminin en kısa süre içerisinde gerçekleşebilmesi, doğru tespit oranının maksimum değerde olması sorularına bu çalışmada cevaplar aranmaktadır.

1.2 ÇALIŞMANIN SINIRLILIKLARI

Bu çalışmada önerilen modelin, önem derecelerine göre seçilen minimum değişken sayısı ile genel sistemin doğruluğunu ve işlem hızını artıracağı beklenmektedir. Bununla birlikte yapılan çalışmada daha az verinin işlenmesinin sistemin karar verme hızını arttıracağı doğal olarak kabul edilmekle birlikte bu konuda net bir hız ölçümü yapılmaması bu tezin kısıtlılıklarından biri olmaktadır.

Oltalama saldırılarının tespitinin gerçek hayattaki yansımasına katkıda bulunacak kullanıcı farkındalığının artırılması çalışmalarına bu tezde yer verilmemiştir. Konunun sosyal bilimler açısından değerlendirilmemesi bu çalışmanın kısıtlılıklarından biri olmuştur.

Oltalama saldırılarının tespitinde kullanılan yazılım tabanlı birden çok teknik bulunmaktadır. Web sayfası görüntü işleme, web sayfası metin işleme, kaynak kodu incelemesi, URL tabanlı inceleme ve üçüncü parti servislerin kullanımı gibi teknikler arasında bu çalışmada sadece URL tabanlı inceleme üzerinde durulmuştur.

2. GENEL KISIMLAR

2.1 MAKİNE ÖĞRENMESİ

Bu çalışmada eldeki mevcut verilerden sonuca ilişkin çıkarımlar yapabilmek amacıyla öğrenebilen sistemlerin oluşturulması ve incelenmesiyle ilgili bir sistematik olarak yapay zekânın bir dalı olan makine öğrenmesi kullanılmıştır. Bu algoritmalar, sonucu örtülü olan bilginin çıkarımını yapmak için büyük miktarlardaki verileri otomatik olarak analiz etme yöntemleri sağlamaktadır. Özellikle, makine öğrenmesi, elde bulunan verilerdeki desenleri keşfetmeyi sağlayan ve daha sonra bu desenleri verilerin yeni örneğinin sonucunu tahmin etmek için kullanan algoritmaların incelenmesidir. Bu sebeple bu çalışmada, çalışmanın amacına uygun olacağı düşünülen makine öğrenmesi ve derin öğrenme algoritmaları kullanılmıştır.

Bu çalışmada olduğu gibi bir makine öğrenmesi sistemi, oltalama saldırısı amacıyla oluşturulmuş sahte internet siteleri ve yasal internet siteleri arasında ayrım yapmayı öğrenmek için bu sitelerin özellikleri üzerinde eğitilebilmektedir. Yine aynı şekilde sistem, eldeki verilerden bir desen öğrendikten sonra yeni bir internet sitesini sınıflandırmak için kullanılabilmektedir.

Makine öğrenmesinde sınıflandırma görevi genellikle denetimli öğrenme olarak adlandırılır. Bu çalışmadaki denetimli öğrenmede, internet sitesi veri kümesinde yer alan URL'ler "Oltalama Sitesi" ve "Yasal Site" sınıflarına uygun olarak etiketlenmektedir. Sınıflandırmayı öğrenme yeteneği ve sonrasında sınıflandırma yapma, insanlara ve bilgisayar programlarına karar verme gücü vermektedir. Bu çalışma boyunca, çoklu özellik seçim yöntemlerinin etkilerini ve uyumluluğunu karşılaştırmak için makine öğrenmesi algoritmalarının performansı analiz edilmiştir.

Bu çalışmada Naive Bayes, J48 Karar Ağacı, Sınıflandırma ve Regresyon Ağacı, Rastgele Orman ve Sıralı Minimal Optimizasyon, Sinir Ağları ve Derin Öğrenme olmak üzere farklı sınıflandırma algoritmaları değerlendirmeye tabi tutulmuştur. Bu bölümde ise kullanılan sınıflandırma algoritmalarına genel bir bakış verilmiştir.

2.1.1 Naive Bayes (NB)

Naïve bayes sınıflandırıcı (Mitchell, 1997), basit ama karşılaştırmalı olarak içgüdüsel bir kavram üzerinde çalışır. Bazı durumlarda, diğer karmaşık algoritmalardan daha iyi performans göstermektedir. Veri örneğinde yer alan değişkenleri tek tek ve birbirinden bağımsız olarak gözlemleyerek kullanır.

Naive Bayes algoritması, Bayes teoremine dayanan ve sınıflandırma problemlerini çözmek için kullanılan denetimli bir öğrenme algoritmasıdır. Olasılıksal bir sınıflandırıcı olup bir nesnenin olasılığına dayanarak öngörüde bulunur. Naive Bayes algoritması metin sınıflandırmasında yaygın kullanıldığı için spam filtreleme gibi siber güvenlik alanı uygulamalarında etkili sonuçlar vermektedir.

Bayes Teoremi, daha önce meydana gelmiş başka bir olayın olasılığı göz önüne alındığında, bir olayın meydana gelme olasılığını bulur. Bayes teoremi matematiksel olarak denklem (2.1) ile ifade edilir:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (2.1)$$

(2.1) nolu denklemde A ve B olaylar olmak üzere P(A) A'nın öncül olasılığını, P(B) B'nin öncül olasılığını, P(A|B) A'nın koşullu olasılığını, P(B|A) B'nin koşullu olasılığını göstermektedir. Bayes teoremi eğitim için kullanılacak verisetine denklem (2.2) ile verildiği şekilde uygulanabilmektedir.

$$P(y|X) = \frac{P(X|y).P(y)}{P(X)} \qquad (2.2)$$

Burada y, sınıf değişkeni ve X, n boyutlu bir bağımlı özellik vektörüdür ($X=(x_1,x_2,\ldots,x_n)$).

Naive Bayes algoritmasının Bayes Teoremi ile ilişkisine gelindiğinde, özelliklerin bağımsız olaylar olarak ön plana çıkması söz konusudur. (2.1) nolu denklemdeki A ve B bağımsız olaylar ise bu iki olay arasındaki ilişki P(A,B)=P(A).P(B) eşitliği şeklinde yazılabildiği için (2.3) nolu denklem elde edilir.

$$P(y|x_1 x_2, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)} \quad (2.3)$$

Bu denklemde belirli bir girdi için payda sabit kaldığından, $P(x_1)$, $P(x_2)$, …, $P(x_n)$ terimleri kaldırılabilir. Bir sınıflandırıcı modeli oluşturmak için, sınıf değişkeni y'nin tüm olası değerleri için verilen girdi setinin olasılığı bulunur ve çıktı maksimum olasılıkla alınır. Bu durum matematiksel olarak (2.4) nolu denklemde verildiği şekilde ifade edilebilir:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \quad (2.4)$$

Burada, P(y) sınıf olasılığı ve P (x_i|y) koşullu olasılık olarak adlandırılmaktadır (Mitchell, 1997).

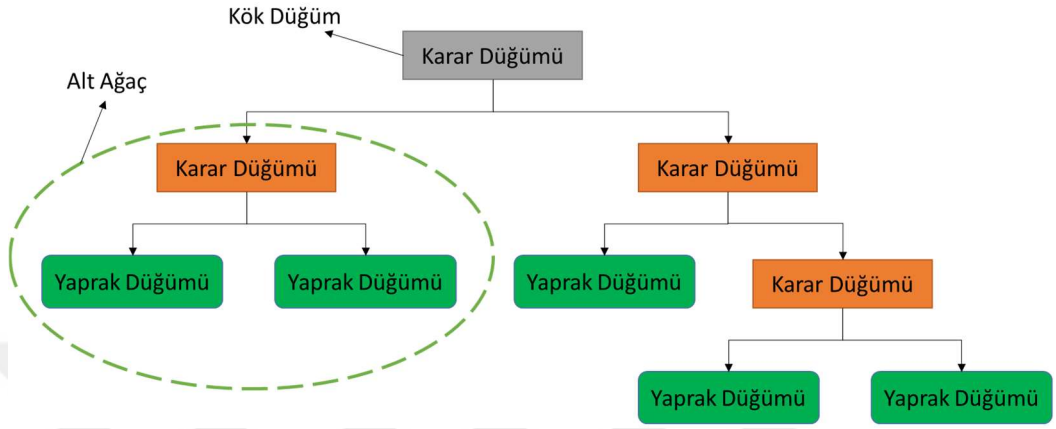
2.1.2 J48 Karar Ağacı (DT)

J48 karar ağacı (Quinlan, 1993), en açık şekilde ayrım yapan eğitim seti niteliklerine dayalı bir ağaç oluşturur. Veri örneklerini en iyi şekilde sınıflandırabilmemiz için bize bilgi verebilen bu özelliğin yüksek bilgi kazanımına sahip olduğu söylenmektedir. Kendi kategorisine giren veri örneklerinin ilgili değişken için yaklaşık aynı değere sahip olduğu olası değerlerinden sonra kendisine atanan hedef değer elde edilir.

Karar ağacı, hem sınıflandırma hem de regresyon problemleri için kullanılabilen, ancak daha çok sınıflandırma problemlerini çözmek için tercih edilen bir denetimli öğrenme tekniğidir. İç düğümlerin bir veri kümesinin özelliklerini, dalların karar kurallarını ve her yaprak düğümün sonucu temsil ettiği ağaç yapılı bir sınıflandırıcıdır.

Bir Karar ağacında, karar düğümü ve yaprak düğümü olmak üzere iki düğüm vardır. Karar düğümleri herhangi bir karar vermek için kullanılır ve birden çok şubeye sahipken, yaprak düğümleri bu kararların çıktısıdır ve başka dallar içermez. Kararlar veya test, verilen veri setinin özelliklerine göre gerçekleştirilir. Karar ağacı, verilen koşullara göre bir soruna/karara tüm olası çözümleri elde etmek için grafiksel bir şekil de sunabilmektedir. Bir ağaca benzer şekilde, daha fazla dalda genişleyen ve ağaç benzeri bir yapı oluşturan kök düğüm ile başladığı için bu

algoritma karar ağacı olarak adlandırılır. Bir karar ağacı basitçe bir soru sorar ve cevaba göre (Evet/Hayır), ağacı alt ağaçlara ayırır. Şekil 2.1'de, bir karar ağacının genel yapısı gösterilmektedir.



Şekil 2.1: Karar Ağacı Algoritmasının Genel Yapısı
(https://www.devops.ae/decision-tree-classification-algorithm, 2022)

Bir karar ağacında, verilen veri kümesinin sınıfını tahmin etmek için algoritma ağacın kök düğümünden başlar. Bu algoritma, kök özniteliğinin değerlerini kayıt (gerçek veri kümesi) özniteliğiyle karşılaştırır ve karşılaştırmaya dayalı olarak dalı izler ve bir sonraki düğüme atlar.

Bir sonraki düğüm için, algoritma yine öznitelik değerini diğer alt düğümlerle karşılaştırır ve daha ileri gider. Ağacın yaprak düğümüne ulaşıncaya kadar işleme devam eder.

Bir Karar ağacı uygularken ana sorun, kök düğüm ve alt düğümler için en iyi özniteliğin nasıl seçileceğidir. Dolayısıyla, bu tür problemleri çözmek için öznitelik seçim ölçüsü olarak adlandırılan teknikler kullanılmaktadır. Bu tekniklerden yaygın kullanılanı "Bilgi Kazanımı" ve "Gini Endeksi"dir.

Bilgi kazanımı, bir veri kümesinin bir özniteliğe göre bölümlendirilmesinden sonra entropideki değişikliklerin ölçülmesidir. Entropi, rastgele bir değişkenin belirsizlik, safsızlık veya düzensizlik derecesi veya bir saflık ölçüsüdür. Keyfi bir örnek sınıfının safsızlığını karakterize eder. Entropi, veri noktalarındaki safsızlıkların veya rasgeleliğin ölçülmesidir. Entropi (2.5) nolu denklem kullanılarak elde edilir.

$$E(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad (2.5)$$

Burada S ile verisetindeki örneklem boyutu, E(S) ile S'nin entropi değeri, p_i ile i sınıfına ait olasılık değeri verilmektedir. Entropi bit cinsinden ölçülen beklenen kodlama uzunluğu olduğundan logaritma 2 tabanındadır. Ayrıca, sınıf sayısını gösteren n değeri büyüdükçe entropinin logaritma değeri de büyümektedir.

Bilgi kazanımı, bir özelliğin bize bir sınıf hakkında ne kadar bilgi verdiğini hesaplar. Bilgi kazancının değerine göre düğüm bölünür ve karar ağacı oluşturulur. Bir karar ağacı algoritması her zaman bilgi kazanımının değerini maksimize etmeye çalışır ve en yüksek bilgi kazancına sahip olan bir düğüm/öznitelik önce bölünür. Bilgi kazanımı, denklem (2.6) kullanılarak hesaplanabilir:

$$Bilgi\ Kazanımı(S, \ddot{O}) = E(S) - \sum_{m}^{n=Değer(\ddot{O})} \frac{|S_m|}{|S|} E(S_m) \quad (2.6)$$

Burada Ö ile S örneklemindeki her bir öznitelik, m ile öznitelikte yer alan her bir benzersiz değer, S_m ile öznitelik içindeki her bir benzersiz değerin sayısı gösterilmektedir. (2.6) nolu denklemde her bir öznitelik için entropi değerleri ayrı ayrı hesaplanarak yaprak düğümleri en uygun şekilde düzenlenir (Mitchell, 1997).

Gini indeksi, CART (Sınıflandırma ve Regresyon Ağacı) algoritmasında bir karar ağacı oluştururken kullanılan safsızlık veya saflığın bir ölçüsüdür. Düşük Gini endeksine sahip bir özellik, yüksek Gini endeksine kıyasla tercih edilmesi gerekir.

Gini indeksinde yalnızca ikili bölmeler oluşturulur ve CART algoritması, ikili bölmeler oluşturmak için Gini indeksini kullanmaktadır. Gini indeksi (2.7) nolu denklem kullanılarak hesaplanabilir:

$$Gini\ indeksi = 1 - \sum_{i=1}^{n} (p_i)^2 \quad (2.7)$$

burada p_i ile i sınıfına ait olasılık değeri verilmektedir.

J48 modeli Waikato Environment for Knowledge Analysis (WEKA) veri madenciliği ortamında kullanılmaktadır (Quinlan, 1993). J48, WEKA aracı kullanılarak uygulanan C4.5 (Witten, 2005) karar ağacının açık kaynaklı bir Java uygulamasıdır. Algoritmada yeni bir öğenin sınıflandırılması için öncelikle mevcut eğitim verilerinin öznitelik değerlerine dayalı bir karar ağacı gerektirmektedir. Eğitim verilerindeki mevcut öğe kümesine bağlı olarak, çeşitli örnekleri en net şekilde sınıflandıran özelliği tanımlar. Veri örnekleri hakkında en çok bilgi veren, yani en iyi sınıflandırmaya yol açabilecek özelliğin en yüksek bilgi kazanımına sahip olması gerekmektedir. Bu özelliğin olası değerlerine bağlı olarak, dallar sonlandırılır ve bir hedef değer atanır. Diğer durumlarda, algoritma en yüksek bilgi kazancını sağlayan diğer öznitelikleri araştırır. Süreç, bir hedef değerin belirlenmesi için belirli bir kural veren niteliklerin kombinasyonuna ilişkin net bir karara ulaşılıncaya kadar bu şekilde devam eder. Bu karar ağacının yardımıyla, tüm ilgili öznitelikler ve değerleri kontrol edilir, böylece tüm yeni örneklerin hedef değerleri atanır veya tahmin edilir (Acharya, 2016).

2.1.3 Sınıflandırma ve Regresyon Ağacı (CART)

Sınıflandırma ve Regresyon Ağacı algoritması CART terimi olarak kısaltılmış ve 1980'lerin başında Leo Breiman tarafından geliştirilmiştir (Breiman, 1984). Bu yöntem veri keşfi ve tahmini için de kullanılmaktadır. CART, geçmiş verileri kullanarak karar ağacı oluşturur ve karar ağacı oluşturmak amacıyla tüm gözlemler için önceden atanmış sınıflarla birlikte geçmiş verilerden oluşan öğrenme örneğini kullanır.

Basit CART, ikili karar ağacı oluşturur, yani yalnızca iki çocuk oluşturur. Entropi, en iyi bölme niteliğini seçmek için kullanılır ve bu algoritma geçmiş verilerin kullanılması durumunda iyi sonuç vermektedir.

CART karar ağacı, sürekli ve nominal öznitelikleri hedefler ve öngörücüler olarak işleyebilen ikili özyinelemeli bir bölümleme prosedürüdür. Veriler ham haliyle işlenir; gruplama gerekmez veya önerilmez. Kök düğümden başlayarak, veriler iki çocuğa bölünür ve her çocuk sırayla torunlara bölünür. Ağaçlar, bir durdurma kuralı kullanılmadan maksimum boyutta büyütülür; temelde ağaç yetiştirme süreci, veri eksikliği nedeniyle daha fazla bölünme mümkün olmadığında durur. Maksimum boyutlu ağaç daha sonra yeni maliyet karmaşıklığı budama yöntemiyle köke geri budanır. Budanacak bir sonraki bölüm, ağacın eğitim verileri üzerindeki

genel performansına en az katkıda bulunan bölümdür ve bu bölümler bir seferde birden fazla şekilde kaldırılabilir. CART mekanizmasının tek bir ağaç değil, her biri en uygun ağaç olmaya aday iç içe budanmış bir dizi ağaç üretmesi amaçlanmıştır. Doğru büyüklükte ağaç, budama sırasındaki her ağacın tahmin performansının bağımsız test verileri üzerinde değerlendirilmesiyle belirlenir. C4.5'ten farklı olarak, CART ağaç seçimi için dahili (eğitim verilerine dayalı) bir performans ölçüsü kullanmaz. Bunun yerine, ağaç performansı her zaman bağımsız test verileriyle ölçülür ve ağaç seçimi yalnızca test verilerine dayalı değerlendirmeden sonra devam eder. Test veya çapraz doğrulama yapılmadıysa, CART sıradaki hangi ağacın en iyi olduğu konusunda kararsız kalır. Bu, eğitim verisi ölçümleri temelinde tercih edilen modelleri üreten C4.5 veya klasik istatistikler gibi yöntemlerle keskin bir zıtlık oluşturmaktadır (Wu, 2009).

2.1.4 Rastgele Orman (RF)

Rastgele Orman algoritması (Breiman, 2001) eğitim zamanında çok sayıda karar ağacı inşa ederek ve tek tek ağaçların çıkardığı sınıfların modu olan sınıfı çıkararak çalışan sınıflandırma için toplu bir öğrenme yöntemidir.

Bir RF sınıflandırıcı, sınıflandırma oranını iyileştirmek için bir dizi karar ağacı kullanmaktadır. Rastgelelik, her bir ağaç için mevcut hale getirilen eğitim verilerinin alt kümesini rastgele örnekleyerek sağlanabilmektedir. Nihai sınıflandırma zamanında, çok sınıflı sınıflandırma sonuçları elde etmek için tüm karar ağaçlarından alınan oylar toplanarak değerlendirilmektedir. Bu çalışma, URL'nin farklı özelliklerini kullanarak RF algoritmasının performansını incelemeyi amaçlamaktadır.

Rastgele orman, hem sınıflandırma hem de regresyon için kullanılan denetimli bir öğrenme algoritmasıdır. Ancak, esas olarak sınıflandırma problemleri için kullanılır. Rastgele orman, popüler makine öğrenmesi yöntemleri arasında başarılı bir tahmin doğruluğu ve model yorumlanabilirliği kombinasyonu sağlar. Rastgele ormanda kullanılan rastgele örnekleme ve topluluk stratejileri, daha iyi genellemelerin yanında doğru tahminler elde edilmesini de sağlar. Bu genelleme özelliği, varyansı azaltarak genellemeyi geliştiren torbalama yönteminden gelirken, yükseltme gibi benzer yöntemler önyargıyı azaltarak bunu başarır (Yang, 2010). Rastgele orman, örneklemelerde torbalama, rastgele değişken alt kümeleri ve çoğunluk oylama yöntemi yoluyla daha iyi sonuçlar elde ederken karar ağaçlarının birçok faydalı özelliğine de

sahip olmaya devam eder (Breiman, 2001). Nihai karar için, RF sınıflandırıcı tek tek ağaçların kararlarını toplar; sonuç olarak, RF sınıflandırıcı iyi bir genelleme sergilemektedir. RF sınıflandırıcı, aşırı uyum sorunu olmadan doğruluk açısından diğer sınıflandırma yöntemlerinin çoğundan daha iyi performans gösterme eğilimindedir.

Rastgele ormanda yaygın olarak kullanılan ilk önem ölçüsü Gini önemidir. Gini önemi, doğrudan sonuçta ortaya çıkan rastgele orman ağaçlarındaki Gini indeksinden (Breiman, 2001) elde edilir. Rastgele orman sınıflandırıcı, ağaç öğrenme aşamasında hangi özniteliğin bölüneceğini belirlemek için Gini indeksi adı verilen bir bölme işlevi kullanır. Gini indeksi, bir düğüme atanan örneklerin, ana düğümdeki bir bölünmeye dayalı olarak safsızlık/eşitsizlik düzeyini ölçer. Gini indeksi denklem (2.8)'deki gibi tanımlanır. İki sınıfın olduğu ikili sınıflandırma durumu altında p, belirli bir k düğümüne atanan pozitif örneklerin kesrini ve (1-p)'de negatif örneklerin kesrini temsil eder.

$$G_k = 2p\,(1-p) \quad (2.8)$$

Bir düğüm ne kadar safsa, Gini değeri o kadar küçüktür. Belirli bir öznitelik kullanılarak her bir düğümün bölünmesi yapıldığında, iki alt düğüm için Gini değeri ana düğümden daha azdır. Ormandaki genel önem, ormandaki tüm ağaçlar arasındaki önem değerinin toplamı veya ortalaması olarak tanımlanır.

Rastgele orman algoritmasının çalışması şöyledir:

Adım 1: İlk olarak, belirli bir veri kümesinden rastgele örnekler seçilir.

Adım 2: Bu algoritma her örneklem için bir karar ağacı oluşturur ve her karar ağacından tahmin sonucunu alır.

Adım 3: Tahmin edilen her sonuç için oylama yapılır.

Adım 4: Sonunda, nihai tahmin sonucu olarak en çok oylanan tahmin sonucu seçilir.

Farklı karar ağaçlarının sonuçlarının ortalamasını alarak veya birleştirerek aşırı uyum probleminin üstesinden gelmesi, çok çeşitli veri öğeleri için tek bir karar ağacından daha iyi çalışması, tek karar ağacından daha az varyansa sahip olması, verilerin ölçeklendirilmesinin

gerekmemesi, ölçeklendirilmeden veri sağlandıktan sonra ya da verilerin büyük bir kısmı eksik olsa bile doğruluğunu iyi yönde koruması rastgele orman algoritmasının avantajlarındandır.

Rastgele orman birçok sınıflandırma ağacını büyütmektedir. Bir giriş vektöründen yeni nesneyi sınıflandırırken, giriş vektörünü ormandaki ağaçların her birine yerleştirir. Her ağaç sınıflandırma sonucu verir ve sonuç ağacın o sınıf için verdiği oy olarak varsayılır. Orman algoritmasıda, ormandaki tüm ağaçların üzerinden en çok oyu alan sınıflandırmayı seçmektedir.

2.1.5 Sıralı Minimal Optimizasyon (SMO)

John C. P. (1998) tarafından önerilen Sıralı Minimal Optimizasyon (SMO), destek vektör makinelerinin (SVM) eğitimi için hızlı bir algoritmadır; burada SVM bir dizi pozitif örneği bir dizi negatif örnekten maksimum payla ayıran bir hiper düzlemdir. SMO büyük problemler için iyi bir performans sergilemektedir.

SMO, destek vektör makinelerinin (SVM) eğitimi sırasında ortaya çıkan ikinci dereceden programlama (QP) problemini çözmek için bir algoritmadır. 1998'de Microsoft Research'te John Platt tarafından icat edilmiştir (John, 1998). SMO, destek vektör makinelerinin eğitimi için yaygın olarak kullanılır ve popüler LIBSVM (Library for SVM) aracı tarafından uygulanır (Chang, 2011) (Luca, 2006).

SMO, SVM QP (Quadratic Programming) problemini herhangi bir ekstra matris depolaması olmadan ve sayısal QP optimizasyon adımlarını hiç kullanmadan hızla çözebilen basit bir algoritmadır. SMO, yakınsamayı sağlamak için Osuna teoremini kullanarak genel QP problemini QP alt problemlerine ayırır. Önceki yöntemlerin aksine, SMO her adımda mümkün olan en küçük optimizasyon problemini çözmeyi seçer. Standart SVM QP problemi için, mümkün olan en küçük optimizasyon problemi iki Lagrange çarpanı içerir, çünkü Lagrange çarpanları doğrusal bir eşitlik kısıtlamasına uymak zorundadır. SMO, her adımda birlikte optimize etmek için iki Lagrange çarpanı seçer, bu çarpanlar için en uygun değerleri bulur ve SVM'yi yeni optimum değerleri yansıtacak şekilde günceller. SMO'nun avantajı, iki Lagrange çarpanı için çözümlemenin analitik olarak yapılabilmesidir. Böylece sayısal QP optimizasyonundan tamamen kaçınılır. Algoritmanın iç döngüsü, tüm bir QP kitaplık yordamını çağırmak yerine kısa bir miktarda C koduyla ifade edilebilir. Algoritma sürecinde daha fazla optimizasyon alt problemi çözülmüş olsa da, her bir alt problem o kadar hızlıdır ki, genel QP problemi hızlı bir şekilde çözülür. Ek olarak, SMO hiçbir şekilde ekstra matris

depolaması gerektirmez. Bu nedenle, çok büyük SVM eğitim sorunları, sıradan bir kişisel bilgisayarın veya iş istasyonunun belleğinin içine sığabilir. SMO, matris algoritmaları kullanmadığından, sayısal kesinlik problemlerine daha az duyarlıdır. İki Lagrange çarpanını çözümleme için analitik yöntem ve hangi çarpanların optimize edileceğini seçmek için buluşsal yöntem olmak üzere SMO'nun iki bileşeni vardır (John, 1998).

2.1.6 Sinir Ağları (NNet)

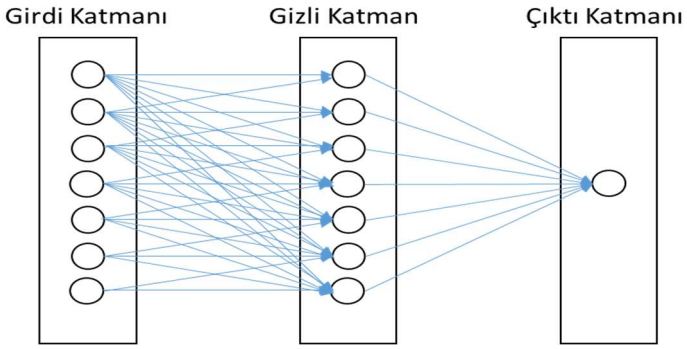
Sinir ağları algoritmasının kökenleri nörobiyoloji alanına dayanmaktadır (McCulloch, 1943). Yapay sinir ağları, beyinde nöronlar aracılığıyla gerçekleşen bilgi işleme yöntemlerine dayanan modellerdir. Bir nöronun temel işlevi, elektrik sinyallerinin toplanması, işlenmesi ve yayılmasıdır. Bu açıdan yapay sinir ağları, beyin fonksiyonları göz önüne alındığında karmaşık sorunların ele alınmasında verimli sonuçlar üretmektedir.

Son zamanlarda yapay sinir ağları, birçok disiplinde sınıflandırma, kümeleme, örüntü tanıma ve tahmin için popüler ve yararlı bir model haline gelmiştir. Yapay sinir ağları makine öğrenmesi için bir model türüdür ve kullanışlılıkla ilgili geleneksel regresyon ve istatistiksel modellere göre nispeten rekabetçi hale gelmiştir (Dave, 2014).

Yapay sinir ağları uygulamasının bir avantajı, modelleri kullanımı kolay ve büyük girdilere sahip karmaşık doğal sistemlerden daha doğru hale getirebilmesidir (Jahnavi, 2017).

Yapay sinir ağlarının, problem çözme ve makine öğrenmesine uygulanan çok yeni ve kullanışlı bir model olduğu bulunmuştur. Yapay sinir ağları, insan beyninin biyolojik sinir sistemi işlevine benzer bir bilgi yöneticisi modelidir (Wang, 2017).

Yapay sinir ağları, öğrenme süreci aracılığıyla veri sınıflandırma veya örüntü tanıma gibi belirli bir uygulama için tasarlanabilmektedir. Genel olarak yapay sinir ağları insan beyninin bir taklidi gibi işlev görmektedit. (Dave, 2014) (Huang, 2017). Tipik bir yapay sinir ağının mimarisi Şekil 2.2'de gösterilmektedir.



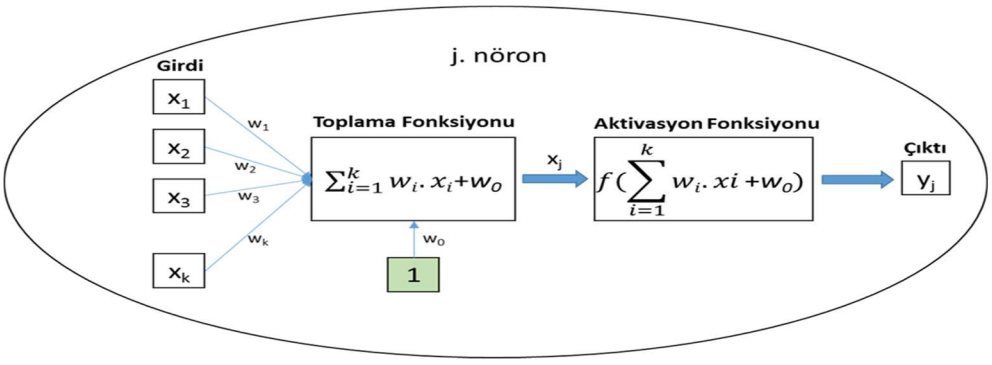
Şekil 2.2: Yapay Sinir Ağının Temel Yapısı
(Kabalcı, 2014)

Giriş düğümleri dış dünyadan ağa bilgi sağlar ve hep birlikte "Girdi Katmanı" olarak adlandırılır. Giriş düğümlerinin hiçbirinde hesaplama yapılmaz, sadece bilgileri gizli düğümlere aktarırlar.

