T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMİ KULLANARAK WEB TABANLI KİMLİK AVI SALDIRILARININ SINIFLANDIRILMASI

Ramazan İNCİR

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

OCAK-2020

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

DERİN ÖĞRENME YÖNTEMİ KULLANARAK WEB TABANLI KİMLİK AVI SALDIRILARININ SINIFLANDIRILMASI

Tez Yazarı

RAMAZAN İNCİR

Danışman

Prof. Dr. Ahmet Bedri ÖZER

OCAK 2020
ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı: Derin Öğrenme Yöntemi Kullanarak Web Tabanlı Kimlik Avı Saldırılarının Sınıflandırılması

Yazarı: Ramazan İNCİR

İlk Teslim Tarihi: 6.12.2019

Savunma Tarihi: 17.1.2020

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

		İmza	
Danışman:	Prof. Dr. Ahmet Bedri ÖZER Fırat Üniversitesi, Mühendislik Fakültesi		Onayladım
Başkan:	Doç. Dr. Taner Tuncer Fırat Üniversitesi, Mühendislik Fakültesi		Onayladım
Üye:	Dr. Öğr. Üyesi Soner KIZILOLUK Munzur Üniversitesi, Mühendislik Fakültesi		Onayladım

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Soner ÖZGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Derin Öğrenme Yöntemi Kullanarak Web Tabanlı Kimlik Avı Saldırılarının Sınıflandırılması" yüksek lisans tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

17/01/2020

Ramazan İNCİR

ÖNSÖZ

Web tabanlı kimlik avı saldırıları geçmiş yıllara göre sürekli artış göstermektedir. Bu yöntem ile web sayfaları taklit edilerek kullanıcılar yanıltılıp dolandırılmaktadır. Kişisel bilgileri ele geçirilen kullanıcılar maddi ve manevi mağdur olmaktadır. Bu çalışmada bu tür durumların önüne geçebilmek amacıyla kimlik avı ve meşru web sitelerin ayrımını yapmak amacıyla bir yöntem önerilmiştir.

Yüksek Lisans danışmanlığımı üstlenerek çalışmalarımın yürütülmesi sırasında yardımlarını esirgemeyen, insani yönüyle örnek aldığım, beraber çalışmaktan onur duyduğum danışman hocam Sayın Prof. Dr. Ahmet Bedri ÖZER'e;

Ayrıca eğitim hayatım boyunca maddi ve manevi desteğini esirgemeyen babam Yusuf İNCİR'e teşekkürü bir borç bilirim.

Ramazan İNCİR

ELAZIĞ-2020

İÇİNDEKİLER

Sayfa

ÖZET

Derin Öğrenme Yöntemi Kullanarak Web Tabanlı Kimlik Avı Saldırılarının Sınıflandırılması

Ramazan İNCİR

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı
Ocak 2020, Sayfa xi+47

Web tabanlı kimlik avı saldırıları, çevrimiçi bir ortamda kullanıcıların kandırılarak bilgilerini istekleri dışında paylaşmalarını sağlamak amacıyla saldırganlar tarafından kullanılan bir yöntemdir. Bu saldırı türü kullanıcıları yanıltmak amacıyla web sitelerin taklit edilmesi ile gerçekleştirilir. Aracı olarak reklam ya da e-posta gibi yöntemler kullanılmaktadır. Mevcut bulunan web sitelerinin kimlik avı olup olmadığının ayrımını yapmak büyük bir problemdir. Bu nedenle kimlik avı web sitelerinde kullanıcıların / müşterilerin hassas bilgilerine farklı kişilerin yetki dâhilinde olmadan erişmesini engellemek ve bu durumu tespit etmek için başarılı sınıflandırma yöntemlerini kullanmak bu problemin çözümünde önemli bir rol oynamaktadır.

Bu çalışmada, yıllara göre artış gösteren web tabanlı kimlik avı saldırılarının önüne geçmek ve kullanıcıların bu saldırı türlerinden etkilenmesini engellemek amaçlanmaktadır. Bu amaç doğrultusunda, sınıflandırma işlemi için derin öğrenme yöntemi kullanılmıştır. Bu yöntem, bilinen verilerin çok katmanlı yapay sinir ağı ile eğitilerek bilinmeyen verilerin sınıflandırılmasını ve analizini yapma imkânı sağlayan bir yaklaşımdır. Sınıflandırma işlemi için 5000 meşru ve 5000 kimlik avı web sitesi bulunan veri seti kullanılmıştır. Bu veri seti kullanılarak eğitilen model ile sınıflandırma yapılmıştır. Sonuç olarak bu sınıflandırma işleminde yüksek doğruluk değerleri elde edilmiş olup derin öğrenme yönteminin başarılı sonuçlar sergilediği görülmüştür.

Anahtar Kelimeler: Kimlik avı, Derin öğrenme, Sınıflandırma, Siber güvenlik

ABSTRACT

Classification of Web-Based Phishing Attacks Using Deep Learning Method

Ramazan İNCİR

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering
January 2020, Pages xi+47

Web-based phishing attacks are a method used by attackers to attain their important information and to trick users in an online media. This type of attack is accomplished by emulating websites to mislead users. Means such as advertising or e-mail are used as mediators. It is a big problem to distinguish whether a website is phishing. Therefore, phishing web sites, users or customers to prevent theft of personal information and to use the classification methods to detect this situation plays an important role in solving the problem.

In this study, it is aimed to prevent web-based phishing attacks that increase in years and to prevent users from being affected by these types of attacks. For this purpose, deep learning method was used for classification process. This method is an approach that allows the classification and analysis of unknown data by training the known data with a multi-layer artificial neural network. For the classification process, a data set with 5000 legitimate and 5000 phishing websites was used. Using this data set, classification was made with a trained model. As a result, high accuracy values were obtained in this classification process and it was seen that deep learning method showed successful results.

Keywords: Phishing, Deep learning, Classification, Cyber security

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

KISALTMALAR LİSTESİ

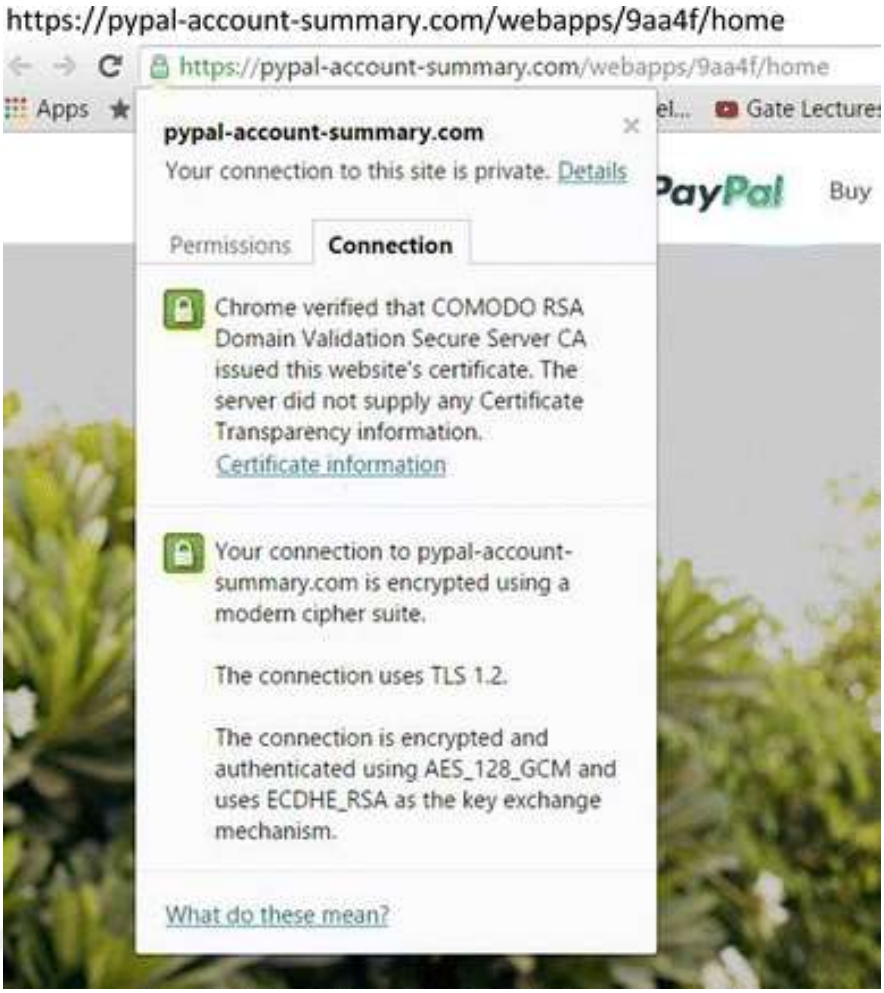
HTTPS : Hypertext Transfer Protocol Secure
URL : Uniform Resource Locator
APWG : Anti-Phishing Working Group
IP : Internet Protocol
HTML : Hyper Text Markup Language
LPD : Legitimate Phishing Detector
DNS : Domain Name Server
HEFS : Hybrid Ensemble Feature Selection
TCP : Transmission Control Protocol
WAF : Web Application Firewall
YSA : Yapay Sinir Ağları
CNN : Convolutional Neural Network
RNN : Recurrent Neural Network
ReLU : Rectified Linear Unit
CSS : Cascading Style Sheets
ARFF : Attribute-Relation File Format
TLD : Top Level Domain
ccTLD : County Code Top Level Domains

1. GİRİŞ

İnsanlar siber dünyada birbirleriyle internete bağlı dijital cihazları kullanarak iletişim kurmaktadır. İnternet, bankacılık işlemleri, resmi olan devlet ile ilgili işleri, alışveriş, gazete, dergi, radyo ve televizyon vb. hizmetlerden yararlanmayı ve dünyanın diğer ucundaki birisiyle kolay bir şekilde iletişim kurabilmeyi, birlikte eğlenmeyi sağlar. İnternetin ortaya çıkması ve teknolojinin gün geçtikçe gelişmesi ile beraber ortaya çıkarılan dijital cihazlar üzerinden çevrimiçi hizmetleri kullanan kullanıcılar gün geçtikçe artmaktadır. Tüm bu gelişmelerin artışı ile birlikte bu durumu fırsata çevirip para kazanma amacıyla kişilerin önemli bilgilerine web siteler aracılığı ile erişmeye çalışan saldırganların sayısında artış bulunmaktadır.

Kimlik avı saldırıları, e-posta, web siteler ve kötü amaçlı yazılımlar ile çeşitli şekillerde gerçekleştirilen, kullanıcıların hassas bilgilerine erişmeye çalışan saldırı yöntemlerinden biridir. E-posta kimlik avında saldırganlar, iletinin güvenilir bir şirketten geldiğini iddia edip, sahte e-postalarla kullanıcıları yanıltmaktadır. En az binlerce kullanıcının buna düşeceği varsayılırsa milyonlarca çevrimiçi kullanıcıya sahte e-posta gönderilmektedir. Web sitesi kimlik avında saldırganlar, meşru siteleri taklit ederek kullanıcıları yanıltmaya çalışmaktadırlar. Yasal sitelerin kopyası gibi görünen bir web sitesi oluşturup kullanıcıları farklı web sitelerinden veya Facebook, Twitter gibi sosyal medya ağları reklamlarından çekmeye çalışmaktadır. Saldırganlar web sayfası aracılığıyla bir yem gönderip ve kullanıcıya ait bilgilerin çalınmasını beklemektedir. Kullanıcılar sahte sayfalara güvenerek, saldırganın hassas bilgileri elde etmesine ve bu konuda başarılı olmasına farkında olmadan imkan sağlamaktadır.

Saldırganlar, bir kısım kullanıcıları yanıltmak adına web sitelerini HTTPS bağlantısı, yeşil asma kilit vb. gibi güvenlik göstergeleri ile yönetebilir. Bir saldırganın kimlik avı web sitesi için HTTPS ve yeşil asma kilit alabildiğini Şekil 1.1'de görebiliriz. Bu nedenle, HTTPS bağlantısının artık bir web sitesinin yasallığına karar vermesi garanti edilmemektedir [1].



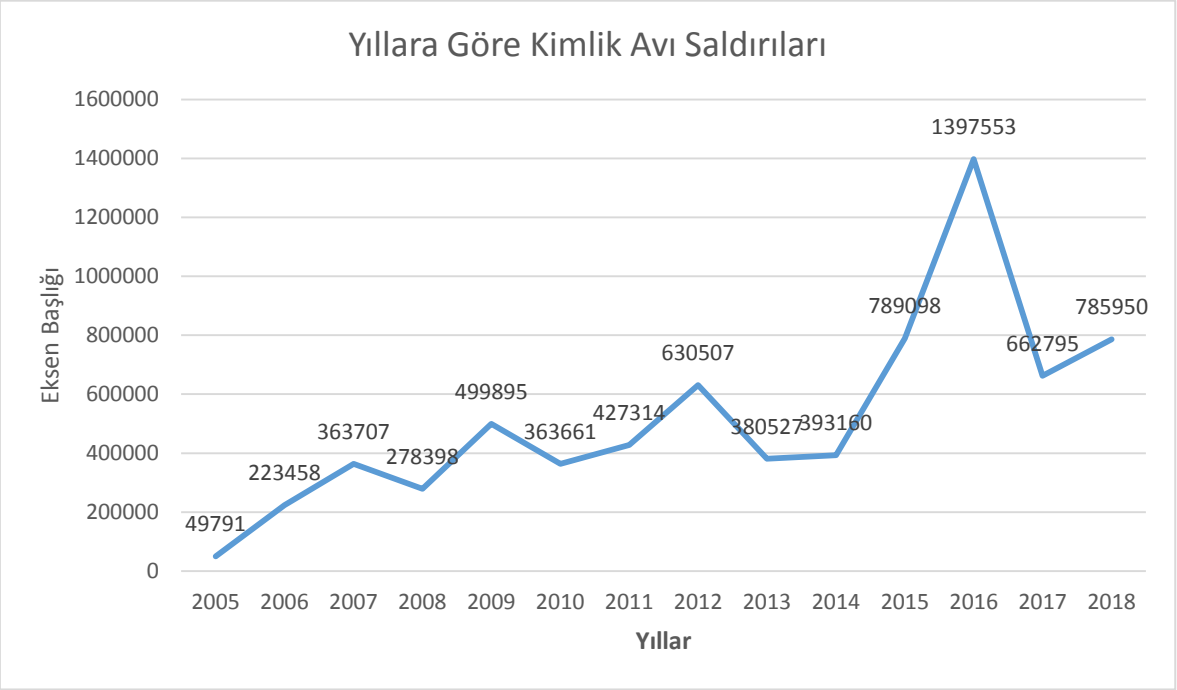
Şekil 1.1. HTTPS Protokolü İle Paypal'ın Kimlik Avı Web Sitesi [1].

Kullanıcılar aşağıdaki maddelerde belirtilen çeşitli faktörlerden dolayı kimlik avı saldırılarına uğramaktadır [2].

1. Bilgisayar sistemleri konusunda yetersiz bilgi,
2. Güvenlik ve güvenlik göstergeleri hakkında yetersiz bilgi,
3. Mevcut araçların güvenliğini zayıflatarak uyarılara ve dikkatlere yeterince özen gösterilmemesi,
4. URL'de ve web sitesi içeriğinde görsel ve metinsel aldatıcı durumlara dikkat edilmemesi (Örneğin, paypal metni, adres çubuğunun URL'inde, telif hakkı veya kimlik avı web sitesinin başlığında paypa1 olarak değiştirilir.)