Gizli düğümlerin dış dünyayla doğrudan bir bağlantısı yoktur (bu nedenle "gizli" adı da buradan gelmektedir). Gizli düğümler hesaplamalar yaparlar ve bilgileri giriş düğümlerinden çıkış düğümlerine aktarırlar. Gizli düğümlerin bir koleksiyonu "Gizli Katman" oluşturur. İleri beslemeli bir ağ yalnızca tek bir giriş katmanına ve tek bir çıktı katmanına sahip olsa da birden çok gizli katmana sahip olabilir.

Çıktı düğümleri topluca "Çıktı Katmanı" olarak adlandırılır ve hesaplamalardan ve ağdan dış dünyaya bilgi aktarımından sorumludur.

Şekil 2.3'te yer verilen yapı, bir yapay sinir ağına ait katman içerisinde yer alan nöronun genel şeklini temsil etmektedir. Bir nöron yapay sinir ağının işlemlerinin yapıldığı ana bileşendir.



Şekil 2.3: Bir Nöronun İçyapısı
(Gupta, 2013)

Şekil 2.3'de gösterilen terimler incelendiğinde, w_i her bir nöronda bulunan ve bir önceki katmanda bulunan girdi değerleriyle işlemlerin yapıldığı ağırlık değerlerini, w_0 modele verilen verilere en iyi uyacak şekilde yardımcı olan önyargı sabitini, x_i girdi değerlerini, y_i çıktı değerlerini temsil etmektedir. Burada önyargının bağımsız bir bağlantı yaptığı ve değerinin "1" olduğu önemli bir noktadır.

Ayrıca Şekil 2.3'de toplama ve aktivasyon olmak üzere iki fonksiyonun yer aldığı görülmektedir. Toplama fonksiyonu, girdi değerlerinin ağırlıklarla çarpımlarının toplandıktan sonra net girdiyi elde ederek aktivasyon fonksiyonu ile işleme hazır hale getiren fonksiyondur. Aktivasyon fonksiyonları, net girdilerin eşik değerine göre çıktısını belirleyen matematiksel denklemlerdir. Aktivasyon fonksiyonlarının ayrıca sinir ağının yakınsama yeteneği ve yakınsama hızı üzerinde büyük bir etkisi vardır. Ayrıca aktivasyon işlevi 1 ila -1 veya 0 ila 1 aralığındaki herhangi bir girişin çıkışını normalleştirmeye yardımcı olur (Tino, 2015).

2.1.7 Derin Öğrenme (DL)

Derin sinir ağlarının (Bengio, 2015) derinlik kavramı sinir ağlarının arttırılmış gizli katman sayısından gelmektedir. Derin sinir ağı mimarisinin katmanları gücünü her bir katmanda bulunan işlem birimleri denilen nöronlardan almaktadır. Nöronlar derin sinir ağları üzerinde hesaplamaların yapıldığı en temel işlem birimleridir. Bir nöronda gerçekleşen işlemlerin matematiksel ifadesi (2.9) nolu denklemle gösterilmektedir.

$$f(\sum x_i w_{ij} + z_j) = \beta_j \quad (2.9)$$

Denklem (2.9)'da *i* indisi ile özellik sayısı, *j* indisi ile nöron sayısı, *x* ile *i* boyutlu giriş vektörü, *z* ile önyargı vektörü, β ile *j* boyutlu çıktı vektörü, *w* ile (ixj) boyutlu ağırlık matrisi, *f* ile aktivasyon fonksiyonu ifade edilmiştir. j. nöronda gerçekleşen işlemde giriş vektöründen gelen tek boyutlu *x* matrisi ile *w* ağırlık matrisi çarpıldıktan sonra *z* önyargı matrisi ile toplanır. Elde edilen sonuç *f* aktivasyon fonksiyonu için girdi değerini oluşturur ve aktivasyon fonksiyonundan β çıktı vektörü elde edilmiş olur.

Önerilen derin sinir ağı modeli çok katmanlı bir yapıya sahip olduğunda (2.9) nolu denklem ile gösterilen form, (2.10) nolu denklemdeki matematiksel ifade ile gösterilmektedir.

$$\beta = h_i(h_{i-1} \dots (h_1(\alpha))) \quad (2.10)$$

Denklem (2.10)'da *h* her bir katmanda işlem yapan fonksiyonu, *i* indisi katman sayısını, α katmanın girdi vektörünü, β çıktı vektörünü temsil etmektedir.

Önerilen derin sinir ağı modeli içerisinde hiperbolik tanjant fonksiyonunun kullanımı tercih edilmiştir. Aktivasyon fonksiyonu olarak tercih edilen hiperbolik tanjant fonksiyonuna ait matematiksel ifadeye (2.11) nolu denklemle yer verilmiştir.

$$tanh\ (x) = \frac{e^{2x}-1}{e^{2x}+1} \qquad (2.11)$$

(2.11) nolu denklemde *tanh* ile hiperbolik tanjant aktivasyon fonksiyonu, *x* ile nöronlarda gerçekleşen toplama fonksiyonu çıktısı, e ile euler sabiti ifade edilmektedir. Hiperbolik tanjant fonksiyonu -1 ile 1 arasındaki değerlerde çıktı üretmektedir. Fonksiyon çıktısının belli sabit değerler arasında hareket etmesi gelen büyük sonuçlardan daha az etkilenmesine yönelik avantaj sağlamaktadır.

Derin sinir ağına ait ağırlıkların ayarlanacak değişkenlerden oluşması sebebiyle öğrenme problemi bir arama veya optimizasyon problemi olarak değerlendirilmektedir. Dolayısıyla yeni ağırlık değerleri bulmak için bir çözüm yolunun istenilen çözüme ne kadar yakın olduğunun tespiti büyük öneme sahiptir. Benzer şekilde, modelin eğitimi sonucu elde edilen çıktı değerlerinin beklenen çıktılardan ne kadar farka sahip olduğunu ölçmek amacıyla kayıp fonksiyonları kullanılmaktadır. Önerilen modelde ikili sınıflandırma problemine çözüm arandığı için kayıp fonksiyonu olarak ikili çapraz entropi fonksiyonu tercih edilmiştir. İkili çapraz entropi fonksiyonuna ait matematiksel ifade denklemlerle gösterilmektedir:

$$q_0 = 1 - \hat{y}, q_1 = \hat{y} \qquad (2.12)$$

$$p_0 = 1 - y, p_1 = y \qquad (2.13)$$

$$L = -\sum p_n \log q_n = -y \log \hat{y} - (1-y) \log (1-\hat{y}) \qquad (2.14)$$

(2.14) nolu denklemde, *L* ile kayıp fonksiyonu, *p* ile gerçek olasılık, *q* ile tahmin edilen olasılık, *y* ile gerçek olasılık değeri, $\hat{y}$ ile tahmin edilen olasılık değeri, *log* ile doğal logaritma, *n* sayısı ile 0 ve 1 tamsayı değerleri ifade edilmektedir.

Kayıp fonksiyonunun uygulanması sonucu elde edilen kaybı en aza indirmek için bir optimizasyon algoritması kullanılmaktadır. Bir yandan, karmaşık bir derin öğrenme modelini

eğitmek çok uzun sürebildiği için optimizasyon algoritmasının performansı, modelin eğitim verimliliğini doğrudan etkilemektedir. Öte yandan, derin öğrenme modellerinin performansını iyileştirmek için optimizasyon algoritmasındaki hiper parametrelerin hedefe yönelik bir şekilde ayarlanması öneme sahiptir.

Bu çalışmada optimizasyon algoritması olarak Adaptive Moment Estimation (ADAM) algoritmasının kullanımı tercih edilmiştir. ADAM (Kingma, 2015), model eğitimi için kullanılan veri setinin küçük parçalara ayrılmış hali olan "minibatch" yapısında daha büyük gözlem setleri kullanılarak vektörleştirmeden kaynaklanan önemli ek verimlilik sağlaması, gereksiz verilere karşı doğal esnekliği ve yakınsamayı hızlandırmak için önceki gradyanları bir sonraki ile toplayan bir mekanizma bulundurması gibi tüm bu teknikleri tek bir etkili öğrenme algoritmasında birleştirmektedir. Bu yüzden derin öğrenmede kullanılacak daha sağlam ve etkili optimizasyon algoritmalarından biri olan popüler bir algoritmadır. ADAM optimizasyon algoritmasının matematiksel ifadelerine aşağıdaki denklemlerle yer verilmiştir.

$$t \in 1,2,\ldots,T \text{ olmak üzere } g_t = \nabla_t f_t(x_{t-1}) \quad (2.15)$$

$$n_t = \beta_1 n_{t-1} + (1-\beta_1) g_t \quad (2.16)$$

$$m_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2 \quad (2.17)$$

$$n'_t = \frac{n_t}{1-\beta_1^t} \; ve \; m'_t = \frac{m_t}{1-\beta_2^t} \quad (2.18)$$

$$g'_t = \frac{\mu \, n'_t}{\sqrt{m'_t} - \epsilon} \quad (2.19)$$

$$x_t = x_{t-1} - g'_t \quad (2.20)$$

Burada $f_t(x)$ x parametreli stokastik amaç fonksiyonu, g_t t. adımdaki gradyanları, n_t birinci momenti, m_t ikinci momenti, $\beta_1, \beta_2 \in [0,1)$ ağırlıklandırma parametrelerini, μ öğrenme oranını, ϵ sıfıra bölünmeyi önlemek için küçük bir değeri, x_t ağırlık vektörünü (başlangıç parametre vektörü) temsil etmektedir. Problemin çözümünde optimum sonuçlar elde edebilmek için denklemde yer alan parametrelerin yaygın seçenekleri μ=0.001, ϵ=10^{-8}, β_1 = 0.9 ve β_2 = 0.999'dur. ADAM'ın temel adımlarından biri, hem birinci momenti hem de gradyanın ikinci momentinin bir tahminini elde etmek için üstel ağırlıklı hareketli ortalamaları kullanmasıdır.

(2.15) nolu denklemde t zaman adımında stokastik hedefe göre gradyanlar alınır. (2.16) nolu denklemde önyargılı ilk moment tahmini güncellenir. (2.17) nolu denklemde önyargılı ikinci moment tahmini güncellenir. (2.18) nolu denklemde önyargıya göre düzeltilmiş ilk moment (n'_t) ve ikinci moment tahminleri (m'_t) hesaplanır. (2.19) ve (2.20) nolu denklemlerle parametrelerin güncellenme işlemi tamamlanır.

2.2 ÖZELLİK SEÇME METOTLARI

Kullanılacak veri seti makine öğrenmesi için uygun hale getirilirse, o zaman sürece ilişkin akışı veya düzeni tespit etme daha kolay ve daha az zaman alıcı olacaktır. Veri setinin en uygun hale getirilmesi ise öğrenilecek görevle ilgisi olmayan veri setinin özelliklerinin süreçten çıkarılmasıdır. Bu süreç özellik seçme süreci olarak tanımlanmaktadır. Özellik seçme yöntemlerinin kullanılması sınıflandırıcı modellerinin anlaşılırlığını önemli ölçüde artırabilmekte ve genellikle daha iyi genelleyen bir model oluşturabilmektedir.

Özellik seçme yöntemleri gereksiz ve alakasız özellikleri ayırarak veri seti boyutunu optimize etmek için uygulanmaktadır. Genel olarak, başarılı bir özellik seçme yöntemi, sınıflandırma analizinde daha etkili öngörülebilirlik ve daha az işlem süresi ile performans kazanımları sağlar.

Özellik seçiminin temel amacı, daha yüksek sınıflandırma doğruluğu üreten bir özellik alt kümesini bulmaktır. Tan vd. (2005) ayrıca nitelik uzayındaki bir azalmanın daha anlaşılır bir modele yol açtığını ve farklı görselleştirme tekniklerinin kullanımını basitleştirdiğini belirtmektedir.

Alternatifler arasından en uygun özellik seçme yönteminin bulunması, sistemin genel doğruluk oranının başarısı için işletilmesi gereken bir süreçtir. Bu adımda özellik seçme yöntemlerinin başarısı veri sınıflandırması için seçilen algoritmalar ile birlikte değerlendirilmiştir. Burada sözü edilen başarı, yöntemin daha az özellik seçmesinden ziyade kurgulanan modele olan katkısını ifade etmektedir.

Bu çalışmada değerlendirilen özellik seçme yöntemleri, nitelik tabanlı ve alt küme tabanlı olarak adlandırılan iki yaklaşıma ayrılmaktadır. Nitelik tabanlı özellik seçme yöntemleri, her bir özelliği birbirinden bağımsız olarak ayrı ayrı değerlendirirken, alt küme tabanlı özellik seçim yöntemlerinde, özellikler arasındaki bağımlılıklar dikkate alınır ve tüm özellik kümesini en iyi şekilde temsil edecek şekilde alt kümeler oluşturulur.

Bu çalışmada, Korelasyon (CFS, Correlation-based Feature Selection) alt küme tabanlı, Tutarlılık (Consistency) alt küme tabanlı, Kazanç Oranı (Gain Ration) nitelik tabanlı ve Relief-F nitelik tabanlı özellik seçme yöntemleri, sınıflandırma algoritmalarına olan performans katkılarına göre değerlendirilmiştir.

Korelasyon tabanlı özellik seçiminin özellik alt kümesinin değerini ölçerken kullandığı yaklaşımda, her özelliğin sınıf etiketini tahmin etmekteki başarısının yanı sıra özelliklerin aralarındaki iç korelasyon değerlerini de dikkate almaktadır. Bu yaklaşım, iyi özellik alt kümelerinin tahmin edilecek ilgili sınıf ile yüksek, diğer taraftan küme içinde yer alan özelliklerin birbirleri ile düşük korelasyona sahip olması gerektiği hipotezine dayanır.

Tutarlılık ölçütü değerlendirmesinde bir özellik alt kümesi için tutarsızlık sayısı hesaplanır. Bu sayı verilerde görünme sayısı ile farklı sınıf özellikleri arasındaki en büyük sayının kullanılmasıyla elde edilir. Bir özelliğin tutarlılık katkısı belirlenmiş bir değerden düşükse özellik kaldırılır aksi takdirde ilgili özellik seçilir. Herhangi bir özellik alt kümesinin tutarlılığı, hiçbir zaman tam özellik kümesinden daha düşük olamaz.

Kazanç oranı özellik seçimi simetrik olmayan bir ölçüt olup, bilgi kazancının çok çeşitli değerlere sahip özellikleri seçme eğiliminin önüne geçmek için kullanılmaktadır. Kazanç oranı [0,1] aralığında değer almaktadır. Oran 1'e eşit olduğunda A bilgisinin tamamen B bilgisini tahmin edebildiğini, 0'a eşit olduğunda ise A ile B arasında hiçbir ilişki olmadığını gösterir.

Relief-F algoritması özelliklerin değerini aralarındaki bağımlılıkları ortaya çıkartarak bulmaktadır. Yöntemin mantığı komşuluk algoritmalarına benzemekte olup, özelliğin bulunduğu örneğin ait olduğu ve olmadığı sınıflarda yer alan en yakın örnekleri ağırlıklandırarak çalışmaktadır.

Bu tez çalışması kapsamında yapılan literatür araştırmasında bu özellik seçme yöntemlerinin URL özelliklerine dayalı oltalama saldırılarının tespitinde kullanıldığına dair bir araştırmaya rastlanmamıştır.

2.3 PERFORMANS METRİKLERİ

Sonuçların değerlendirilmesi aşamasında, kullanılan sınıflandırma sistemlerinin genel başarısını ölçüm metriklerinin değeri üzerinden göstermek amaçlanmıştır. Makine öğrenmesi

alanında hata matrisi olarak da bilinen karışıklık matrisi, genellikle denetimli bir öğrenme algoritmasının performansının görselleştirilmesine izin veren özel bir tablo düzenidir. Hata matrisi veya karışıklık matrisi ismi, sistemin test edilen iki sınıfı karıştırıp karıştırmadığını görmeyi kolaylaştırmasından kaynaklanmaktadır. Değerlendirmede kullanılan karışıklık matrisi ve ölçüm metrikleri Tablo 2.1'de gösterilmiştir.

Tablo 2.1: Karışıklık Matrisi

		GERÇEK DURUM	
		Pozitif	Negatif
TEST SONUCU	Pozitif	**Gerçek Pozitif (TP)**	**Yanlış Pozitif (FP)**
	Negatif	**Yanlış Negatif (FN)**	**Gerçek Negatif (TN)**

Test edilmek istenen sınıfların karşılaştırmasını yapmak için yaygın olarak kullanılan birçok değerlendirme ölçütü vardır. Tablo 2.1'den bir sınıflandırıcının performansını değerlendirmek için hata, doğruluk, duyarlılık, geri çağırma ve F metrikleri aşağıdaki şekilde tanımlanmaktadır.

Hata oranı genel olarak sınıflayıcının ne sıklıkta yanlış tahmin ettiğinin bir ölçüsüdür ve 2.21 nolu denklem ile elde edilir.

$$\text{Hata} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (2.21)$$

Doğruluk, tüm testlerde doğru tespit edilen vakaların sayısıdır. 2.22 nolu denklem ile tanımlanan doğruluk, doğru tespitlerin tüm tespitlere oranlanması ile elde edilir.

$$\text{Doğruluk} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (2.22)$$

2.23 nolu denklem tarafından tanımlanan duyarlılık, doğru tespit edilen şüpheli vakaların şüpheli olarak tahmin edilen tüm vakalara oranıyla elde edilir.

$$\text{Duyarlılık (Kesinlik)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (2.23)$$

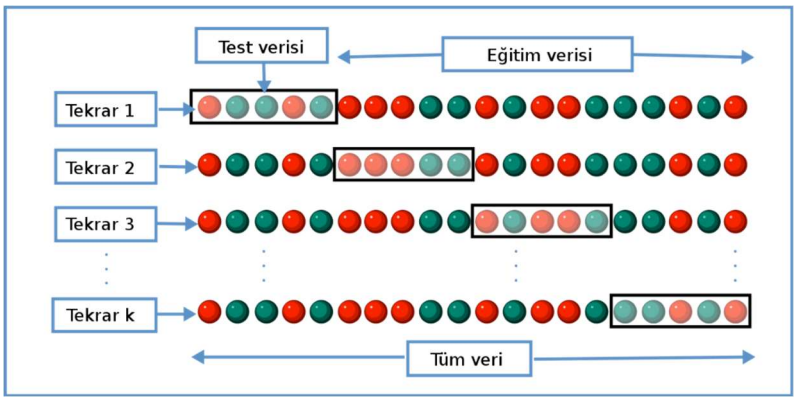
2.24 nolu denklem ile tanımlanan geri çağırma, doğru tespit edilen şüpheli vakaların tüm gerçek şüpheli vakalara oranı ile elde edilir.

$$\text{Geri Çağırma} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (2.24)$$

Duyarlılık ve geri çağırmanın harmonik ortalamasını ölçen istatistiksel bir yöntem olan F metrik değeri de 2.25 nolu denklemdeki gibi hesaplanmaktadır.

$$\text{F} = 2 \times \frac{\text{Duyarlılık} \times \text{Geri Çağırma}}{\text{Duyarlılık} + \text{Geri Çağırma}} \qquad (2.25)$$

Çapraz doğrulama yöntemi ise tek bir veri kümesinden birden çok örnek oluşturarak beklenen tahmin hatasını öngören yöntem olarak bilinmektedir. Çapraz doğrulama gerçekleştirmenin en bilinen yaygın yolu k-kat çapraz doğrulama olarak tanımlanmaktadır. K-kat çapraz doğrulama, makine öğrenmesi modeli oluşturmada kullanılan ve çapraz doğrulamanın verinin farklı kısımları kullanılarak k kez gerçekleştirildiği en yaygın doğrulama yaklaşımıdır. Öncelikle eğitim verileri k ayrık sete bölünür. Bölünen setlerden birisi doğrulama için kullanılırken kalanlar eğitim için kullanılmaktadır. Sonraki aşamada tüm k doğrulamalarının performans ortalaması alınır ve bu ortalama kurulan modelin performansı olarak kabul edilir. Literatürde tespit edilen ve en yaygın kullanılan k değerleri 5 ve 10'dur.



Şekil 2.4: Çapraz Doğrulama
(https://tr.wikipedia.org, 2020)

2.4 LİTERATÜR TARAMASI

Bu kapsamda incelenen ve yararlanılan kaynaklar Google Scholar ve Scopus veri tabanlarında, farklı anahtar kelimeler kullanılarak tez süreci boyunca yapılan taramalar sonucunda elde edilmiştir. Tarama sürecinde; “machine learning”, “phishing website”, “detection”, “phishing URL”, “deep learning” anahtar kelimeleri birleştirilerek kullanılmıştır.

Aburrous vd. (2010), beş farklı veri madenciliği algoritmasıyla (C4.5, JRip Ripper, PART, PRISM ve CBA) birleştirilmiş bir bulanık mantığa dayanan ve elektronik bankacılık için hazırlanmış sahte oltalama sitelerinin tespiti kurgulanan esnek katmanlı bir sistem sunmuştur. Bu yaklaşım, altı kriterin toplam 27 özelliğine dayanmaktadır: (i) URL ve Etki Alanı Kimliği (ii) Güvenlik ve Şifreleme (iii) Kaynak Kodu ve Java Betiği (iv) Sayfa Stili ve İçeriği (v) Web Adres Çubuğu ve (vi) Sosyal İnsan Faktörü. Bu çalışmadaki değerlendirme üç adımda yapılmıştır. (1) Bulanıklaştırma – her özellik için yüksek, orta, düşük vb. gibi bir sözel değer atanır ve girdilerin geçerli aralıkları bulanık kümelere bölünür; (2) Sınıflandırma algoritmalarını kullanarak kural oluşturma - oltalama saldırısı amacıyla oluşturulmuş sahte internet sitelerinin arşivlenmiş verilerinde önemli oltalama özelliklerini elde etmek için veri madenciliği sınıflandırma ve ilişkilendirme kurallarını kullanma (APWG arşivinden ve PhishTank'tan toplam 606 oltalama internet sitesi elde edilmiştir); ve (3) Durulaştırma (defuzzification) - belirsiz bir çıktının, bir internet sitesinin Kesin Yasal, Yasal, Şüpheli, Sahte veya Kesin Sahte olup olmadığını belirleyen bir skaler değere dönüştürülmesi. Yazarlar bu çalışmada test modu olarak 10 kat çapraz doğrulama kullanmışlar ve P.A.R.T. ile %86.381 algılama doğruluğu elde etmişlerdir.

Alkhozae ve Batarfi (2011), oltalama saldırısı amacıyla oluşturulmuş sahte sitelerin tespit edilmesi amacıyla https, resimler, şüpheli URL'ler, etki alanı, IFrame, komut dosyası ve açılır pencereler dâhil olmak üzere internet sayfası kaynak kodunu kontrol etmeye dayanan bir yöntemi önermiştir. Yazarların yaklaşımı, internet sayfası kaynak kodunun her satırında bir oltalama özelliğiyle karşılaşılması durumunda ilgili güvenlik ağırlığının azaltılmasıyla elde edilen nihai güvenlik ağırlığına dayalı olarak söz konusu internet sitesinin güvenlik yüzdesinin hesaplanmasını içermektedir. Ancak araştırmacılar, yöntemlerini gerçek bir veri seti üzerinde test etmemişlerdir; dolayısıyla, algoritmalarının doğruluğu ve başarı oranı da hesaplanmamıştır.

Basnet vd. (2011), gözlemlenen her paketi bir dizi kuralla karşılaştırarak ağları izlemeyi içeren bir yaklaşımdan esinlenen, kurala dayalı bir oltalama tespit yöntemi önermiştir. Yazarlar, Karar Ağacı (DT) ve Lojistik Regresyon (LR) öğrenme algoritmaları ile geçici veri setlerine uyguladıkları makine öğrenmesi özelliklerini kullanarak oltalama saldırısı gerçekleştirenler tarafından kullanılan çeşitli tekniklerin ve püf noktalarının sezgisel gözlemlerine dayanarak kural setlerini oluşturmuşlardır. Kurallarını aşağıdaki kategorilerde gruplamışlardır: (a) Arama Motoru Tabanlı Kurallar: URL'nin listelenip listelenmediğini kontrol etmek için web sayfalarını tarama (Google, Yahoo! ve Bing'den en iyi 30 sonuç) (b) Kırmızı İşaretli Anahtar Kelime Tabanlı Kural: URL'deki kelimelerden herhangi birinin, eğitim veri kümelerinden oluşturdukları, oltalama saldırganları tarafından sıkça kullanılan 62 kelimelik listede olup olmadığını kontrol etme (c) Gizlemeye Dayalı Kurallar: Oltalama saldırganları tarafından URL'leri gizlemek için yaygın olarak kullanılan "-", "_", "@" gibi belirli karakterlerin mevcut olup olmadığını kontrol etme (d) Kara Listeye Dayalı Kural: internet sayfasının Google Güvenli Tarama API'si tarafından kara listeye alınıp alınmadığını kontrol etme (e) İtibara Dayalı Kural: URL'nin en popüler oltalama siteleri olarak listelenip listelenmediğini veya IP'sinin veya etki alanının PhishTank, StopBadware.org, vb. tarafından rapor edilip edilmediğinin kontrol edilmesi ve (f) İçeriğe Dayalı Kurallar: internet sayfasının HTML içeriğinin TLS/SSL kullanmadan yapılan parola giriş alanının incelenmesi, dahili bağlantılardan daha fazla harici bağlantı olması, kötü HTML biçimlendirmeleri, kara listede bulunan bir URL'ye sahip IFrame, kara listede bulunan meta etiketi ve harici etki alanı kontrolü. Yazarlar, kural tabanlı yaklaşımı kullanmanın temel faydalarından birinin, sürekli değişen oltalama taktiklerini tespit etmek için gerektiğinde kuralların kolayca uyarlanabileceğine dikkat çekmişlerdir. Yaklaşımlarını 40.000'den fazla oltalama ve yasal URL üzerinde test etmişler ve %95'den fazla doğruluk elde etmişlerdir.

Chen vd. (2011), saldırının risk düzeyini ve şirketin oltalama uyarılarının neden olduğu potansiyel mali kaybını değerlendirerek bir oltalama saldırısının ciddiyetini tahmin etmeye çalışan hibrit bir yöntem önermiştir. Yazarların hibrit yaklaşımları, oltalama uyarılarından elde edilen metin verilerini ve hedeflenen şirketin finansal verilerini kullanan, metin çıkarma ve denetimli sınıflandırma veri madenciliği yöntemlerini birleştirmektedir. Ham veri setinde bulunan toplam 168 finansal değişkenden, sınıflandırma amacıyla ilk 25 değişken seçilmiştir. Yazarlar 3 × 3 × 2 deneysel bir tasarım kullanmışlardır: (i) üç set girdi verisi (metinsel, mali ve ikisinin birleşimi) (ii) üç sınıflandırıcı - karar ağacı (DT), destek vektör makineleri (SVM) ve

sinir ağı (NN) ve (iii) iki sınıflandırma görevi - risk seviyesi ve toplam anormal getiri. Bir oltalama uyarısının şirketin değeri üzerindeki etkisini anında ölçtüğü için toplam anormal getiriyi kullanmışlardır. Modellerinin performansı 10 kat çapraz doğrulama kullanılarak test edilmiştir. Elde edilen en iyi sınıflandırma doğruluğu, birleştirilmiş metinsel ve finansal verilerin kullanıldığı karar ağacı modeliyle %89.18'e kadar çıkmıştır.

Khonji vd. (2011), belirli bir URL'den tüm alan adlarının çıkarılmasına ve tespit edilen tüm alan adlarının sayfa sıralarının arama motorları kullanılarak hesaplanmasına dayanan basit bir sezgisel oltalama saldırısı sınıflandırma yöntemi önermiştir. Sezgisel olarak algılanan alan adlarının belirli bir farkla gerçek alan adından daha büyük bir sıraya sahip olması durumunda ve gerçek alan adının sıralaması belirli bir eşiğin altındaysa, URL bir oltalama URL'si olarak sınıflandırılır. Yazarların yaklaşımları, oltalama URL'lerinin çoğunun sıralamalarını yeterince yükseltecek kadar uzun yaşamadıkları gerçeğine dayanıyor. Sezgisel alan adı çıkarma işlemi, belirli bir URL'nin sözcüksel olarak analiz edilmesiyle ve dizenin herhangi bir bölümünün bir Üst Düzey Alan Adları (TLD) ile eşleşip eşleşmediğini kontrol etmek için önceden tanımlanmış bir TLD listesi kullanılarak gerçekleştirilir. İki adet ana bilgisayar etki alanı (host) çıkarma modu vardır: (i) katı - sıralama, tüm alt etki alanları dâhil olmak üzere tam ana bilgisayar adına göre hesaplanır ve (ii) gevşek - alt alanlar hariç tutulur (daha yüksek sıralar üretir). Amaçları, mevcut çözümlerin sınıflandırma verimliliğini artırmak ve yanlış pozitif oranları azaltmaktır. Yöntemlerini PhishTank'tan toplanan 60.000'den fazla URL'de ve 8 gönüllüden toplanan 160.000 URL'de test etmişler ve gevşek mod için %62.9 ve katı mod için %83.31 oltalama saldırısı algılama doğruluğu elde etmişlerdir. Bu çalışmanın en büyük dezavantajını, önerilen çözümün HTTP istemcileri üzerinde verimsiz olması, çünkü arama motorları kullanıldığı için mevcut internet veriminden etkilenmesi olarak ifade etmişlerdir.

Khonji vd. (2011), simge ayırıcı olarak herhangi bir karakterin yanı sıra büyük/küçük harf duyarlı karakter dizileri, alt çizgi (_) ve kısa çizgi (-) olarak tanımladıkları URL simgelerinin sözcüksel analizine dayanan bir web sitesi sınıflandırma yöntemi önermiştir. Yazarların yaklaşımlarının arkasındaki fikir, yeni oltalama URL'lerini daha önceki oltalama URL'lerinin sözcüksel analiziyle tanımlamanın mümkün olmasına dayanmaktadır. Çünkü bu sitelerin daha önce görülmemiş olan simgelere göre aynı simgeleri yeniden kullanma olasılıkları çok daha yüksektir. Ayrıca oltalama saldırısı amacıyla oluşturulan sahte sitelerde daha önce görülen simgelerin yasal URL'lerde görülme oranı da çok küçüktür. Yazarla ne kadar küçük veya

belirsiz olursa olsun simgelerin göreceli konumlarını dikkate almışlardır, ancak hem oltalama hem de yasal internet sitelerinde ortak kullanıldığını düşündükleri tüm Üst Düzey Etki Alanlarını (TLD'ler) hariç tutmuşlardır. Simgeler gerçek sayıların atandığı "oltalama" göstergeleri olarak kabul edilir. İlk öğrenme aşaması, her bir simgenin iki boyutlu bir vektör olarak temsil edildiği ve bir sınıflandırma modeli çıkarabilmesi için önceden tanımlanmış sınıfla birlikte istatistiksel sınıflandırıcıya gönderildiği, hem sahte hem de yasal URL'ler içindeki simge dağılımının sözcüksel analizini içermektedir. Bundan sonra sınıflandırılmamış URL'ler, şüpheli URL'lerde her bir simgenin oluşumlarını hesaplayarak çıktı olarak tahmin edilen ikili sınıftan birini (Sahte veya Yasal) döndüren sınıflandırıcıya beslenir. Tüm simgelerinin dolandırıcılık değerlerinin ortalaması olan bir URL'nin toplam dolandırıcılık değeri belirli bir eşiği aşarsa, şüpheli bu URL sahte bir internet sitesi olarak sınıflandırılır. Yazarlar, yöntemlerini PhishTank ve Halifa Üniversitesi'nin HTTP erişim kayıtlarından toplanan 70.000'den fazla yasal ve oltalama URL'si üzerinde değerlendirmiş ve yöntemlerinin sınıflandırma doğruluğunun %97,31 olduğunu ifade etmiştir.

Martin vd. (2011), (i) URL ve Etki Alanı Kimliği (ii) Güvenlik Şifrelemesi (iii) Kaynak Kodu ve Java betiği (iv) Sayfa Stili ve İçeriği (v) Web Adres Çubuğu ve (vi) Sosyal İnsan Faktörü şeklinde altı kıstas altında gruplandırılabilen 27 özellik ve göstergeye dayanan Sinir Ağları tekniklerini kullanan yeni bir elektronik bankacılık oltalama internet sitesi algılama modeli çerçevesi önermektedir. Yöntemleri, rastgele ağırlıklarla bir Sinir Ağının başlatılmasını ve ardından arşivlenmiş verilerden makine tarafından anlaşılabilir formatta bir dizi girdi ile eğitilmesini içermektedir. Bu çalışmada kullanılan algoritma, her aşamada sırasıyla çıktının kontrol edildiği ve ağırlıkların buna göre ayarlandığı çok katmanlı bir Sinir Ağlarında ağırlıkları güncellemek için mevcut en iyi hipoteze dayanan bir algoritmadır. Bu yöntemin tahmin hata oranını azaltacağını ve bir Sinir Ağlarının doğası nedeniyle daha iyi bir sınıflandırma sunacağını öngörmüşlerdir fakat sonuç bildirmemiş ve diğer yöntemlerle bir karşılaştırma yapmamışlardır.

Shahriar ve Zulkernine (2011), mevcut oltalama tespit yaklaşımlarını en yaygın beş bilgi kaynağına göre analiz etmiş ve sınıflandırmışlardır: beyaz liste, kara liste, hibrit, bağımsız ve rastgele. Amaçları, yeni oltalama URL'lerini tespit etmek için mevcut oltalama önleme yaklaşımlarının başarısızlığı gibi sorunların ele alınmasına yardımcı olacak bu yöntemlerin kapsamlı bir analizini yapmaktır; yani, oltalamayı önleme konusunda hala çözülmemiş olan

sorunların mevcut olduğu düşünülmektedir. Çalışmaları, beyaz listeye alınmış bilgi tabanlı yaklaşımların, kullanıcı tarafından sağlanan bilgiler oldukça küçük olduğunda yararlı olduğunu, ancak çok büyük miktarlarda bilgi depolamayı gerektirdiğini ve dinamik internet sayfaları için pratik olmadığını göstermektedir. Diğer taraftan, kara listeye alınmış bilgiye dayalı yaklaşımlar, etkin tespit için zamanında güncel bilgilerin paylaşılmasını gerektirir ki bu pratikte böyle olmayabilir. Bununla birlikte, hibrit yaklaşımlar, her iki yaklaşımın avantajlarını birleştirdikleri için çok daha iyi performans ve doğruluğa sahiptir, ancak aynı zamanda beyaz ve kara listedeki bilgilerin güncelliği ile ilgilenmeleri gerekir. Rastgele bilgilere dayalı yaklaşımlar, oltalama sitelerinin yalnızca sınırlı sayıda rastgele kimlik bilgisi gönderimine izin verdiği durumlarda etkili değildir. Kısacası, sonuçlar, farklı bilgi kaynaklarını birleştirmenin genel olarak daha iyi koruma sağladığını göstermektedir.

Xiang vd. (2011), makine öğrenmesi tekniklerini, HTML Belge Nesne Modeli'ni (DOM), arama motorlarını, üçüncü taraf hizmetlerini ve gerçek pozitif oranın iyileştirilmesi hedefiyle 16 internet sayfası özelliğini kullanan CANTINA+ adlı katmanlı bir oltalama saldırısı algılama sistemi geliştirmişlerdir. CANTINA+ üç ana modülden oluşmaktadır: (1) internet sitesinin bilinen oltalama saldırılarına benzerliğini inceleme; (2) sınıflandırma aşamasından önce, hassas bilgiler isteyen giriş formları olmayan internet sitelerini filtrelemek için sezgisel yöntemler kullanma ve (3) üç kategoride düzenlenmiş 16 özelliği kullanarak makine öğrenmesi tekniklerini kullanma: URL (6 özellik), HTML içeriği (4 özellik) ve internet sitesi hakkında bilgi için web'de arama (6 özellik). Sistemi 8118 oltalama ve 4883 yasal internet sitesi üzerinde test etmişlerdir. CANTINA+, benzersiz testlerde %92'nin üzerinde gerçek pozitif başarı oranı elde etmiştir.

Balamuralikrishna vd. (2012), iki aşamaya dayalı bir oltalama saldırısı önleme yöntemi geliştirmiştir: URL Etki Alan Kimliği ve Görüntü Tabanlı Web Sayfası Eşleştirme. Yazarlar bir web sayfasının 3 özelliğini göz önünde bulundururlar: (i) metin parçaları ve stilleri (ii) sayfaya gömülü resimler ve (iii) sayfanın genel görsel görünümü. URL etki alanlarını ve IP adreslerini tanımlamak için bölme kuralı yaklaşımını ve yaklaşık dize eşleştirme algoritmasını kullanmışlardır. Eğer IP adresleri birbirinden farklıysa, şüpheli web sayfasının anlık görüntüsü, görüntü eşleştirme için ikinci aşamaya aktarılır. İkinci aşamada, anlık görüntünün göze çarpan noktalarını hesaplamak için köşe algılama yöntemi kullanılır. Daha sonra yasal web sayfasının aynı özellikleriyle karşılaştırılan sahte web sayfasının özelliklerini çıkarmak için Kontrast

Bağlam Histogram (CCH) tanımlayıcısı kullanılır. Bu eşleştirme sırasında seçilen eşik seviyesi aşılırsa, şüpheli internet sitesi bir oltalama saldırısı amacıyla oluşturulmuş bir internet sitesi olarak sınıflandırılır. Yazarlar, yöntemlerinin diğer mevcut araçlardan daha iyi performans gösterdiğini iddia ediyorlar, ancak kanıt olarak herhangi bir test sonucu sunmamışlardır.

Basnet vd. (2012), gerçek oltalama veri kümeleri üzerinde 177 özelliği (38 içerik tabanlı ve 139 URL tabanlı) kullanarak oltalama saldırısı tespiti için yaygın kullanılan iki adet özellik seçme tekniğini (korelasyon tabanlı ve sarmalayıcı tabanlı) karşılaştırmışlardır. Amaçları, etkili bir özellik seçme yöntemini keşfetmek ve Naïve Bayes, Lojistik Regresyon ve Rastgele Orman gibi sınıflandırma modellerinin oltalama saldırısı tespit doğruluğunu ve eğitim sürelerini önemli ölçüde iyileştirebileceğini göstermektir. Deneylerinde, PhishTank'a ait 16.000'den fazla oltalama internet sitesini ve Yahoo! ve DMOZ dizinlerinden 32.000'den fazla yasal internet sitesini kullanmışlardır. Sonuçlar, seçilen daha küçük alt kümeler yerine tam özelliklerin kullanılmasının daha iyi sonuçlar verdiğini göstermiştir. Ayrıca, sarmalayıcı tabanlı teknik, daha yavaş sınıflandırıcılar için son derece yavaş olduğu kanıtlanmış olsada, korelasyon tabanlı özellik seçimine kıyasla sınıflandırıcı doğruluklarını önemli ölçüde geliştirmiştir.

Basnet ve Sung (2012), farklı web servislerini kullanarak URL'lerden meta verileri madencilik yaparak çıkarma ve bunları URL'lerin gerçek zamanlıya yakın değerlendirmesini sunduğu Lojistik Regresyon (LR) sınıflandırıcısı için özellikler olarak kullanarak bir sezgisel oltalama saldırısı amaçlı oluşturulan URL tespit şeması geliştirmiştir. Meta veri toplama için çalışmada kullanılan web madenciliği tabanlı sezgisel yöntem, en önde gelen üç arama motorunu (Google, Yahoo! ve Bing) ve aynı zamanda güvenilir kaynaklar tarafından yayımlanan geçmiş raporları, istatistikleri ve kara listeleri içermektedir. Sezgisel yöntemleri, yasal bir internet sitesinin tümü olmasa da en az bir büyük arama motorunun ilk 30 arama sonucunda listelenecek kadar olacağı varsayımına dayanır; dolayısıyla, bu amaçla web tarama ve filtreleme tekniklerini kullanırlar. İtibar tabanlı sezgisel yöntemleri, PhishTank.com'dan (İlk 10 Alan Adı, İlk 10 IP ve En Popüler 10 Hedef), StopBadware.org (bildirilen URL'lerin ilk 50 IP adresi), hpHosts ve Google Güvenli Tarama API'sinden (URL'leri Google'ın sürekli güncellenen kara listelerine göre kontrol etme) çeşitli istatistiklerin toplanmasını içermektedir. Yazarlar yaklaşımlarını, 16.000'den fazla oltalama URL'sinden ve 31.000 yasal URL'den oluşan gerçek veri kümesi üzerinde 10 kat çapraz doğrulama yöntemini kullanarak test etmişler ve %97,2'lik genel bir doğruluk oranı elde

etmişlerdir. Yöntemleri yalnızca URL'ye dayandığından, URL'nin gömülebildiği herhangi bir ortama uygulanabilmektedir.

Huang vd. (2012), 23 özellikten oluşan bir özellik vektörü üzerinde modellenen destek vektör makinesine (SVM) dayalı bir oltalama URL tespit yöntemi önermiştir: (i) URL yapısıyla ilgili 4 özellik - IP adresi, ana bilgisayar (host) adı uzunluğu, uzun bağlantıdaki <path> içindeki noktaların sayısı ve <hostname> içindeki tire sayısı (ii) 9 sözcüksel özellik - onayla, güvenli, giriş yap, oturum aç, http, bankacılık vb. gibi sözcükler ve (iii) 10 marka adı özelliği - eBay, PayPal, Sulake, Facebook, Orkut, Santander, Mastercard, Warcraft, Visa ve Bradesco. İlk eğitim aşamasında, özellikler uygun bir formatta bir özellik vektörü oluşturmak için çıkarılır ve bunlar daha sonra, SVM'nin ilgili URL'nin bir oltalama sayfasına ait olup olmadığına karar verdiği sınıflandırma aşamasında kullanılır.

Maurer ve Höfer (2012), oltalama saldırısı amacıyla oluşturulmuş sahte sitelerin tespit edilmesi için URL benzerliğine dayalı bir yaklaşım geliştirmişlerdir. Yöntemleri, dört farklı URL teriminin (temel alan adı, alt alan adı, yol alan adı ve marka adı) çıkarılmasını ve bu adların bilinen arama motorlarının yazım düzeltme ve belirsiz bir sorgu durumunda aranan kelimeye en yakın olacak şekilde olası orijinal internet sitesinin adını arama önerisi yeteneğini kullanarak doğrulanmasını içermektedir. Önerilen arama motoru destekli yöntemi 8370 gerçek oltalama saldırısı URL'inden oluşan geniş bir sette test etmişlerdir ve %54,3 oltalama saldırısı URL tespit başarı doğruluğu sonucunu elde etmişlerdir; bu da onları, bu yöntemin kendi başına bir oltalama saldırısı tespit mekanizması olarak yeterli olmadığı, bunun yerine diğer yöntemlerle birlikte kullanılması gerektiği sonucuna varmalarına yol açmıştır. Buna ilave olarak, URL'de beklenmedik konumlarda düzgün yazılmış alan adlarının daha gelişmiş yöntemlerle çıkarılmasının ve daha iyi bir marka adı denetleyicisinin de yöntemlerini iyileştirebileceğine dikkat çekmişlerdir.

Sadi vd. (2012), URL ve HTML kodundaki belirli özellikleri kontrol etmeye dayalı bir oltalama saldırısı tespit yöntemi önermiştir. Önerilen yöntem şu adımlara dayanmaktadır: ilk olarak, verilen URL içindeki nokta ve "@" simgelerinin sayısı sayılır. Eğer bu sayılardan herhangi biri 5'ten büyükse, olası oltalama saldırısı uyarısı verilmektedir. Sonraki adım, URL'nin PhishTank internet sitesinde yer alan listede olup olmadığını kontrol etmeyi içermektedir. Eğer söz konusu URL listede bulunursa bu sitenin oltalama amaçlı oluşturulmuş bir site olduğu sonucuna varılmaktadır. Eğer söz konusu URL PhishTank listesinde yer almıyorsa, sonraki adımda URL

ve HTML kodları içerisindeki bağlantılar ve ilgili metinler çıkarılarak bunları yine PhishTank internet sitesi üzerinden karşılaştırılmaktadır. Eğer çıkarılan bağlantılardan herhangi biri PhishTank'ta bulunursa, söz konusu internet sitesi oltalama saldırısı amacıyla kurulmuş olarak değerlendirilmektedir. Çıkarılan bağlantılar PhishTank'ta bulunmuyorsa, önerilen algoritma IP tabanlı URL'leri aramakta ve eşleştirmeye çalışmaktadır. Eğer bir eşleşme bulunursa bu durum da oltalama amaçlı bir site olduğuna ilişkin gösterge olarak kabul edilir. Hiçbiri bulunmazsa, sonraki adım, HTML kaynak kodunda yer alan bağlantılar tarafından atıfta bulunulan sayfaları kaydetmeyi ve bu sayfaların oturum açma/kayıt sayfaları olup olmadıklarını ve https bağlantılı olup olmadıklarını kontrol etmeyi içermektedir. Eğer bu kontrollerden oturum açma/kayıt sayfaları bulunursa, elde edilen her sayfanın içeriği verilen URL'nin içeriğiyle eşleştirilmekte ve %90 eşleşmesi durumunda sayfanın oltalama amaçlı oluşturulduğu sayılmaktadır. Önerilen yöntemin hem çevrimiçi hem de çevrimdışı çalışmakta, ayrıca kullanıcıları kötü niyetli ve istenmeyen bağlantılardan korumakta olduğu ifade edilmiştir. Yazarlar tarafından önerilen yöntemin daha hızlı performans gösterdiği ve mevcut bilinen oltalama saldırısı tespit yöntemlerine kıyasla yanlış pozitif oranlarını azalttığı iddae edilmiştir. Buna karşın önerilen algoritmada oltalama saldırısı tespit doğruluğu %82 olarak bulunmuştur.

Santhana Lakshmi and Vijaya (2012), bir internet sitesinin URL ve HTML kodlarından internet sitesinin kimliğine ilişkin özelliklerin çıkarımını içeren, ilave olarak arama motoru, 'Whois' sorgusu ve oltalama saldırısı şüpheli internet sitelerinin 'Kara Liste' veritabanı gibi üçüncü parti servislerini de kullanan makine öğrenmesi yöntemlerine dayalı oltalama saldırısı amacıyla oluşturulmuş sahte sitelerin tahmin edilmesi sistemi geliştirmiştir. Öğrenme modeli, Çok katmanlı Algılayıcı, Karar Ağacı ve Naïve Bayes sınıflandırmasının denetimli öğrenme algoritmalarını kullanarak ve hem yasal hem sahte sitelerin özellikleriyle eğitilerek oluşturulmuştur. Modellerin 10 kat çapraz doğrulama tekniği kullanılarak değerlendirilmesinden sonra, Karar Ağacı sınıflandırıcısının, tahmin doğruluğu %97,5'e kadar ulaşan en iyi tahmin sonuçlarını verdiği ortaya çıkmıştır. Bu araştırmada özellik çıkarma ve seçme yöntemlerine ve veri setine ilişkin yeterli bilgi bulunmadığı gözlemlenmiştir.

Zhang vd. (2012), mantık regresyonuna dayalı istatistiksel makine öğrenmesi sınıflandırıcısını kullanan önceki oltalama URL tespit yöntemlerini iyileştirmek için geçmişe dayalı bir aktarım öğrenme yöntemi sunmaktadır. Özelliklerin dağılımı alanlar arasında oldukça farklı olduğundan, yazarlar gelişmiş tespit yöntemlerini tasarlarken bu farklılıkları da hesaba

katmaktadırlar. Katkıları, istatistiksel makine öğrenmesi algoritmasının mevcut veri kaynaklarını kullanarak 15 URL özelliğine dayalı sınıflandırma modellerini eğitmeyi bitirdikten sonra, aktarım öğrenme algoritmasını kullanarak eğitimli modelleri ayarlamayı ve bunları eşleşen alanlara yüklemeyi içermektedir. Yöntemlerini 193.175 oltalama ve oltalama olmayan URL'ler üzerinde yapılan bir dizi deney aracılığıyla değerlendirmişlerdir. Sonuçlar, önerilen yöntemin %97'den fazla doğruluk sağlayarak geleneksel makine öğrenmesi algoritmasından daha iyi performansa sahip olduğunu göstermiştir.

Zhuang vd. (2012), bir internet sayfasını temsil eden 10 özelliği (Başlık, <h1> - <h6> arası etiketlerdeki içerik, Anahtar kelime bilgileri, Sayfa açıklaması, Telif hakkı bilgisi, Bağlantı metni, URL adresi, Resmin URL adresi, Resmin açıklama metni ve sayfanın diğer tüm görünür dizisini) ve hiyerarşik kümeleme algoritması aracılığıyla oltalama saldırısının kategorilendirilmesinde kullanan IPDCM'yi (Intelligent Phishing Website Detection and Categorization Model - Akıllı Oltalama Web Sitesi Algılama ve Sınıflandırma Modeli) uygulamıştır. Modellerinin ana bileşenleri şunlardır: (i) Özellik Çıkarıcı (ii) Sınıflandırıcı Eğitim Modülü - eğitim için Naive Bayes Sınıflandırıcı ve SVM (Destek Vektör Makinesi) algoritması kullanır; (iii) Topluluk Sınıflandırma Modülü ve (iv) Küme Eğitim Modülü - ağırlıklandırma şeması ile frekans vektörleri üzerine hiyerarşik kümeleme algoritması uygulanır. Modellerini, Kingsoft Internet Security Lab'dan toplanan gerçek oltalama saldırısı internet sitelerinden oluşan büyük veri kümesinde test etmişler ve modellerinin, %97,72 duyarlılık doğruluğu elde ederek diğer popüler oltalama önleme yöntemlerinden ve araçlarından (Kaspersky Anti-phishing gibi) daha iyi performans gösterdiğini ifade etmişlerdir.

Aburrous ve Khelifi (2013), sınıflandırma kurallarını çıkarmak amacıyla oltalama saldırılarının veri özelliklerini ve modellerini işlemek için basit veri madenciliği ilişkisel sınıflandırma algoritmaları ile bulanık mantık modeli kullanan denetimli makine öğrenmesine dayalı bir oltalama elektronik bankacılık internet sitesi algılama sistemi geliştirmiştir. Bulanık akıl yürütme, kesin olmayan ve dinamik oltalama özelliklerini belirleme ve oltalama bulanık kurallarını sınıflandırma yeteneği sağlar. Sistemleri, altı kritere (URL ve Etki Alanı Kimliği, Güvenlik ve Şifreleme, Kaynak Kodu ve Java Komut Dosyası, Sayfa Stili ve İçeriği, Web Adres Çubuğu ve Sosyal İnsan Faktörü) dayalı 27 oltalama internet sitesi özelliği ve deseni çıkarır ve oltalamaya olan korunmasızlığı çıkarılan her bir özelliğe karşılık gelen bulanık değişkenlerle (Yüksek, Orta ve Düşük) çapraz kontrol ederek belirtilen bulanık kümelere dayalı doğrular.

Yazarlar, sistemlerini üç katman halinde tasarlamış ve işlem süresini optimize etmek için bir budama tekniği kullanmışlardır. Yazarlar, yaklaşımlarını değerlendirmek için bir araç çubuğu eklentisi tasarlamışlardır ve 160 farklı elektronik bankacılık internet sitesinden oluşan bir örnek kullanarak test etmişlerdir. Sonuç olarak düşük yanlış pozitif alarmlarla %86 tespit doğruluğuna ulaşmışlardır.

Barraclough vd. (2013), Bulanık Mantık ve Sinir Ağı'nı birleştirerek hem sayısal hem de sözel özellikleri kullanan, çevrimiçi işlemler için bir hibrit Neuro-Fuzzy sahte internet sitelerini algılama ve bu sitelerden korunma sistemi geliştirmiştir. Bu çalışmada oltalama sitelerinin gerçek zamanlı olarak tespit edilmesine olanak tanıyan oltalama tekniklerini ve stratejilerini temsil eden girdiler kullanılmıştır. Önerilen bu modelde 2 katlı çapraz doğrulama tekniği kullanılmıştır. Elde edilen sonuçlar, modelin doğruluğunun yüksek olduğunu (%97,5) göstermiş ve oltalama, şüpheli veya yasal internet siteleri arasında doğru ayrım yapılmasına izin vermiştir.

Li vd. (2013), web görüntüsü özelliklerini ve belge nesne modeli (DOM) nesnelerini ve özelliklerini hesaba katan bir tür yarı denetimli öğrenme algoritması olan dönüştürücü destek vektör makinesine (TSVM) dayalı bir oltalama tespit tekniği önermektedir. Web görüntüsü özellikleri şunları içermektedir: gri tonlamalı histogram, renk histogram ve alt grafikler arasındaki uzamsal ilişki. Diğer taraftan belge nesnesi modeli nesnelerinden çıkarılan özellikleri ise şunları içermektedir: İstek URL'si (burada farklı etki alanı istek URL'lerinin yüksek yüzdesi, oltalama olasılığının yüksek olduğunu gösterir), Bağlantı özellikleri (çok sayıda boş ve dış bağlantı, yüksek bir oltalama olasılığını gösterir), Form işleyici (harici bir siteye başvuran bir form işleyici tarafından kişisel bilgilerin istenmesi oltalamayı gösterir); SSL sertifikası (bir oltalama sayfasının sertifikasının, kimliğine bürünülen yasal sayfanın kimliğiyle eşleşmediği durumlarda); ve DNS kaydı (bir sayfadaki bağlantıların güvenilir sitelere atıfta bulunup bulunmadığını kontrol etmek için kullanılır). TSVM sınıflandırıcı, çıkarılan bu özelliklere göre internet sayfasını oltalama veya yasal olarak sınıflandırır. Yazarlar yaklaşımlarını PhishTank'tan elde edilen 200 oltalama sayfasında ve Google Whitelist'ten elde edilen 200 yasal sayfada test etmişlerdir. Sonuçlar, önerilen yöntemin oltalama saldırısı tespitinde %95.5 oranında doğruluğa ulaştığını göstermiştir.

Mohammad vd. (2014) tarafından yapılan çalışmada, yapay sinir ağlarına, özellikle de kendi kendini yapılandıran sinir ağlarına dayanan oltalama saldırılarını tahmin etmek için akıllı bir

model önerilmiştir. Oltalama web sayfalarının türünü belirlemede önemli özelliklerin sürekli olarak değiştiği sorunuyla başa çıkabilmek için ağ yapısını sürekli iyileştirilmesi gerekmektedir. Bu nedenle önerilen model, ağı yapılandırma sürecini otomatikleştirerek bu sorunu çözmekte ve gürültülü veriler için yüksek kabul, hata toleransı ve yüksek tahmin doğruluğu göstermektedir. Bu sinir ağı modeli ilk önce, gizli katmanın yalnızca bir nörona sahip olduğu en aza indirgenmiş üç katmanlı bir sinir ağı kurmakta ve ardından model eğitimi üzerine geri bildirim yoluyla gizli katman nöronlarını kademeli olarak arttırmaktadır. Oltalama sitesi tespitinde yapılan deneyler sonucunda %92,18 doğruluğa ulaşıldığı gözlemlenmiştir. Sonuçlardan, üretilen tüm yapıların yüksek genelleme kabiliyetine sahip olduğunu görülmüştür.

Kazemian ve Ahmed'in (2015) yaptığı çalışmada kötü amaçlı web sayfalarını tespit etmek için K-en yakın komşular, Destek Vektör Makineleri ve Naive Bayes gibi üç denetimli makine öğrenme tekniği ile K-ortalamalar ve Yakınlık Yayılımı gibi iki denetimsiz makine öğrenme tekniği kullanılmıştır. Tüm bu makine öğrenimi teknikleri, çok sayıda kötü amaçlı ve güvenli web sayfasını analiz etmek için tahmine dayalı modeller oluşturmak için kullanılmıştır. Deney sonuçlarında, denetimli teknikler için %98'e varan doğruluk ve denetimsiz teknikler için %96'ya yakın siluet katsayısı üretilmiştir.

Islam ve Nihad (2016) tarafından yapılan çalışmada Naive Bayes, Destek Vektör Makinesi, Sinir Ağı, Rastgele Orman, K-en yakın komşular ve Karar Ağacı dahil olmak üzere farklı türde makine öğrenimi tabanlı sınıflandırma algoritmaları uygulanarak performansları doğruluk açısından ölçülmüş ve karşılaştırılmıştır. Deney veri seti 11055 web sitesi örneğinden oluşmakta olup her örnek 31 öznitelik içermektedir. Deney sonucunda Naive Bayes sınıflandırma algoritması için %93,09 ile en düşük sınıflandırma doğruluğu ve Random Forest algoritması için %97,47 ile en yüksek sınıflandırma doğruluğu elde edilmektedir. Karar ağacı için 95,61 Destek Vektör Makinesi için 94,84 Sinir ağı için 96,65 ve K-en yakın komşular için ise 97,07 doğrulukları elde edilmiştir.

Bahnse vd. (2017) yaptığı çalışmada oltalama sitesi tahmini için uygulanan makine öğrenmesi modelleri için girdi olarak URL'lerin kullanımı araştırılmıştır. Bu şekilde, bir özellik mühendisliği yaklaşımını ve ardından rastgele bir orman sınıflandırıcısını tekrarlayan sinir ağlarına dayanan yeni bir yöntemle karşılaştırılmıştır. Tekrarlayan sinir ağı yaklaşımının, manuel özellik oluşturmaya gerek kalmadan %98,07 doğruluk oranı sağladığı belirlenmiştir ve bu doğruluk oranı ile rastgele orman yöntemi %5 geride kalmıştır. Çalışma sonucunda, tam

içerik analizi gerektirmeyen ölçeklenebilir ve hızlı hareket eden proaktif bir algılama sisteminin mümkün olduğu anlaşılmıştır.

Zhang ve Li'nin (2017) yaptığı çalışma, kimlik avı önleme ve kontrolünün geliştirme geçmişini analiz etmekte ve kimlik avını tespit etmek için bir Borderline-Smote (Synthetic Minority Over-sampling Technique) DBN (Deeping Belief Network) yöntemi sunmaktadır. Yöntem, tanıma doğruluğunu %1 iyileştirmek için web belgeleri içeriğine ve diğer özelliklere dayalı derin öğrenme kimlik avı algılama yöntemini kullanır. Ayrıca çalışma, kimlik avı algılama eğitiminde dengesiz veri sorununu çözmek için Borderline-Smote'u kullanmakta ve F değeri ve geri çağırma oranını %2 daha da iyileştirmektedir. İlk olarak, kimlik avı sayfalarının üç grup özelliği çıkarılır; bunlar URL'nin, web sayfalarının ve görüntülerin özellikleridir.

Chen vd. (2018) tarafından yapılan çalışmada oltalama saldırılarını etkili bir şekilde tespit etmek için LSTM tekrarlayan sinir ağlarını kullanan yeni bir algılama sistemi tasarlanmıştır. LSTM'nin tercih sebebi veri zamanlamasını ve uzun vadeli bağımlılıkları yakalama avantajına sahip olması, güçlü bir öğrenme yeteneği bulunması, karmaşık yüksek boyutlu büyük veri karşısında güçlü bir potansiyeli olmasıdır. Çalışmada kimlik avı sayfaları toplanmakta ve ardından LSTM tekrarlayan sinir ağı algoritması 12 özellik ile eğitilmektedir. Deneysel sonuçlar, bu modelin diğer sinir ağı algoritmalarından daha yüksek doğruluğa ulaştığını göstermektedir.

Selvaganapathy vd. (2018) tarafından yapılan çalışmada kötü amaçlı URL'lerin tespiti ve sınıflandırılması için önerilen metodoloji, ikili sınıflandırma için derin sinir ağıyla özellik seçimi için yığılmış sınırlı Boltzmann makinesini kullanır. Birden fazla sınıf için, sınıflandırma için IBK-kNN, İkili İlişki ve SVM'li Etiket Güç Kümesi kullanılır. Önerilen yaklaşım 27.700 URL örneği ile test edilmiştir ve sonuçlar, derin öğrenmeye dayalı özellik seçimi ve sınıflandırma tekniklerinin ağı hızlı bir şekilde eğitebildiğini ve yanlış pozitifleri azaltarak tespit edebildiğini göstermektedir. DBN5model, "sigm" veya "tanh" aktivasyon fonksiyonları ve 100'den büyük ön eğitim ve alternatif Gibbs örnekleme yinelemeleri ile 0.8 öğrenme oranında %75'lik bir doğruluk oranı üretmiştir.

Vazhayil vd. (2018) makine öğrenmesi ve derin öğrenme algoritmaları ile oltalama internet sitesi URL'lerini tespit etmek için göreceli bir çalışma yapmıştır. Çalışmada CNN ve CNN-LSTM sınıflandırma modelinin mimarisi oluşturulmuştur. Sistem, makine öğrenmesi ve derin

öğrenme modeli için Keras ile birlikte TensorFlow gibi araçlar kullanılarak tasarlanmıştır. Veri kümesi, daha iyi ölçeklenebilirlik sağlamak için birden çok kaynaktan aktarılmıştır. Oltalama sitesi URL veri kümesi OpenPhish ve Phishtank'tan platformlarından elde edilirken, kötü amaçlı veya gereksiz internet sitesi URL'leri Malware Domains platformundan aktarılmıştır. Burada, klasik makine öğrenmesi tekniği lojistik regresyon, CNN gibi derin öğrenme teknikleri ve kötü amaçlı URL'leri tespit etmek için kullanılan mimariler olarak CNN-LSTM arasında karşılaştırmalı bir çalışma sunulmuştur. Karşılaştırma sonucunda, oltalama URL'lerinin sınıflandırılması için CNN-LSTM yaklaşık %98'lik doğruluk ile en iyi sonucu verdiği görülmüştür.