Kimlik Avı Koruması Çalışma Grubu (APWG), kendisine üye olan çeşitli şirketler tarafından bildirilen kimlik avı saldırılarını inceleyen, kar amacı gütmeyen bir organizasyondur. Bu grup, dünya çapında yer alan kötü amaçlı etki alanları ve kimlik avı saldırılarıyla ilgili istatistiksel bilgileri analiz edip raporları periyodik olarak (üç aylık ve yarım yıllık) yayınlamaktadır. Şekil

1.2’de 2005’den 2018’e geçen zaman diliminde her yıl belirtilen periyot raporlarındaki toplam kimlik avı site saldırıları belirtilmiştir [3].



Şekil 1.2. 2005'den 2018'e Kimlik Avı Saldırıları [3]

Şekil 1.2’deki değerler incelendiğinde kimlik avı saldırılarının yıllara göre genel olarak artış gösterdiği ve belirtilen rakamların ciddiyeti görülmektedir. Şekil 1.2’de belirtilen bilgiler dışında en son yapılan APWG (2019) 2. çeyrek raporundaki kimlik avı saldırı sayıları incelendiğinde geçmiş üç raporda belirtilen saldırı sayılarının hayli üstünde olduğu gözlemlenmiştir. En son yapılan APWG 2. çeyrek 2019 raporuna göre toplam 182.465, 1.çeyrek 2019 raporuna göre toplam 180.768, 4. çeyrek 2018 raporuna göre toplam 138.328 ve 3. çeyrek raporuna göre toplam 151.014 kimlik avı sitesi olduğu belirtilmiştir. Veriler incelendiğinde 2018 yılına göre 2019 yılında ciddi bir artış gözlemlenmiştir [3].

Bu çalışmada yıllara göre artış gösteren web tabanlı kimlik avı saldırılarının önüne geçmek ve kullanıcıların bu saldırı türlerinden etkilenmesini engellemek amaçlanmaktadır. Bu durum için klasik makine öğrenme yöntemlerinden farklı olarak derin öğrenme yöntemi ile sınıflandırma yapılmıştır. Yapılan testler sonucunda başarılı sonuçlar elde edilmiş ve daha önceden aynı veri seti ile yapılan çalışmalarla kıyaslandığında daha yüksek doğruluk değerleri elde edildiği gözlenmiş ve uygulama başlığı altında sonuçlar belirtilmiştir.

2. KİMLİK AVI TESPİTİ İLE İLGİLİ ÇALIŞMALAR

Kimlik avı ile mücadele için birçok yöntem geliştirilmiştir. Bunlar teknik ve teknik olmayan yöntemler olmak üzere iki kimlik avı çözüm gurubuna ayrılabilir. Teknik olmayan yöntemler genel olarak yasal yöntemler ve eğitim çalışmalarını içermektedir. Teknik yöntemler ise genel olarak kara liste / beyaz liste, görsel benzerlik, arama motoru ve makine öğrenmesi yöntemleri olmak üzere sıralanabilir. Bu çalışmaların içerikleri aşağıda özetlenmiştir.

2.1. Yasal Çözüm Yöntemi

Kimlik avı saldırılarına çözüm arayan ülkeler yasa çıkararak bu saldırılara engel olmaya çalışmaktadır. 2004 yılında Amerikalı tüketicilerin korunması amacıyla Federal Ticaret Komisyonu tarafından kimlik avı saldırıları suç listesine eklenmiştir. Farklı olarak siber suçlarla mücadele etmek amacıyla Avustralya hükümeti, 2005 yılında savaş stratejileri için Microsoft ile ortaklık imzalamıştır. İngiltere hükümeti ise 2006 yılında kimlik avı saldırılarını cezalandırmıştır. Yasal çözümler geçmişten günümüze bir dereceye kadar engel olabilmesine rağmen, tam olarak engellemek konusunda zayıf kalmaktadır. Gelecekte gelişen teknoloji ve yöntemlere karşı ilgili yasal çözüm yönteminin kısa ömürlü olacağı düşünülmektedir. Kimlik avı saldırılarını takip etmek zordur. Bu nedenle kimlik avı saldırılarına engel olabilmek için diğer yöntemlerle birleştirmek gerekir [4].

2.2. Eğitim Yöntemi

İnsanların çoğu kimlik avı saldırılarının bir kısmı hakkında bilinçlidirler. Bu sebepten saldırıya uğramaları kolaylaşmaktadır. İnsanlar bu saldırılar karşısında bilinçlendirilip eğitim süzgecinden geçirilse bile bir dereceye kadar saldırılar azaltılabilir. Kimlik avı konusunda eğitim yöntemi ile çözüm bulmaya çalışan çalışmalar incelendiğinde, Kumaraguru ve arkadaşları (2009), kullanıcıların eğitim süzgecinden geçirilmesi sonucu tehlikeli web site ve e-postaların ayırt etme becerilerinin gelişebileceğini savunmuştur [5]. Başka bir çalışmada ise kullanıcıların bir web sitesine kişisel bilgilerini vermeden önce eğitilmesini önermiş olup kullanıcıların eğitimindeki kilit faktörün ise iyi bir iletişimin kurulması gerektiği belirtilmiştir [6]. Eğitim yönteminde belirtilen başka bir çözüm ise geliştirilen oyunlarla kullanıcıların eğitilmesidir. Oyun ile eğitim çalışmalarında kimlik avı saldırılarında bilgi sahibi olup, kullanıcıların farkındalık edinmelerini sağlamışlardır [7]. Bu alanda yapılan çalışmaların geneli incelendiğinde eğitim aşamasında çok zaman harcandığını ve bu yöntemin kimlik avı teknolojisinin gerisinde kaldığı düşünülmektedir.

2.3. Liste Tabanlı Algılama Yöntemi

Liste tabanlı algılama tekniklerinde kimlik avı web sayfalarını tespit edebilmek için beyaz liste ve kara liste olmak üzere iki liste yöntemi kullanılır.

Beyaz liste tabanlı kimlik avı yöntemi gerekli bilgileri sağlamak için yasal ve meşru web sitelerini barındırır. Beyaz listede bulunmayan her web sitesi şüpheli olarak kabul edilir. Cao ve diğerleri, kullanıcıların ziyaret ettiği kullanıcı arabirimi olan sitelerin IP adresini kaydeden bir sistem geliştirmiştir. Bu sayede kullanıcılar web sitesini ziyaret ettiğinde aksi bir durum varsa sistem uyarmaktadır. Eğer bu siteler ilk kez ziyaret ediliyorsa web sitesi meşru olsa bile şüpheli olarak kabul edilmektedir [8]. Jain ve Gupta, webde otomatik olarak kendini güncelleyen ve bu şekilde kullanıcıları uyaran bir yöntem geliştirmişlerdir. Yöntem domain-Ip adresi eşleştirme modülü ve kaynak kodundaki linklerin özelliklerinin çıkarılması aşamalarından oluşmaktadır. Bu yöntem ile % 86.02 doğruluk elde edilmiştir [9].

Kara listeler kimlik avı web site URL kayıtlarından oluşmaktadır. Bu listelerin URL kayıtları kullanıcılar, spam algılama sitemi ve diğer kuruluşlar gibi çeşitli kaynaklardan oluşturulmaktadır. Kara liste yöntemi ile daha önceden tespit edilen, saldırı amacı güden kimlik avı web sitesi aynı URL ve IP adresi ile saldırılması engellenir. Güvenlik mekanizması kötü niyetli URL ve IP'leri tespit ederek kara listeyi günceller. Kara listeye dayalı yöntemler ilk saldırıyı saptama gücüne sahip değildir. Bu yöntemin kimlik avı sitelerini tespit başarısı %20'dir [10]. Bu başarı oranı nedeniyle kara listeye dayalı çözüm sistemlerinin verimli olmadığı görülmektedir. Google Güvenli Tarama API'si, PhishNet gibi bazı şirketler kara listeye dayalı kimlik avı saldırısı tespit sistemlerine hizmet eder. Bu sistemler, şüpheli URL'nin kara listede olup olmadığını kontrol etmek için bir eşleştirme algoritması kullanmaktadırlar. Kara listeye dayalı çözüm yöntemleri sık güncelleme gerektirir. Ek olarak, kara liste yönteminin hızlı bir şekilde olgunluğa ulaşması için aşırı sistem kaynakları gerektirmektedir [11].

Beyaz listede meşru olan sitelere bile şüphe ile yaklaşıp kimlik avı olarak değerlendirilmesi, kara listenin sürekli güncellenmesinin gerekmesi, sistem kaynaklarının aşırı kullanılması, yapılan çalışmalar değerlendirildiğinde düşük yüzdelikte doğruluk oranı vermesi bu çözüm yönteminin zayıf olduğunu ortaya koymaktadır.

2.4. Görsel Benzerlik Yöntemi

Görsel benzerlik teknikleri, kimlik avı web siteleri görünümleri ile meşru web sitelerinin görünümleri arasında ayrım yapmaktadır. Dhamija ve Tygar, Blok seviyesi benzerliği, genel stil benzerliği ve düzen benzerliğinin web sitelerini tanımak için kullanılabileceğini belirtmiştir [12]. Liu ve arkadaşları, kullanıcıları korumak için bir tarayıcı eklentisi geliştirmiştir. Kullanıcılar, kimlik avı web sitelerini bir görüntü doğrulama mekanizması ile tanımaktadır [13]. Medvet ve

arkadaşları, metin parçaları ve stilleri, gömülü görüntüler ve sayfanın genel görsel görünümü olmak üzere üç önemli özellik ortaya koymuş, bu sayede kimlik avı web sitesi ile meşru web sitelerinin ayırt edilebileceğini belirtmiştir [14]. Jain ve Gupta (2017a) görsel benzerlik teknikleri çalışması, metin bağlamı, HTML etiketleri, basamaklı stil sayfası (CSS) ve resim içeren bir özellik setine dayanmaktadır [15]. Ancak Jain ve Gupta (2017b), bu görsel benzerlik tekniklerinin sıfır saatlik bir kimlik avı saldırısını tanıyamayacaklarına dair bir eksiklik olduğunu öne sürmektedir [16].

2.5. Arama Motoru Yöntemi

Arama motoru temelli teknikler, web sayfasından başlık, metin, logo, telif hakkı, etki alanı adı vb. gibi kimlik özelliklerini çıkarır ve web sayfasının yasallığını kontrol etmek için arama motorunu kullanmaktadır. Huh ve Kim (2011), kimlik avını web arama motorlarındaki arama sonuçlarına göre meşru olabileceğini önermektedir [17]. Chang ve arkadaşları (2013), web sitesinin kimlik avı olup olmadığını belirlemek için Google görüntü veri tabanını kullanan bir yaklaşım önermiştir [18]. Kang ve arkadaşları (2015), web sitesinin meşruluğunu ayırt edebilmek için logo görüntülerini kullanarak logo tabanlı bir yöntem önermiştir [19]. Varshney ve arkadaşları, davranış-tepki mekanizması ve yanıt süresi içeren hafif bir Kimlik Avı Dedektörü (LPD) algılama sistemi önermiştir [20]. Bununla birlikte Jain ve Gupta (2017b) tarafından LPD yöntemi ele alınmış ve performansının dil tarafından etkilendiğinin kanısına vararak gerçek LPD oranının düşük olduğunu belirtmiştir [16]. Tan ve arkadaşları, web sitesinin metin içeriğinden anahtar kelimeleri tespit eden yeni bir ağırlıklı URL belirteç sistemi olan PhishWHO'yu önermiştir [21]. Ancak Jain ve Gupta (2017b), PhishWHO'nun yanlış pozitif oranının çok yüksek olduğunu belirterek hızlı ve dil bağımsızlığı avantajına sahip iki seviyeli bir kimlik doğrulama yaklaşımı önermiştir [16].

2.6. Makine Öğrenmesi Yöntemi

Makine öğrenmesi yöntemleri, orijinal bir web sitesini kimlik avı web sitesinden ayırt edebilen özelliklere sahip bir sınıflandırma algoritması geliştirmektedir. Bir web sitesinin tasarımı önceden belirlenmiş özelliklerle eşleşiyorsa kimlik avı olarak belirtilmektedir. Bu çözümlerin performansı, eğitim verilerine, ayarlanan özelliklere ve sınıflandırma algoritmasına bağlıdır. Makine öğrenmesi yönteminde kullanılan özellikler URL, sayfa kaynağı, web sitesi trafiği, arama motoru, DNS, vb. gibi çeşitli kaynaklardan elde edilir. Bu özelliklerden bazıları, erişilmesi zor, yavaş ve üçüncü şahıslara bağlı olabileceğinden, kaynaklardan özellikleri elde etme aşamaları zorlu olabilmektedir.

Pan ve Ding, kimlik avı web sitelerini tespit edebilmek için şirket adı ve sayfa kategori bilgilerini web sayfasından alarak SVM yöntemiyle bir yöntem önermiştir [22]. Miyamoto ve arkadaşları (2009), saldırganların oluşturmuş olduğu kimlik avı sitelerinin tespiti için dokuz makine

öğrenme metodu (AdoBoost, Naif Bayes, Rastgele Ormanlar, Yapay Sinir Ağları, Bayesian Katkı Regresyon Ağaçları, Torbalama, Sınıflandırma ve Regresyon Ağaçları, Bayesian Katkı Arka Ağaçları ve Destek Vektörü Makineleri) performansını değerlendirmiştir. Deney sonuçlarında, dokuz makine öğrenme yönteminden yedisinin geleneksel yöntemlerden daha iyi olduğunu göstermiştir [23]. Mohammad ve arkadaşları (2013), kimlik avı sitelerini tanımlamak için geliştirmiş oldukları bir yapıyla web sitelerinden otomatik olarak 17 özellik geliştirip her özelliğin önemi deneysel yöntemlerle analiz edilmiştir [24]. Mohammad ve arkadaşları (2014), kimlik avı sitelerini tanımlamak için kendi kendini yapılandıran sinir ağını temel alan bir yöntem önermişlerdir. Deney sonuçlarına göre "Hold-Out" validation yöntemiyle doğruluğun %92.48'e ulaştığı görülmüştür [25]. Hadi ve arkadaşları (2016), kimlik avı web sitelerini saptamak için ilişkilendirme kurallarına dayalı bir sınıflandırma algoritması önermişlerdir. Bu yeni algoritmanın sonuçlarına göre 10 kat çapraz doğrulama test yöntemiyle % 92 ile 93 arasında yüksek bir doğruluk elde edildiği belirtilmiştir [26]. Hanbay ve Kaytan (2017), aşırı öğrenme makinesine (ELM) dayalı kimlik avı web sitelerini tespit etmek için önerdikleri modelde 10 kat çapraz doğrulama test yöntemiyle ortalama %95,05 kesinliğe ulaştıklarını belirtmişlerdir [27].

Jain ve Gupta (2018), kimlik avı web sitelerini meşru sitelerden ayırmak için 19 özellik çıkararak makine öğrenmesini kullanan bir kimlik avı koruma yöntemi sunmuştur. Alexa popüler web sitelerinden, bazı çevrimiçi ödeme ağ geçitlerinden ve bazı üst düzey banka web sitelerinden 1918 meşru web sitesi ve PhishTank (2018) ile Openfish'ten (2018) 2141 kimlik avı web sitesi kullanmışlardır. Önerilen makine öğrenmesi yöntemi yaklaşımlıyla %99,39 doğruluk elde edilmiştir [28]. Rao ve Pais (2018), sadece makine öğrenme yaklaşımları ile birlikte görüntü kontrolünü kullanarak bir karma yöntem sunmuştur. Köprü tabanlı özellikler, üçüncü taraf tabanlı özellikler ve URL gizleme özellikleri olmak üzere üç özellik kategorisi kullanmışlardır. Üçüncü taraf hizmetlerin kullanımı zaman konusunda dezavantaj olsa da, sistemin doğruluk oranını %99,55'e kadar yükseltmiştir [1].