Yang vd. (2019) tarafından yapılan çalışmada derin öğrenmeyi kullanarak hızlı bir algılama yöntemine dayalı çok boyutlu bir oltalama tespit yaklaşımı önerilmektedir. Çalışmanın ilk adımında verilen URL'nin karakter dizisi özellikleri çıkarılmakta ve bu özellikler derin öğrenme ile hızlı sınıflandırma için kullanılmaktadır. Bu adım, üçüncü taraf yardımı veya oltalama hakkında herhangi bir ön bilgi gerektirmemektedir. İkinci adımda, URL istatistiksel özellikleri, web sayfası kod özellikleri, web sayfası metin özellikleri ve derin öğrenmenin hızlı sınıflandırma sonucu çok boyutlu özellikler halinde bileştirilmektedir. Yaklaşım, bir eşik belirlemek için algılama süresini azaltabilmektedir. Veri kümesi üzerindeki testte doğruluk %98,09'a ulaşmakta ve yanlış pozitif oranı ise %0,59 çıkmaktadır. Deneysel sonuçlar eşiği makul bir şekilde ayarlayarak, algılama verimliliğinin iyileştirilebileceğini göstermektedir.

Adebowale vd. (2020) tarafından yapılan çalışmada hem CNN (Convolutional Neural Network – Evrişimli Sinir Ağı) hem de LSTM (Long Short-Term Memory Network – Uzun Kısa Süreli Bellek Ağı) yöntemlerinin derin öğrenme algoritmasını kullanan hibrit yaklaşım benimsenmiştir. Hem CNN hem de LSTM'nin kombinasyonu, büyük bir veri seti ve daha yüksek sınıflandırıcı tahmin performansı sorununu çözmek için uygun görülmüştür. Bu nedenle, iki yöntemi birleştirmek, LSTM ve CNN mimarisi için daha az eğitim süresi ile daha iyi bir sonuca ulaşırken, görüntü, çerçeve ve metin özelliklerini model tespiti için de idealdir. Bu yüzden, CNN ve LSTM algoritması, Akıllı Oltalama Tespit Sistemi (IPDS) adı verilen karma bir sınıflandırma modeli oluşturmak için kullanılmıştır. Özelliğin türü, yanlış sınıflandırma sayısı ve bölünme sorunları gibi çeşitli faktörler göz önünde bulundurularak önerilen modelin hassasiyeti belirlenmiştir. Akıllı Oltalama Tespit Sisteminin büyük veri kümelerine uygulandığında oltalama internet sayfalarını ve saldırılarını algılamadaki etkinliğini

değerlendirmek ve karşılaştırmak için kapsamlı bir deneysel analiz gerçekleştirilmiştir. Deney sonuçları, modelin %93,28'lik bir doğruluk oranına ve 25 sn'lik bir ortalama algılama süresine ulaştığını göstermiştir.

Sountharrajan vd. (2020) tarafından yapılan çalışmada, Derin Boltzmann Makinesi (DBM), Yığınlanmış Otomatik Kodlayıcı (SAE) ve Derin Sinir Ağı (DNN) gibi derin öğrenme prosedürlerini kullanarak gerçek ve oltalama URL'lerinin farklı özelliklerini de araştırarak oltalama internet sitesi ayırt etme stratejileri önerilmiştir. DBM ve SAE, öznitelik belirleme için veri tasviriyle modelin önceden hazırlanması için kullanılmıştır; bunların arasında SAE, daha düşük yanlış sınıflandırma hatası yapmakta ve azalan bir öznitelik listesi içermektedir. Sahte veya gerçek bir URL olduğu bilinmeyen bir URL'yi ayırt etmede iki katlı gruplama için DNN kullanılmıştır. Önerilen çalışmada, diğer makine öğrenmesi stratejilerinden daha düşük yanlış pozitif oranıyla %94'lük daha yüksek doğruluk oranına ulaşılmıştır.

Butt vd. (2021) yaptığı çalışmada yasal ve oltalama URL'leri ayırt edilmesi için URL'lerin temel özelliklerinin keşfedilmesi üzerine yoğunlaşılmaktadır. Oltalama URL'lerinin örüntüsünün belirlenmesi için URL veri kümelerine derin öğrenme tekniği uygulanmıştır. Modelin eğitiminde çeşitli URL dizilerinde gradyanla güçlendirilmiş karar ağaçları algoritması ve Adam optimizasyon algoritmasının kullanıldığı derin sinir ağı uygulanmıştır. En sık bulunan örüntüler, sistemin oltalama URL'lerini algılamasına ve oltalamadan kaçınmasına yardımcı olmuştur. Model değerlendirmesi sonucunda, eğitilen modelin yaklaşık %92 doğruluk ve %94 f-skoru elde ettiği görülmektedir.

Lakshmi vd. (2021) yaptığı çalışma, ilgili web sitesindeki HTML sayfasının kaynak kodunda bulunan köprüleri kullanarak kimlik avı web sitelerini tanımlamak için yenilikçi bir yaklaşım sunmaktadır. Önerilen yöntem, kötü huylu web sayfalarını tespit etmek için 30 parametreli bir özellik vektörü kullanmaktadır. Bu özellikler, sahte web sitelerini gerçek web sitelerinden ayırmak için denetimli Derin Sinir Ağı modelini Adam optimizer ile eğitmek için kullanır. Adam optimizasyon algoritması ile önerilen derin öğrenme modeli, kimlik avı web sitelerini ve gerçek web sitelerini sınıflandırmak için Listwise yaklaşımını kullanır. Önerilen yaklaşımın performansı, SVM, Adaboost, AdaRank gibi diğer geleneksel makine öğrenmesi yaklaşımlarıyla karşılaştırıldığında daha iyi sonuçlara sahiptir. Sonuçlar, önerilen yaklaşımın kimlik avı web sitelerini tespit etmede daha doğru sonuçlar sağladığını göstermektedir. Ayrıca deneyler sonucunda ADAM, RMSPROP ve SGD optimizasyon algoritmaları için sırasıyla

%96, %94 ve %92 yaklaşık değerler elde edilmiş ve ADAM optimizasyon algoritmasının daha başarılı olduğu sonucuna ulaşılmıştır.

Mourtaji vd. (2021) çalışmasında, oltalama sorununu algılayabilen ve kontrol edebilen altı farklı algoritma modelini birleştirerek yeni bir hibrit kural tabanlı çözüm geliştirmek amaçlanmaktadır. Çalışma, kara liste yöntemi, sözlüksel ve ana bilgisayar yöntemi, içerik yöntemi, özdeşlik yöntemi, kimlik benzerliği yöntemi, görsel benzerlik yöntemi ve davranışsal yöntem olmak üzere altı farklı yöntemden çıkarılan 37 özniteliği içermektedir. Ayrıca, CART (Classification and Regression Trees), KNN (K-Nearest Neighbor), SVM (Support Vector Machine) içeren farklı makine öğrenmesi yöntemleri ile MLP (Multi Layer Perceptron - Çok Katmanlı Algılayıcı) ve CNN gibi derin öğrenme modelleri arasında karşılaştırmalı analiz yapılmıştır. Modellerin uygulanması için PhishTank platformundan oltalama URL'lerin ve Alexa platformundan da güvenli URL'lerin toplandığı bir veri seti oluşturulmuştur. Veri setinin %80'i eğitim ve %20'si test verisi olarak tanımlanmıştır. Güvenilir bir model elde etmek için on kat çapraz doğrulama yöntemi kullanılmıştır. Elde edilen bulgulara göre, makine öğrenmesi yöntemlerinden SVM için %91.132, KNN için %92.189 ve CART için %92.915 doğruluk elde edilirken, derin öğrenme yöntemlerinden en yüksek doğruluk değerine sahip olan CNN için %97.945, MLP için %93.216 doğruluk görülmektedir.

Srinivasan vd. (2021) yaptığı çalışmada, mevcut kötü amaçlı URL'lerin türevlerini tespit etmek için URL karakter seviyesi yerleştirme kullanılarak kodlanan CNN temelli DeepURLDetect (DURLD) modeli ve n-gramlı DNN modeli önerilmektedir. Karakter seviyesi yerleştirme, karakterleri sayısal biçimde temsil etmek için doğal dil işlemede (NLP) kullanılmaktadır. DeepURLDetect mimarisindeki gizli katmanlar URL'nin kötü niyetli olma olasılığını tahmin etmek için ReLU (Rectified Linear Units) kullanmaktadır. Çalışmada, kötü amaçlı URL algılaması için DeepURLDetect ve n-gramlı DNN yöntemleri, öğrenme oranı 0.001 olan ve 500 epoch'a kadar yürütülen deneyler yapılarak değerlendirilmiş ve optimal derin öğrenme tabanlı karakter seviyesi yerleştirme modeli seçilmiştir. Deneyler sonucunda elde edilen başarı oranı %93 ile %98 arasında bulunmuştur.

3.MALZEME VE YÖNTEM

Bu bölümde öncelikle çalışmada kullanılan veri setleri ve analizleri hakkında bilgi verilmiş, daha sonrasında çalışmada kullanılan yöntemlerin açıklaması yapılmıştır.

3.1 VERİ SETİ

Tez çalışmasının girdisini internet sitelerinin URL adres bilgilerinin bulunduğu veri setleri oluşturmaktadır. Bu çalışmada kullanılan ve karşılaştırılan özellik seçimli çoklu makine öğrenmesi ve derin öğrenme modelleri yurtdışı kaynaklı aynı veri setini kullanmaktadır. Bununla birlikte çalışma sonunda daha başarılı sonuç veren modelin Türkiye verisi ile de test edilmesi için ayrıca yurtiçi kaynaklı veri seti kullanılmıştır.

3.1.1 Yurtdışı Kaynaklı Veri Seti

Her iki modelde de kullanmak üzere elde ettiğimiz veri setimiz, dünya genelinde gerçekleşen oltalama saldırılarında en çok hedeflenen marka isimleriyle ilişkili oluşturulmuş sahte internet sitelerine odaklanmaktadır. En çok hedeflenen marka isimleri ve bu isimlere ait gerçek oltama saldırısı amacıyla oluşturulmuş sayfaların URL'leri PhishTank sitesinden (www.phishtank.com) elde edilmiştir.

OpenDNS tarafından işletilen PhishTank, internette yer alan sahte siteler hakkında veri ve bilgi paylaşımı amacıyla kurulmuş bir işbirliği platformudur. Bu platformda şüpheli bir internet sitesinin URL'i paylaşıldıktan sonra bu internet sitesinin oltalama saldırısı amacıyla oluşturulmuş bir internet sayfası olup olmadığı platforma kayıtlı kullanıcılar tarafından doğrulanmaktadır.

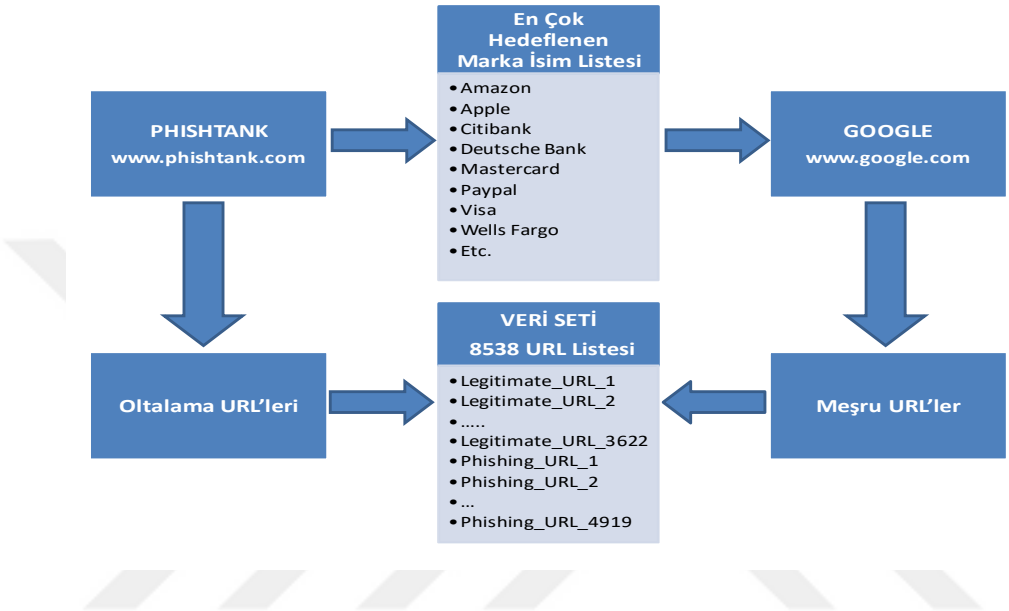
Oltalama saldırılarına en fazla hedef olan marka isimleri PhishTank sitesinden belirlenerek Tablo 3.1'de sunulmuştur.

Tablo 3.1: Oltalama Saldırılarında En Fazla Hedefte Olan Marka İsimleri

Allegro	Key Bank
Alliance Bank	Littlewoods
Allied Bank Limited	Lloyds Bank
Amazon	Mastercard
American Express	Microsoft

AOL	National Australia Bank
Apple	Nationwide Building Society
ASB Bank Limited	NatWest Bank
AT&T	Nets
Australia and New Zealand Banking Group	Orange
Banco De Brasil	PagSeguro
Bank of America Corporation	PayPal
Barclays Bank PLC	PKO Polish Bank
Blizzard	Poste Italiane
Bradesco	Rabobank
Caixa	Regions Bank
Capitec Bank	Royal Bank of Canada
Cariparma Credit Agricole	Royal Bank of Scotland
Cartasi	RuneScape
CIMB Bank	Safra Bank
Cielo	Santander UK
Citibank	South African Revenue Service
Commonwealth Bank of Australia	Standard Bank Ltd.
Co-operative Bank	Steam
Craigslist	Sulake Corporation
Deutsche Bank	Swedbank
Dropbox	TAM Fidelidade
eBay, Inc.	TD Canada Trust
Facebook	Tesco
Google	TSB
Groupon	Twitter
GuildWars2	Verizon Wireless
Halifax	Very
Her Majesty's Revenue and Customs	Visa
Hotmail / Live	Vodafone
HSBC Group	WalMart
ING Direct	Wells Fargo
Internal Revenue Service	Western Union
Itau	Westpac
JPMorgan Chase and Co.	Yahoo

En çok hedeflenen marka isimlerini belirledikten sonra, oltalama saldırısı amacıyla oluşturulmuş internet sayfalarına ilişkin veri seti oluşturmak amacıyla PhishTank internet sitesinden 4919 adet doğrulanmış sahte internet sitesi adresi indirilmiştir. (internet sitesi erişim adresi: http://www.phishtank.com/developer_info.php). Şekil 3.1'de hedeflenen markalar için veri toplama şeması gösterilmektedir.



Şekil 3.1: En Çok Hedeflenen Markalar İçin Veri Toplama Şeması

Bu marka isimlerinin kendi isimleri adına oluşturdukları meşru URL'lerini elde etmek için Google arama motoru kullanılmıştır. Bu çalışmada ilk olarak marka adları Google arama motoruna yazılarak arama çalıştırılmış ve arama sonuçlarından ortaya çıkan ilk bağlantı URL'si aranan meşru internet sitesinin ana sayfa bağlantısı olarak değerlendirilmiştir. Saldırılarda en çok hedeflenen markaların bu ana sayfa bağlantılarını aldıktan sonra, tekrar Google ile web taraması/sürünmesi yaparak söz konusu internet sitelerine ait daha fazla bağlantı alma çalışması yürütülmüştür. Bu çalışma sonucunda, 3622 meşru ve 4919 onaylanmış oltalama saldırısı URL'si olmak üzere toplam 8538 URL adresi toplanmıştır.

3.1.2 Yurtiçi Kaynaklı Veri Seti

Bu çalışmada önerilen özellik seçimli derin öğrenme modelinin ülkemize özel bir model olarak değerlendirilmesi için söz konusu model yurtiçi kaynaklı veri seti kullanılarak analiz edilmiştir. Bu kapsamda kullanılacak oltalama siteleri ile ilgili veri setini oluşturan URL adresleri Bilgi

Teknolojileri ve İletişim Kurumu bünyesinde kurulmuş olan Ulusal Siber Olaylara Müdahale Merkezi (USOM) tarafından tespit edilmiş oltalama sitelerinden elde edilmiştir (internet sitesi erişim adresi: https://usom.gov.tr/zararli-baglantilar/20.html). Özellik seçimli derin öğrenme modeli kapsamında URL adreslerine ilişkin listede bulunan açıklama sütunu kısmında "bankacılık" ifadesine yer verilen oltalama siteleri kullanılmıştır.

Python'da yazılan bir program aracılığıyla USOM internet sitesinde yer alan URL adresleri metin dosyasına kaydedilmiştir. Bu çalışma için 107 meşru ve 4427 onaylanmış oltalama saldırısı URL'si olmak üzere toplam 4534 URL adresi toplanmıştır.

3.2 VERİ ANALİZİ

Bu bölümde oltalama URL'lerinin toplanma aşamasından bu URL'lerin modellerin eğitimi için kullanılabilir bir veri setine dönüştürülmesi aşamasına kadarki geçen süreç hakkında bilgi verilmektedir.

Program.cs

MaydinPhd.ConsoleApp — Main()

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

namespace MaydinPhd
{
    class ConsoleApp
    {

        private const string UrlListFilePath = @"C:\MaydinPhd\UrlList.txt";

        static void Main()
        {
            //var p =new Program();
            //var dataList = p.ReadData(PhishingFilePath);
            //foreach (var data in dataList)
            //{
            //    data.ApplyRule5();
            //}

            string[] lines = System.IO.File.ReadAllLines(UrlListFilePath);

            using (var file = new System.IO.StreamWriter(@"C:\MaydinPhd\Output1.txt"))
            {
                foreach (var line in lines)
                {
                    file.WriteLine(line.Count(x => x == '-'));
                }
            }
            using (var file = new System.IO.StreamWriter(@"C:\MaydinPhd\Output2.txt"))
            {
                foreach (var line in lines)
                {
                    file.WriteLine(line.Count(x => x == '_'));
                }
            }
            using (var file = new System.IO.StreamWriter(@"C:\MaydinPhd\Output3.txt"))
```

Şekil 3.2: Özellik Çıkarma Amaçlı Kullanılan Yazılım Kod Örneği

URL veri setinden özellik çıkarmak için Microsoft Visual Studio programı kullanılmıştır. Öncelikle, daha önce seçilen URL'ler metin dosyasına (.txt uzantılı) kaydedilmiş ve oluşturulan bu dosya, özellik matrisi oluşturmak için bir girdi dosyası olarak kullanılmıştır. Şekil 3.2'de örnek olarak Microsoft Visual Studio programında kullanılan bazı kodlar gösterilmektedir.

3.2.1 URL Gelişmiş Özellikleri

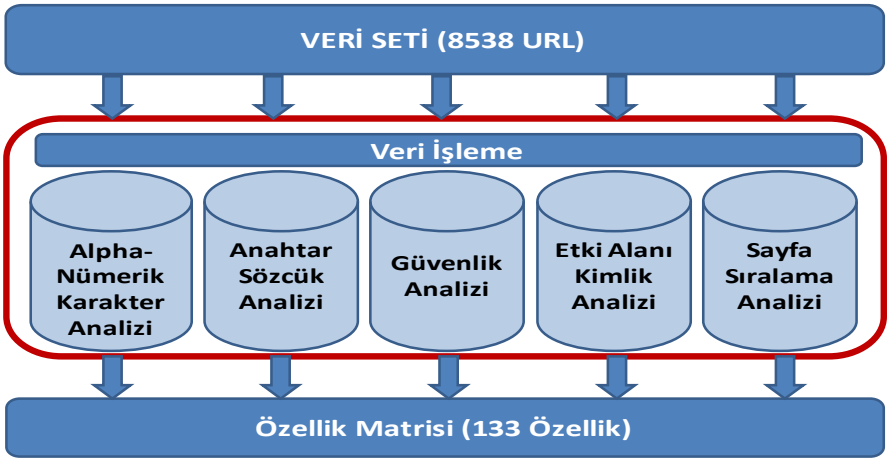
Oltalama siteleri genellikle bireysel verileri ele geçirmekte ve kişileri maddi zararlara uğratmaktadır. Bu yüzden bu sitelerin kişisel kullanım seviyesine indirgenerek engellenmesi çok önemlidir. Oltalama sitelerinin engellenmesine yönelik çalışmalara başlamadan tespit başarısını arttırmak amacıyla öncelikle bu sitelerin belirgin ve ortak özellikleri tespit edilmelidir. Oltalama sitelerinin en belirgin tespit edilebilir özelliklerinden birisi URL içeriği olmaktadır.

Bu çalışmada "özellik", bir internet sitesinde aramak istediğimiz bir değişken olarak tanımlanmaktadır. Özellik tabanlı yaklaşımın arkasındaki amaç, oltalama saldırısı algılama tekniğini olabildiğince yalın ve modüler hale getirmektir. Temel olarak, bu çalışmada kullanılan "özellik" seti uzman gözlemlerine ve oltalama saldırısı tespitine ilişkin çeşitli mevcut akademik kaynaklara dayanarak oluşturulmuştur. Sonuç olarak, veri setimizde bulunan URL'ler ile ilgili 133 farklı özellik elde edilmiş ve Ek-1'de verilmiştir. Bu bölümde yer verilen veri setindeki özelliklerin sayısal değerlere dönüştürme adımlarında kullanılan programlar da tamamen çalışmaya özgün oluşturulmuştur.

Ek-1'de bahsedilen özellikler A, B, C, D ve E grubu olmak üzere 5 ana kriterden oluşmaktadır. Derin öğrenme modelinde D grubunun 8. özelliği ve E grubu özellikleri veri seti hazırlama işlemlerinden çıkartılmış, geri kalan A, B, C ve D özellikleri ise yapılan çalışmaya uygun olacak şekilde uyarlanmıştır.

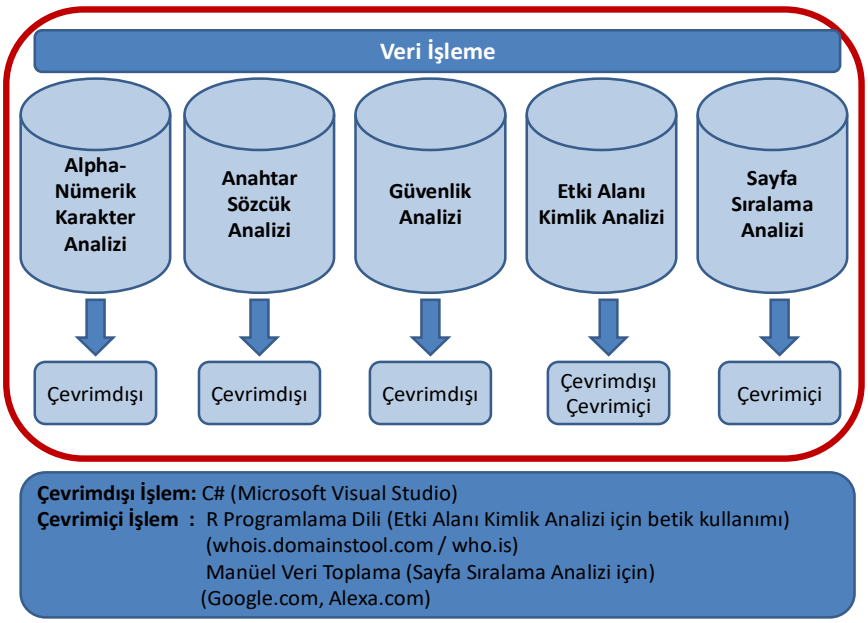
3.2.2 Özellik Çıkartma

Bu çalışmadaki performans analizinin iş akışı birbirinden ayrı bir dizi süreçten oluşmaktadır. İlk süreç, elde edilen veri setinde yer alan internet sitelerine ait URL'ler ile ilgili özellikleri çıkarmakta ve çıkarılan bu özelliklerle süreç içerisinde kullanılacak özellik matrisini oluşturmaktadır.



Şekil 3.3: URL Özellik Çıkartma Şeması

Şekil 3.3'de de görüleceği üzere, eldeki özellikler onları karakterize eden "Alfanümerik Karakter Analizi", "Anahtar Sözcük Analizi", "Güvenlik Analizi", "Etki Alanı Kimlik Analizi" ve "Sayfa Sıralama Analizi" olmak üzere beş farklı analiz altında kategorize edilmiştir.



Şekil 3.4: Özelliklere İlişkin Veri İşleme Şeması

Bu özelliklerin çoğu, URL'nin kendisinin metinsel özellikleridir, diğerleri ise, "whois sorgusu" ve "Google ve Alexa sayfa sıralama puanları" gibi üçüncü taraf hizmetlerine dayanmaktadır. Şekil 3.4'de verilen veri işleme şemasında kategorize edilmiş özelliklerin hangi yöntemlerle elde edildiği ve bu yöntemlerde kullanılan araçlar hakkında bilgi verilmektedir.

Eldeki verilerin metinsel özelliklerini elde etmek için Microsoft Visual Studio programında C # programlama dili kullanarak kodlar yazılmıştır. Öte yandan, diğer özellikleri üçüncü taraf hizmet sağlayıcılardan almak için çevrimiçi işlem adımları gerçekleştirilmiştir. Bu adımda "whois kaydını" almak için R programlama dili betiği yazılmıştır. Son olarak manuel yöntemlerle Google ve Alexa web sitelerinden ilgili URL'ler hakkında sayfa sıralama skoru verisi toplanmıştır.

R yazılımı açık kaynak kodlu bir yazılımdır ve konu üzerine uzmanlaşmış bir yazılım geliştirme ekibi ve geniş bir kullanıcı topluluğu tarafından desteklenmektedir. R yazılım dili, istatistiksel yazılım ve veri analizi geliştirmek için istatistikçiler ve veri madencileri arasında yaygın olarak kullanılmaktadır. R yazılımı, doğrusal ve doğrusal olmayan modelleme, klasik istatistiksel testler, zaman serisi analizi, sınıflandırma, kümeleme ve diğer benzer konular dâhil olmak üzere çok çeşitli istatistiksel ve grafiksel analiz teknikleri sağlamaktadır. Bu platformda kullanılan standart fonksiyonların çoğu R yazılımının kendi dilinde yazılmıştır, bu da kullanıcıların yapılan algoritmik seçimleri takip etmesini kolaylaştırmaktadır.

Yurtdışı kaynaklı veri seti kullanılarak yapılan İngilizce anahtar sözcük analizinde yine C# programlama dili kullanılarak URL bilgileri içerisinde geçen kelimeler çıkartılmış olup yapılan manuel analiz ile en çok dikkat çeken kelimeler elde edilerek Ek-1'de yer alan B grubu özellikler oluşturulmuştur.

Diğer taraftan yurtiçi kaynaklı veri seti kullanılarak yapılan analizlere aşağıda ayrıntılı olarak yer verilmektedir. Bu bölümde her bir özelliğin oltalama adresleri üzerinde uygulanmasıyla veri setinde yer alacak sayısal değerlerin belirlenmesi için oluşturulan kriterler ve algoritmalar detaylı olarak incelenmektedir.

A grubu özellikler

Alfabetik ve numerik karakterlerin ve kombinasyonlarının şüpheli şekilde bulunma durumunun araştırıldığı bölümdür.

1 ile 24 numara arasındaki özellikler oltalama sitelerinde kullanımı muhtemel karakterler olduğu için bu karakterlerin URL'de geçme sayısını bulan bir program parçacığı hazırlanmıştır. Bu özelliklerin tespiti için Python programlama dilinin düzenli ifadelerinden yararlanılmıştır.

25 numaralı özellik 1 ile 24 numara arasındaki özelliklerin geçme sayılarının toplamını elde etmektedir. Bunun için 1 ile 24 numara arasındaki özelliklerde tespit edilen sayıların toplamını kaydeden bir kod yazılmıştır.

26 numaralı özellikte oltalama URL'lerinin toplam karakter uzunluğu araştırılmaktadır. Bunu elde etmek için Python programlama dilinin *len()* fonksiyonundan yararlanılmıştır.

27 numaralı özellikte URL'in host kısmında yer alan [0,9] arası tamsayıların adedi araştırılmaktadır. Bu adedi elde etmek için düzenli ifadeler kullanılmıştır.

28 numaralı özellikte URL'in host kısmında yer alan "." karakterinin adedi araştırılmaktadır. Bu adedi elde etmek için Python programlama dilinin liste içinde aranan karakterin geçme sayısını bulan *count()* fonksiyonundan yararlanılmıştır.

29 numaralı özellik ile oltalama URL'in host kısmındaki toplam karakter uzunluğu araştırılmaktadır. Bunu elde etmek için Python programlama dilinin *len()* fonksiyonundan yararlanılmıştır.

30 numaralı özellikte URL içerisindeki en uzun kelimenin karakter uzunluğu incelenir. Bu özellikte istenilen durumu elde edebilmek için URL içindeki kelimeleri bulan *host_longest_word()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, URL içinde alfabetik ve numerik olmayan ifadelere göre URL'yi bölerek kelimeleri elde etmektedir. Bölme işlemi Python programlama dilinde yer alan *split()* fonksiyonu aracılığıyla, böldükten sonraki en uzun kelimeyi bulmak için ise *sort()* sıralama hazır fonksiyonu kullanılmıştır.

31 numaralı özellikte URL içerisinde yer alan kelime sayısı araştırılmaktadır. İstenilen durumu elde etmek için *url_words()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, *split()* fonksiyonu ile alfabetik ve numerik olmayan karakterlere göre kelimeleri böldükten sonra elde edilen kelimeleri listeye ekler. Daha sonra Python programlama dilinin *len()* fonksiyonu ile listede yer alan kelimelerin sayısı bulunur.

32 numaralı özellikte URL'in host kısmında yer alan kelime sayısı araştırılmaktadır. İstenilen durumu elde etmek için *host_words()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, *split()* fonksiyonu ile alfabetik ve numerik olmayan karakterlere göre kelimeleri böldükten sonra elde edilen kelimeleri listeye ekler. Daha sonra Python programlama dilinin *len()* fonksiyonu ile listede yer alan kelimelerin sayısı bulunur.

33 numaralı özellik ile URL içerisinde harf-rakam-harf deseninin bulunma durumu araştırılmaktadır. Bu deseni tespit etmek için Python programlama dilinin düzenli ifadeler kütüphanesine başvurulmuştur.

34 numaralı özellik ile URL içerisinde rakam-harf-rakam deseninin bulunma durumu araştırılmaktadır. Bu deseni tespit etmek için Python programlama dilinin düzenli ifadeler kütüphanesine başvurulmuştur.

B grubu özellikler

Bu grubun özellikleri, oltalama URL adresleri içerisinde yer alan kelimelerin analizini kapsamaktadır. Oltalama siteleri için inandırıcı olmak ya da dikkat çekmek amacıyla dolandırıcılar tarafından çeşitli kelimeler kullanılmaktadır. Bu grubun özellikleri sayesinde URL adresinde yer alan kelimeler modelin eğitiminde etkili olarak kullanılabilmektedir.

Bu grubun özelliklerinin çıkarımıyla ilgili bir kelime modülü tasarlanmıştır. Bu kelime modülü tamamen çalışmaya özgü olarak hazırlanmış özgün bir programdır. Kelime modülü sayesinde kelimeler dinamik olarak elde edildiğinden bu grubun özellik sayısı değişkenlik göstermektedir. Fakat veri seti hazırlama programının tasarımı sayesinde B grubu özelliklerinin sayısındaki değişiklikler modelin eğitimini hiçbir şekilde yanıltmamaktadır.

Kelime Modülü

Bu bölümde oltalama URL'lerinde geçen kelimelerin herhangi bir manuel müdahale olmadan program tarafından otomatik olarak çıkarılmasından bahsedilmektedir.

Kelime modülü, çok sayıdaki oltalama URL'leri içinden kelimelerin çıkartılmasındaki zorluklar ile başa çıkmak ve manuel olarak incelenmesi sonucu gözden kaçırılmasını engellemek için tasarlanmış bir programlama algoritmasıdır. Bu modül sayesinde oltalama

URL'leri hangi miktarda olursa olsun oltalama kelimeleri zaman kaybedilmeden hızlı bir şekilde elde edilebilmektedir.

Bu çalışmada tasarlanan kelime modülü hazır metin madenciliği kütüphanelerinden önemli farklara sahiptir. Bir metin madenciliği kütüphanesi kelimeler arasında boşluk ve noktalama işaretleri varken kelimeleri tespit edebilmektedir. Fakat URL adresleri gibi bitişik kelimelerin yer aldığı sözcük öbeklerinde bu kütüphaneler yardımıyla URL içinde yer alan kelimelerin algılanması mümkün olmamaktadır. Kelime modülü bu durumla başa çıkabilmek için bu çalışmaya özel olarak tasarlanmıştır. Kelime modülü karakterler arasında boşluk olmasından ve noktalama işaretlerinin ve benzeri işaretlerin bulunmasından etkilenmeyecek şekilde tasarlanmıştır.

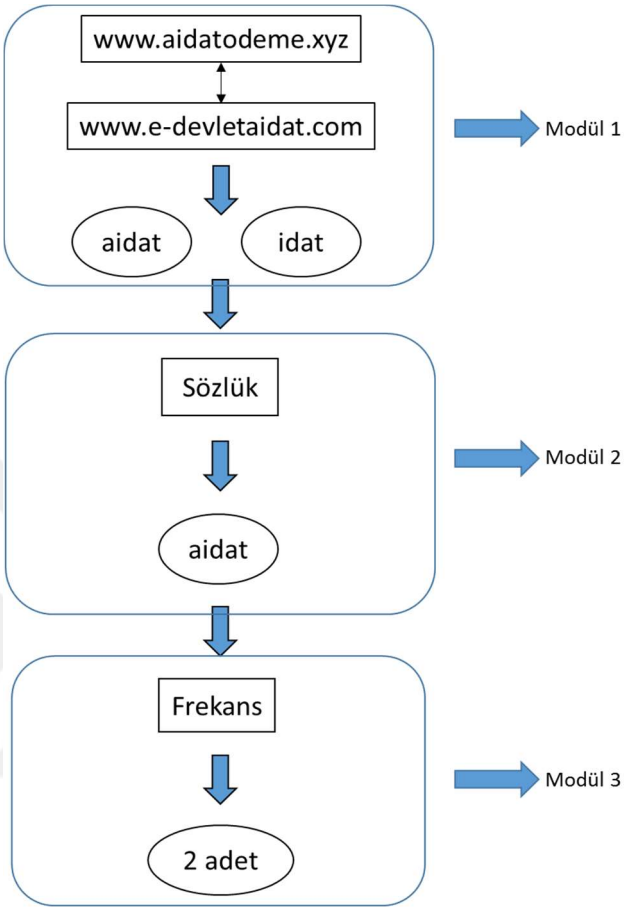
Göz alıcı kelimeler için oluşturulmuş B grubu özelliklerin belirlenmesi için kelime modülü kritik rol oynamaktadır. B grubu özelliklerin kelimeleri bu modülde yapılan değerlendirme neticesinde belirlenmektedir. Kelime modülü hem İngilizce hem de Türkçe'ye uyumlu olacak şekilde iki farklı tasarımda işletilmiştir. İlerleyen bölümlerde sadece Türkçe uyumluluk tasarımından bahsedilmektedir.

Tasarım

Oltalama URL'lerine göre hazırlanan B grubu özelliklerin belirlenmesi için kelime modülü kendi içinde üç temel işlevden oluşmaktadır. Kelime modülünün 3 farklı işleve sahip bölümlerine Şekil 3.5'de yer verilmektedir.

İlk işlev oltalama URL'lerini birbirleriyle karşılaştırarak içinden anlamlı ve anlamsız bütün kelimeleri bulmak için tasarlanmıştır. Bu işlev sayesinde tekrar eden harf öbeklerinin elde tutulmasıyla önemli bir göz alıcı kelimenin dikkatten kaçması engellenmiş olur. İkinci işlev sözlük kontrolü işlemidir. Bu işlevde ilk işlevden gelen anlamlı ve anlamsız ortak bütün harf öbeği veri tabanındaki Türkçe sözlükte aratılır. Sözlükte yer almayan anlamsız kelimeler elenerek elde ettiğimiz kelime listesinin kullanılabilir hale gelmesi sağlanır. Son olarak üçüncü işlevin görevi ise oltalama URL'lerinde geçen anlamlı kelimelerin toplamdaki kullanım sıklığını kaydetmektir. Ayrıca kelimelerin kullanım sıklığı üzerinde yapılan değerlendirmeler sonucu eşik değerinin belirlenmesi ve bu eşik değerinin üstünde kalan kelimelerin seçiminin yapılması yine bu işlevle gerçekleştirilmektedir. Kelime modülünün bahsedilen işlemleri

sonrasında modelin eğitimi için hazırlanan B grubu özelliklerinin göze alıcı kelimeleri belirlenmektedir.



Şekil 3.5: Kelime Modülünün İçyapısı ve Bölümleri

Algoritma

Bu bölümde kelime modülü tarafından oltalama URL'lerinde yer alan kelimelerin nasıl elde edildiği ile ilgili adımlardan oluşan algoritmalara ve işleyişe ait detaylı bilgilere yer verilmektedir. Kayıtlı olan URL listesinden anlamlı ve anlamsız kelimelerin tespit edilmesini sağlayan ilk işleve ait algoritmanın adımları aşağıda yer almaktadır. URL listesi sisteme yüklendikten sonra:

Adım 1: Listedeki URL baştan itibaren sırayla tutulur.

Adım 2: Listedeki bir sonraki URL benzerlik araması için alınır.

Adım 3: İlk URL harfleri sırasıyla bir sonraki URL içinde aratılır.

Adım 4: İlk URL'in sıradaki harfi, alınan URL içinde bulunursa Adım 5'den, bulunmazsa Adım 3'ten devam edilir.

Adım 5: Harfin bulunduğu konum kaydedilir.

Adım 6: İlk URL'in sıradaki ikinci harfi, diğer URL'in kaydedilen konumunun bir sonrakinde yer alan harfiyle eşleştirilir.

Adım 7: Bu işlem benzer şekilde yapıldıktan sonra üç harfli kelimeden fazla olmazsa Adım 3'ten, fazla olursa Adım 8'den devam eder.

Adım 8: Üç harften fazla olan kelime, kelime listesine kaydedilir

Adım 9: Bulunan kelimenin ikinci harfinden başlanarak Adım 3'den devam edilir.

Adım 10: İlk URL'de aratılacak kelimeler biterse Adım 2'den devam edilir.

Adım 11: Adım 2'deki URL'ler sona gelinirse Adım 1'den devam edilir. Elde edilen kelime listesi kaydedilir.

Anlamlı ve anlamsız kelimeler listesi yukarıda verilen algoritma yardımıyla elde edildikten sonra ikinci işlev olan anlamsız kelimelerin elenmesi sürecinden geçirilmektedir.

Bu işlevde anlamlı anlamsız kelimeler listesi github'tan elde edilen Türkçe veri tabanından sorgulanarak anlamlı kelimeler seçilir. URL'ler yapısı gereği İngilizce karakterlerden oluştuğu için elde edilen Türkçe kelimelerde geçen "ç","ğ","ı","ö","ü","ş" harfleri İngilizce alfabe formatındadır. Türkçe sözlük ile İngilizce formatında elde edilen Türkçe kelimeler arasındaki uyuşmazlığı kaldırmak için öncelikle dönüşüm işlemi yapılması gereklidir. Bu yüzden Türkçe veri tabanındaki kelimeler İngilizce karakterlere uygun olarak (ç harfi c'ye, ğ harfi g'ye, ı harfi i'ye, ö harfi o'ya, ş harfi ş'ye, ü harfi u'ya) dönüştürülmüştür.

İkinci işlevin işleyişine ait algoritma adımları aşağıda yer almaktadır:

Adım 1: Kelime listesindeki kelimeler sırayla okunur.

Adım 2: Kelime listesinden gelen kelime İngilizce alfabeye göre dönüştürülmüş Türkçe sözlükte aratılır.

Adım 3: Kelime bulunursa Adım 4'e geçilir, bulunmazsa Adım 1'e dönülür.

Adım 4: Kelime anlamlı kelimeler listesine kaydedilir.

Bu işlem sonucunda elimizde sadece anlamlı kelimelerden oluşan bir liste bulunmaktadır. Fakat URL listesi ne kadar çok sayıda olursa benzerlik gösterecek çok sayıda kelime ortaya çıkacaktır. Bu kadar çok kelimenin özellik olarak kullanılması hem gereksiz hem de sistemi yanıltma potansiyeline sahip olabilmektedir. Bu yüzden bu kelime listesi içerisinden eğitimde kullanılacak göz alıcı özellik olan B grubu kelimelerin seçilmesi için kriter belirlenmesi çok önemlidir. Bu kriterin belirlenmesinde kelimelerin toplam URL listesinde geçme sıklığının baz alınmasına karar verilmiştir.

Üçüncü işlevin görevi bulunan anlamlı kelimelerin tüm URL içerisinde geçme sıklığını kontrol etmesi ve bulduğu sıklık sayılarını kaydetmesidir. Daha sonra her kelime için kaydedilen geçme sıklıklarının ortalaması alınarak bir eşik değeri belirlenmiştir. Bu eşik değerinin üstünde kalan kelimeler B grubu özelliğinin göz alıcı kelimeleri listesinde yer almaktadır.

Ortalamaya göre eşik değerinin belirlenmesi gibi bir yapı, aynı zamanda kendi kendini sürekli güncelleyen bir dinamik özelliği beraberinde getirmiştir. Bu yapı sayesinde modası geçen ya da yeni trend olan kelimeler otomatik olarak B grubu özellik listesini güncelleyerek değişiklik sağlayabilecektir.

C grubu özellikler

Bu grubun özellikleri oltalama URL adreslerinde yer alan sitenin güvenlik sertifikaları, URL yolunda yer alan dosya uzantıları gibi güvenlik unsurlarını analiz etmektedir.

1 numaralı özellik sitede güvenlik sertifikasını temsil eden “https” kelimesini araştırmaktadır. Python programlama dilinin *find()* fonksiyonu ile “https” kelimesi aratılarak varlığı tespit edilmektedir.

2 numaralı özellikte URL adresinde “http” karakterinin birden fazla geçme durumu araştırılmaktadır. Bu adedi elde etmek için python programlama dilinin karakter dizisi içinde aranan karakterin geçme sayısını bulan *count()* fonksiyonundan yararlanılmıştır.

3 numaralı özellikte URL adresinde “https” karakterinin birden fazla geçme durumu araştırılmaktadır. Bu adedi elde etmek için Python programlama dilinin karakter dizisi içinde aranan karakterin geçme sayısını bulan *count()* fonksiyonundan yararlanılmıştır.

4 numaralı özellikte URL adresinde “www” karakterinin birden fazla geçme durumu araştırılmaktadır. Bu adedi elde etmek için Python programlama dilinin karakter dizisi içinde aranan karakterin geçme sayısını bulan *count()* fonksiyonundan yararlanılmıştır.

5 numaralı özellikte URL adresinde yer alan muhtemel tehlikeye sahip program dosya uzantıları araştırılmaktadır. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

6 numaralı özellikte URL adresinde yer alan tehlikeli olabilecek script dosya uzantıları incelenmektedir. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

7 numaralı özellikte URL adresinde yer alan olası tehlikeye yol açacak kısa yol uzantıları incelenmektedir. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

8 numaralı özellikte URL adresinde yer alan tehlikeli olabilecek ofis programı makro uzantıları incelenmektedir. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

9 numaralı özellikte URL adresinde yer alan tehlikeli olması muhtemel diğer uzantılar incelenmektedir. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

D grubu özellikler

Bu grubun özellikleri oltalama URL adreslerini alan adını tanımlayan bilgilere göre incelemektedir.

1 numaralı özellik URL adresi içerisinde yer alan herhangi bir IP adresinin varlığını araştırmaktadır. Bu dosyaların tespiti ile ilgili olarak Python programlama dilindeki düzenli ifadeler kullanılmıştır.

2 numaralı özellik URL adresi içerisinde birden fazla üst seviye alan adının *(top level domain)* bulunup bulunmadığını araştırmaktadır. Bu özelliği işletebilmek için *topLevelCount()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, bütün üst seviye alan adlarının kayıtlı olduğu metin dosyasındaki kelimelerin URL içerisinde aratılması sonucu bulunan üst seviye alan adlarının sayısını kaydederek çalışmaktadır.

3 numaralı özellik URL adresi içerisinde herhangi bir üst seviye alan adının bulunup bulunmadığını araştırmaktadır. Bu fonksiyon, bütün üst seviye alan adlarının kayıtlı olduğu metin dosyasındaki kelimelerin URL içerisinde aratılması sonucu çıktı "0" ise muhtemel oltalama sitesi olduğunu belirtmek için özellik "1" değerini alır.

4 numaralı özellikte host kısmı dışında kalan URL yolunda marka adının varlığı araştırılmaktadır. Marka adının tespit edilebilmesi için *outOfHostCount()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, kayıtlı yasal banka URL adresleriyle şüpheli URL adreslerini düzenli ifadeler aracılığıyla kıyaslamaktadır.

5 numaralı özellikte URL'in host kısmında yer alan marka adının yanında herhangi bir kelimenin varlığı araştırılmaktadır. Marka adının haricindeki kelimenin tespit edilebilmesi için *substringHost()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, kayıtlı yasal banka URL adreslerinde yer alan kelimeleri çıkarttıktan sonra şüpheli URL adreslerini içerisinde düzenli ifadeler aracılığıyla aratır. Daha sonra bulunan marka kelimesinin uzunluğu dışındaki karakterlerin uzunluğu 0'dan büyükse şüpheli kelimenin varlığı tespit edilmiş olur.

6 numaralı özellikte URL'in host kısmında yer alan alan adının yanında herhangi bir alt alan adının varlığı araştırılmaktadır. Bu özellik için kullanılan *substringHost()* fonksiyonu 5 numaralı özellikte bahsedildiği şekilde tespit yapmaktadır.

7 numaralı özellikte URL'in adres kısaltma servislerinden yararlanılıp yararlanılmama durumu incelenmektedir. Bu özelliğin işleyebilmesi için *shorteningURL()* adında bir fonksiyon hazırlanmıştır. Bu fonksiyon, tüm kısaltma servislerine ait URL içinde kullandıkları kelimeleri

veri setinde URL içinde aratarak çalışmaktadır. Eğer kısaltma mevcutsa muhtemel şüpheli olarak özellik "1" değerini vermektedir.

Özellik seçimli derin öğrenme modelinin yurtiçi kaynaklı veri setine ilişkin yapılan analizde, kelime modülünün oltalama URL veri setine uygulanması sonucu 342 adet göz alıcı kelime elde edilmiştir. Bu 342 adet göz alıcı kelimenin toplam URL içerisinde geçme sıklığı tablo 3.2'de gösterilmektedir.

Tablo 3.2: Yurtiçi Veri Setinden Elde Edilen Göz Alıcı Kelimelerin Frekansı

Kelimeler	Frekans	Kelimeler	Frekans	Kelimeler	Frekans	Kelimeler	Frekans	Kelimeler	Frekans
aidat	309	istem	24	vakif	10	bilgilendirme	5	guvenlik	3
iade	270	sistem	24	arac	9	bili	5	haziran	3
turk	263	arti	23	duyuru	9	birlik	5	iran	3
devlet	236	baskanlik	23	emen	9	bonus	5	kamusal	3
vuru	209	form	23	esis	9	destekci	5	kara	3
basvuru	207	bakanligi	22	guzel	9	diyesi	5	kayit	3
deste	167	adem	21	hazine	9	erce	5	mani	3
turkiye	158	edik	21	hizli	9	eris	5	medya	3
destek	153	ineb	21	iletisim	9	guvenli	5	name	3
hizmet	124	ergi	20	sigar	9	hazir	5	onayli	3
evde	121	geri	20	yeni	9	ilgilendirme	5	onda	3
iris	117	kuru	20	yukleme	9	isler	5	ortak	3
giris	116	urus	20	yuru	9	kalk	5	sayfa	3
andemi	107	denizde	19	denizden	8	kurumsal	5	tala	3
pandemi	106	erin	19	geldi	8	meri	5	tarife	3
diye	87	mail	19	genel	8	payi	5	tozel	3
kredi	87	ceki	18	iska	8	tekci	5	vuruk	3
bank	85	ilim	18	kala	8	vize	5	yetim	3
deme	85	isim	18	kilisi	8	vurut	5	yilbasi	3
internet	84	ramazan	18	temmuz	8	yasa	5	zamani	3
odeme	75	tali	18	yara	8	aktif	4	abak	2
kart	70	atlari	17	ziraat	8	alim	4	agun	2
hediye	69	bayram	17	agit	7	asla	4	ahiz	2
anli	64	cekilis	17	aile	7	ayli	4	akla	2
islem	64	finans	17	arak	7	bakiye	4	anapara	2
kanli	64	kilis	17	besi	7	birlikte	4	artis	2
orgu	62	atil	16	daire	7	deva	4	asit	2
sorgu	62	bireysel	16	elek	7	diba	4	avize	2
sosyal	62	ekler	16	emis	7	dijital	4	bile	2
saglik	60	ortali	16	etsiz	7	diyen	4	borcu	2
orta	57	takip	16	hemen	7	edim	4	comert	2
azan	54	tekler	16	hesap	7	erden	4	cumhurbaskanligi	2
lama	54	vatandas	16	ilen	7	esiz	4	deneme	2

merkez	54	vergi	16	ileri	7	ezim	4	diyet	2
ulama	52	vurus	16	ilik	7	halka	4	ekonomik	2
deniz	50	atilim	15	ipci	7	ikrar	4	elleme	2
islemler	48	katilim	15	orum	7	imce	4	elli	2
paket	41	bedava	14	takipci	7	isleri	4	emekli	2
erim	40	dava	14	ucretsiz	7	istikrar	4	eneme	2
halk	40	firsat	14	acilik	6	izin	4	evin	2
kampanya	40	irsat	14	aktarim	6	kamus	4	eyin	2
yardim	40	kasi	14	alin	6	kont	4	genelge	2
para	39	talep	14	andemik	6	laka	4	hepsi	2
sorgulama	39	destegi	13	bankacilik	6	mika	4	kalb	2
yapi	39	kamu	13	birli	6	onay	4	karar	2
kazan	35	servis	13	borc	6	plaka	4	komunikasyon	2
urum	35	uygulama	13	dagitim	6	saglikli	4	kontor	2
akan	34	dirimli	12	dene	6	sonuc	4	kontrol	2
anka	34	indirimli	12	dimi	6	tural	4	lira	2
bakan	34	sana	12	emek	6	uruk	4	meni	2
cumhur	33	tasi	12	emik	6	urunu	4	miza	2
kapi	32	zira	12	enli	6	yisa	4	olusturma	2
mide	32	anti	11	eter	6	abil	3	ongun	2
banka	31	aranti	11	haber	6	alma	3	sanal	2
etleri	31	arin	11	icin	6	anali	3	sayi	2
dirim	29	atis	11	isba	6	arda	3	sinirsiz	2
indir	29	garanti	11	mayis	6	arife	3	site	2
mobil	29	guncel	11	operator	6	atim	3	size	2
sube	29	guven	11	ramazanda	6	ayit	3	ster	2
kapisi	28	ajans	10	sanayi	6	bakanlik	3	telekomunikasyon	2
pisi	28	akif	10	sati	6	banko	3	trol	2
hayat	27	alba	10	tarim	6	basi	3	ulke	2
indirim	27	alka	10	urun	6	bini	3	vatandaslarin	2
anlik	26	bilgi	10	yardimi	6	dede	3	veri	2