Chiew ve arkadaşları HEFS adını verdikleri makine öğrenmeye dayalı kimlik avı tespit sistemi önermişlerdir. Bu çalışmada özellik seçim algoritması ile birlikte 48 özellik 10 özelliğe indirgenmiştir. Tüm özelliklerin yalnızca %20,8'i kullanılarak %94,6 oranında başarı elde edilmiştir. 48 özelliğin tamamını kullanarak %96,17 oranında bir başarı elde edildiği belirtilmiştir [29].

Feng ve arkadaşları (2018), Monte Carlo algoritması kullanarak kimlik avı web sitelerinin tespiti için yeni bir sinir ağı tabanlı sınıflandırma yöntemi sunmuşlardır. Seçilen özellikler dört ana guruba ayrılmıştır. Adres çubuğuna dayalı özellikler, anormal temelli özellikler, HTML ve JavaScript tabanlı özellikler ve etki alanı tabanlı özellikler olmak üzere 30 özellik kullanılmıştır. Deney sonuçlarına göre %97,71 oranında doğruluk değeri elde edildiği belirtilmiştir [4].

3. SİBER GÜVENLİK

Siber kelimesi insanların artık günümüzde sürekli kullandığı bir kelime haline gelmiştir. Kabaca elektronik ortamları ifade etmektedir. Siber güvenlik ise siber dünyada kullanıcıların kullanmış olduğu bilişim sistemlerinin saldırı türlerinden korunmasını, bilginin gizlilik, bütünlük ve erişilebilirlik gibi güvenlik unsurlarını güvence altına almayı, saldırıların tespit edilmesini, saldırılara karşı saldırı tespit mekanizmalarını devreye almasını ve saldırıdan önce sistem nasıl ise o hale geri döndürülmesi olarak tanımlanabilir.

Bu başlık altında siber güvenliğin olmazsa olmazları olan siber güvenlik unsurlarının ne olduğu, siber saldırı tehdit türleri hakkında bilgi verilerek bu saldır türlerine karşı alınabilecek tedbirlerden bahsedilmiştir.

3.1. Siber Güvenlik Unsurları

Siber güvenliği yüksek seviye de tutmak için olmazsa olmaz olan aşağıda belirtilen unsurlara dikkat edilmeli ve bu unsurlar yerine getirilmelidir. Bu unsurlar Şekil 3.1'de gösterilmiş olup aşağıda kısaca bilgi verilmiştir [30].

- **Gizlilik,** verinin yetkisiz kişilerin erişimine açık olmaması anlamına gelmektedir. Diğer bir ifade ile sadece erişim yetkisi olan kişilerin veriye erişebilmesini garanti etmektir. Bu unsur şifreleme algoritmaları aracılığı ile sağlanmaktadır.
- **Bütünlük,** siber güvenlik unsurlarından bir diğeridir. Verinin yetkisiz kişilerce silinmemesi, değiştirilmemesi ya da hiçbir şekilde zarar görmemesine neden olacak saldırılardan korunuyor olması anlamına gelir. Kısaca bütünlük verinin kazara ya da kasıtlı bir şekilde bozulmaması olarak tanımlanır. Özetleme fonksiyonları ile ya da farklı fonksiyonlar kullanılarak bütünlüğün bozulup bozulmadığı sağlanır.
- **Kimlik doğrulama,** yetkili olan kullanıcıların kimliğini doğrulaması ve o kişi olduğunu garanti edebilmesi ya da kullanıcı kimliğinin belirlenmesi veya doğrulanması olarak tanımlanabilir. Bu unsur, elektronik imza ile sağlanır.
- **İnkâr edememe,** mesaj veya bilgi kaynağının gönderdiği mesajı yalanlayamamasıdır. Diğer bir ifade ile yetkilendirilmiş kullanıcının mesaj gönderim-alım işlemlerinin ispatlanması ve bu işlemi inkâr edememesini sağlamaktır. Bu işlem açık anahtar altyapısı ile sağlanır.
- **Erişilebilirlik,** yetkili kullanıcıların bilgiye ve ilgili kaynaklara erişim hakkına sahip olmalarının garanti edilmesi, yetkili kullanıcıların sisteme güvenli olarak erişmelerinin garanti edilmesi olarak tanımlanmaktadır. Herhangi bir sorun ya da problem çıkması durumunda bile erişebiliyor olması gerekmektedir.

Şekil 3.1. Siber Güvenlik Unsurları [30]

3.2. Siber Saldırı Tehditleri ve Tedbirleri

İnternet birçok bilgisayar sisteminin birbirine bağlı olduğu, dünya çapında yaygın olan ve sürekli büyüyen iletişim ağına denir. İnternet, insanların her geçen gün gittikçe artan, üretilen bilgiyi saklama/paylaşma ve ona kolayca ulaşma istekleri sonrasında ortaya çıkmış bir teknolojidir. Bu teknoloji yardımıyla insanlar pek çok alandaki bilgilere kolay, ucuz, hızlı ve güvenli bir şekilde erişebilmektedir. İnternetin geniş kitlelere ulaşması ile insanların her türlü bilgiye ulaşabildikleri bir dünya oluşmuştur. Bu dünya, sanal dünya ve siber uzay gibi terimlerle anılmaktadır. Siber uzayın sınırlarının olamaması, ağ şeklinde bir yapıya sahip olması, herhangi bir devletin veya idarenin elinde olmayışından bir devlete ya da bireye verilecek ceza kararını almayı zorlaştırabilir. Bu nedenden ötürü teknik, sosyal, hukuki alanda düzenlemeler yapılmalıdır. Söz konusu siber uzay olunca bir devletin sınırlarını aşıp başka bir devletin sınırları içerisine girmektedir [31].

Siber ortam, kara, deniz, hava ve uzay ortamlarından sonra beşinci savaş alanı olarak belirtilebilir. Günümüzde hemen hemen her bilgi iletişim araçlarında depolanmaktadır. Teknolojik araçların gelişimi siber uzaydaki olanakları arttırıp kötü amaçlı kullanılabilir. Bu durum insanları, devletleri ve kurumları tehdit edebilecek boyuta gelebilmektedir. Tehdit durumunda olan ilgili yerlerin savunma hazırlıkları yapmaları gerekmektedir [31].

Çok çeşitli siber güvenlik tehdit unsurları bulunmaktadır. Virüs, solucan, Truva atı, casus yazılım ya da kendisini bir bankanın resmi sitesi olarak gösteren birebir kopya ve sahte bir internet sitesi tehdit unsurları arasında gösterilebilir. Bireyler siber saldırıların hem vasıtası hem de hedefi konumundadır. Bilinçsiz bir birey bir bağlantıya tıklaması ile verilerini ve bilgisayarını bir saldırgana teslim edebilir. Bunun sonucunda bilgisayarının saldırganlar tarafından ciddi bir suç unsuru yaratacak durumda kullanma olasılığı ortaya çıkmaktadır. Böylelikle bir suça aracı olma konumuna düşmektedir [32].

Siber tehditler sadece bilgi hırsızlığı, casusluk ya da saldırı sonucunda verilecek bir zarar veya elde edilecek haksız bir kazançla sınırlı olmayıp, internetin bilgi çarpıtma maksatlı olarak kullanımı ve siber teröristler tarafından bir iletişim ve propaganda aracı olarak kullanılması da siber güvenlik tehdidi oluşturmaktadır. İstenmediği halde gönderilen elektronik postalar da bu mahiyette değerlendirilmektedir. Bunlar bazen sadece bir reklam, bir ürün pazarlama şekli ve bazen de propaganda aracı olabilmekte iken, bazen de posta eklerinde yer alan dosyalarda bireysel ve ülke varlıklarına çok ciddi zararlar verebilecek virüs, Truva atı gibi sistemleri hedef alan zararlı yazılımlar olabilmektedir. Bu yolla zararlı yazılımlar kendilerini bilgisayardan bilgisayara, ülkeden ülkeye yayabilmektedirler [32].

Teknolojinin sürekli gelişimi ile birlikte bilişim sistemlerine yapılan saldırılar da sürekli gelişmektedir. Bilişim sistemlerinde alınan güvenlik önlemleri de gelişen teknolojiye karşı yetersiz kalmaktadır. Bu alanda gelişen teknoloji ile birlikte bilinen siber saldırı tehditlerine karşı güvenlik önlemleri de arttırılmalıdır. Bazı siber saldırı tehditleri şu şekilde sıralanabilir;

Oltalama: Yasa dışı yollarla kullanıcılara ait şifre veya kredi kartı bilgileri gibi hassas bilgileri elde etmek için saldırganlarca kullanılan bir yöntem olarak tanımlanır. Oltalama yöntemi genellikle, önemli kurumlardan geldiği intibası veren reklam, elektronik posta gibi yöntemlerle yapılmaktadır. 1995 yılında ortaya atılıp, kelime olarak balık avı anlamına gelmektedir. Bu kavrama göre, internet üzerinden kullanıcılara ait hassas bilgileri elde etmek isteyen saldırganlar, balıkçı gibi olta atıp oltaya takılan kullanıcıların hassas bilgilerini elde etmektedir. Bu saldırı türü genel olarak üç aşamadan oluşmaktadır. Birinci aşamada Oltalama saldırısını yapacak olan saldırganlar, bir gerçek kuruma ait sayfayı taklit ederek sahte web sayfaları yapmaktadırlar. İkinci olarak, büyük kullanıcı kitlelerine gerçek bir kuruma aitmiş izlenimi veren reklam, e-posta vb. gibi yöntemlerle kullanıcıları yanıltarak bu sayfalara girmelerini sağlamaktadır. Bu noktada kullanıcıların bu sayfada kendilerine ait kullanıcı adı, şifre gibi özel bilgilerini girmelerini sağlayarak bu bilgilere sahip olmaktadır. Saldırının üçüncü aşamasında saldırganların kullanıcılara ait hassas bilgileri kullanarak kişilerin kredi kartlarını kullanmakta veya hesaplarında bulunan para alınmaktadır.

Şebeke Trafiğinin Dinlenmesi (Sniffing): Kelime olarak koklama anlamına gelmektedir. Bir ağdaki dijital cihazlar arasındaki veri trafiğinin dinlenmesi olarak bilinir. Saldırganların amacı

ağdaki veriyi dinleyerek elde ettiği veriyi çözümleyip ele geçirmektir. Bu saldırı türünden korunmak için dijital cihazlar arasındaki bağlantının şifreli olması gerekmektedir. Şifreleme yapılarak güvenlik arttırılır. Veri dinlenip ele geçirilebilir fakat içerik ele geçirilse bile şifreli olması nedeniyle bir şey anlam ifade etmeyecektir.

Virüs: Virüs, bilgisayar kullanıcılarının sık karşılaştığı bir terimdir. Kapsayıcı olarak bütün zararlı yazılımları ifade eden bir genel ifade olarak kullanılmaktadır. Fakat bu mantıkta yaklaşmak yanlıştır. Bütün zararlı yazılımlar virüs olarak ifade edilmemektedir. Virüs yayılan ve farklı dosyalara bulaşan özel zararlı yazılım olarak ifade edilmektedir [32]. İlk zamanlar bilgisayara virüs CD, disket vb. yapılarla yayılmaktayken, günümüzde internetin hayatımıza girmesiyle beraber bilgisayarlara birçok virüs girmekte ve farklı dosyalara bulaşmaktadır. İnternet vasıtasıyla indirilen programlar, gelen dosyalar, e-posta olarak gönderilen mailler, tıklanılan siteler üzerinden virüsler dijital cihazlarımıza bulaşmaktadır.

Solucan: Solucan virüsü diğer virüs göre daha karmaşık bir yapıda olmaktadır. Bulaşma şekli daha çok e-mail, bazı web sayfalarına girişte ya da ortak ağ üzerinde bulunan dosyaların paylaşımı esnasında bulaşmaktadır. Bu virüs bilgisayar bulaştıktan sonra kullanıcının sistemini kullanarak kendisini kopyalamaya başlamaktadır. Kullanıcının e-mail ya da diğer iletişim kaynaklarını kullanarak daha fazla bilgisayara yayılmaktadırlar. Bir sistemde aktif olduklarında çeşitli sıkıntılara neden olabilirler. Kurulmuş olan bir programı pasif hale getirme, var olan dosyaları silme, sistem başarımını düşürme gibi sıkıntılar oluşturabilmektedir. Bunları yaparken kullandığı bilgisayarın yavaşlamasına neden olmaktadır. Ağ kaynaklarında yavaşlama meydana getirirler. Solucan virüsünün bulaşmasını sağlayan saldırgan bilgisayarlardan bilgi alabilir ve bilgisayarları kontrol edebilir.

Truva: Truva, güvenli görünen ancak giriş yaptığı sistemin arka planına sızarak bilgisayarda olan biten tüm verileri internet korsanlarına sunabilen virüs ve casusluk yazılımlarıdır. Truva atı bulaşmış bir bilgisayarda şifre girerek kişisel hesaplarınızı açmak ve hatta bilgisayarı açık ve internete bağlı vaziyette tutmak dahi büyük risk teşkil etmektedir. Bu yapı ile saldırganlar kullanıcılara ait şifre, belge, kredi kartı bilgileri gibi bilgilere erişerek kendi cihazlarına yönlendirmektedir. Virüsler gibi kendilerini kopyalama özellikleri yoktur. Aktif olmaları için çalıştırılması gerekmektedir [33].

Kod istismarı: Bu yöntemle saldırganlar donanımsal ve yazılımsal sistemler üzerindeki güvenlik zafiyetlerinden, aksaklıklardan veya hatalardan yararlanabilirler. Kod istismarı, komut diziniyle veya bir yazılım parçası ile gerçekleşebilir [34].

Belirtilen tehditler dışında MAC aldatmacası, IP aldatmacası, DNS zehirlenmesi, ARP zehirlenmesi, HTTP taşması, SYN taşması, ACK/FIN/PUSH taşması, UDP taşması, DNS taşması, işletim sistemi tarama, IP tarama, kod istismarı, kök dizin, hizmet engelleme ve CSRF gibi durumlarda siber saldırı tehditleri olarak belirtilebilir.

Siber savunma tehditlerine karşı alınabilecek tedbirler için bu bölümde belli başlı bazı yöntemlere değinilmiştir. Bu yöntemlerden bazıları şu şekildedir;

Saldırı tespit sistemi: Olası ihlalleri ve tehditleri tespit etmek için ağda oluşan aktivitelerin izlenmesi ve trafik analizi yapılmasını sağlayan güvenlik mekanizmalarından biridir. Saldırı tespit sistemi ağ üzerinden yapılacak saldırılara karşı kullanılan yazılım veya donanım bileşenlerinden oluşturulmuş saldırganın sisteme sızmasını önlemek için kullanılan bir yapıdır. Saldırı tespit sistemi sayesinde bir saldırganın profili, güvenlik zaafiyetleri ve saldırı türü gibi bilgiler elde edilmektedir.

Saldırı engelleme sistemi: Saldırı engelleme sistemleri saldırıları saptamak önlenmek gibi işlemleri kapsayan ağ güvenlik sistemleridir. Saldırı engelleme sisteminde önceden yapılan saldırı imzaları bulunur. Bu saldırı imzalarıyla eşleşen yeni saldırıların tespiti durumunda TCP işlemi sonlandırma, paket düşürme gibi işlemleri yapabilirler. Güvenlik duvarı üzerinden geçen paketlerin incelenmemesi ve yalnızca tablodaki kurallar çerçevesince paketlerin geçişine izin verilmesi nedeniyle yapılan saldırıların sadece güvenlik duvarıyla tespit edilip engellenmesi mümkün değildir. Bu nedenden ötürü ağ üzerinden geçen paketlerin izlenip incelenmesi, gerektiği durumda izin verip ya da reddedip sistemi koruyan güvenlik uygulamaları ihtiyacı duyulmaktadır.

Güvenlik duvarı: Güvenlik duvarı, bir ağ üzerinde, gelen ve giden trafiği izleyen ve ağdaki muhtemel güvenlik açıklıklarından bilgisayarları koruyan bütün yazılımlara verilen addır. Güvenlik duvarının tek görevi ağ üzerinden gelen tehditlere karşı kullanıcıların bilgisayarlarını korumaktır. Bu mekanizma aktif olduğu takdirde kullanıcılar internetten gelecek olan tehlike ve tehditlere karşı bu yazılımlar aracılığı ile korunabilir.

Web uygulama güvenlik duvarı: Bir web uygulama güvenlik duvarı olan (WAF), bir web uygulamasına veya sunucuya gelen HTTP trafiğini izleyebilen, filtreleyen veya engelleyebilen bir güvenlik duvarıdır. Web uygulama güvenlik duvarı sayesinde web sunucularına ve hizmetlerine yapılabilecek saldırılar engellenebilir.

Veri tabanı güvenlik duvarı: Veri tabanı güvenlik duvarı, gelecek olan saldırıların veri tabanına ulaşmasını engellemeye yardımcı olan bir savunma hattıdır. Ağ güvenlik duvarı, dış erişime karşı veri merkezine kötü niyetli olan yaklaşımlardan koruma konusunda önemli rol oynamaktadır. Fakat veri merkezlerine karşı oluşan tehditler zamanla daha tehlikeli ve detaylı olmuştur. Bazı güvenlik açıklıklarından faydalanarak sisteme sızılmış, kullanıcı yetkileri ele alınarak sistem yönetimi ele geçirilebilir hale gelmiştir. Bu nedenle veri tabanı içeriğinin korunması için güvenlik seviyesini üst düzeye çıkarmak zorunluluk haline gelmiştir. Veri tabanı güvenlik duvarı uygulama davranışlarını monitorize edip bir savunma katman yapısı oluşturup ve kötü niyetli olarak veri tabanına ulaşmaya çalışma girişimlerini engelleyip bir güvenlik mekanizması oluşturmaktadır.

Bu tedbirler dışında ağ erişim denetimi, yük dengeleyici, URL filtreleme, vekil sunucu, zafiyet tarama, risk analizi yönetim sistemi, güvenlik bilgileri ve olay yönetimi, veri kaçağı önleme gibi tedbirlerde ilgili tehditlere karşı tedbir olarak alınmaktadır.

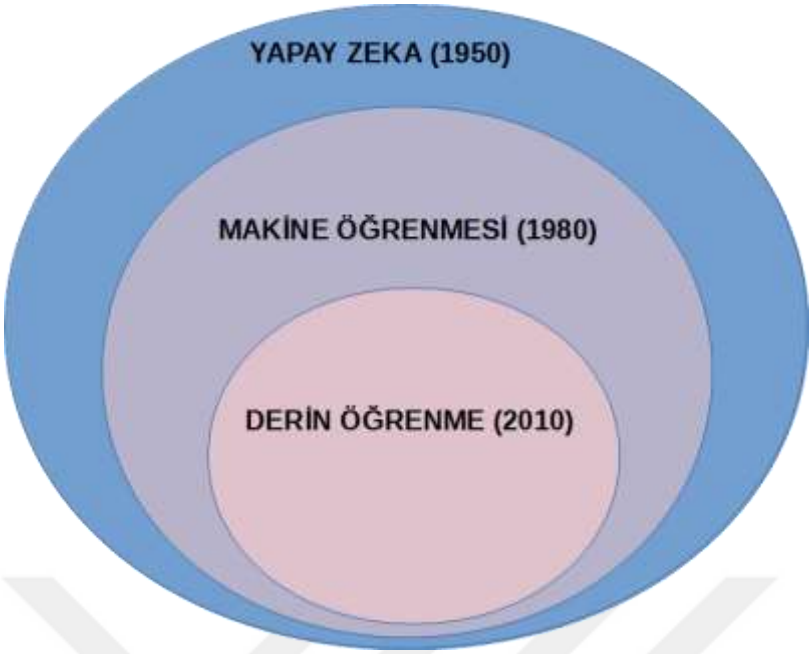
4. MATERYAL VE METOT

4.1. Derin Öğrenme

Derin öğrenme, insan beyninin zor problemleri çözmek için kullandığı yöntem ve kabiliyetlerden esinlenerek, büyük miktarda veriden faydalanarak, özellik çıkarma ve sınıflandırma gibi işlemleri yapma yeteneğine sahip makine öğrenmesinin alt dalıdır. Yapay sinir ağı araştırmalarının tarihsel süreç içindeki gelişmelerinden ortaya çıkan derin öğrenme kavramı "yeni nesil sinir ağları" olarak anılmaktadır. Yapısal olarak insan gibi düşünüp karar veren yapılar oluşturmayı hedefleyip bu hedef doğrultusunda sayısal veri, metin, ses gibi veri türlerini kullanır.

Günümüz çalışmalarında karmaşıklığa sahip olan yapılarda derin öğrenme kullanımının, karmaşık matematiksel problemleri çözümlemede sığ mimarilere göre daha iyi sonuçlar verdiği görülmüştür. Temelde derin öğrenme bir yapay sinir ağıdır. Derin olarak belirtilmesinin sebebi ise yapay sinir ağı yapısından kaynaklanmaktadır. İlk olarak 1970'lerde yapay sinir ağıları ortaya çıktığında birkaç katmandan oluşmaktayken günümüzde bu katman sayısı gittikçe artmaktadır. Yakın tarihlerde yapay sinir ağlarının eğitiminin zorlu olduğu belirtiliyordu. Donanımsal eksiklikler sebebiyle bu zorlukların oluştuğu görülmekteydi. Teknolojinin gün geçtikçe gelişmesiyle beraber yaşanılan zorluklar giderilerek katman sayısı arttırılmaya başlanmıştır. Bu katmanların artması ile derin ağ yapıları oluşmuştur. Deney sonuçlarına göre derin ağlar bir, iki gizli katmana sahip sinirsel ağlardan daha iyi performans göstermektedir. Derin öğrenme ise diğer yapılara göre, karmaşık yapıdaki durumları çözmede, analiz yapmada, çok büyük veri örneklerini ve etkisiz verileri değerlendirmede diğer yöntemlere nazaran daha iyi sonuçlar vermektedir. Derin öğrenme yaklaşımları, sınıflandırma, doğal dil işleme, görüntü işleme, konuşma tanıma vb. konularda kullanılmaya uygun olup ve bu yapı sığ mimarilere kıyasla çok daha güçlü olduğu belirtilmektedir [35].

Derin öğrenme, yapay zeka ve makine öğrenmesi kavramları birbirleriyle iç içe olan kavramlardır. Şekil 4.1'de yapay zeka, makine öğrenmesi ve derin öğrenemeye genel bakış belirtilmiştir.



Şekil 4.1. Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenemeye Genel Bakış [36]

4.1.1. Makine Öğrenmesi

Makine öğrenmesi temel olarak bilgisayar biliminin yapay zekâda sayısal öğrenme ve model tanıma çalışmalarından geliştirilmiş bir alt daldır. Makine öğrenme algoritmaları öğrenen ve girdiler üzerinden tahmin yapan bir yapıdır. Bu yapılar standart programlama talimatlarından farklı olarak girilen bilgilerden tahmin ve karar verme amaçlı model oluşturarak çalışırlar. Makine öğrenme algoritmalarında temel dayanak, giriş verisi alan algoritma oluşturma ve çıktıları yeni girişler olukça güncelleme ve çıktıların tahmini için istatistiksel analiz kullanmaktadır [37].

İnsanlar, internet üzerinden paylaşılan alışveriş vb. gibi reklamların yayınlanmasından makine öğrenimini bilmektedir. Bu durumun sebebi, öneri motorları anlık olarak kullanıcıların aramalarına göre kişiselleştirmiş reklam yayınlamak için makine öğrenim yöntemini kullanmasıdır. Bu durum dışında makine öğrenimi, sahtekârlık tespitinde, spam filtrelemede, ağ güvenliğini tehdit eden unsurları algılamada vb. gibi alanlarda kullanılmaktadır [38].

Makine öğreniminin en temel amacı verilerin olasılık dağılımını elde etmek olduğundan, dağılımı oluşturmak için birçok farklı algoritma kullanılmaktadır. Bu algoritmalar, denetimsiz ve denetimli öğrenme olarak iki gruba ayrılmaktadır.

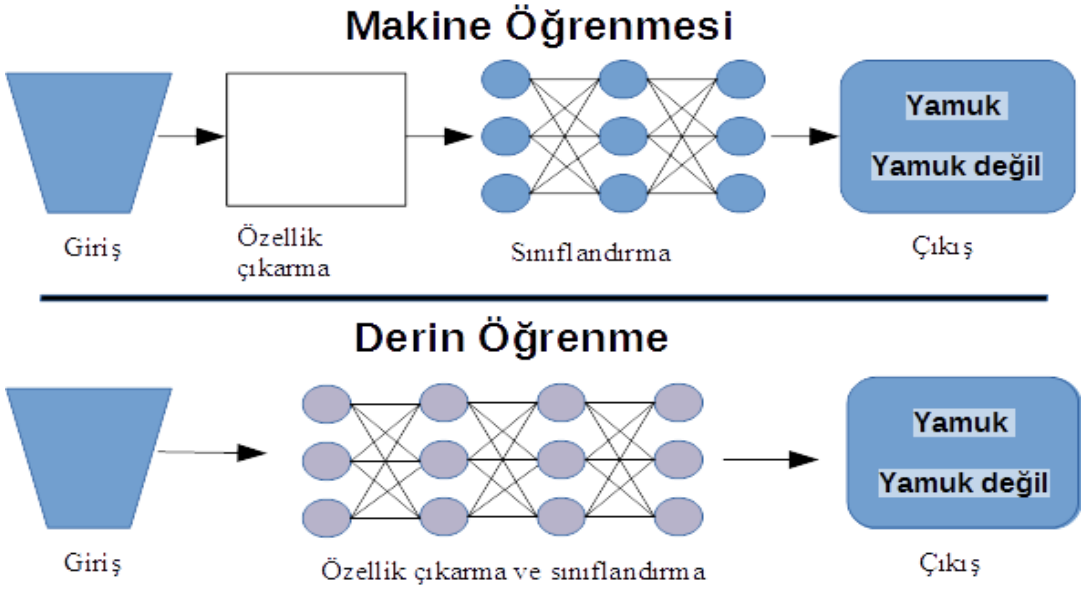
Denetimli öğrenmede veri kümesi ve bu veri kümesinden istenilen çıktıların nasıl olması gerektiği bilinmektedir. Denetimli öğrenme, verileri ve o verilerden çıkan sonuçları makineye tekrar vererek elde edilen bilgilerden bir fonksiyon çıkarılmasını sağlamaktadır. Bu sayede makine veriler arasındaki ilişkiyi öğrenebilmektedir. Denetimli öğrenme problemleri regresyon ve

sınıflandırma olmak üzere ikiye ayrılmaktadır. Bir sınıflandırma probleminde, her bir gözleme bir kategori/sınıf atması yapılmaktadır. Sınıflar sayısal değildir ve birbirlerine yakın/uzak olmaları gibi bir durum söz konusu değildir. Tümörlü bir hasta göz önüne alınarak, tümörün kötü huylu ya da iyi huylu olup olmadığını ön görmek örnek verilebilir. Bir regresyon probleminde ise, her bir gözlem için öğrenilene bakılarak reel bir değer tahmini yapılmaktadır. Resmi verilen bir kişinin, verilen resimlere göre yaşını tahmin etmek örnek verilebilir [39].

Denetimsiz öğrenmede, verilerden elde edilmek istenen çıkış bilgisinin nasıl olduğu hakkında az ya da hiç fikir olunmayan durumlarda kullanılan yaklaşımlardır. Değişkenlerin etkisinin bilinmediği verilerden model oluşturulabilir. Denetimsiz öğrenmede sadece veriler vardır ve hakkında bilgi verilmemektedir. Bu verilerden sonuçlar çıkarılmaya çalışılmaktadır. Veriler hakkında herhangi bir bilgi verilmediği için sonuçların kesinlikle doğru olduğu ifade edilememektedir [39].

4.1.2. Derin Öğrenme ve Makine Öğrenmesi Farklılıkları

Derin öğrenme algoritması çok fazla veriye ihtiyaç duymaktadır. Bu veriler üzerinden kendi özelliklerini kendisi çıkarmaktadır. Makine öğrenmesi yönteminde özelliklerin verilmesi, belirtilmesi gerekmektedir. Derin öğrenmede özelliklerin otomatik olarak çıkarılması en temel fark olup Şekil 4.2'de de görülebileceği gibi geleneksel makine öğrenme modelinde sınıflandırma yapmadan önce her sınıfın özelliğinin belirtilmesi gerekirken, derin öğrenme de bu özellikler otomatik olarak çıkartılır ve öğrenilir. Başka bir deyişle, derin öğrenme de denetimsiz öğrenme yapma kabiliyeti vardır [40].



Şekil 4.2. Derin Öğrenme ve Makine Öğrenmesi Farkı [40].

Tablo 4.1'de makine öğrenmesi ve derin öğrenmenin veri bağımlılığı, donanımsal bağımlılıklar, özellik çıkarma ve uygulama süresi gibi durumlar için karşılaştırılması yapılmıştır. Yapılan karşılaştırılmaya göre derin öğrenme algoritmalarının daha iyi başarım gösterebilmesi için çok sayıda veriye güçlü donanımlara ihtiyaç duymakta ve bu nedenden ötürü uygulama süresini çok daha uzun sürdüğü görülmektedir. Tüm bu durumlar dezavantaj olarak düşünülebilir fakat bu bağımlıklar derin öğrenme algoritmalarının en önemli avantajı durumunda olan özellik çıkarma kabiliyetinin maliyeti olarak oluşmuştur.