ceza	26	duyur	10	yeter	6	ekli	3	virus	2
trafik	26	ilgi	10	aninda	5	ekonomi	3	yeme	2
fatura	25	kurum	10	anlar	5	enba	3		
ozel	25	maliye	10	atin	5	girisim	3		
tura	25	uyur	10	bayi	5	gunler	3		

Tüm veri setinden elde edilen 342 adet kelimenin bir kısmı, az sayıda sıklığa sahip olduğundan ve modelin eğitimi üzerinde gereksiz yük teşkil edeceğinden elenmesinin gerekli olduğu düşünülmüştür. Bu sebeple B grubu özelliği kelimeleri olarak hangi sıklık miktarından itibaren seçileceği eşik değeri sıklıkların aritmetik ortalamasına göre belirlenmiştir. 342 adet kelimenin sıklık ortalamaları alındığında eşik değeri yaklaşık 21 olarak hesaplanmıştır. Hesaplanan eşik

değeri baz alındığında 342 kelime içerisinden 78'nin göz alıcı kelime olarak kullanımı uygun görülmüştür. Tablo 3.3'de veri setinde kullanılacak 78 adet göz alıcı kelimelerin hangileri olduğu ve B grubu özelliklerden hangilerine karşılık geldiği gösterilmektedir.

Tablo 3.3: Eşik Değerinin Üstünde Kalan Göz Alıcı Kelimeler

Özellik	Kelime	Özellik	Kelime	Özellik	Kelime	Özellik	Kelime
B1	adem	B21	destek	B41	iris	B61	pandemi
B2	aidat	B22	devlet	B42	işlem	B62	para
B3	akan	B23	dirim	B43	işlemler	B63	pisi
B4	andemi	B24	diye	B44	istem	B64	saglik
B5	anka	B25	edik	B45	kampanya	B65	sistem
B6	anli	B26	erim	B46	kanli	B66	sorgu
B7	anlik	B27	etleri	B47	kapi	B67	sorgulama
B8	arti	B28	evde	B48	kapisi	B68	sosyal
B9	azan	B29	fatura	B49	kart	B69	sube
B10	bakan	B30	form	B50	kazan	B70	trafik
B11	bakanligi	B31	giris	B51	kredi	B71	tura
B12	bank	B32	halk	B52	lama	B72	turk
B13	banka	B33	hayat	B53	merkez	B73	turkiye
B14	baskanlik	B34	hediye	B54	mide	B74	ulama
B15	basvuru	B35	hizmet	B55	mobil	B75	urum
B16	ceza	B36	iade	B56	odeme	B76	vuru
B17	cumhur	B37	indir	B57	orgu	B77	yapi
B18	deme	B38	indirim	B58	orta	B78	yardim
B19	deniz	B39	ineb	B59	ozel		
B20	deste	B40	internet	B60	paket		

Göz alıcı kelimeler için eşik değeri değişken olduğundan daha sık geçen ve güncel kelimeler otomatik olarak eskilerinin yerini alabilecektir. Tablo 3.3'de belirtilen B grubu özelliklerinin sayısı da eşik değerine göre değişiklik gösterecek şekilde dinamiktir. Bu çalışmada kullanılan kelime modülünün aynı işlem adımlarını içerecek şekilde İngilizceye uyarlanması sonrası elde edilen göz alıcı kelimelere Ek-2'de yer verilmektedir.

Önerilen derin öğrenme modelinin eğitimi için veri hazırlamanın bir parçası olarak normalleştirme tekniği uygulanmaktadır. Normalleştirme, veri setindeki sütunların sayısal değerlerini, bilgi kaybına sebep olmadan ya da değer aralıklarında yer alan farklılıkları bozmaksızın ortak bir ölçek kullanımı için değiştirme işlemidir.

Orijinal veri setindeki sayıların ölçeğinde yer alan büyük farklılıklar, modelleme esnasında değerlerin özellik bazında birleştirilmesi sorunlara neden olabilir. Normalleştirme işlemi, hem model için kullanılan bütün sayısal değer içeren sütunlara uygulanan ölçek bazındaki değerleri korurken, hem de orijinal verilerdeki genel dağılımı ve aralarındaki oranları korumak için yeni değerler oluşturarak karşılaşılacak sorunları engeller. Ayrıca optimizasyon algoritmaları gibi algoritmaların verileri doğru biçimde modelleyebilmesi için normalleştirme işlemi gereklidir.

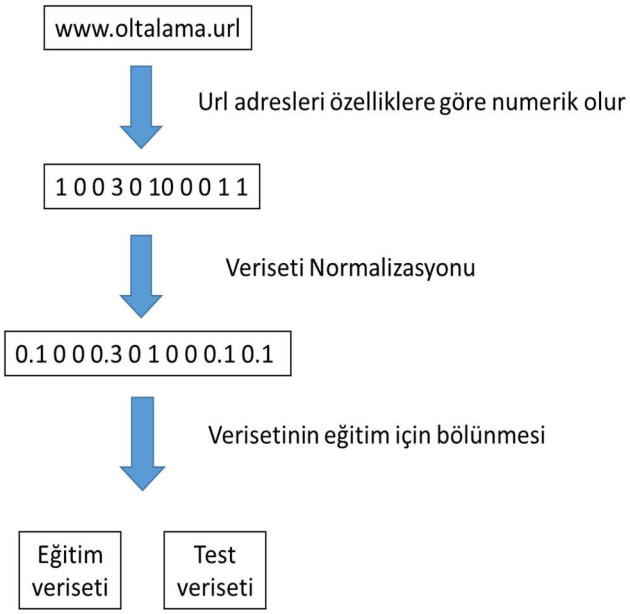
Hazırlanan veri setindeki değerlerin normalizasyonu için her özelliği doğrusal olarak [0,1] aralığında yeniden ölçeklendiren min-maks normalleştiricisi kullanılmıştır. 3.1 nolu denklemde sütundaki değerlerin [0,1] aralığında yeniden ölçeklendirme formülü gösterilmektedir.

$$z = \frac{x - min\ (x)}{max(x) - min\ (x)} \qquad (3.1)$$

Burada x her bir özelliğe karşılık gelen değeri, min(x) veri setindeki en küçük değeri, max(x) veri setindeki en büyük değeri temsil etmektedir. Negatif değerlerden kurtulmak için minimum değer “0” olacak şekilde değerler kaydırılır ve orijinal maksimum ve minimum değerler arasındaki fark olan yeni maksimum değere her bir özelliğin değeri bölünür.

Bir sonraki aşama olarak modelin eğitimi için veri setinin eğitim-test veri setleri olarak bölünmesi gerekmektedir. Veri setinin bölünme oranları için belli bir standart bulunmamakla birlikte yaygın tercih edilen %70-%30 oranı için ayrı ayrı deney yapılmıştır. Genel olarak sistemin aşırı öğrenmeye gitmemesi için bu oranların kullanımı ideal bulunmaktadır.

Veri seti hazırlama aşamasında oltalama URL’lerine sırasıyla belirlenen özelliklerin uygulanması sonucu URL adresler sayısal veri setine dönüştürülmüştür. Şekil 3.6’da genel olarak veri önişleme adımlarına yer verilmiştir.



Şekil 3.6: Veri Önişleme Adımları

Önerilen modelin eğitimi için 4427 oltalama URL'si ve 107 adet yasal URL kullanılmış ve bu URL'ler veri seti hazırlama aşamasında bahsedilen özellikler baz alınarak derin öğrenmenin uygulanabileceği numerik değerler veri setine dönüştürülmüştür. Veri seti model için eğitime uygun hale geldikten sonra Tablo 3.4'de gösterildiği gibi veri seti eğitim ve test olmak üzere %70-%30 oranında iki bölüme ayrılmıştır. Sistemin aşırı öğrenmeye maruz kalmaması için modelin eğitimi aşamasında eğitim veri setinin doğrulama testi için pareto optimumu olarak bilinen %80-%20 oranı tercih edilmiştir. Bu modelde optimizasyon algoritması olarak adam ve adamax, aktivasyon fonksiyonu olarak relu, sigmoid ve tanh, kayıp fonksiyonu olarak binary cross entropy kullanılmıştır.

Tablo 3.4: Veri Setinin Bölünmesi

	Yasal	Oltalama
Eğitim	75	3098
Test	32	1329
Toplam	107	4427

Bu modelin eğitimi aşamasında daha doğru değerlendirme yapmak için katman ve düğüm sayıları değiştirilerek 6 farklı model kurulmuştur. Bu mimarilerde katman sayısının ve düğüm sayısının model eğitimindeki başarısı araştırılmıştır.

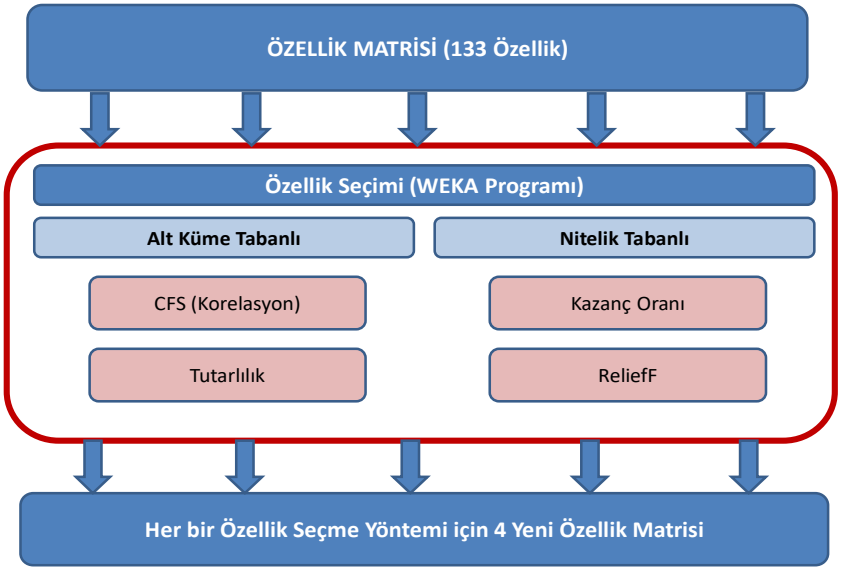
Önerilen derin öğrenme modelinin hazırlanan veri setiyle eğitilebilmesi için Python geliştirme ortamında kullanılan “keras” ve “tensorflow” makine öğrenme hazır kütüphanelerindeki fonksiyonlar kullanılmaktadır. Hazırlanan veri seti formatının bu fonksiyonlarla uyumlu işlem yapılabilmesi için “pandas” kütüphanesinin DataFrame() fonksiyonu kullanılarak veri seti uygun formata dönüştürülmüş olur. Ayrıca veri seti, çalışma içerisinde titiz bir şekilde hazırlandığı için yanıltıcı değerler ve boş değerler bulunmamaktadır.

Önerilen derin öğrenme modelinin tasarımında programlama dili olarak Python 3.8 versiyonu kullanılmıştır. Python programlama dili, makine öğrenmesi ve derin öğrenme ile ilgili denenmiş ve olgunluğa erişmiş çok sayıda kütüphaneye sahip olduğu için tercih edilmiştir. Derin öğrenme çalışmasına yardımcı olması için hazırlanmış kütüphanelerden Keras, Numpy, Pandas ve Scikit-Learn kütüphaneleri tez çalışması için kullanılmaktadır. Modelin tasarımı için geliştirme ortamı olarak ise Anaconda Spyder IDE tercih edilmiştir. Model eğitiminin yapıldığı deney için kullanılan ana makine; 1.8GHz - Intel Core i7 8550U serisi işlemci, 16GB ana bellek kapasitesi, 512GB SSD harici disk kapasitesi ve Windows 10 Home işletim sistemi özelliklerine sahiptir.

3.2.3 Özellik Seçimi

Özellik seçimi başarılı bir analiz için gerekli bir süreçtir. Diğer taraftan alternatifler arasından en uygun özellik seçim yönteminin bulunması bir diğer süreç olarak karşımıza çıkmaktadır. Bu adımda özellik seçme yöntemlerinin başarısı veri sınıflandırması için seçilen algoritmalar ile değerlendirilmektedir. Burada sözü edilen başarı, yöntemin daha az özellik seçmesinden ziyade genel olarak kurgulanan modele olan katkısı olarak ifade edilmektedir. Çoklu makine öğrenmesi modelinde en iyi modeli elde etmek için bu çalışmada bir dizi özellik seçim yöntemi ve sınıflandırma algoritması değerlendirmeye tabi tutulmuştur.

Bu çalışmada, korelasyon ve tutarlılık alt küme tabanlı, kazanç oranı ve Relief-F nitelik tabanlı özellik seçme yöntemleri, sınıflandırma algoritmalarına olan performans katkılarına göre değerlendirilmiştir. Bahsedilen dört özellik seçme yöntemi, toplanan 8538 URL'nin belirlenmiş farklı özelliklerini içeren veri kümesinde ayrı ayrı çalıştırılmıştır. Bu özellik seçme yöntemlerini uyguladıktan sonra farklı sayıda özelliğe sahip yeni dört özellik matrisi elde edilmiştir. Şekil 3.7’de özellik seçme metotları genel kullanım mimarisi görülmektedir.



Şekil 3.7: Özellik Seçme Metotları Kullanım Mimarisi

Bu çalışmada, bu süreç WEKA veri madenciliği ve sınıflandırma yazılım aracı ile analiz edilmiştir. WEKA açık kaynak kodlu, içerisinde pek çok sınıflandırma, regresyon, kümeleme, bağıntı kuralları, yapay sinir ağları algoritmaları ve önişleme metotları barındıran, yaygın kullanımlı bir veri madenciliği aracıdır. WEKA, ham verinin işlenmesi, öğrenme metotlarının veri üzerinde istatistiksel olarak değerlendirilmesi, ham verinin ve ham veriden öğrenilerek çıkarılan modelin görsel olarak izlenmesi gibi veri madenciliğinin tüm basamaklarını desteklemektedir. Ayrıca geniş bir öğrenme algoritmaları yelpazesine sahip olduğu gibi pek çok veri önişleme filtrelerini de içermektedir.

3.3 YÖNTEMLER

Makine öğrenmesi ve istatistikte sınıflandırma, kategorisi bilinen gözlemleri içeren bir eğitim veri setine dayalı olarak yeni bir gözlemin kategorisini tanımlama konusudur. Bu çalışmada kullanılacak sınıflandırmada belirli bir internet sitesini "oltalama internet sitesi" veya "yasal internet sitesi" sınıflarına atamak üzerine durulmuştur. Bu çalışmada, çalışmanın amacına uygun olacağı düşünülen makine öğrenmesi ve derin öğrenme algoritmaları kullanılmıştır.

Çalışmada iki model esas alınmıştır. Özellik Seçimli Çoklu Makine Öğrenmesi Modeli ile birden fazla özellik seçimi ve sınıflandırma algoritması hibrit şekilde kullanılarak en iyi ikili

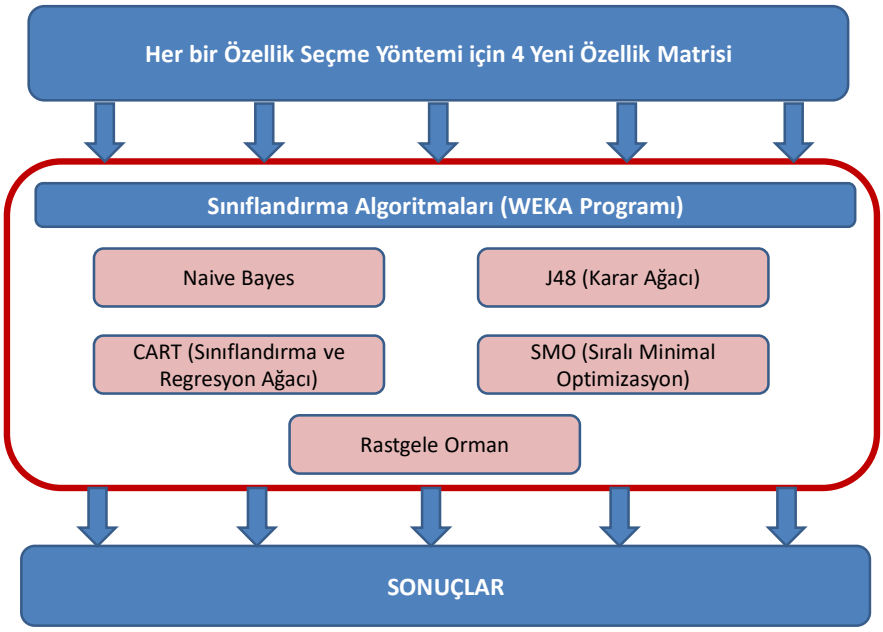
ile başarı elde edilmeye çalışılmıştır. Diğer model olan Özellik Seçimli Derin Öğrenme Modelinde ise özellik seçimini kendi içinde yaptığı için genel model mimarisi üzerinde çalışılmıştır.

3.3.1 Özellik Seçimli Çoklu Makine Öğrenmesi Modeli

Çalışmanın bu adımında oltalama saldırısı amaçlı oluşturulan internet sitesi tespiti veri kümesi üzerinde bazı özel sınıflandırma algoritmalarının performansı analiz edilmiştir. Araştırmadaki temel amaç, farklı sınıflandırma algoritmalarının ve farklı özellik seçme yöntemlerinin birbirleriyle olan en iyi uyumluluğunu bularak sahte internet sitesi tespit doğruluğunu maksimize etmektir. Elde edilen verilerin işlenmesi sonucunda sınıflandırma algoritmalarına girdi oluşturmak amacıyla belirli özellikler seçilmiş ve hangi özelliklerin (özniteliklerin) sınıflandırmada en çok yardımcı olduğu tespit edilmeye çalışılmıştır. Genel olarak özellik seçimi algoritmaları, öğrenme için en önemli özellikleri belirleyerek, verilerin gelecek tahmini için en yararlı olan yönlerine odaklanmaktadır. Özellik seçimi, alakasız ve gereksiz verileri ortadan kaldırmakta ve çoğu durumda öğrenme algoritmalarının performansını iyileştirmektedir.

Çalışmada kullanılacak veri setini oluşturduktan sonra özellik seçme yöntemleri kullanılmıştır. Ardından ilk veri seti özellik seçim yöntemlerinden elde edilen sonuçlara göre yeniden düzenlenmiştir. Yeni veri setleri oluşturulduktan sonra, kullanılan özellik seçme yöntemlerine göre belirlenen her bir sınıflandırma algoritmasının performansı değerlendirilmiştir. Veri setine özellik seçme yöntemlerini uyguladıktan sonraki adımda genel analizin sınıflandırma adımına girdi olarak önceki adımda elde edilen yeni dört veri seti ayrı ayrı kullanılmıştır.

Bu çalışmada, Naïve Bayes, SMO (Sıralı Minimal Optimizasyon), CART (Sınıflandırma ve Regresyon Ağacı), J48 (Karar Ağacı) ve Rastgele Orman olmak üzere beş tür sınıflandırma algoritması üzerine çalışılmıştır. Bu algoritmalar her bir veri kümesinde çalıştırılarak performans analizi için kullanılmıştır. Bu beş sınıflandırma algoritması WEKA yazılımı kullanılarak incelenmiştir. Sonuçların geçerliliğini iyileştirmek amacıyla veri kümelerinde eğitim ve test verilerini bölmek için 10 kat çapraz doğrulama kullanılmıştır. Bu sınıflandırma süreci, çevrimdışı bir eğitim ve test sürecinde geliştirilmiştir. Şekil 3.8'de sınıflandırma genel şeması verilmektedir.



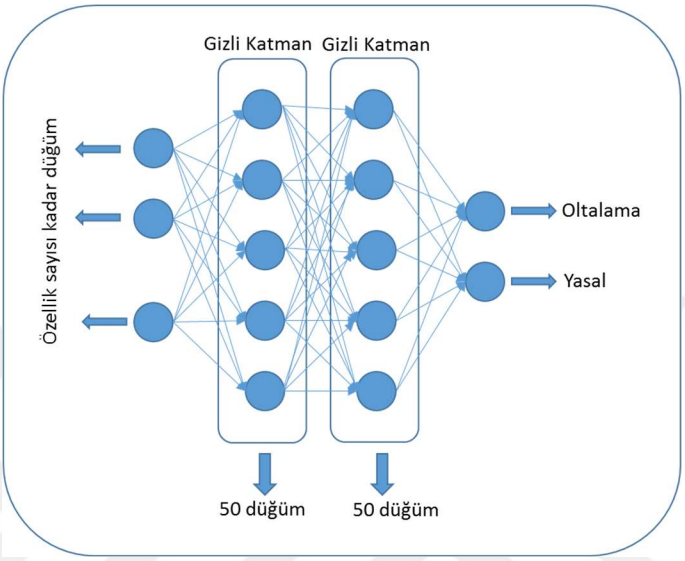
Şekil 3.8: Özellik Seçimli Çoklu Makine Öğrenmesi Modeli Sınıflandırma Şeması

3.3.2 Özellik Seçimli Derin Öğrenme Modeli

Çalışmanın bu modelinde oltalama sitelerinin tespiti için derin öğrenme modeli olarak ileri beslemeli derin sinir ağlarının kullanımı tercih edilmiştir. İleri beslemeli derin sinir ağları temelde çok katmanlı algılayıcıların (MLP) altyapısına dayanmaktadır. MLP, özellikle giriş birimleri ve çıkış katmanının bir ara gizli katmanla birbirine bağlandığı yapıya sahip olan en yaygın kullanılan sinir ağı yapısıdır.

Önerilen derin sinir ağı modeli girdi katmanı, gizli katmanı ve çıktı katmanından oluşmaktadır. Her katmanda işlem birimi olarak kullanılan nöron sayısı farklılık göstermektedir. Girdi katmanında bulunan nöron sayısı, veri setinde yer alan özellik sayısına eşit olmak zorundadır. Bu sayede veri setinde bulunan her bir sütuna karşılık gelen değerler sinir ağında işleme girmek için hazırlanmış olur. Girdi katmanındaki girdi değerleri sinir ağının asıl işlem yapısı olan gizli katmanlara aktarılır. Derin öğrenme yöntemlerinde mimari kurulurken gizli katman ve nöron sayısında belli bir standart olmadığından bu çalışma için yaygın kullanılan yapılar tercih edilmiştir. Son olarak, gizli katmanlarda yapılan işlemlerin sonucunun aktarıldığı bir çıktı katmanı yer almaktadır. Bu katmanda genellikle problem için aranan çözüme göre nöron sayısı

belirlenmektedir. Dolayısıyla bu çalışmada iki olasılığın yer alacağı oltalama sitelerin tespiti gibi bir varlık yokluk kavramı kullanıldığından çıktı katmanında iki nöron tercih edilmiştir. Oltalama sitelerinin tespitinde kullanılmak üzere örnek derin öğrenme mimarisi Şekil 3.9'da verilmiştir.



Şekil 3.9: Özellik Seçimli Derin Öğrenme Örnek Model Mimarisi

Bu modelin başarısına katman ve düğümlerin etkisini araştırmak için 6 adet farklı deneysel mimari hazırlanmıştır. Toplamda bu 6 adet farklı mimariye sahip derin öğrenme modelleri oltalama veri setiyle eğitilmiş ve önerilen modeler arasından en optimum çözüm sağlayan model seçilmek istenmiştir.

4. BULGULAR

Özellik seçimli çoklu makine öğrenmesi modelinde kullanılan dört özellik seçme yönteminin veri setine uygulanması sonucu elde edilen özellik sayıları Tablo 4.1'de verilmektedir.

Tablo 4.1: Özellik Seçme Metotları Bazında Seçilen Özellik Sayıları

Özellik Seçme Metotu	Seçilen Özellik Sayısı
Korelasyon Alt Kümesi	17
Tutarlılık Alt Kümesi	25
Kazanç Oranı Nitelik Tabanlı	36
Relief-F Nitelik Tabanlı	58

Kullanılan özellik seçme yöntemlerinin hangi özellikleri seçtikleri Ek-3'de tablo olarak sunulmaktadır. Bunun birlikte her metotun da seçtiği ortak 9 özellik Tablo 4.2'de verilmektedir.

Tablo 4.2: Seçilen Ortak Özellikler

Seçilen Ortak Özellikler
A_03
A_27
B_79
B_80
C_01
D_03
D_08
E_01
E_02

Tablo 4.3'de özellik seçimli çoklu makine öğrenmesi modelinde test edilen 5 farklı algoritmanın herbir özellik seçme metotu sonrası elde edilen yeni veri setine uygulanması sonucu elde edilen performans metrikleri sonuçları gösterilmektedir.

Tablo 4.3: Sınıflandırma Algoritmalarına Dayalı Analiz Sonuçları

Özellik Seçme Yöntemleri	Sınıflandırma Algoritmaları	Genel Doğruluk	Genel Hata	Geri Çağırma	Kesinlik	F Skoru
ReliefF Nitelik Tabanlı	J48	98,47%	1,53%	98,50%	98,50%	0,9850
Tutarlılık Alt Kümesi		98,46%	1,54%	98,50%	98,50%	0,9850
Kazanç Oranı Nitelik Tabanlı		97,18%	2,82%	97,20%	97,20%	0,9720
CFS Alt Kümesi		96,73%	3,27%	96,70%	96,70%	0,9670
CFS Alt Kümesi	Naive Bayes	88,17%	11,83%	88,20%	90,00%	0,8909
Kazanç Oranı Nitelik Tabanlı		87,08%	12,92%	87,10%	89,40%	0,8824
Tutarlılık Alt Kümesi		83,69%	16,31%	83,70%	87,20%	0,8541
ReliefF Nitelik Tabanlı		81,99%	18,01%	82,00%	86,80%	0,8433
Tutarlılık Alt Kümesi	Rastgele Orman	98,85%	1,15%	98,90%	98,90%	0,9890
ReliefF Nitelik Tabanlı		98,85%	1,15%	98,90%	98,90%	0,9890
Kazanç Oranı Nitelik Tabanlı		97,35%	2,65%	97,40%	97,40%	0,9740
CFS Alt Kümesi		96,86%	3,14%	96,90%	96,90%	0,9690
Tutarlılık Alt Kümesi	CART	98,18%	1,82%	98,20%	98,20%	0,9820
ReliefF Nitelik Tabanlı		98,11%	1,89%	98,10%	98,10%	0,9810
Kazanç Oranı Nitelik Tabanlı		97,10%	2,90%	97,10%	97,10%	0,9710
CFS Alt Kümesi		96,63%	3,37%	96,60%	96,60%	0,9660
ReliefF Nitelik Tabanlı	SMO	96,42%	3,58%	96,40%	96,50%	0,9645
Kazanç Oranı Nitelik Tabanlı		95,95%	4,05%	96,00%	96,00%	0,9600
Tutarlılık Alt Kümesi		95,39%	4,61%	95,40%	95,40%	0,9540
CFS Alt Kümesi		94,67%	5,33%	94,70%	94,70%	0,9470

Bu çalışmada uygulanan özellik seçme yöntemleri, kullanılan sınıflandırma algoritmasına bağlı olarak farklı çıktı değerleri üretmiştir. Örneğin, Naïve Bayes, SMO, CART ve J48 algoritmalarının, sırasıyla CFS Alt Kümesi, Relief-F Nitelik Tabanlı, Tutarlılık Alt Kümesi ve Relie-F Nitelik Tabanlı özellik seçme yöntemleriyle birlikte kullanıldığında en iyi doğruluk sonuçlarını ortaya çıkardığı görülmüştür. Diğer taraftan Rastgele Orman algoritması ise tahmin edilen doğruluk açısından en iyi sonuçları Tutarlılık Alt Kümesi ve Relief-F Nitelik Tabanlı özellik seçme yöntemleriyle vermiştir.

Tablo 4.4: Sınıflandırma Algoritması ve Özellik Seçme Metotu Uyumluluk Tablosu

Sınıflandırma Algoritması	Özellik Seçme Metotu
Naïve Bayes	Korelasyon Alt Kümesi
Sıralı Minimal Optimizasyon	Relief-F Nitelik Tabanlı
Sınıflandırma ve Regresyon Ağacı	Tutarlılık Alt Kümesi
Karar Ağacı	Relief-F Nitelik Tabanlı
Rastgele Orman	Tutarlılık Alt Kümesi

Seçilen özellik sayıları açısından bir değerlendirme yapıldığında ise daha az sayıda özellik seçen Tutarlılık Alt Kümesinin Rastgele Orman algoritması için en uygun özellik seçme yöntemi olduğu sonucuna varılmaktadır. Özellik seçme metotları ile algoritmaların uyumluluk çiftine Tablo 4.4'de yer verilmiştir.

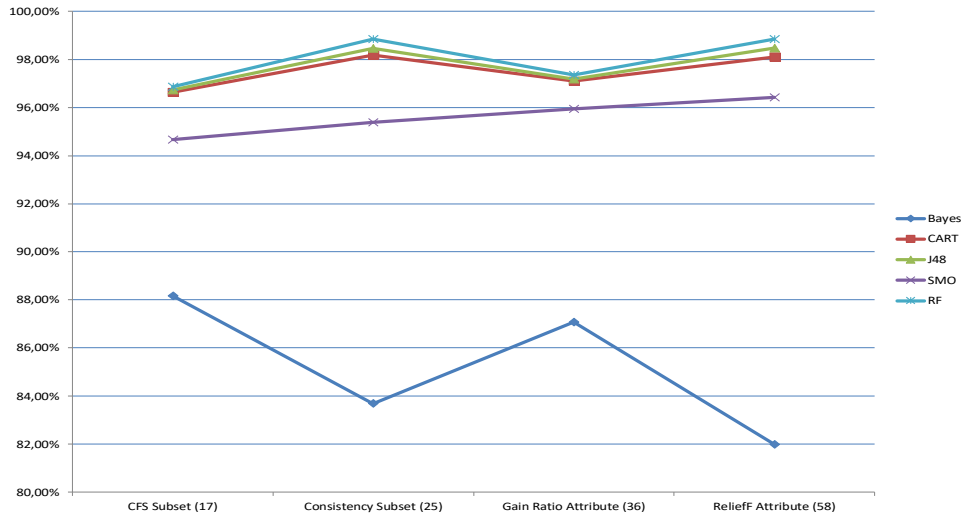
Naïve Bayes algoritması, kendi yüksek performansını %88,17 genel doğruluk metriği ile göstermiştir. SMO algoritması en iyi uyumluluğu %96,42 ile ve CART algoritması ise kendi en iyi performansını %98,18 ile göstermiştir.

J48 algoritması, Relief-F Nitelik Tabanlı özellik seçme algoritması ile kullanıldığında %98,47 ve Tutarlılık Alt Kümesi ile kullanıldığında ise %98,46 yüksek doğruluğunu göstermiştir. Ve son olarak Rastgele Orman algoritması, her iki durumda da %98,85 genel doğrulukla sırasıyla 25 ve 58 özellikli Tutarlılık Alt Kümesi ve Relief-F Nitelik Tabanlı özellik seçme yöntemleri ile en iyi performansı göstermiştir. Bu, analizde elde edilen en yüksek genel doğruluk değeri olmuştur. Analiz sonuçlarına Tablo 4.5'de yer verilmiştir.

Tablo 4.5: Doğruluk Metriğine Dayalı Analiz Sonuçları

Özellik Seçme Yöntemleri	Sınıflandırma Algoritmaları	Genel Doğruluk	Genel Hata	Geri Çağırma	Kesinlik	F Skoru
Tutarlılık Alt Kümesi	Rastgele Orman	98,85%	1,15%	98,90%	98,90%	0,9890
ReliefF Nitelik Tabanlı	Rastgele Orman	98,85%	1,15%	98,90%	98,90%	0,9890
ReliefF Nitelik Tabanlı	J48	98,47%	1,53%	98,50%	98,50%	0,9850
Tutarlılık Alt Kümesi	J48	98,46%	1,54%	98,50%	98,50%	0,9850
Tutarlılık Alt Kümesi	CART	98,18%	1,82%	98,20%	98,20%	0,9820
ReliefF Nitelik Tabanlı	CART	98,11%	1,89%	98,10%	98,10%	0,9810
Kazanç Oranı Nitelik Tabanlı	Rastgele Orman	97,35%	2,65%	97,40%	97,40%	0,9740
Kazanç Oranı Nitelik Tabanlı	J48	97,18%	2,82%	97,20%	97,20%	0,9720
Kazanç Oranı Nitelik Tabanlı	CART	97,10%	2,90%	97,10%	97,10%	0,9710
CFS Alt Kümesi	Rastgele Orman	96,86%	3,14%	96,90%	96,90%	0,9690
CFS Alt Kümesi	J48	96,73%	3,27%	96,70%	96,70%	0,9670
CFS Alt Kümesi	CART	96,63%	3,37%	96,60%	96,60%	0,9660
ReliefF Nitelik Tabanlı	SMO	96,42%	3,58%	96,40%	96,50%	0,9645
Kazanç Oranı Nitelik Tabanlı	SMO	95,95%	4,05%	96,00%	96,00%	0,9600
Tutarlılık Alt Kümesi	SMO	95,39%	4,61%	95,40%	95,40%	0,9540
CFS Alt Kümesi	SMO	94,67%	5,33%	94,70%	94,70%	0,9470
CFS Alt Kümesi	Naive Bayes	88,17%	11,83%	88,20%	90,00%	0,8909
Kazanç Oranı Nitelik Tabanlı	Naive Bayes	87,08%	12,92%	87,10%	89,40%	0,8824
Tutarlılık Alt Kümesi	Naive Bayes	83,69%	16,31%	83,70%	87,20%	0,8541
ReliefF Nitelik Tabanlı	Naive Bayes	81,99%	18,01%	82,00%	86,80%	0,8433

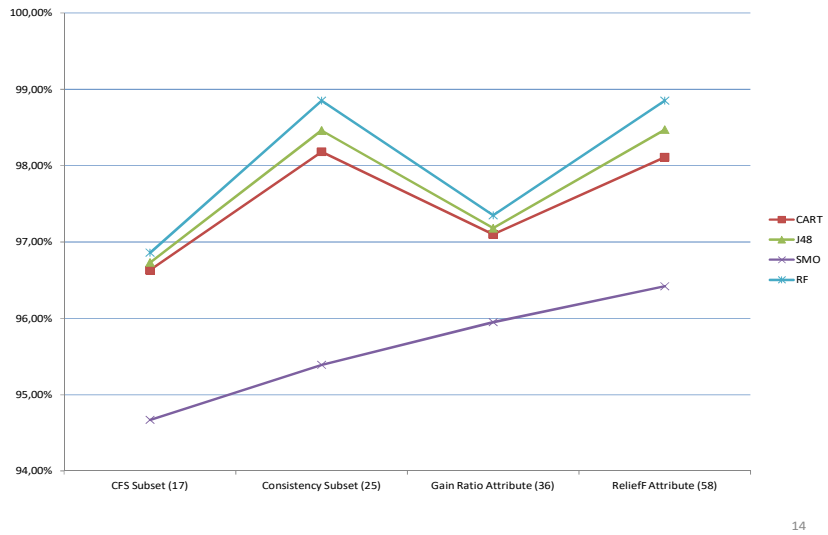
Rastgele Orman algoritması, tüm özellik seçme yöntemlerinde en iyi performansı göstermiştir. İlave olarak, J48 algoritması ikinci, CART algoritması ise üçüncü en iyi sınıflandırma algoritması olarak öne çıkmıştır. SMO algoritması, ilk üç algoritma ile karşılaştırıldığında daha düşük performanslı olarak değerlendirilebilir. Naïve Bayes algoritması ise tüm özellik seçme yöntemlerinde zayıf performans ortaya koymuştur. Bu algoritmaların karşılaştırmalı grafikleri ilk 5 algoritma için Şekil 4.1'de, biraz daha ayrıntılı olacak şekilde ilk 4 algoritma için Şekil 4.2'de sunulmaktadır.



Şekil 4.1: İlk 5 Sınıflandırma Algoritmasının Karşılaştırmalı Grafiği

Korelasyon Alt Kümesi yöntemi, Naïve Bayes dışındaki tüm sınıflandırma algoritmalarında en düşük genel doğrulukla en zayıf performansı sergilemiştir. Diğer taraftan, bu yöntem en iyi performansını Naïve Bayes'te sergilemektedir ancak bu performans diğer tüm sonuçlardan daha düşük çıkmıştır.

Sınıflandırma algoritmalarının değerlendirilmesi ile elde edilen sonuçlar, URL özelliklerine dayalı oltalama saldırılarının tespitinde Rastgele Orman, J48 ve CART algoritmalarının tercih edilebileceğini ortaya koymuştur. Diğer taraftan sonuçlar, bu çalışmanın sınıflandırma analizinde Naïve Bayes algoritmasından kaçınılması gerektiğini düşündürmektedir.



Şekil 4.2: İlk 4 Sınıflandırma Algoritmasının Karşılaştırmalı Grafiği

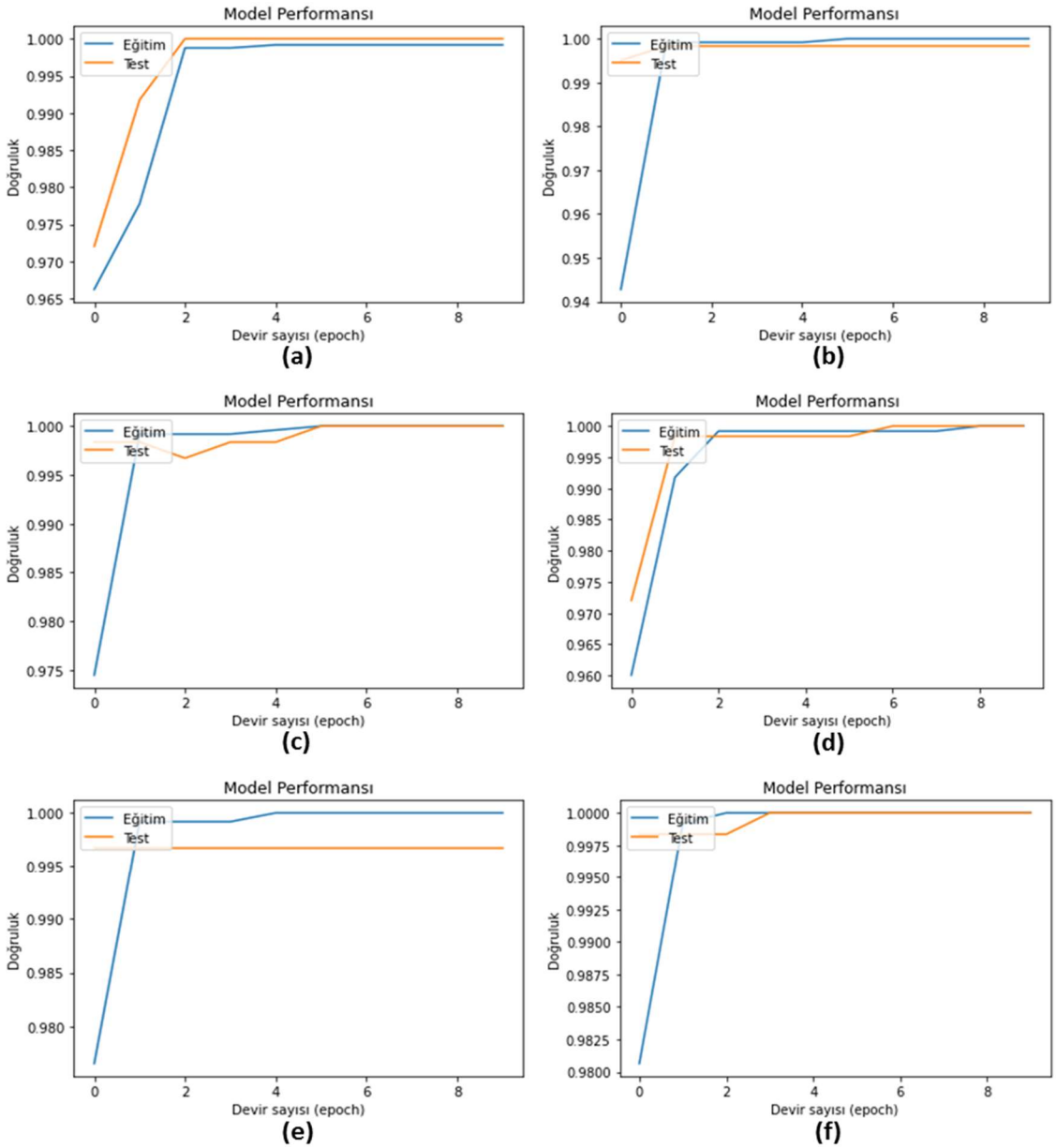
Oltalama saldırılarının etkili bir şekilde engellenebilmesi amacıyla belirlenecek ideal veri kümesi boyutunun ve maksimum performansın, seçilen özellik seçme yöntemine ve sınıflandırma algoritmasına bağlı olarak optimize edilebileceği düşünülmektedir. Mevcut çalışmanın sonuçlarının, oltalama saldırılarının tespitinde gelecekteki araştırmalar için temel referans oluşturabileceği beklenmektedir.

Özellik seçimli derin öğrenme modelinde ise 6 farklı mimari kurgulanmıştır. Kurgulanan mimarilerde katman sayısı ve düğüm sayıları değiştirilerek elde edilen test sonuçlarına Tablo 4.6'da yer verilmiştir.

Tablo 4.6: Derin Öğrenme Modelinin Yurtdışı Veri Seti Test Sonuçları

Mimari	Katman	Düğüm	Doğruluk	Kesinlik	Geri Çağırma	F Skoru
1	2	10	0.9913	0.9965	0.9891	0.9928
2	2	50	0.9892	1.0	0.9824	0.9911
3	2	100	0.9851	1.0	0.9756	0.9876
4	3	10	0.9876	0.9993	0.9803	0.9897
5	3	50	0.9934	0.9993	0.9898	0.9945

Tablo 4.6'da gösterilen değerler incelendiğinde geri çağırma değerlerinde Mimari 6 göze çarpmaktadır. Kesinlik değeri açısından değerlendirme yapıldığında Mimari 2, Mimari 3 ve Mimari 6 en yüksek değeri almakta fakat performans açısından daha az katman ve düğüm Mimari 2'de bulunmaktadır. Modelin doğruluğu ve F skoru değerlendirildiğinde ise Mimari 6 yine öne çıkmaktadır.



Şekil 4.3: a) Mimari 1 (b) Mimari 2 (c) Mimari 3 (d) Mimari 4 (e) Mimari 5 (f) Mimari 6

Diğer taraftan 6 farklı mimari için modelin eğitimi aşamasında doğrulama veri seti ile yapılan performans ölçümü sonuçlarına Şekil 4.3'de yer verilmektedir. Grafikler incelendiğinde 6 mimari arasında çok ciddi farklar olmamakla birlikte Şekil 4.3.b'de yer alan Mimari 2'nin ilk devir sayısında en yüksek doğruluğa ulaştığı ve istikrarlı bir şekilde eğitime devam ettiği görülmektedir.

Modelin oltalama URL sitelerini tespit etmedeki başarı oranının %99,46 olması yapılmış diğer çalışmalardan daha yüksek bir orana sahip olduğunu kanıtlamıştır. Elde edilen başarı oranlarının %90'ın üzerinde olması bu modelin oltalama URL tespiti için uygulanabilir olduğunu göstermektedir.

Çalışmanın bir diğer aşamasında ise yurtiçi kaynaklı veri seti kullanılarak derin öğrenme modeli ülkemize göre modellenmektedir. Bu modelde de 6 farklı mimari kurgulanmıştır. Kurgulanan mimarilerde katman sayısı ve düğüm sayıları değiştirilerek elde edilen test sonuçlarına Tablo 4.7'da yer verilmiştir.

Tablo 4.7: Derin Öğrenme Modelinin Yurtiçi Veri Seti Test Sonuçları

Mimari	Katman	Düğüm	Doğruluk	Kesinlik	Geri Çağırma	F Skoru
1	2	10	0.9987	0.9993	0.9993	0.9993
2	2	50	0.9993	1.0	0.9993	0.9997
3	2	100	0.9993	1.0	0.9993	0.9997
4	3	10	0.9987	0.9993	0.9993	0.9993
5	3	50	0.9987	0.9993	0.9993	0.9993
6	3	100	0.9993	1.0	0.9993	0.9997