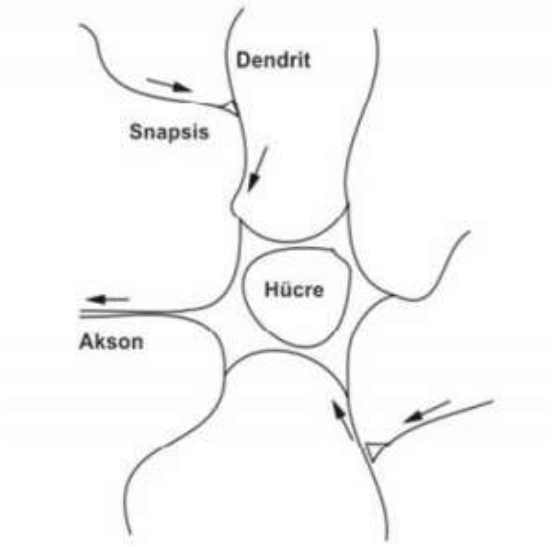
Tablo 4.1. Makine Öğrenmesi ve Derin Öğrenme Karşılaştırması [40]

Karşılaştırma Parametresi	Derin Öğrenme	Makine Öğrenmesi
Veri Bağımlılığı	Büyük bir veri setinde mükemmel performans	Küçük/orta ölçekli veri kümesinde mükemmel performans
Donanımsal Bağımlılıklar	Güçlü bir makine gerektirir, tercihen GPU: çok sayıda matris çarpımı gerçekleştirme kabiliyetine sahip bir donanıma ihtiyaç duyar	Düşük kaliteli bir makinede çalışabilir
Özellik çıkarma	Verileri temsil eden en iyi özelliği bilmeye ihtiyaç duymaz	Verilerin temsil ettiği özellikleri bilmeye ihtiyaç duyar
Uygulama Süresi	Haftalar sürebilir. Bunun sebebi yapay sinir ağının önemli miktarda ağırlık hesaplamasının gerekmesidir	Birkaç dakikadan saatlere kadar sürebilir

4.1.3. Yapay Sinir Ağları

Bilgisayar teknolojisinin gelişmesiyle beraber insanlara neredeyse bütün işlemleri bu gelişen teknoloji ile yapmakta ve yeni yöntemler bulunmasına imkân sağlamaktadır. 1980'li yıllarda belirtilen makinaların insanlara benzer düşünebilmesi fikri, 1990'lı yıllarda Yapay Sinir Ağları (YSA) teknolojisi adıyla büyük bir gelişme görülmüş ve bayağı hızlanmıştır. YSA, Yapay Zeka kavramının altında oluşan konu olup, bu teknolojiye ilgi duyan araştırmacıların odak noktası haline gelmiştir.

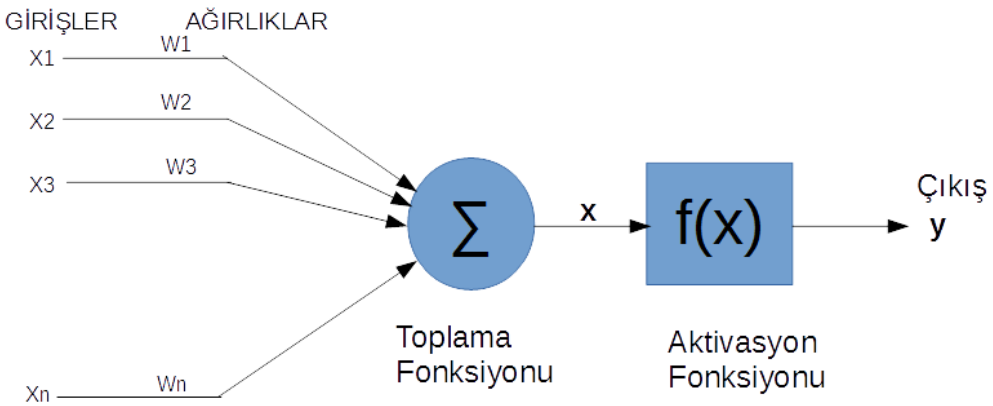
YSA, insan beyni özelliği olan öğrenme yolu ile yeni bir bilgi türetme, yeni bilgiler oluşturabilme ve keşfedebilme gibi özellikleri, herhangi bir yardım almayıp otomatik olarak gerçekleştirebilmek için ile geliştirilen bilgisayar sistemleridir. İnsan beyninin biyolojik nöronundan esinlenip öğrenme işlevi tasarlanmıştır. Biyolojik sinir hücresi yapı olarak Şekil 4.3'te gösterilmiştir.



Şekil 4.3. Biyolojik Sinir Hücresi – Nöron [41]

Biyolojik sinir sistemi temel yapısını oluşturan nöronlar, dendrit, akson, çekirdek ve snapsis adında dört ana kısımdan oluşmaktadır. Dendritler sinir hücresinin ucunda bulunmaktadır. Ağaç kökü görünümlü bir şekildedir. Dendritler kendilerine bağlı halde bulunan bir nöron veya duyu organından aldığı sinyali çekirdeğe göndermektir. Çekirdek ise dendritten aldığı sinyalleri bir araya toplayıp sonrasında aksona gönderir. Akson toplanan bu sinyalleri işleyip, nöronun diğer ucundaki snapsislere gönderir. Yeni üretilen bu sinyalleri snapsis ise diğer nöronlara gönderir.

Biyolojik nöron ile benzer şekilde, yapay sinir hücreleri Şekil 4.4'de belirtildiği gibi bilgileri toplama fonksiyonu vasıtasıyla toplar, bu toplamı bir aktivasyon fonksiyonundan geçirerek anlamlı bir çıktı üretir ve bu çıktıyı ağın bağlantıları ile ağın diğer elemanları ile paylaşır. Bu aşamalarda ihtiyaca göre değişik toplama ve aktivasyon fonksiyonları kullanılabilir [42].



Şekil 4.4. Yapay Sinir Ağı Yapısı

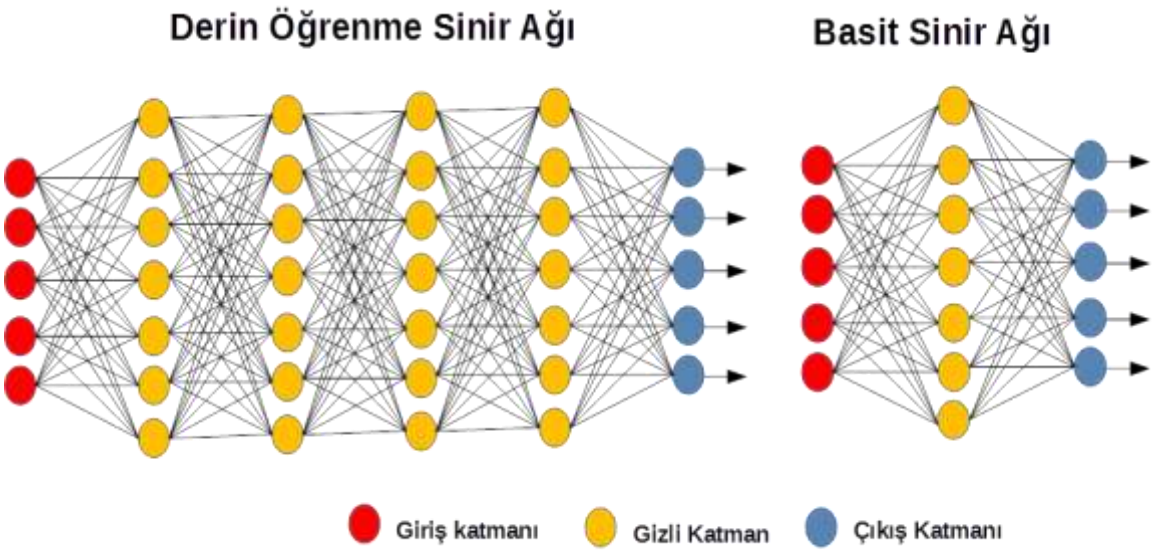
Bir biyolojik sinir hücresindeki sinaps YSA'da ağırlıkları, dentrit toplama fonksiyonunu, çekirdek aktivasyon fonksiyonunu ve akson çıktıyı temsil etmektedir. YSA 3 katmandan oluşur.

Bunlar:

- Giriş katmanı,
- Ara(gizli) katman,
- Çıkış katmanıdır.

Giriş katmanı, konuyla ilgili verilerin dış dünyadan geldiği katmandır. Bu katmanda genellikle veri üzerinde herhangi bir işlem uygulanmadan kendinden sonra gelen katmanlara aktarılır. Ara (gizli) katman ise giriş katmanının çıktılarını girdi olarak alır. Bu katmanda, hesaplama karmaşıklığını ve zaman performansını etkileyecek hususlar söz konusudur. YSA'da gizli katmanlar birden çok olabilip ya da hiç olmayabilir. Gizli katmanlardaki nöron sayıları birbirinden faklı olabilir. Ara katman ve nöronların sayılarının artması hesaplama karmaşıklığını ve zamanını arttırabilir. Bu maliyetin artmasına rağmen karmaşık problemlerin çözümünde kullanılabilir. Çıkış katmanı ise gizli katmandan gelen bilginin çıktısını dış dünyaya gönderen katmandır.

1970'lerde YSA ortaya çıktığında birkaç katmandan oluşmaktayken günümüzde bu katman sayısı gittikçe artmaktadır. Katman sayısının armasıyla derin öğrenme ortaya çıkmıştır. Derin öğrenme ağ yapısını basit bir YSA'dan ayıran en temel özellik daha karmaşık ağ yapısına sahip olması ve birden fazla gizli katmana sahip olmasıdır. Basit sinir ağıyla derin öğrenme sinir ağı karşılaştırması Şekil 4.5'te gösterilmiştir.



Şekil 4.5. Basit Sinir Ağıyla Derin Öğrenme Sinir Ağı Karşılaştırması [43]

4.1.4. Derin Öğrenme Mimarileri

YSA'daki katman sayısı arttırılarak oluşturulan çok farklı türde derin öğrenme mimarileri bulunmaktadır. Sık kullanılan mimarilerden bazıları şu şekildedir;

- **Konvolüsyonel sinir ağları:** Konvolüsyonel Sinir Ağları (Convolution Neural Network-CNN) hayvanların görme merkezinden esinlenilerek ortaya atılmıştır. Konvolüsyonel sinir ağları görüntü sınıflandırma, nesne tanımlama gibi görüntü tabanlı çalışmalarda oldukça başarılıdır.
- **Tekrarlayan sinir ağları:** Tekrarlayan sinir ağlarının (RNN) amaç olarak ardışık bilgileri kullanmaktır. Bir sinir ağında giriş ve çıkışlar birbirinden bağımsız olarak varsayılır. RNN'ler tekrarlayan olarak adlandırılmakta çünkü bir diziye ait her öğe aynı görevi yerine getirmekte ve çıktı önceki hesaplamalara bağlı olmaktadır. Gizli katmanda çıkış tekrar aynı katmana ve sonra gelen katmana giriş olarak verilmektedir. RNN'ler birçok doğal dil işlemede büyük umut vaat eden popüler modellerdir.
- **Sınırlı boltzmann makineleri:** Sınırlı boltzmann makineleri girdi seti üzerinde olasılık dağılımını öğrenebilen üretken bir rastgele YSA'dır. İki katmanlı yapıya sahip olup, ilk katman giriş ve ikinci katman ise gizli katman olarak adlandırılmaktadır. Boyut indirgeme, işbirlikçi filtreleme, sınıflandırma, konu modelleme ve özellik öğrenimi gibi farklı konular için kullanışlı bir algoritmadır.
- **Derin oto-kodlayıcılar:** Diabolo ağı olarak adlandırılan Oto-kodlayıcılar, denetimsiz öğrenme için kullanılan bir özel YSA'dır. Oto-kodlayıcılar, bir veri kümesi için boyut indirgeme amacıyla bir temsil (kodlama) öğrenmeyi hedefler. Oto-kodlayıcılar, kabaca girdi verisinin sıkıştırılmış gösteriminden en iyi özelliklerin öğrenilmesini hedefleyen bir ileri beslemeli sinir ağıdır.

4.1.5. Derin Öğrenme Süreçleri

Bir problemin derin öğrenme ile çözümü uygun olmayabilir. Bunun için öncelikle problemin belirlenmesi ve derin öğrenme algoritmaları ile çözümünün mümkün olup olmadığı değerlendirilir. Eğer uygun ise derin öğrenme için belirli süreç adımları sırasıyla yapılması gerekir. Derin öğrenmenin süreçleri şu şekilde sıralanabilir [44].

I. Problemin tanımı ve derin öğrenme ile çözümünün uygun olup olmadığının tespiti
II. İlgili veri kümelerinin tanımı ve analize hazır hale getirilmesi
III. Uygun derin öğrenme mimarisinin seçilmesi
IV. Veri seti ve seçilen derin öğrenme mimarisi kullanılarak sistemin eğitilmesi
V. Eğitilen sistemin eğitimde kullanılmayan test verileriyle performansının test edilmesi

4.1.6. Derin Öğrenme İçin Kullanılan İşletim Sistemleri, Programlama Dilleri ve Kütüphaneler

Derin öğrenmenin işletim sistemi bağımlılığı bulunmamaktadır. Windows, Linux, Mac OSX, Android ya da iOS kullanılabilir. Bu çalışmada Windows işletim sistemi üzerinde geliştirme yapılmaktadır.

Derin öğrenmede farklı farklı programlama dilleri kullanılmaktadır. Genel olarak Python, MATLAB, C/C++, Java dilleri tercih edilmektedir. Bu çalışmada Python programlama dili tercih edilmiştir.

Derin öğrenmede kullanılan programlama diline göre pek çok kütüphane mevcuttur. Yaygın olarak kullanılan kütüphaneler Tablo 4.2'de gösterilmiştir.

Tablo 4.2. Derin Öğrenme Kütüphaneleri[40]

Kütüphane	Arayüz	Sahibi
TensorFlow	Python, C++, Java	Google
Theano	Python,C++	Montreal Institute for Learning Algorithms(MILA)
Caffee	Python, C++, Matlab	Berkeley Vision and Learning Center (BVLC)
Torch/PyTorch	Lua, Python	Ronan Collobert Diğerleri
CNTK	Python, C++	Microsoft
Deeplearning4j	Java, Scala, C	Skymind
MatConvNet	Matlab	Andrea Vedaldi, Karel Lenc

4.2. Derin Öğrenmede Kullanılan Hiper-Parametreler

Hiper-parametre, ne olması gerektiği model tasarımcısına bırakılmış, problem, aktivasyon fonksiyonu, öğrenme katsayısı, katman sayısı, nöron sayısı gibi tercihlere denilmektedir. Bu parametreler veri setine göre değişiklik göstermektedir. Bu başlık altında bazı hiper-parametrelere değinilmiştir.

4.2.1. Veri Setinin Boyutu

Bir veri setinin büyüklüğü çeşitliliği öğrenme için kilit faktördür. Veri seti büyüklüğü arttıkça öğrenme oranı da artacaktır. Veri seti boyutu arttıkça öğrenme için harcanan zamanda artmaktadır. Sürekli eğitilmeyen ve depolama sorunu olmayan projelerde öğrenme başarısını arttırmak için zaman ve depolama durumu göz ardı edilebilmektedir. Veri seti büyüklüğü arttıkça öğrenmede artmakta fakat sonsuza kadar devam etmemektedir. Belirli bir andan itibaren küçük küçük artmaktadır. Eğer başarım düşük boyutlu bir veri setinden elde edilebiliyorsa, depolama alanı da tasarımcı için önemli bir durumsa veri seti düşük boyutta tutulabilir.

4.2.2. Optimizasyon Algoritması Seçimi

Derin öğrenmede öğrenme işlemi temelinde bir optimizasyon problemi mantığı bulunmaktadır. Doğrusal olmayan problemlerinin çözümünde optimum değerleri bulma amacı ile optimizasyon yöntemleri kullanılmaktadır. Derin öğrenmede genel olarak stochastic gradient descent, adagrad, adadelta, adam, adamax gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoritmaların arasındaki fark başarı ve hızdır.

4.2.3. Eğitim Tur (Epoch) Sayısı

Derin öğrenme uygulamalarında bir modelin eğitimi sırasında verilerin tamamı eğitime katılmayıp bölük bölük parçalar halinde eğitime katılmaktadır. İlk parça eğitildikten sonra model başarımı test edilerek başarım oranına bağlı olarak geri yayılım (backpropagation) ile ağırlıklar güncellenip, yeni eğitim kümesi ile eğitim devam ederek ağırlıklar yeniden güncellenir. Bu işlem eğitim sırasında her adımda tekrarlanarak en uygun ağırlıklar hesaplanır. Eğitim işlemindeki her adıma epoch denilmektedir. Epoch değeri probleme göre değişmektedir. Epoch sayısı arttıkça başarım oranı da artmaktadır. Belirli bir epoch değerinden sonra öğrenme başarımında küçük artışlar gözlenmektedir. Küçük miktarlarda artışlar gözlendikten sonra eğitim sonlandırılabilir.

4.2.4. Katman Sayısı

Derin öğrenme yöntemini YSA'dan ayıran en önemli özelliği karmaşık problemlere başarılı sonuç vermesidir. Bu durumu sağlayan özellik ise derinlik kavramının ortaya çıkmasını sağlayan katman sayısıdır. Katman sayısının artmasıyla beraber verinin genel hatlarındaki diğer özellikler sonraki katmanlarda öğrenilmektedir. Katman sayısının artması öğrenmeyi arttırmakta fakat belirli katmanlardan sonra geri besleme etkisi ilk katmanlara daha az ulaşabilmektedir. Bu nedenle belirli bir katman sayısından sonra öğrenme başarımına çok fazla etki etmemektedir.

4.2.5. Nöron Sayısı

Derin öğrenme uygulamasında kullanılacak nöron hafızada tutulacak bilgi sayısını belirtmektedir. Bu sayının fazla olması bellek ihtiyacını ve hesaplama zamanını arttırmaktadır. Nöron sayısının az olması ise yetersiz uyuma sebep olmaktadır.

4.2.6. Aktivasyon Fonksiyonları

Bu fonksiyonlar hücreye gelen girişi işleyip hücrenin girişe karşılık üreteceği çıkışı belirler. Derin öğrenme, doğrusal yapıda olmayan (non-linear) problemlerin çözümünde diğer yöntemlere

göre daha başarılı olduğundan, derin öğrenme yöntemleriyle çözülmeye çalışılan problemler genel olarak doğrusal olmayan non-linear bir problemdir. Matris çarpımında elde edilen değerin non-linear hale dönüştürülmesinde aktivasyon fonksiyonu kullanılmaktadır. Ayrıca aktivasyon fonksiyonları geri türev alma işleminde gizli katman çıktılarının normalize edilmesi için kullanılmaktadır [43].

Bir projede kullanılacak aktivasyon fonksiyonun seçimi modelin öğrenme başarısında büyük etkiye sahiptir. Doğru ve mantıklı sonuçlara varmak için modelin katmanlarında farklı farklı aktivasyon fonksiyonları kullanılabilir ancak katman için seçilen aktivasyon fonksiyonu katmandaki tüm nöronlar için geçerlidir.

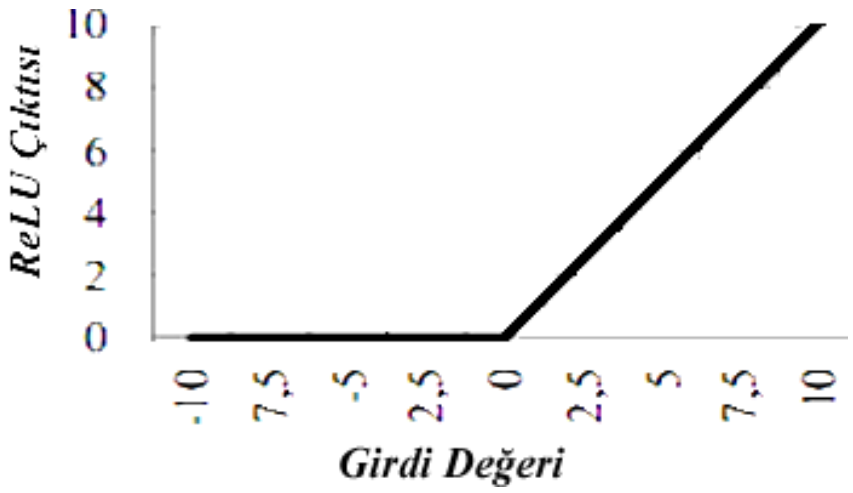
Yaygın aktivasyon fonksiyonları: Sigmoid, Softmax, ReLU, TanH, SoftPlus, ELU, PReLU, Swish'dir. En yaygın kullanılan ReLU, Softmax ve Sigmoid'dir [35].

- **ReLU (Rectified Linear Unit) Fonksiyonu:**

Doğrultulmuş lineer birim (Rectified Linear Unit-ReLU) doğrusal olmayan bir fonksiyondur. ReLU fonksiyonunda girdilerin negatiflik durumunda 0 değerini alırken, pozitiflik durumunda k değerini almaktadır.

$$f(x) = \begin{cases} k \text{ eğer}, & k \geq 0 \\ 0 \text{ eğer}, & k < 0 \end{cases}$$

Şekil 4.6'da ReLU fonksiyonu grafiği gösterilmektedir.



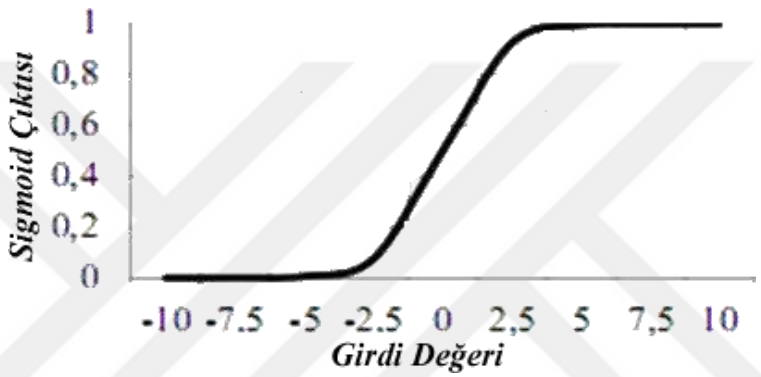
Şekil 4.6. ReLU Fonksiyonu [35]

- **Sigmoid Fonksiyonu:**

Sigmoid aktivasyon fonksiyonu sürekli ve türev alınabilir bir fonksiyondur. Bu fonksiyon her girdi değerini için 0 ile 1 aralığında değer üretmekte. İkili sınıflandırmada kullanılmakta ve genel olarak son katmanda bulunmaktadır.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Şekil 4.7'de Sigmoid fonksiyonu grafiği gösterilmektedir.



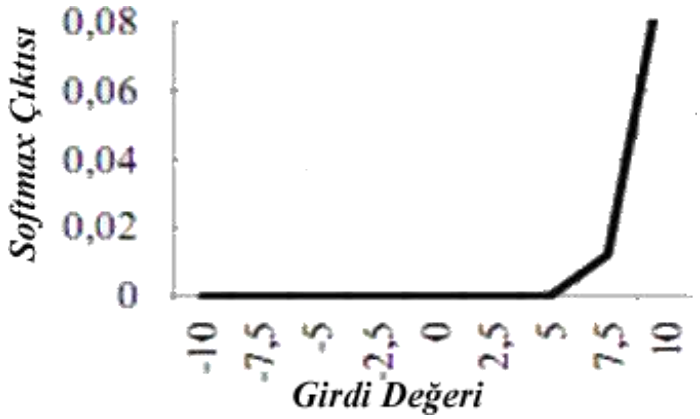
Şekil 4.7. Sigmoid Fonksiyonu [35]

- **Softmax Fonksiyonu:**

Çoklu sınıflandırma durumlarında kullanılan bu fonksiyondur. Verilen her bir girdinin bir sınıfa ait olma olasılığını gösteren [0,1] aralığında değerler üretmektedir.

$$f(x_i) = \frac{e_i^x}{\sum_i^k e_i^x}\ , i = 0,1,2,3, \dots, k$$

Şekil 4.8'de Softmax fonksiyonu grafiği gösterilmektedir.



Şekil 4.8. Softmax Fonksiyonu [35]

4.3. Veri Seti

Gerekli çalışmanın yapılıp özelliklerin incelenmesi için iki ayrı zaman diliminde, Ocak-Mayıs 2015 ve Mayıs-Haziran 2017 arasında web siteleri toplanmıştır. Web siteleri özellikle PhishTank ve OpenPhish'ten URL'leri temel alan 5000 kimlik avı web sitesi ve 5000 meşru web sitesi seçilmiştir [29].

Web sayfalarının toplanması aşamasında işlemleri otomatikleştirmek adına GNU Wget6 aracı ve Python betiği kullanılmıştır. HTML belgeleri haricinde indirilen web sitelerinin tarayıcıda düzgün bir şekilde çalıştırılması için CSS, JavaScript ve görüntüler vb. kaynaklar indirilmiştir. Ayrıca her web sitesinin ekran görüntülerini daha fazla inceleme yapılabilmesi için saklanmıştır [29].

Kimlik avı tespiti için yapılan çalışmalarda çıkarılan özellikler dahili ve harici özellikler olmak üzere iki kategoriye ayrılabilir. Dahili özellikler bir web sitesi HTML kaynak kodu ve URL'sinden oluşturulur. Dahili özelliklerin tamamı web sitesinden oluşturulabilmektedir. Harici özellikler ise etki alan kaydı, arama motoru, WHOIS vb. gibi kayıtlardır. Harici özellikler üçüncü taraf hizmetlerin sorgulanması ile elde edilebilir. Bu veri seti oluşturulurken sadece dahili özellikler baz alınmıştır. Harici özelliklerin incelemeye alınmamasının nedenleri şöyle sıralanmaktadır [29]:

a) Web sitelerin kimlik avı olup olmaması hususunu değerlendiren en temel veri web site URL'i ve HTML kaynak kodudur. Veri setinde sürekli erişilebilir özelliklerin kullanılması, kimlik avı tespiti yapan araştırmacıların veri setine ilgili olmasını sağlayacağı düşünülmektedir.

b) Harici veriler sürekli değişim halindedir. Arama motorlarının sonuçları değişim halinde olması örnek olarak gösterilmektedir.

c) Harici verilere sürekli ulaşımın olmaması durumunun söz konusu olabileceği düşünülmektedir. Google PageRank sorgulanması 2016'da durdurulması örnek olarak gösterilmektedir. Bu nedenle Google PageRank sorgulamasını kullanan kimlik avı saptama tekniklerinin etkilendiği açıkça ortadadır.

Özellik çıkarımı işlemini otomatikleştirmek için tarayıcı komut dosyası tabanlı Selenium WebDriver adlı bir yapı kullanılmıştır. Selenium WebDriver, yapılacak test işlemleri için yerel bilgisayarımızda kullanabileceğimiz bir API'dir. Selenium WebDriver, herhangi bir web sayfası üzerinde herhangi bir kullanıcının yapabileceği bütün işlemleri otomatize edilebilmektedir. Web sayfalarını tarayıcıya yükleyebilmek için bir Python betiği kullanılmıştır. Bu yapılar sayesinde web sayfaları analiz edilerek özellik değerleri çıkarılıp bir metin dosyasına kaydedilmiştir [29].

HTML ve web sayfası URL'lerinden toplam 48 özellik çıkarılmıştır. Bu özellikler çıkarılırken kimlik avı web sayfası tespiti için yapılmış olan çalışmalar incelenmiştir. Çalışmanın veri setinin tamamı [45] adresindeki Attribute-Relation File Format (ARFF) dosyası olarak indirilebilir.

4.3.1. Veri Seti Özellikleri

Veri setine ait 48 özellik hakkında aşağıda maddeler halinde bilgi verilmiştir [29].

1) **NumDots (Nokta Sayısı):** Web sayfası URL'indeki nokta sayısını sayılmaktadır.

2) **SubdomainLevel (Alt alan seviyesi):** Web sayfası URL'indeki alt alan seviyesini sayılmaktadır.

3) **PathLevel (URL yol derinliği):** Web sayfası URL'indeki yolun derinliğini sayılmaktadır.

4) **UrlLength (URL uzunluğu) :** Saldırganlar tarafından oltalama saldırısı yapılırken kullanıcıların şüpheli kısımları fark edememeleri için uzun URL kullanabilmektedir. Bu sayede uzun URL içine şüpheli kısımlar gizlenebilmektedir. Bu özellik ile URL'lerin karakter sayıları sayılmaktadır.

5) **NumDash (Etki alanındaki "-" simgesi) :** Genellikle yasal sitelerin çoğunda tire simgesi çok az kullanılmaktadır. Saldırganlar tarafından URL içine tire işareti kullanılarak ön ek veya son ek eklenip kullanıcıları yanıltarak web sayfasının yasal olduğu yanılgısına neden olmaktadır. Bu özellikte kaç adet tire sayısı olduğu belirtilmektedir.

6) **NumDashInHostname (Anabilgisayar adı bölümündeki tire sayısı):** Web sayfası URL'inin ana bilgisayar adı bölümündeki "-" işareti sayısı belirtilmektedir.

7) **AtSymbol (@ Simgesi kullanımı):** "@" işaretinin kullanılıp kullanılmadığı kontrol edilmektedir.

8) **TildeSymbol (Tilde Sembolü):** Web sayfası URL'inde "~" işaretinin olup olmadığını kontrol edilmektedir.

9) **NumUnderscore (Alt tire sayısı):** Web sayfası URL'indeki "_" işareti sayısı sayılmaktadır.

10) **NumPercent (Yüzde işareti sayısı):** Web sayfası URL'indeki "%" işareti sayısı sayılmaktadır.

11) **NumQueryComponents (Sorgu bölümleri sayısı):** Web sayfası URL'indeki "%" sorgu bölümlerinin sayısı sayılmaktadır.

12) **NumAmpersand (Ve işareti sayısı):** Web sayfası URL'indeki "&" işareti sayısı sayılmaktadır.

13) **NumHash (Sharp işareti sayısı):** Web sayfası URL'indeki "#" işareti sayısı sayılmaktadır.

14) **NumNumericChars (Sayısal karakter sayısı):** Web sayfası URL'indeki sayısal karakterlerin sayısı sayılmaktadır.

15) **NoHttps (HTTPS kullanma):** Web sayfası URL'inde HTTPS olup olmadığı kontrol edilir. HTTPS sunucular arasında güvenli bir bağlantı kurmayı hedefleyen protokoldür. Protokol açılımı "Secure Hypertext Transfer Protocol" olup, türkçe anlam olarak "Güvenli hiper metin protokolü" şeklinde ifade edilmektedir. Bir web sitesinin meşru olmasında HTTPS protokolünün varlığına önem verilmektedir.

16) **RandomString (Rastgele dize kontrolü):** Web sayfası URL'inde rastgele dizeler olup olmadığını kontrol edilmektedir.

17) **IpAddress (IP adresi kullanma):** IP adresinin, web sayfası URL'inin ana bilgisayar adı bölümünde kullanılıp kullanılmadığını kontrol edilmektedir. Bir IP adresi URL'de alan adının alternatifi olarak kullanılabilir. Bazen de IP adresi yerine hexadecimal sayı tabanına dönüştürülüp kullanılabilir. Bu gibi durumlarda bir kullanıcının kişisel bilgileri risk altında olabilir.

Örnek:

http://126.88.4.136/sayfa.html

http://0x61.0xBC.0xCD.0x52/2/visaa.brr/mainn.html

18) **DomainInSubdomains (TLD veya ccTLD'nin alt etki alanında kontrolü):** Web sayfası URL'inde alt etki alanının bir parçası olarak TLD veya ccTLD'nin kullanılıp kullanılmadığını kontrol edilmektedir.

19) **DomainInPaths (Etki alanı uzantısı):** Web sayfası URL uzantısında TLD veya ccTLD kullanılıp kullanılmadığını kontrol edilmektedir. ccTLD (country-code Top Level Domain) internet ülke alan kodu anlamına gelir ve internet adreslerinin hangi ülkeye ait olduğunu gösteren iki harflik en son uzantıdır. TLD ise alan adının en son kısmıdır. Alan adı uzantıları olarak da isimlendirilirler. URL'de noktadan sonra gelen kısımdır.