Tablo 4.7'de gösterilen değerler incelendiğinde geri çağırma değerlerine katman ve düğüm sayısının etki etmediği görülmektedir. Kesinlik değeri açısından değerlendirme yapıldığında Mimari 2, Mimari 3 ve Mimari 6'nın en yüksek değeri almakta fakat Mimari 2'de daha az katman ve düğüm bulunmaktadır. Modelin doğruluğu ve F skoru değerlendirildiğinde Mimari 2, Mimari 3 ve Mimari 6 daha başarılı olmuştur.

Yurtiçi veri seti için dengeli veri kullanılarak başarı oranları tekrar karşılaştırılmıştır. Bu çerçevede eldeki 107 adet yasal web sitesi arttırılarak veri boyutunda denge oluşturulmuştur. Bu amaçla SMOTE (Sentetik Azınlık Aşırı Örnekleme Tekniği) kullanılmıştır. Bu yöntem azınlık sınıfı için hali hazırda var olanlara dayalı olarak sentezleme öğelerinden oluşur. Azınlık sınıfından rastgele bir nokta seçip bu nokta için en yakın komşuları hesaplayarak çalışmaktadır. Sentetik noktalar, seçilen nokta ile komşuları arasına eklenmektedir. Bu yöntemle 3035 adet

sentetik yasal veri oluşturulmuştur. Bu çalışma sonucunda 3142 yasal 4427 oltalama verisi ile test tekrarlanmıştır.

Bu çalışmada da katman ve düğüm sayıları değiştirilerek 6 farklı model kurulmuştur. Bu mimarilerde katman sayısının ve düğüm sayısının model eğitimindeki başarısı araştırılmıştır. 6 farklı mimari ile elde edilen deney sonuçlarına Tablo 4.8'de yer verilmiştir.

Tablo 4.8: Derin Öğrenme Modelinin Yurtiçi Veri Seti Test Sonuçları (Dengeli Veri)

Mimari	Katman	Düğüm	Doğruluk	Kesinlik	Geri Çağırma	F Skoru
1	2	10	0.9943	0.9930	0.9961	0.9945
2	2	50	0.9939	0.9945	0.9937	0.9941
3	2	100	0.9939	0.9961	0.9922	0.9941
4	3	10	0.9931	0.9907	0.9961	0.9934
5	3	50	0.9951	0.9976	0.9930	0.9953
6	3	100	0.9935	0.9937	0.9937	0.9937

Tablo 4.8'de gösterilen değerler incelendiğinde geri çağırma değerlerinde Mimari 1 ve Mimari 4 biraz öne çıkmaktadır. Kesinlik değeri açısından değerlendirme yapıldığında Mimari 5 en yüksek değeri almaktadır. Modelin doğruluğu ve F skoru değerlendirildiğinde Mimari 5 yine daha başarılı olmuştur.

Sentetik olarak üretilen verilerle elde edilen sonuçlar değerlendirildiğinde sistemin genel doğruluk oranında bir miktar düşme gözlenmiştir. Hem dengeli hem de dengesiz veri sonuçlarına bakıldığında sistemin oldukça başarılı sonuçlar ürettiği görülmektedir.

5. TARTIŞMA VE SONUÇ

Oltalama saldırıları, kimlik avcılarının sahte bir internet sitesini yasal bir internet sitesi gibi göstererek internet kullanıcılarını inandırdığı saldırılardır. Son dönemde kişilerden hassas ve değerli bilgiler elde etmek amacıyla, bilgi girişi yapılabilen yasal internet sitelerini taklit etmek için pek çok oltalama internet siteleri oluşturulmuştur. Kişilerin özellikle finansal hassas verilerini elde etmek amacıyla oluşturulan bu sitelerde anlık olarak gerçekleştirilen bu tür saldırılar oltaya gelen kişilerin maddi varlığı için ciddi tehditler oluşturmaya devam etmektedir.

Bugüne kadar çeşitli kaynaklar tarafından sunulan istatistikler oltalama sitelerinin sayısının sürekli artma eğiliminde olduğunu göstermektedir. Yaşanan toplam oltalama vakaları incelendiğinde de son yıllara doğru ciddi bir artış olduğu dikkat çekmektedir. Dolayısıyla günümüzde oltalama saldırılarının hala trend saldırılardan biri olarak yerini koruduğu gözlemlenebilmektedir. Kaspersky'in 2020 spam ve oltalama raporuna göre kurumsal sektöre yönelik hedefli saldırıların genel olarak artan eğiliminin sonraki yıllarda da devam edeceği belirtilmiştir. Bunun sebebi ise popüler hale gelen uzaktan çalışma modelinin çalışanları daha savunmasız hale getirmesidir.

Dünya genelinde sık kullanılan siber saldırılardan biri olan oltalama sitelerinin ülkemizde de yaygın bir dolandırıcılık aracı olarak kullanıldığı ve önemli bir artışta olduğu görülmektedir. Emniyet Genel Müdürlüğü Siber Suçlarla Mücadele Daire Başkanlığı ile Bilgi Teknolojileri ve İletişim Kurumu Başkanlığı tarafından açıklanan verilere göre kapatılan oltalama sitelerinin sayısı sürekli artmaktadır. Ayrıca dolandırıcılar tarafından oltalama sitelerinin genellikle trend olan olaylar için oluşturulduğu ve çoğunlukla kamu kurumları ile bankalara ait yasal sitelerin kopyalanarak hassas verilere erişildiği gözlemlenmektedir.

Ulusal Siber Olaylara Müdahale Merkezinin sitesinde yer alan tespit edilmiş oltalama URL adresleri incelendiğinde bankacılık sektörünü hedef alacak anahtar kelimelerin sıkça kullanıldığı görülmektedir. Oltalama vakalarında yaşanan maddi kayba gelindiğinde bu oranın da göz ardı edilemeyecek kadar büyük olduğu bankaların dolandırıcılık verilerinden ortaya çıkmaktadır.

Elde edilen istatistikler ve yukarıda paylaşılan bilgiler genel olarak değerlendirildiğinde oltalama saldırıları gerek küresel çapta gerek Türkiye çapında mücadele edilmesi gereken kritik siber güvenlik konularından birisi olmaya devam etmektedir.

Oltalama siteleri genellikle bireysel verileri ele geçirmekte ve kişileri maddi zarara uğratmaktadır. Bu yüzden bu sitelerin kişisel kullanım seviyesine indirgenerek engellenmesi çok önemlidir.

Oltalama sitelerinin engellenmesine yönelik çalışmalara başlamadan tespit başarısını arttırmak amacıyla öncelikle bu sitelerin belirgin ve ortak özellikleri tespit edilmelidir. Oltalama sitelerin en belirgin tespit edilebilir özelliklerinden birisi de URL içeriği olmaktadır.

Bu çalışmada, bir internet sitesinin URL özelliklerini kullanarak oltalama amacıyla oluşturulmuş sahte sayfaların gerçek sayfalardan ayırt edilerek tespit edilmesi amaçlanmıştır. Bu amaç doğrultusunda literatürde kabul edilen performans metriklerine bağlı olarak yüksek başarı oranına sahip bir sınıflandırıcı iş akışı modeli önerisi hedeflenmiştir.

Oltalama sitelerinin tespiti için çeşitli yöntemler denenmekte olup bunların içinde yaygın kullanılanlardan birisi de makine öğrenmesi yöntemleridir. Bu çalışmada oltalama amaçlı kullanılan URL adreslerinin tespiti için 2 farklı model oluşturularak karşılaştırılmıştır.

Çalışmanın ilk modelinde oltalama saldırısı amaçlı oluşturulan internet sitesi tespiti veri kümesi üzerinde bazı özel sınıflandırma algoritmalarının performansı analiz edilmiştir. Araştırmadaki temel amaç, farklı sınıflandırma algoritmalarının ve farklı özellik seçme yöntemlerinin birbirleriyle olan en iyi uyumluluğunu bularak sahte internet sitesi tespit doğruluğunu maksimize etmektir. Bu çalışmada değerlendirilen özellik seçme yöntemleri, nitelik tabanlı ve alt küme tabanlı olarak adlandırılan iki yaklaşıma ayrılmaktadır. Nitelik tabanlı özellik seçme yöntemleri, her bir özelliği ayrı ayrı değerlendirirken, alt küme tabanlı özellik seçim yöntemlerinde, özellikler arasındaki bağımlılıklar dikkate alınır ve tüm özellik kümesini en iyi şekilde temsil edecek şekilde alt kümeler oluşturulmaktadır. Bu çalışmada, Naïve Bayes, SMO (Sıralı Minimal Optimizasyon), CART (Sınıflandırma ve Regresyon Ağacı), J48 (Karar Ağacı) ve Rastgele Orman olmak üzere beş tür sınıflandırma algoritması üzerine çalışılmıştır. Bu algoritmalar her bir veri kümesinde çalıştırılarak performans analizi için kullanılmıştır. Bu beş sınıflandırma algoritması WEKA yazılımı kullanılarak incelenmiştir. Rastgele Orman algoritması, tüm özellik seçme yöntemlerinde en iyi performansı göstermiştir. İlave olarak, J48

algoritması ikinci, CART algoritması ise üçüncü en iyi sınıflandırma algoritması olarak öne çıkmıştır. SMO algoritması, ilk üç algoritma ile karşılaştırıldığında daha kötü olarak değerlendirilebilir. Naïve Bayes algoritması ise tüm özellik seçme yöntemlerinde zayıf performans ortaya koymuştur. En iyi sonuç doğruluk parametresi özelinde %98,85 olmuştur.

Çalışmanın diğer modelinde ise oltalama sitelerinin tespiti için derin öğrenme modeli olarak ileri beslemeli derin sinir ağlarının kullanımı tercih edilmiştir. İleri beslemeli derin sinir ağları temelde çok katmanlı algılayıcıların altyapısına dayanmaktadır. Bu modelin başarısına katman ve düğümlerin etkisini araştırmak için 6 adet farklı deneysel mimari hazırlanmıştır. Toplamda bu 6 adet farklı mimariye sahip derin öğrenme modelleri oltalama veri setiyle eğitilmiş ve en optimum çözüm tespit edilmiştir. En iyi sonuç doğruluk parametresi özelinde %99,46 olmuştur.

Bu modellerle yapılan testlerde kurgulanan özellik seçimli derin öğrenme modeli diğer modellerden daha başarılı olmuştur. Literatürde yer alan oltalama URL adreslerinin derin öğrenmeyle tespiti üzerine yapılmış diğer çalışmalar incelendiğinde ise bu modelin diğer derin öğrenme modellerinin sahte URL tespitinden daha iyi sonuçlar elde ettiği çıkarılmaktadır. Literatürde yer alan çalışmalar göz önünde bulundurulduğunda oltalama URL adreslerinin tespiti için derin öğrenme modeli kullanımının doğru bir yaklaşım olduğu kanıtlanmaktadır. Derin öğrenme mimarisinin tercih edilme sebebi sinir ağlarının karakteristik yapısı sayesinde kendi içerisinde hem özellik çıkarma hem de sınıflandırma işlemini bir arada bulundurması olmuştur.

Önerilen model oltalama URL adreslerinin tespitinde %99,46 doğruluğa ulaşmış ve diğer çalışmalar içinde daha yüksek bir başarı göstermiştir. Modelin başarılı olmasında en büyük etkenlerden birisi de URL adreslerini iyi analiz eden bir özellik setinin uygulanmış olmasıdır.

Bu çalışmada karşılaşılan ve yeni ortaya çıkan oltalama saldırısı taktiklerini dikkate alacak bir sistem tasarlanmış ve bu sistem özellik kümesinin istenildiği gibi değiştirilmesine imkân verecek şekilde esnek ve modüler hale getirilmiştir.

Ayrıca analizi yapılan 2 model arasından tercih edilen derin sinir ağları modelinin Türkiye verisi ile de testi gerçekleştirilmiştir. USOM kaynaklı Türkiye'de bulunan oltalama URL'lerinden oluşmuş veri seti kullanılarak ülkemize özel bir model sunulmuştur. Uygulanacak veri setini ayarlamak amacıyla önceki modeller için bahsedilen oltalama URL'leri için çıkarılan özellikler Türkiye'de kullanılan sahte sitelere göre tekrardan uygun şekilde

düzenlenmiştir. Türkçe'ye özel özgün kelime modeli tasarımı sayesinde dinamik olarak en çok kullanılan URL kelimeleri çıkarımı sağlanmıştır. Bu test için de toplamda 6 adet farklı mimariye sahip derin öğrenme modelleri oltalama veri setiyle eğitilmiş ve en optimum çözüm tespit edilmiştir. En iyi sonuç doğruluk parametresi özelinde %99,93 olmuştur.

Bununla birlikte bu model ile çalışma boyunca kullanılan klasik makine öğrenmesi yöntemlerine ek olarak derin öğrenme yöntemi olan derin sinir ağları modelinin test edilmesi, derin sinir ağları sayesinde benzer çalışmalara nispeten daha küçük veri setinde geneli temsil eden sonuçların elde edilmesi, oltalama sitesi saldırıları için Türkiye'ye uyarlanmış oltalama URL veri setinin hazırlanması ve oltalama URL'leri içinde geçen göz alıcı kelimeleri otomatik çıkaran özgün bir kelime modülü uygulaması geliştirilmesi sağlanmıştır.

Bu çalışmada kullanılan verilerin kaynakları açık ve net bir dille ifade edilerek bir çalışmanın izlenebilirliği prensibini uygun davranılmış ve yöntemler ve sonuçlar her aşamada izah edilmiştir. Tezin ana konusu itibariyle çalışmanın objektif başarısı için gerçek veriler kullanılarak, özgün yazılım kodlama yapılarak ve alanında kabul gören yazılım araçlarıyla doğruluk ve başarı oranı hesaplanmıştır. Çalışmanın her aşamasında kullanılan veriler için kapsamlı bir veri toplama süreci yürütülmüştür. Gerek web sürünmesi yöntemi, gerek betik yazma gerekse manuel çalışmalarla en doğru çalışmayı yapmak üzere gerçek veriler toplanmıştır.

Literatürde yer alan bazı çalışmalarda kullanılan üçüncü parti kullanımı bu çalışmada da değerlendirilmiştir. Bu süreçte özellikle üçüncü parti destekli veri toplamada profesyonel destek talebinde bulunulan Domaintools şirketinden talep edilen URL adreslerinin whois sorgusu için teklif edilen 12.000 ABD doları karşılığı hizmet yerine söz konusu veriler kendi özgün çalışmamızla toplanmış ve söz konusu özelliğin oltama saldırılarındaki etkisi analiz edilmiştir. Sonuç olarak önerilen modelde üçüncü parti kullanımı olmadan da yüksek başarı elde edilmiş ve sistemin başarısını etkileyebilecek erişilebilirlik ve internet hızından etkilenme problemleri bu çalışmada önerilen modelde söz konusu olmamıştır.

Dolandırıcılar tarafından oltalama sitelerinin genellikle trend olan olaylar için oluşturulduğu ve çoğunlukla kamu kurumları ile bankalara ait yasal sitelerin kopyalanarak hassas verilere erişildiği gözlemlenmektedir. Trend olaylara göre kendini uyarlayamayan modeller zaman içerisinde problemin çözümünde yetersiz kalmaktadır. Kendi kendine öğrenmeyen sistemler

kullanıldığında pratik hayatta başarı sağlamak mümkün olmayacaktır. Her defasında yeni verilerle ve yeni özeliklerle çevrimdışı çalışmak yerine yapay zekâ tabanlı sistemlerin geliştirilmesi önem arz etmektedir ve bu çalışmanın ana hedeflerinden biri de çevrim içi hızlı karar alabilen sistemler geliştirmek olmuştur. Bu çalışmada yer alan kelime işleme sürecinde derin öğrenme modeli geliştirilmiş olup sistemin kendi kendine öğrenmesi de kurgulanmıştır. Çalışmanın ülkemiz verilerine dayanarak yapılan analiz bölümünde ise son zamanda tespit edilen gerçek verilerle pandemi döneminde gerçekleşen oltalama saldırılarına ilişkin problemlere de çözüm üretilmiştir.

Bu çalışma boyunca elde edinilen kazanımlar sonrasında URL ve alan adı standardının oltalama saldırılarının önlenmesinde önemli olduğu sonucuna varılmaktadır. Mevcut durumda yasal internet sitesi olmasına rağmen sahte site gibi filtrelere takılan URL'lerin standart dışı isimlendirilmesi dikkat çekmektedir. Bu kapsamda yasal şirketler için açık ve anlaşılır olacak şekilde alan adı standardı getirilmesi tavsiye edilmektedir.

Çalışmanın son bölümünde Türkçe veri seti kullanılarak analiz yapılmıştır. Bununla birlikte oltaya gelmesi beklenen kişinin ana dili üzerinden oltalama saldırısı yapmak daha yaygın olsa da kullanılan dilden bağımsız olarak kişilerin oltaya gelmesi mümkün olabilmektedir. Bu düşünceyle sonraki çalışmalarda dil ayrımı olmaksızın Türkçe-İngilizce veri seti kullanılarak hibrit çalışma yapılabileceği ve sonuçların analiz edilebileceği düşünülmektedir.

Oltalama saldırılarının tespitinin gerçek hayattaki yansımasına katkıda bulunacak kullanıcı farkındalığının artırılması çalışmalarına bu tezde yer verilmemiştir. Konunun sosyal bilimler açısından değerlendirilmemesi bu çalışmanın kısıtlılıklarından biri olmuştur. Bu kapsamda, bundan sonra yapılabilecek çalışmalarda konunun insan davranış faktörü açısından değerlendirilerek katkı sağlayabilecek farklı öneriler getirilmesi mümkün görünmektedir.

Bir sonraki çalışma olarak derin öğrenme tabanlı çalışan anlık oltalama sitesi tespiti yapan bir sistemin bir internet tarayıcısına entegrasyonu planlanmaktadır. Böylece, kullanıcılar tarayıcılarına bir URL girdiği zaman ya da bir URL tıklayarak açmak istediklerinde URL adresindeki siteye yönlendirilmeden önce bu adresin güvenilirliği için kullanıcıya risk puanı gösterilebilecek ve olası zarar oluşmadan oltalama saldırısının önüne geçilebilecektir.

KAYNAKLAR

Acharya T.D., Lee D.H., Yang I.T., and Lee J.K., 2016, Identification of Water Bodies in a Landsat OLI Image Using a J48 Decision Tree, Sensors, 16(7):1075, https://doi.org/10.3390/s16071075.

Adebowale, M.A., Lwin, K.T. and Hossain, M.A., 2020, Intelligent phishing detection scheme using deep learning algorithms, *Journal of Enterprise Information Management*, https://doi.org/10.1108/JEIM-01-2020-0036.

Abdelhamid N., Ayesh A., and Thabtah F., 2014, "Phishing detection based Associative Classification data mining", *Expert Systems with Applications*, vol. 41, Issue 13, pp. 5948–5959.

Abraham, D., and Raj N.S., 2014, "Approximate string matching algorithm for phishing detection," in Advances in Computing, *Communications and Informatics (ICACCI), International Conference*, pp.2285-2290.

Abunadi, A., Akanbi O., and Zainal A., 2013, "Feature extraction process: A phishing detection approach," Intelligent Systems Design and Applications (ISDA), *2013 13th International Conference*, pp.331-335.

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, pp. 7913–7921.

Aburrous M., and Khelifi A., 2013, "Phishing Detection Plug-In Toolbar Using Intelligent Fuzzy-Classification Mining Techniques," *The International Journal of Soft Computing and Software Engineering [JSCSE],* Vol. 3, No. 3, Special Issue: The Proceeding of International Conference on Soft Computing and Software Engineering, pp. 54-61.

Afroz, S., and Greenstadt R., 2011, "PhishZoo: Detecting Phishing Websites by Looking at Them," in Semantic Computing (ICSC), *Fifth IEEE International Conference*, pp.368-375.

Aggarwal, A., Rajadesingan, A., and Kumaraguru, P., 2012, "PhishAri: Automatic realtime phishing detection on twitter," *eCrime Researchers Summit* (eCrime), pp.1,12, 23-24.

Alkhozae M. G., and Batarfi O. A., 2011, " Phishing Websites Detection based on Phishing Characteristics in the Webpage Source Code" *International Journal of Information and Communication Technology Research*, Volume 1 No. 6, pp. 283-291.

Anti-Phishing Working Group (APWG), 2015, *"Phishing Activity Trends Report - 3rd Quarter 2014",* www.apwg.org, Published March 30, 2015.

Anti-Phishing Working Group (APWG), 2015, *"Global Phishing Survey 2H2014: Trends and Domain Name Use",* www.apwg.org, Published March 30, 2015.

Aydin M., Butun I., Bicakci K. and Baykal N., 2020, Using Attribute-based Feature Selection Approaches and Machine Learning Algorithms for Detecting Fraudulent Website URLs, *10th Annual Computing and Communication Workshop and Conference (CCWC),* pp. 0774-0779, doi: 10.1109/CCWC47524.2020.9031125.

Aydin, M., and Baykal, N., 2015, Feature Extraction and Classification Phishing Websites Based on URL, *IEEE Conference on Communications and Network Security (CNS) Italy*, pp. 769-770.

Aydin, M., and Baykal, N., 2014, Improved Phishing Attack Detection Using Hybrid Machine Learning Algorithms Based On URL, *Structure, International Conference on Software Engineering and Digital Technology (SEDT) Japan*.

Bahnse A. C., Bohorquez E. C., Villegas S., Vargas J. and Gonález F. A., 2017, Classifying phishing URLs using recurrent neural networks, Proc. *IEEE APWG Symp. Electron*. Res. (eCrime), pp. 1-8.

Balamuralikrishna T., Raghavendrasai N., and Sukumar M. S., 2012, "Mitigating Online Fraud by Ant phishing Model with URL & Image based Webpage Matching", *International Journal of Scientific & Engineering Research*, vol. 3, Issue 3, pp. 1-6.

Barraclough P.A., Hossain M.A, Tahir M.A, Sexton G., and Aslam N., 2013, "Intelligent phishing detection and protection scheme for online transactions," *Expert Systems with Applications*, 40, 4697–4706.

Basnet R. B., Sung A. H., and Liu Q., 2012, *"Feature Selection for Improved Phishing Detection"*, H. Jiang et al. (Eds.): IEA/AIE 2012, LNAI 7345, pp. 252–261. Springer-Verlag Berlin Heidelberg.

Basnet R. B., and Sung A. H., 2012, "Mining Web to Detect Phishing URLs,", *11th International Conference on Machine Learning and Applications.*

Basnet R. B., Sung A. H., and Liu Q., 2011, "Rule-Based Phishing Attack Detection", *International Conference on Security and Management (SAM)*, Las Vegas, NV.

Belabed A., Aimeur E., and Chikh, A., 2012, "A Personalized Whitelist Approach for Phishing Webpage Detection," Availability, Reliability and Security (ARES), *Seventh International Conference,* pp. 249-254.

Bengio Y., LeCun Y., and Hinton G., 2015, "Deep Learning", Nature, 521 (7553): 436–444, doi:10.1038/nature14539

Breiman L., Friedman J., Stone C.J., and Olshen R.A., 1984, Classification and Regression Trees Chapman and Hall/CRC.

Breiman L., 2001, Random forests. Mach. Learn. 45, 5–32, doi: 10.1023/A:1010933404324.

Butt M. H. F., Li J. P., Saboor T., Arslan M., and Butt M.A.F., 2021, "Intelligent Phishing Url Detection: A Solution Based On Deep Learning Framework," *18th International*

Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 434-439, doi: 10.1109/ICCWAMTIP53232.2021.9674162.

Chang C. C., and Lin C. J., 2011, "LIBSVM: A library for support vector machines", ACM Transactions on Intelligent Systems and Technology, 2(3).

Chen C. M., Huang J. J., and Ou Y. H., 2014, "Efficient suspicious URL filtering based on reputation", *Journal of Information Security and Applications I – II*.

Chen, W., Zhang, W., Su, Y., 2018, Phishing Detection Research Based on LSTM Recurrent Neural Network, *ICPCSEE Communications in Computer and Information Science*, pp. 638–645, vol 901. Springer, Singapore. https://doi.org/10.1007/978-981-13-2203-7_52.

Chen X., Bose I., Leung A. C. M., and Guo C., 2011, "Assessing the severity of phishing attacks: A hybrid data mining approach" *Decision Support Systems 50*, pp. 662–672

Choon Lin T., Kang Leng C., and San Nah S., 2014, "Phishing website detection using URL-assisted brand name weighting system," *Intelligent Signal Processing and Communication Systems (ISPACS), International Symposium*, pp.054,059.

Chu W., Zhu B.B., Xue F., Guan X., and Cai Z., 2013, "Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs," *Communications (ICC), IEEE International Conference*, pp.1990-1994.

Dave V. S., and Dutta K., 2014, Neural network-based models for software effort estimation: a review, Artif. Intell. Rev., 42 (2), pp. 295-307.

Fahmy H.M.A., and Ghoneim S.A., 2011, "PhishBlock: A hybrid anti-phishing tool," Communications, *Computing and Control Applications (CCCA), International Conference*, pp. 1-5.

Faris H., and Yazid S., 2021, "Phishing Web Page Detection Methods: URL and HTML Features Detection" IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pp. 167-171, doi: 10.1109/IoTaIS50849.2021.9359694.

Feroz, M.N., and Mengel, S., 2014, "Examination of data, rule generation and detection of phishing URLs using online logistic regression," *Big Data IEEE International Conference,* pp.241-250.

Geng G. G., Lee X. D., Wang W., and Tseng S. S., 2013, "Favicon - a clue to phishing sites detection," *in eCrime Researchers Summit (eCRS),* pp.1-10.

Gowthama R., and Krishnamurthib I, 2014, "A comprehensive and efficacious architecture for detecting phishing webpages", *Computers & Security*, vol. 40, pp. 23–37.

Gupta, N., Aggarwal, A., and Kumaraguru, P., 2014, "bit.ly/malicious: Deep dive into short URL based e-crime detection," Electronic Crime Research (eCrime), *APWG Symposium*, pp.14-24.

Gündoğmuş Y. N., 2020, 14 bin 120 'oltalama' sitesi erişime kapatıldı https://www.aa.com.tr/tr/turkiye/14-bin-120-oltalama-sitesi-erisime-kapatildi/1850849. (Erişim Tarihi: 22.03.2021)

Huang H., Qian L., and Wang Y., 2012, "A SVM-based Technique to Detect Phishing URLs," *Information Technology Journal* 11(7), 921-925.

Huang T. J., 2017, Imitating the brain with neurocomputer a "new" way towards artificial general intelligence, Int. J. Autom. Comput., 14 (5), pp. 520-531.

Islam M., and Nihad C., 2016, "Phishing websites detection using machine learning based classification techniques", International Conference on Advanced Information and Communication Technology At: Chittagong, Bangladesh.

John C. P., 1998, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Advances in Kernel Methods - Support Vector Learning

Kabalcı E., 2014, Yapay Sinir Ağları, https://ekblc.files.wordpress.com/2013/09/ysa.pdf

Kazemian H.B., and Ahmed S., 2015, "Comparisons of machine learning techniques for detecting malicious webpages", Expert Systems with Applications, volume 42, issue 3, pages 1166-1177, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2014.08.046.

Khonji M., Iraqi Y., and Jones A., 2011, "Lexical URL Analysis for Discriminating Phishing and Legitimate Websites", *International Conference for Internet Technology and Secured Transactions (ICITST),* pp. 422 – 427.

Khonji M., Iraqi Y., and Jones, A., 2013, "Phishing Detection: A Literature Survey," *Communications Surveys & Tutorials, IEEE* , vol.15, no.4, pp.2091-2121.

Khonji M., Jones A., and Iraqi Y., 2011, "A Novel Phishing Classification Based On URL Features" *IEEE GCC Conference and Exhibition*, pp. 221-224.

Kingma, D.P., & Ba, J., 2015, Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kordestani, H., and Shajari, M., 2013, "An entice resistant automatic phishing detection," Information and Knowledge Technology (IKT), *2013 5th Conference*, pp.134,139.