Örnek:

ccTLD	TLD
.tr Türkiye	.com
.de Almanya (Deutschland)	.net
.fr Fransa	.org
.eu Avrupa Birliği	.biz

20) **HttpsInHostname (Anabilgisayar adı bölümündeki Https'in varlığı):** HTTPS'in web sayfası URL'inin ana bilgisayar adı bölümünde gizlenip gizlenmediğini kontrol edilmektedir.

21) **HostnameLength (Anabilgisayar adı uzunluğu):** Web sitesi URL'inin ana bilgisayar adı bölümündeki toplam karakterler sayılmaktadır.

22) **PathLength (Yol Uzunluğu):** Web sayfası URL'inin yolundaki toplam karakterleri sayılmaktadır.

23) **QueryLength (Sorgu Uzunluğu):** Web sayfası URL'inin sorgu bölümündeki toplam karakterleri sayılmaktadır.

24) **DoubleSlashInPath (Slash işareti kontrolü):** Web sitesi URL'inin yolunda "//" olup olmadığını kontrol edilmektedir.

25) **NumSensitiveWords (Hassas kelimelerin sayısı):** Web sayfasındaki "güvenli", "hesap", "webscr", "giriş", "bankacılık", "onay" gibi hassas kelimelerin sayısını sayılmaktadır.

26) **EmbeddedBrandName (Gömülü marka adı):** Marka adının alt etki alanlarında ve web sayfası URL yolunda görünüp görünmediğini kontrol edilmektedir. Buradaki marka adı, web sayfasının HTML içeriğinde en sık kullanılan alan adı olarak kabul edilmektedir.

27) **PctExtHyperlinks (Dış köprü sayısı):** Web sayfasının HTML kaynak kodundaki dış köprülerin yüzdesi sayılmaktadır.

28) **PctExtResourceUrls (Dış kaynak URL sayısı):** Web sayfasının HTML kaynak kodundaki dış kaynak URL'lerin yüzdesini sayar.

29) **ExtFavicon (Favicon kullanımı):** Bir web sayfasının başlığında ve sayfa adres satırında bulunmaktadır. Genel olarak 16x16 boyutlarında bulunmaktadır. Harf, resim, simge şeklinde tasarlanmaktadır. Bu semboller web sitesinin tanınmasına yardımcı olmaktadır. Favicon, web sayfası URL etki alanı haricinde farklı bir etki alanı adından yüklenip yüklenmediği denetlenmektedir.

30) **InsecureForms (Formların güvenliği):** Form eylemi niteliğinin HTTPS protokolü olmayan bir URL içerip içermediğini kontrol edilmektedir.

31) **RelativeFormAction (Göreceli form eylemi):** Form eylemi niteliğinin göreceli bir URL içerip içermediği kontrol edilmektedir.

Örnek:

iletişim

32) **ExtFormAction (Harici form eylem niteliği):** Form eylemi niteliğinin harici etki alanı URL'ini içerip içermediği kontrol edilmektedir.

33) **AbnormalFormAction (Form eylemi özniteliği):** Form eylemi işlemlerinde gönderilen bir bilgiye karşılık bir eylemin alınması gerekir. Bu durumun kontrolü için form eylemi özniteliğinin "#", "about: blank", boş bir dize veya "javascript: true" içerip içermediğini kontrol edilir.

34) **PctNullSelfRedirectHyperlinks (Boş ve kendine yönlendirme vb. bağlantıların yüzdesi):** Boş değer, "#" gibi kendi kendine yönlendirme değeri, geçerli web sayfasının URL'i "file://E:/" gibi veya normal olmayan bir değer içeren bağlantıların yüzdesi sayılmaktadır.

35) **FrequentDomainNameMismatch (Sık kullanılan alan adı karşılaştırma):** HTML kaynak kodunda en sık kullanılan alan adının web sayfası URL alan adıyla eşleşip eşleşmediğini kontrol edilmektedir.

36) **FakeLinkInStatusBar (Durum çubuğunda sahte URL varlığının kontrolü):** Saldırganlar tarafından adres çubuğunda meşru bir URL gösterilebilmektedir. Bu durumu yapmak için JavaScript kodu kullanılabilir. HTML kaynak kodu incelenerek, durum çubuğunda sahte bir URL'in varlığını öğrenmek için onMouseOver'da JavaScript komutu içerip içermediği kontrol edilmektedir.

37) **RightClickDisabled (Sağ tıklamayı devre dışı bırakma):** Saldırganlar tarafından sayfa kaynak koduna erişimi ve kaydedilmesini engellemek adına sağ tıklama fonksiyonu devre dışı bırakılabilmektedir. HTML kaynak kodunun sağ tıklama işlevini devre dışı bırakmak için JavaScript komutu içerip içermediği kontrol edilmektedir.

38) **PopUpWindow (Açılır pencere kullanma):** Bazı web sitelerde açılır pencere kullanılmaktadır. Bu özellik meşru sitelerde belirli bir amaca hizmet etmektedir. Genellikle meşru sitelerde uyarı amacıyla, kullanıcıları bilgilendirme ve kullanıcıları karşılama amacıyla kullanılmaktadır. Meşru olmayan sitelerde ise açılır pencerelerde genellikle kullanıcıların kişisel bilgilerini göndermesini istenmesi gözlemlenmiştir. Kişisel bilgileri isteme gibi durumların hiçbiri meşru sitelerde açılır pencerelerde gözlemlenmemiştir. Bu özellikte ise HTML kaynak kodunun, açılır pencereleri başlatmak için JavaScript komutu içerip içermediğini kontrol edilmektedir.

39) **SubmitInfoToEmail (Epostaya bilgi gönderme):** Bir kullanıcı bilgileri bir sunucuya göndermek için web formu kullanılmaktadır. Saldırganlar ise oltalama yapma amacıyla kullanıcıların bilgilerini kendi epostalarına yönlendirebilmektedir. Bu özellikte HTML kaynak kodunun "mailto" işlevini içerip içermediği kontrol edilmektedir.

Örnek:

```
mail("xyz@example.com","konu","mesaj")
<a
href="mailto:abc@example.com?konu=o%20konu&body=o%20mesaj">Gonder </a>
```

40) **IframeOrFrame (Iframe veya Frame kullanımı kontrolü):** Iframe ile bir web sayfası içerisinde başka bir web sayfası çağırma işine yarayan bir HTML etiketidir. Bu HTML kodunu kullanarak hem web siteyi hemde web site sayfasını görüntüleyebiliriz. Oltalama saldırısında iframe etiketi kullanılabilir. Web sayfasında oluşturulan iframe çerçeve kalınlığı olmadan görünmez hale getirilebilir. Bu işlem içinde tarayıcıda görüntülemede frameBorder özelliği

kullanılmaktadır. Bu özellikte HTML kaynak kodunda iframe veya frame'in kullanılıp kullanılmadığını kontrol edilmektedir.

Örnek:

<iframe src=http://www.abc.com width=442 height=325

frameborder=0 scrolling=no </iframe>

41) **MissingTitle (Kayıp başlık):** HTML kaynak kodunda başlık etiketinin boş olup olmadığı kontrol edilmektedir.

42) **ImagesOnlyInForm (Yanlızca görüntü içeren form):** HTML kaynak kodundaki form kapsamında yalnızca görüntülerden oluşan hiç metin içermeyen alanlar kontrol edilmektedir.

43) **SubdomainLevelRT (Ana bilgisayar adındaki nokta sayısı):** Web sayfası URL'inin ana bilgisayar adı bölümündeki nokta sayısı sayılmaktadır. Değer üretmek için kurallar ve eşik değerleri uygulanmaktadır.

44) **UrlLengthRT (URL uzunluğuna kural ve eşik değeri belirleme):** Saldırganlar tarafından oltalama saldırısı yapılırken kullanıcıların şüpheli kısımları fark edememeleri için uzun URL kullanabilmektedir. Bu sayede uzun URL içine şüpheli kısımlar gizlenebilmektedir. Bu özellikte web sayfası URL'indeki toplam karakterler sayılmaktadır. Değer üretmek için kurallar ve eşikler değerleri uygulanır.

Örnek:

http://ornek.com/32/a/x32r3r476r72616b327k56e884c674567c5990097656d/?cmd=_root&z;update=33555c66a6c67a7cf2f8d3f9cc3267/89&klh/qa8rw09e8r;c8er87er8er98er787y87re@xyz.html

Örnek kural:

URL karakter uzunluk < 54 → yasal

URL karakter uzunluk ≥ 54 ve ≤ 75 → şüphe

Aksi takdirde → oltalama

45) **PctExtResourceUrlsRT (HTML kaynak kodundaki farklı URL'ler) :** Bir web sayfasında kullanılan farklı URL'lerin sayfayla ilgili olmama durumu olabilir. Bu tür durumlarda oltalama işlemi olasılığına neden olma ihtimali bulunmaktadır. Bu tür bir durumun etkisini ortaya çıkarabilmek adına bu özellikte web sayfası HTML kaynak kodundaki dış kaynak URL'lerinin yüzdesini sayılıp belirli kural ve eşik değerleri uygulanmaktadır.

46) **AbnormalExtFormActionR (Form eylemi kontrolü):** Form eylemi özniteliğinin yabancı bir etki alanı, "about: blank" veya boş bir dize içerip içermediğini kontrol edilmiştir. Değer üretmek için belirli kural ve eşik değerleri uygulanmıştır.

47) **ExtMetaScriptLinkRT (Etiketlerdeki bağlantılar):** Web sayfalarının kaynak kodları incelenmiştir. Web sayfasında kullanılan etiketlerin farklı farklı amaçları vardır. <meta> etiketi genellikle sayfa yapısı ve içerik hakkında bilgi sunmak amaçlı kullanılmakta. <script> HTML kodları arasına yerleştirilen küçük kodlardır. <link> etiketi diğer web kaynaklarını almak için

kullanılmaktadır. Bu etiketlerin web sayfasının aynı etki alanına bağlı olması beklenmektedir. Bu özellikte, özniteliklerdeki harici URL içeren meta, script ve link etiketlerinin yüzdesi sayılmaktadır. Değer üretmek için kurallar ve eşik değerleri uygulanmaktadır.

Örnek kural:

<script> ,<link> ve <meta> 'daki bağlantı %'si < %17 → yasal

<meta>, <script> ve <link> 'teki bağlantı %'si ≥ %17 ve ≤ %81 → şüphe

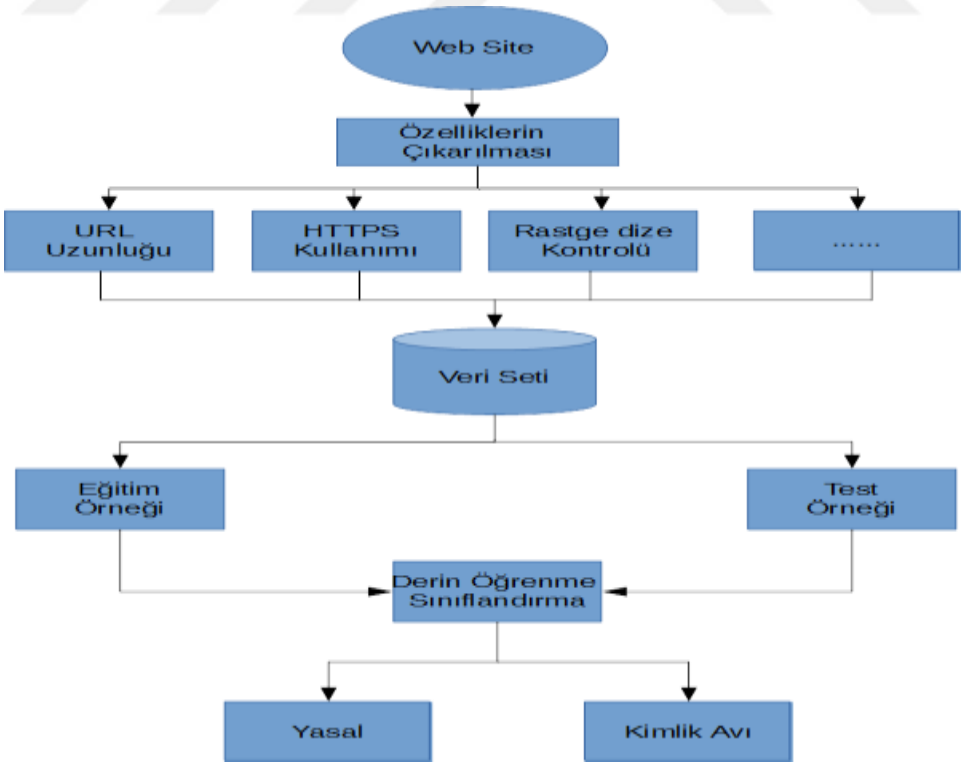
Aksi takdirde → oltalama

48) **PctExtNullSelfRedirectHyperlinksRT (Bazı köprülerin yüzdeleri):** Bu özellikte HTML kaynak kodundaki farklı alan adlarını kullanan, "#" ile başlayan veya "JavaScript::void(0)" kullanan köprülerin yüzdesini sayılmaktadır. Değer aralığı üretmek için bazı kural ve eşik aralığı uygulanır.

5. BULGULAR

5.1. Derin Öğrenme Yöntemi İle Web Tabanlı Oltalama Saldırılarının Tespiti

Teknolojinin ilerlemesi ile gün geçtikçe gelişmekte olan derin öğrenme yöntemi, bilinen verilerin çok katmanlı YSA ile eğitilerek bilinmeyen verilerin sınıflandırılmasını ve analizini yapma imkânı sağlayan bir yaklaşımdır. Bu çalışmada sınıflandırma işlemi için derin öğrenme yöntemi kullanılmıştır. 5000 meşru ve 5000 kimlik avı web site barındıran veri seti sınıflandırma için kullanılmıştır. Veri setinde web sitelere ait 48 özelliğin değerleri bulunmaktadır. Oluşturulan derin öğrenme modelinde 1 adet giriş katmanı, 3 gizli katman ve 1 çıkış katmanı vardır. Katman çıkışlarında fonksiyonu olarak ReLU aktivasyon fonksiyonu $ReLU: f(x) = max(x, 0)$ kullanılmaktadır. Çıkış katmanında ise Softmax fonksiyonu kullanılmıştır. $f(x_i) = \frac{e_i^x}{\sum_i^k e_i^x}, i = 0,1,2,3, \dots, k$ fonksiyonu [0,1] arasında çıktılar üretmektedir. Model eğitimi aşamasında, ağırlık değerlerini en düşük hale indirgemek için optimize edilmeye çalışılmıştır. Bu optimizasyon işleminde iş yükünü hesaplamak için Sparse Categorical Crossentropy kullanılmış ve elde edilen maliyeti global minimum seviyeye indirmek için Adam optimizasyon fonksiyonu kullanılmıştır. Bu çalışmada sunulan sistemin yapısı Şekil 5.1'de belirtilmiştir.



Şekil 5.1. Sistem Tasarımı

Kimlik avı web sitelerinin tespiti amacı ile yapılan bu çalışmada Python programlama dili ile birlikte derin öğrenme tekniğini uygulamak için Keras Deep Learning Library (Keras Derin Öğrenme Kütüphanesi) kullanılmıştır.

5.2. Uygulama-I

Kimlik avı saldırıların belirlenmesine yönelik oluşturulan modelde kullanılan veri setinin yüzdelik oranları Tablo 5.1'de belirtilmiş olup, model parametreleri Tablo 5.2'de belirtilmiştir.