Kumar, A., Roy S.S., Saxena, S., and Rawat S. S. S., 2013, "Phishing Detection by Determining Reliability Factor Using Rough Set Theory," *Machine Intelligence and Research Advancement (ICMIRA), International Conference*, pp. 236-240.

Lakshmi L., Reddy M.P., and Santhaiah, C. et al, 2021, *Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM*, https://doi.org/10.1007/s11277-021-08196-7.

Le A., Markopoulou A., and Michalis F., 2011, "PhishDef: URL names say it all," INFOCOM, *2011 Proceedings IEEE* , pp.191,195.

Li Y., Xiao R., Feng J., and Zhao L., 2013, "*A semi-supervised learning approach for detection of phishing webpages*".

Lin M., Chiu C., Lee Y., and Pao H., 2013, "Malicious URL filtering — A big data application," *Big Data, IEEE International Conference*, pp.589-596.

Luca Z., Thomas S., and Gaetano Z., 2006, Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems. J. Mach. Learn. Res. 7, 1467–1492.

Jahnavi M., 2017, Introduction to Neural Networks, Advantages and Applications, towards Data Science, https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207.

James, J., Sandhya, L., and Thomas, C., 2013, "Detection of phishing URLs using machine learning techniques," *Control Communication and Computing (ICCC) International Conference*, pp.304-309.

Marchal, S., Francois, J., State, R., and Engel, T., 2014, "PhishStorm: Detecting Phishing with Streaming Analytics," *Network and Service Management, IEEE Transactions*, vol.11, no.4, pp.458-471.

Manek, A.S., Sumithra, V., Shenoy, P.D., Mohan, M.C., Venugopal, K.R., and Patnaik, L.M., 2014, "DeMalFier: Detection of Malicious web pages using an effective classifier," *Data Science & Engineering (ICDSE), International Conference*, pp.83-88.

Martin A., Anutthamaa N. B., Sathyavathy M., Saint Francois M. M., and Venkatesan P., 2011, "A Framework for Predicting Phishing Websites Using Neural Networks" *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 2, ISSN (Online): 1694-0814.

Maurer M. E., and Höfer L., 2012, "*Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity against Phishing*" Springer-Verlag Berlin Heidelberg, pp. 414-426.

Mitchell T. M., 1997, Machine Learning, Publisher: New York: McGraw-Hill, Edition: Print book: English.

Mohammad R. M., Thabtah F.and Mccluskey L., 2014, Predicting phishing websites based on self-structuring neural network, *Neural Comput*. Appl., vol. 25, no. 2, pp. 443-458.

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*, Article ID 8241104, 24 pages. https://doi.org/10.1155/2021/8241104.

Nguyen L. A. T.; To B. A., Nguyen H. K, and Nguyen M. H. (2014), "A novel approach for phishing detection using URL-based heuristic," Computing, *Management and Telecommunications (ComManTel), International Conference*, pp.298-303.

Number of unique phishing sites detected worldwide from 3rd quarter 2013 to 1st Quarter 2020, https://www.statista.com/statistics/266155/number-of-phishing-domain-names-worldwide/ (Erişim Tarihi: 15.09.2020)

Phishing By Industry Benchmarking Report, 2020, http://www. https://info.knowbe4.com/phishing-by-industry-benchmarking-report, (Erişim Tarihi: 15.03.2021)

Quinlan R., 1993, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo.

Ranganayakulua D., and Chellappan C., 2013, "Detecting Malicious URLs in E-mail – An Implementation", *AASRI Conference on Intelligent Systems and Control AASRI Procedia*, vol. 4, pp. 125–131.

Sadi M. S., Khan M. R., Islam M., Srijon S. B., and Mia M. H., 2012, "Towards Detecting Phishing Web Contents for Secure Internet Surfing", *APR International Conference on Informatics*, Electronics & Vision.

Sainlez M., Heyen G., 2011, "Recurrent neural network prediction of steam production in a Kraft recovery boiler", Editor(s): E.N. Pistikopoulos, M.C. Georgiadis, A.C. Kokossis, *Computer Aided Chemical Engineering*, Elsevier, vol. 29, pp. 1784-1788, ISSN 1570-7946, ISBN 9780444538956, https://doi.org/10.1016/B978-0-444-54298-4.50135-5.

Santhana Lakshmi V., and Vijaya M. S., 2011, "Efficient prediction of phishing websites using supervised learning algorithms*" International Conference on Communication Technology and System Design, Procedia Engineering 30*, pp. 798 – 805.

Selvaganapathy S., Nivaashini M., and Natarajan H., 2018, Deep belief network based detection and categorization of malicious URLs, Information Security Journal: *A Global Perspective*, 27:3, pp. 145-161, doi: 10.1080/19393555.2018.1456577.

Shahriar H., and Zulkernine M., 2011, "Information Source-based Classification of Automatic Phishing Website Detectors," *2011 IEEE/IPSJ International Symposium on Applications and the Internet*.

Shcherbakova T., Vergelis M., and Demidova N., "Spam and Phishing in the First Quarter of 2015," https://securelist.com/analysis/quarterly-spam-reports/69932/spam-and-phishing-in-the-first-quarter-of-2015/, (Erişim Tarihi: 13.05.2015)

Sorio E., Bartoli A., and Medvet E., 2013, "Detection of Hidden Fraudulent URLs within Trusted Sites Using Lexical Features," Availability, *Reliability and Security (ARES), Eighth International Conference,* pp.242-247.

Sountharrajan S., Nivashini M., Shandilya S.K., Suganya E., Bazila Banu A., Karthiga M., 2020, *Dynamic Recognition of Phishing URLs Using Deep Learning Techniques*. In: Shandilya S., Wagner N., Nagar A., Advances in Cyber Security Analytics and Decision Systems. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-19353-9_3.

Spam and phishing in 2020, Kaspersky, https://securelist.com/spam-and-phishing-in-2020/100512/ (Erişim Tarihi: 10.01.2021)

Srinivasan S., Vinayakumar R., Arunachalam A., Alazab M., Soman K., 2021, DURLD: *Malicious URL Detection Using Deep Learning-Based Character Level Representations*. In: Stamp M., Alazab M., Shalaginov A. (eds) Malware Analysis Using Artificial Intelligence and Deep Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-62582-5_21

State of the Phish Report, 2020, Annual Report proofpoint.com (Erişim Tarihi: 15.03.2021)

USOM, *Zararlı Bağlantılar*, 2021, https://usom.gov.tr/zararli-baglantilar/20.html, (Erişim tarihi: 14.02.2021).

Tino P., Benuskova L., and Sperduti A., 2015, Artificial Neural Network Models, In: Kacprzyk J., Pedrycz W., Springer Handbook of Computational Intelligence. Berlin, Heidelberg, https://doi.org/10.1007/978-3-662-43505-2_27.

Wang D., He H., and Liu D, 2017, Adaptive critic nonlinear robust control: a survey, IEEE Trans. Cybern., 47 (10), pp. 3429-3451.

Warburton D., and Dockter P., 2020, "Phishing and Fraud Report Phishing During A Pandemic", Editor: Debbie Walkowski

Witten I.H., Frank E., 2005, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Inc., San Francisco, CA, USA.

Wu X., "CS 332: Data Mining – Feature Selection," *Lecture Notes, Department of Computer Science*, University of Vermont, USA, www.cs.uvm.edu/~xwu/PPT/FeatureSelection-06.ppt, (Erişim tarihi: 09.08.2015).

Wu X., and Kumar V., 2009. The Top Ten Algorithms in Data Mining (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781420089653.

Xiang G., Hong J., Rose C. P., and Cranor L., 2011, CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites, *ACM Transactions on Information and System Security*, Vol. 14, No. 2, Article 21.

Vazhayil A., Vinayakumar R., and Soman K., 2018, "Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks, *9th International Conference on Computing, Communication and Networking Technologies*, ICCCNT, pp. 1–6.

Yang P., Hwa Yang Y., Zhou B., Zomaya Y., et al., 2010, A review of ensemble methods in bioinformatics. Current Bioinformatics 5(4), 296–308.

Yang P., Zhao G. and Zeng P., 2019, Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning, *in IEEE Access*, vol. 7, pp. 15196-15209, doi: 10.1109/ACCESS.2019.2892066.

Yue T., Sun J., and Chen H., 2013, "Fine-Grained Mining and Classification of Malicious Web Pages," *Digital Manufacturing and Automation (ICDMA) Fourth International Conference*, pp.616-619.

Zhang D., Yan Z., Jiang H., Kim T., 2014, "A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites", *Information & Management* ,51, 845–853.

Zhang J., and Li X., 2017, "Phishing Detection Method Based on Borderline-Smote Deep Belief Network. In: Wang G., Atiquzzaman M., Yan Z., Choo KK., Security, Privacy, and Anonymity in Computation, Communication, and Storage. *Lecture Notes in Computer Science*, vol 10658. Springer, Cham. https://doi.org/10.1007/978-3-319-72395-2_5

Zhang J., Ou Y., Li D., and Xin Y., 2012, "A Prior-based Transfer Learning Method for the Phishing Detection" *Journal of Networks*, vol. 7, no. 8, pp. 1201-1207.

Zhang J., and Wang Y., 2012, "A real-time automatic detection of phishing URLs," *Computer Science and Network Technology (ICCSNT) 2012 2nd International Conference*, pp.1212, 1216.

Zhuang W., Jiang Q., and Xiong T., 2012, "An Intelligent Anti-phishing Strategy Model for Phishing Website Detection", *32nd International Conference on Distributed Computing Systems Workshops*, pp. 51-56.

EKLER

Ek-1 URL Gelişmiş Özellikleri

A. Alfa Nümerik Karakter Analizi

Öznitelik A-1 ila A-24: Aşağıdaki tabloda açıklanmıştır.

Öznitelik 1	İnternet sitesinin URL'i kaç tane	tire	-	karakteri içermektedir?
Öznitelik 2		alt tire	_	
Öznitelik 3		rakam	0-9	
Öznitelik 4		et	@	
Öznitelik 5		çift tırnak	"	
Öznitelik 6		noktalı virgül	;	
Öznitelik 7		eğik çizgi	/	
Öznitelik 8		ünlem	!	
Öznitelik 9		soru işareti	?	
Öznitelik 10		ile	&	
Öznitelik 11		yüzde	%	
Öznitelik 12		iki nokta üstüste	:	
Öznitelik 13		etiket (hashtag)	#	
Öznitelik 14		eşittir	=	
Öznitelik 15		yaklaşık (tilda)	~	
Öznitelik 16		artı	+	
Öznitelik 17		dolar	$	
Öznitelik 18		nokta	.	
Öznitelik 19		yıldız	*	
Öznitelik 20		sol parantez	(	
Öznitelik 21		sağ parantez	)	
Öznitelik 22		dikey çubuk	|	

Öznitelik 23		boşluk		
Öznitelik 24		çift eğik çizgi	//	

Eğer bir internet sitesinin URL'si bu karakterlerden bireysel olarak fazlaca içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-25: Eğer bir internet sitesinin URL'si toplamda Öznitelik 1 ila 24 arasında ifade edilen karakterlerden fazlaca içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-26: Eğer bir internet sitesinin URL'si çok uzunsa (toplamda çok fazla alfanümerik karakter içeriyorsa) bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-27 ile A-28: Aşağıdaki tabloda açıklanmıştır.

Öznitelik 27	URL'in host bölümü kaç tane	rakam	0 - 9	karakteri içermektedir?
Öznitelik 28		nokta	.	

Eğer bir URL'nin host bölümü bu karakterlerden bireysel olarak fazlaca içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-29: Eğer bir URL'nin host bölümü çok uzunsa (toplamda çok fazla alfanümerik karakter içeriyorsa) bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-30: Eğer bir URL'in içinde yer alan en uzun kelime çok uzunsa (toplamda çok fazla alfanümerik karakter içeriyorsa) bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-31: Eğer bir URL içinde çok sayıda kelime varsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-32: Eğer bir URL'in host bölümü içinde çok sayıda kelime varsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-33: Eğer bir URL çok sayıda "harf-rakam-harf" desenli kelime içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik A-34: Eğer bir URL çok sayıda "rakam-harf-rakam" desenli kelime içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

B. Anahtar Sözcük Analizi

Öznitelik B-1 ila B-20: Aşağıdaki tabloda açıklanmıştır.

Öznitelik 1	İnternet sitesinin URL'i yandaki "göz alıcı" kelimelerden	"access"	kaç tane içermektedir?
Öznitelik 2		"account"	
Öznitelik 3		"assets"	
Öznitelik 4		"click"	
Öznitelik 5		"id"	
Öznitelik 6		"logon"	
Öznitelik 7		"online"	
Öznitelik 8		"payment"	
Öznitelik 9		"purchase"	
Öznitelik 10		"refund"	
Öznitelik 11		"submit"	
Öznitelik 12		"transaction"	
Öznitelik 13		%bonus%	
Öznitelik 14		%confirm%	
Öznitelik 15		%free%	
Öznitelik 16		%login%	
Öznitelik 17		%love%	

Öznitelik 18		%secure%	
Öznitelik 19		%sexy%	
Öznitelik 20		%signin%	

Eğer bir internet sitesinin URL'si bu kelimeleri içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-21 ila B-48: Aşağıdaki tabloda açıklanmıştır.

Öznitelik 21	İnternet sitesinin URL'i yandaki "riskli" kelimelerden	"content"	kaç tane içermektedir?
Öznitelik 22		"poste"	
Öznitelik 23		"reset"	
Öznitelik 24		"security"	
Öznitelik 25		%abuse%	
Öznitelik 26		%auth%	
Öznitelik 27		%blogspot%	
Öznitelik 28		%component%	
Öznitelik 29		%detail%	
Öznitelik 30		%dispatch%	
Öznitelik 31		%fail%	
Öznitelik 32		%file%	
Öznitelik 33		%image%	
Öznitelik 34		%include%	
Öznitelik 35		%info%	
Öznitelik 36		%internationals%	
Öznitelik 37		%plugin%	
Öznitelik 38		%redirect%	
Öznitelik 39		%reply%	
Öznitelik 40		%resolution%	

Öznitelik 41		%session%	
Öznitelik 42		%setting%	
Öznitelik 43		%success%	
Öznitelik 44		%updat%	
Öznitelik 45		%upload%	
Öznitelik 46		%user%	
Öznitelik 47		%validat%	
Öznitelik 48		%verif%	

Eğer bir internet sitesinin URL'si bu kelimeleri içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-49 ila B-76: Aşağıdaki tabloda açıklanmıştır.

Öznitelik 49	İnternet sitesinin URL'i yandaki "sistem" kelimelerden	"bin"	kaç tane içermektedir?
Öznitelik 50		"css"	
Öznitelik 51		"js"	
Öznitelik 52		"jscript"	
Öznitelik 53		"logs"	
Öznitelik 54		"net"	
Öznitelik 55		"run"	
Öznitelik 56		"server"	
Öznitelik 57		"ssl"	
Öznitelik 58		"wp"	
Öznitelik 59		%admin%	
Öznitelik 60		%cache%	
Öznitelik 61		%cmd%	
Öznitelik 62		%dll%	

Öznitelik 63		%doc%	
Öznitelik 64		%img%	
Öznitelik 65		%libraries%	
Öznitelik 66		%module%	
Öznitelik 67		%php%	
Öznitelik 68		%process%	
Öznitelik 69		%script%	
Öznitelik 70		%system%	
Öznitelik 71		%template%	
Öznitelik 72		%theme%	
Öznitelik 73		%tmp%	
Öznitelik 74		%webapp%	
Öznitelik 75		%webscr%	
Öznitelik 76		%website%	

Eğer bir internet sitesinin URL'si bu kelimeleri içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-77: Eğer bir internet sitesinin URL'si fazlaca "göz alıcı" kelimeler içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-78: Eğer bir internet sitesinin URL'si fazlaca "riskli" kelimeler içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-79: Eğer bir internet sitesinin URL'si fazlaca "sistem" kelimeleri içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik B-80: Eğer bir internet sitesinin URL'si fazlaca "göz alıcı", "riskli" ve "sistem" kelimeleri içeriyorsa, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

C. Güvenlik Analizi

Öznitelik C-1: Eğer bir internet sitesinin URL'si güvenli bağlantılı metin aktarım protokolü (https) kullanıyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi **olmaması** olasıdır.

Öznitelik C-2: Eğer bir internet sitesinin URL'si birden fazla "http" (potansiyel harici bağlantı) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-3: Eğer bir internet sitesinin URL'si birden fazla "https" (potansiyel harici bağlantı) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-4: Eğer bir internet sitesinin URL'si birden fazla "www" (potansiyel harici bağlantı) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-5: Eğer bir internet sitesinin URL'si fazlaca potansiyel riskli program dosyası uzantılarını (.exe, .pif, .application, .gadget, .msi, .msp, .scr, .hta, .cpl, .msc, .jar) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-6: Eğer bir internet sitesinin URL'si fazlaca potansiyel riskli betik dosyası uzantılarını (.bat, .cmd, .vb, .vbs, .vbe, .js, .jse, .ws, .wsf, .wsc, .wsh, .ps1, .ps1xml, .ps2, .ps2xml, .psc1, .psc2, .msh, .msh1, .msh2, .mshxml, .msh1xml, .msh2xml) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-7: Eğer bir internet sitesinin URL'si fazlaca potansiyel riskli kısa yol uzantılarını (.scf, .lnk, .inf) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-8: Eğer bir internet sitesinin URL'si fazlaca potansiyel riskli ofis makro uzantılarını (.doc, .xls, .ppt, .docm, .dotm, .xlsm, .xltm, .xlam, .pptm, .potm, .ppam, .ppsm, .sldm) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik C-9: Eğer bir internet sitesinin URL'si fazlaca potansiyel riskli diğer uzantıları (.reg) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

D. Etki Alanı Kimlik Analizi

Öznitelik D-1: İnternet sitesinin URL'i IP (İnternet Protokolü) olarak mı verilmiş? Eğer bir internet sitesinin URL'i IP olarak verilmişse, bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik D-2: Eğer bir internet sitesinin URL'si birden fazla Üst Seviye Etki Alanı (TLD) (.com, .org, .net, .gov, .int, .mil, .edu, .post, .co) içeriyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik D-3: Eğer URL'in host bölümü Üst Seviye Etki Alanı (TLD) (.com, .org, .gov, .post, .co) **içermiyorsa** bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Öznitelik D-4: Eğer bir internet sitesinin URL yolunda (host bölümü dışında) hedefte olan markaların etki alanı ismi yer alıyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Ör: http://www.phishing.com/refund.**paypal.com**

Öznitelik D-5: Eğer bir URL'in host bölümünde hedefte olan markaların ismi yer alıyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Ör: http://www.login**paypal**.com/

Öznitelik D-6: Eğer bir URL'in host bölümünde hedefte olan markaların etki alanı ismi yer alıyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

Ör: http://www.accounts.**paypal.com**.esy.es/

Öznitelik D-7: Eğer bir internet sitesinin URL'si "URL Kısaltma Servisi"ni (gerçek host ismini gizlemek için kullanılır) kullanıyorsa bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

WHOIS-Etki Alanı Sorgusu

Öznitelik D-8: Eğer bir etki alanı isminin "Oluşturulma Tarihi" son 12 ay içindeyse bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi olması olasıdır.

E. Sayfa Sıralama Analizi

Öznitelik E-1: Eğer bir etki alanı isminin Google'daki "Sayfa Sıralama Skoru" yüksekse bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi **olmaması** olasıdır.

Öznitelik E-2: Eğer bir etki alanı isminin Alexa'daki "Global Sayfa Sıralama Skoru" düşükse bu internet sitesinin potansiyel olarak oltalama saldırılarında kullanılan bir internet sitesi **olmaması** olasıdır.

Ek-2 Yurtdışı Veri Setinden Elde Edilen Göz Alıcı Kelimeler

Özellik	Kelime	Özellik	Kelime	Özellik	Kelime	Öznitelik	Kelime
B1	account	B21	eric	B41	media	B61	solution
B2	admin	B22	file	B42	ment	B62	stem
B3	ages	B23	form	B43	mini	B63	temp
B4	america	B24	google	B44	name	B64	template
B5	apple	B25	graph	B45	online	B65	templates
B6	apps	B26	group	B46	paypal	B66	tent
B7	bill	B27	home	B47	photo	B67	theme
B8	brad	B28	host	B48	plate	B68	themes
B9	cent	B29	http	B49	plates	B69	tiny
B10	comp	B30	image	B50	plugins	B70	tion
B11	components	B31	images	B51	port	B71	tions
B12	content	B32	incl	B52	press	B72	update
B13	count	B33	includes	B53	rica	B73	upload
B14	date	B34	index	B54	screen	B74	verification
B15	direct	B35	info	B55	security	B75	vice
B16	docs	B36	late	B56	service		
B17	drive	B37	libraries	B57	session		
B18	ebay	B38	line	B58	sign		
B19	edit	B39	load	B59	sing		
B20	enter	B40	login	B60	site		

Ek-3 Her Bir Özellik Seçme Metodu İçin Seçilen Özellikler

Tabloda yer alan her bir özellik için ifade edilen "1" değeri söz konusu özelliğin ilgili özellik seçme metodu tarafında önemli, "0" değeri de önemsiz özellik olarak seçildiğini ifade etmektedir.

Özellik	CFS Alt Küme Tabanlı	Tutarlılık Alt Küme Tabanlı	Kazanç Oranı Nitelik Tabanlı	ReliefF Nitelik Tabanlı	Toplam
A_03	1	1	1	1	4
A_27	1	1	1	1	4
B_79	1	1	1	1	4
B_80	1	1	1	1	4
C_01	1	1	1	1	4
D_03	1	1	1	1	4
D_08	1	1	1	1	4
E_01	1	1	1	1	4
E_02	1	1	1	1	4
A_15	1	0	1	1	3
A_32	1	0	1	1	3
A_34	1	0	1	1	3
B_78	1	0	1	1	3
D_02	0	1	1	1	3
D_04	1	1	0	1	3
D_05	1	1	0	1	3
D_06	0	1	1	1	3
A_01	0	1	0	1	2
A_04	0	0	1	1	2
A_07	0	1	0	1	2
A_09	0	0	1	1	2
A_12	0	0	1	1	2
A_14	0	0	1	1	2
A_18	0	1	0	1	2
A_25	0	1	0	1	2
A_26	0	1	0	1	2
A_28	0	1	0	1	2
A_29	0	1	0	1	2
A_30	0	1	0	1	2

A_31	0	1	0	1	2
A_33	0	0	1	1	2
B_05	0	0	1	1	2
B_16	0	0	1	1	2
B_22	0	0	1	1	2
B_25	0	0	1	1	2
B_27	0	1	0	1	2
B_34	0	0	1	1	2
B_36	0	0	1	1	2
B_41	0	1	0	1	2
B_54	0	0	1	1	2
B_58	0	0	1	1	2
B_61	0	0	1	1	2
B_67	0	1	1	0	2
B_77	0	0	1	1	2
D_01	0	0	1	1	2
D_07	0	0	1	1	2
A_10	0	0	1	0	1
A_11	0	0	1	0	1
A_13	1	0	0	0	1
B_21	0	0	0	1	1
B_28	0	0	0	1	1
B_30	0	0	0	1	1
B_35	0	0	0	1	1
B_37	0	0	0	1	1
B_44	0	0	1	0	1
B_46	0	0	0	1	1
B_49	0	0	0	1	1
B_55	0	0	0	1	1
B_59	0	0	0	1	1
B_63	0	0	0	1	1
B_70	0	0	0	1	1
B_75	0	0	0	1	1
B_76	0	0	0	1	1
C_04	1	0	0	0	1
A_02	0	0	0	0	0
A_05	0	0	0	0	0
A_06	0	0	0	0	0
A_08	0	0	0	0	0

A_16	0	0	0	0	0
A_17	0	0	0	0	0
A_19	0	0	0	0	0
A_20	0	0	0	0	0
A_21	0	0	0	0	0
A_22	0	0	0	0	0
A_23	0	0	0	0	0
A_24	0	0	0	0	0
B_01	0	0	0	0	0
B_02	0	0	0	0	0
B_03	0	0	0	0	0
B_04	0	0	0	0	0
B_06	0	0	0	0	0
B_07	0	0	0	0	0
B_08	0	0	0	0	0
B_09	0	0	0	0	0
B_10	0	0	0	0	0
B_11	0	0	0	0	0
B_12	0	0	0	0	0
B_13	0	0	0	0	0
B_14	0	0	0	0	0
B_15	0	0	0	0	0
B_17	0	0	0	0	0
B_18	0	0	0	0	0
B_19	0	0	0	0	0
B_20	0	0	0	0	0
B_23	0	0	0	0	0
B_24	0	0	0	0	0
B_26	0	0	0	0	0
B_29	0	0	0	0	0
B_31	0	0	0	0	0
B_32	0	0	0	0	0
B_33	0	0	0	0	0
B_38	0	0	0	0	0
B_39	0	0	0	0	0
B_40	0	0	0	0	0
B_42	0	0	0	0	0
B_43	0	0	0	0	0
B_45	0	0	0	0	0

B_47	0	0	0	0	0
B_48	0	0	0	0	0
B_50	0	0	0	0	0
B_51	0	0	0	0	0
B_52	0	0	0	0	0
B_53	0	0	0	0	0
B_56	0	0	0	0	0
B_57	0	0	0	0	0
B_60	0	0	0	0	0
B_62	0	0	0	0	0
B_64	0	0	0	0	0
B_65	0	0	0	0	0
B_66	0	0	0	0	0
B_68	0	0	0	0	0
B_69	0	0	0	0	0
B_71	0	0	0	0	0
B_72	0	0	0	0	0
B_73	0	0	0	0	0
B_74	0	0	0	0	0
C_02	0	0	0	0	0
C_03	0	0	0	0	0
C_05	0	0	0	0	0
C_06	0	0	0	0	0
C_07	0	0	0	0	0
C_08	0	0	0	0	0
C_09	0	0	0	0	0

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Mustafa AYDIN
Doğum Yeri	
Doğum Tarihi	Tarih girmek için tıklayın veya dokunun.
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	
E-Posta Adresi	
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	Gazi Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Elektrik-Elektronik Mühendisliği
Mezuniyet Yılı	Tarih girmek için tıklayın veya dokunun.

Yüksek Lisans	
Üniversite	Hacettepe Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Elektrik-Elektronik Mühendisliği Anabilim Dalı
Programı	Elektrik-Elektronik Mühendisliği

Yüksek Lisans	
Üniversite	Temple University, Philadelphia, USA
Enstitü Adı	Fox School of Business
Anabilim Dalı	Management and Information Systems
Programı	Information Technology Audit and Cyber Security

Yüksek Lisans	
Üniversite	Ahmet Yesevi Üniversitesi
Enstitü Adı	İktisadi ve İdari Bilimler Fakültesi
Anabilim Dalı	İşletme Anabilim Dalı
Programı	İşletme

Doktora	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik Anabilim Dalı
Programı	Enformatik Programı

Makale ve Bildiriler
Aydın, M., Butun, I., Bicakci, K., and Baykal, N., ***"Using Attribute-based Feature Selection Approaches and Machine Learning Algorithms for Detecting Fraudulent Website URLs***", IEEE 10th Annual Computing and Communication Workshop and Conference, Las Vegas, USA, January 2020
Çil, A.E., Aydın M***., "Kurum İçi Uygulamaların EBYS ile Entegrasyonunda Yapay Zekanın Önemi Üzerine Bir İnceleme",*** Endüstri 4.0 Sürecinde Bilgi Yönetimi ve Bilgi Güvenliği 4. e-Beyas Sempozyumu, Ankara Üniversitesi, Ekim 2019, Bilgi Yönetimi Dergisi (e-ISSN 2636-8544)
Aydın, M., and Baykal, N., ***"Feature Extraction and Classification Phishing Websites Based on URL***", IEEE Conference on Communications and Network Security (CNS), pp. 769-770, Florence, Italy, September 2015
Dilek, S., Aydın, M., and Çakır, H., ***"Applications of Artificial Intelligence Techniques to Combating Cyber Crimes: A Review",*** International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 6, No. 1, pp. 21-39, January 2015
Aydın, M., and Baykal, N., ***"Improved Phishing Attack Detection Using Hybrid Machine Learning Algorithms Based On URL Structure",*** International Conference on Software Engineering and Digital Technology (SEDT), Tokyo, Japan, December 2014
Aydın M., Yazgan E., Arıöz U., Cila A., and Oğuz K., ***"Biomedical Image Fusion with Selection Opreration in the Laplacian Pyramid Domain"***, The 11th International Biomedical Science and Technology Conference, Ankara, Türkiye, 2005
Aydın M., Yazgan E., Arıöz U., Cila A., and Oğuz K., ***"Laplace Piramit Bölgesinde Seçim Tekniği ile Biyomedikal Görüntü Füzyonu"***, IEEE 12th Signal Processing and Communications Applications Conference, pp. 106-110, Kuşadası, Türkiye, 2004