Tablo 5.1. Uygulamada Kullanılan Veri Seti

Veri Seti	% Oranı	Örnek Adedi
Toplam	%100	10000
Eğitim	%70	7000
Test	%30	3000

Tablo 5.2. Model Parametreleri

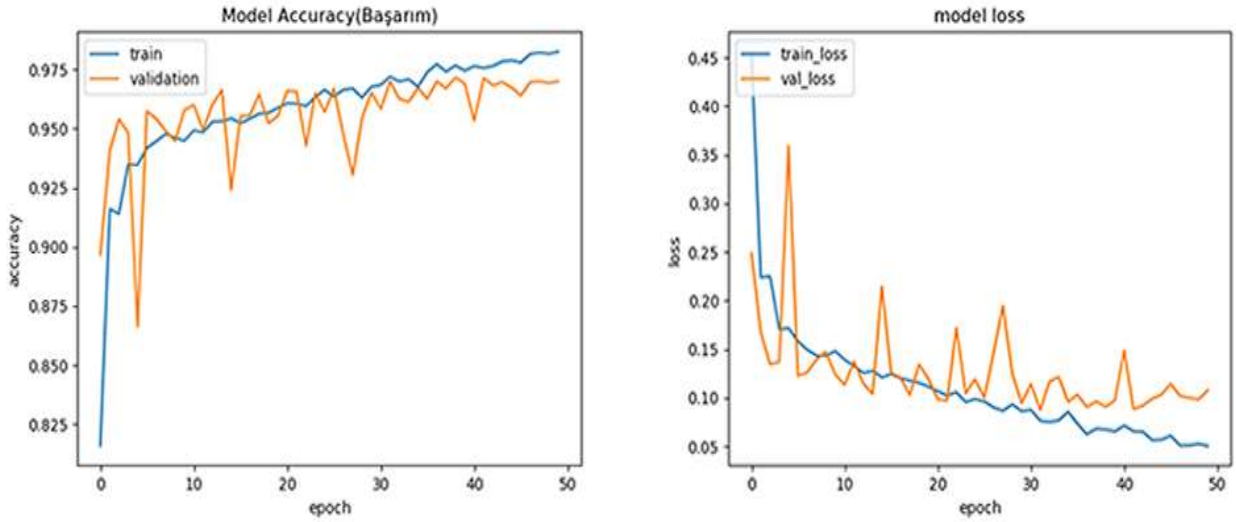
Parametreler	Değerler
Katman sayısı	5
Katmanlardaki nöron sayısı	48-256-180-128-2
Adım sayısı (Epoch)	50-100-150
Aktivasyon fonksiyonu	Relu – Softmax(Çıkış)
Optimizasyon foksiyonu	Adam
Kayıp fonksiyonu	sparse_categorical_crossentropy
Batch size	32

10000 veri ve 48 özelliğe sahip veri seti ile model eğitilip 50-100-150 adım sayısı değerlerine göre başarım değerleri elde edilmiştir. Yapılan testler sonucunda adım sayısının başarı oranını önemli ölçüde etkilediği görülmüştür. Adım sayısı modelin kaç kez eğitildiğini temsil etmektedir. Farklı adım sayıları ile elde edilen sonuçlar Tablo 5.3'te belirtilmiştir.

Tablo 5.3. Veri Seti Uygulama Sonuçları

Model	Son Başarım	En İyi Başarım	Adım Sayısı (Epoch)
Model 1	%97.00	%97.17	50
Model 2	%97.30	%97.60	100
Model 3	%97.70	%97.77	150

Tablo 5.3'te belirtilen değerlere göre model 1'e ait başarım ve hata oran grafiği Şekil 5.2'de, başarım ve hata oran çıktıları Şekil 5.3'te gösterilmiştir.

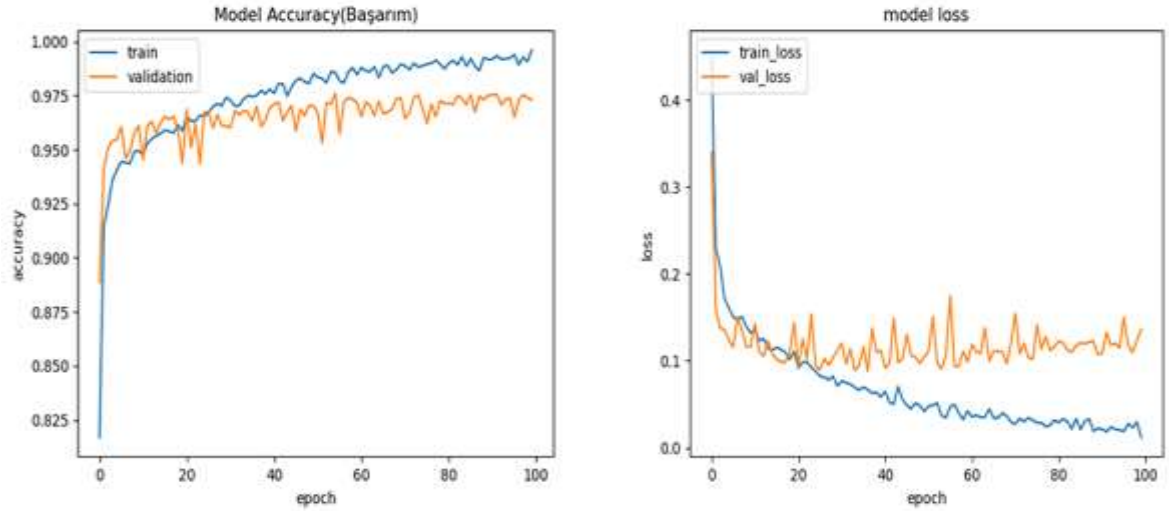


Şekil 5.2. Model 1 Başarım(Solda) ve Hata Oranları(Sağda) Grafiği

```
Epoch 40/50
 - 2s - loss: 0.0656 - acc: 0.9746 - val_loss: 0.0972 - val_acc: 0.9690
Epoch 41/50
 - 1s - loss: 0.0716 - acc: 0.9764 - val_loss: 0.1489 - val_acc: 0.9533
Epoch 42/50
 - 2s - loss: 0.0656 - acc: 0.9757 - val_loss: 0.0885 - val_acc: 0.9713
Epoch 43/50
 - 2s - loss: 0.0652 - acc: 0.9766 - val_loss: 0.0923 - val_acc: 0.9683
Epoch 44/50
 - 2s - loss: 0.0566 - acc: 0.9784 - val_loss: 0.0992 - val_acc: 0.9697
Epoch 45/50
 - 1s - loss: 0.0571 - acc: 0.9789 - val_loss: 0.1034 - val_acc: 0.9673
Epoch 46/50
 - 1s - loss: 0.0612 - acc: 0.9780 - val_loss: 0.1145 - val_acc: 0.9640
Epoch 47/50
 - 2s - loss: 0.0511 - acc: 0.9816 - val_loss: 0.1022 - val_acc: 0.9697
Epoch 48/50
 - 2s - loss: 0.0513 - acc: 0.9821 - val_loss: 0.1000 - val_acc: 0.9700
Epoch 49/50
 - 1s - loss: 0.0528 - acc: 0.9817 - val_loss: 0.0985 - val_acc: 0.9693
Epoch 50/50
 - 1s - loss: 0.0507 - acc: 0.9826 - val_loss: 0.1078 - val_acc: 0.9700
```

Şekil 5.3. Model 1 Başarım ve Hata Oranı Çıktıları

Tablo 5.3'te belirtilen değerlere göre model 2'ye ait başarım ve hata oran grafiği Şekil 5.4'te, başarım ve hata oran çıktıları Şekil 5.5'te gösterilmiştir.

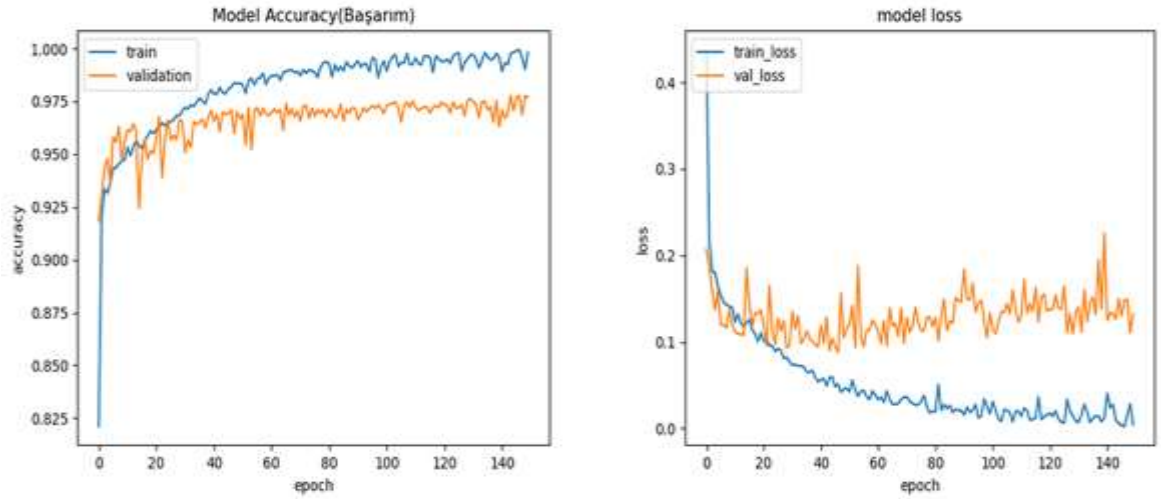


Şekil 5.4. Model 2 Başarım(Solda) ve Hata Oranları(Sağda) Grafiği

```
Epoch 90/100
 - 3s - loss: 0.0222 - acc: 0.9916 - val_loss: 0.1076 - val_acc: 0.9750
Epoch 91/100
 - 4s - loss: 0.0213 - acc: 0.9919 - val_loss: 0.1083 - val_acc: 0.9753
Epoch 92/100
 - 3s - loss: 0.0183 - acc: 0.9934 - val_loss: 0.1325 - val_acc: 0.9757
Epoch 93/100
 - 2s - loss: 0.0237 - acc: 0.9917 - val_loss: 0.1182 - val_acc: 0.9707
Epoch 94/100
 - 2s - loss: 0.0213 - acc: 0.9920 - val_loss: 0.1200 - val_acc: 0.9733
Epoch 95/100
 - 2s - loss: 0.0205 - acc: 0.9923 - val_loss: 0.1151 - val_acc: 0.9737
Epoch 96/100
 - 2s - loss: 0.0188 - acc: 0.9940 - val_loss: 0.1501 - val_acc: 0.9650
Epoch 97/100
 - 2s - loss: 0.0275 - acc: 0.9891 - val_loss: 0.1182 - val_acc: 0.9727
Epoch 98/100
 - 2s - loss: 0.0232 - acc: 0.9927 - val_loss: 0.1096 - val_acc: 0.9753
Epoch 99/100
 - 2s - loss: 0.0296 - acc: 0.9907 - val_loss: 0.1234 - val_acc: 0.9740
Epoch 100/100
 - 2s - loss: 0.0118 - acc: 0.9960 - val_loss: 0.1364 - val_acc: 0.9730
```

Şekil 5.5. Model 2 Başarım ve Hata Oranı Çıktıları

Tablo 5.3'te belirtilen değerlere göre model 3'e ait başarım ve hata oran grafiği Şekil 5.6'da, başarım ve hata oran çıktıları Şekil 5.7'de gösterilmiştir.



Şekil 5.6. Model 3 Başarım(Solda) ve Hata Oranları(Sağda) Grafiği

```
Epoch 140/150
 - 2s - loss: 0.0122 - acc: 0.9959 - val_loss: 0.2258 - val_acc: 0.9630
Epoch 141/150
 - 3s - loss: 0.0405 - acc: 0.9891 - val_loss: 0.1249 - val_acc: 0.9710
Epoch 142/150
 - 1s - loss: 0.0236 - acc: 0.9931 - val_loss: 0.1330 - val_acc: 0.9670
Epoch 143/150
 - 2s - loss: 0.0277 - acc: 0.9924 - val_loss: 0.1338 - val_acc: 0.9693
Epoch 144/150
 - 2s - loss: 0.0096 - acc: 0.9976 - val_loss: 0.1262 - val_acc: 0.9777
Epoch 145/150
 - 1s - loss: 0.0056 - acc: 0.9983 - val_loss: 0.1499 - val_acc: 0.9707
Epoch 146/150
 - 1s - loss: 0.0036 - acc: 0.9989 - val_loss: 0.1301 - val_acc: 0.9770
Epoch 147/150
 - 1s - loss: 0.0023 - acc: 0.9994 - val_loss: 0.1482 - val_acc: 0.9777
Epoch 148/150
 - 1s - loss: 0.0154 - acc: 0.9954 - val_loss: 0.1489 - val_acc: 0.9687
Epoch 149/150
 - 2s - loss: 0.0283 - acc: 0.9901 - val_loss: 0.1106 - val_acc: 0.9777
Epoch 150/150
 - 2s - loss: 0.0047 - acc: 0.9981 - val_loss: 0.1324 - val_acc: 0.9770
```

Şekil 5.7. Model 3 Başarım ve Hata Oranı Çıktıları

Tablo 5.3'te üç farklı adım sayısı ile elde edilmiş sonuçlar belirtilmiştir. 150 adım sayısı kullanılan veri setine en iyi uyumu sağlamakta ve % 97.70 oranında bir son başarım değeri üretmektedir. 150 üzeri adım sayısı için genel olarak belirtilen çıktı seviyesinde başarım değeri üretmektedir. Bu durum sistemin belirli bir uygunluk noktasına ulaştığı öğrenmenin en üst seviyede olduğu anlamına gelmektedir.

Bir model oluşturulurken veri setini eğitim ve test olarak ayırmanın amacı, modelin daha önceden görmediği veri seti üzerinde nasıl performans gösterdiğini anlamak içindir. Modelin eğitim ve test aşamasında veri dağılımdan kaynaklı bazı sapmalar olabilir. Bu sapmaları minimum seviyeye indirmek için k-katlamalı çapraz doğrulama yöntemi (k-fold cross validation) kullanılır. K-katlamalı çapraz doğrulama yöntemi bir veri kümesi üzerinde yapılan sınıflandırma işleminin sonuçlarının tutarlı olması için kullanılmaktadır. Bu yöntem ile veri seti rasgele K parçaya bölünür. K-1 parça eğitim için kullanılırken 1 parça test için kullanılır ve K defa bu işlem tekrarlanır. Her tekrarda elde edilen değerler toplanıp ortalaması alınır ve modelin performansı değerlendirilir.

Derin öğrenme yöntemiyle sınıflandırma işleminin sonuçlarının daha tutarlı olması 10-kat, 5-kat ve 3-kat değerli modeller oluşturulmuştur. Oluşturulan modelin parametreleri Tablo 5.4'te belirtilmiştir.

Tablo 5.4. Çapraz Doğrulama Model Parametreleri ve Veri Seti Yüzdelik Oranları

Parametreler	**Değerler**
Katman sayısı	5
Katmanlardaki nöron sayısı	48-256-180-128-2
Adım sayısı (Epoch)	150
Aktivasyon fonksiyonu	Relu – Softmax(Çıkış)
Optimizayon foksiyonu	Adam
Kayıp fonksiyonu	Sparse categorical crossentropy
Batch size	32
Veri seti toplam örnek	10000 (%100)
10-Kat eğitim-test örnek	9000 (%90) – 1000 (%10)
5-Kat eğitim-test örnek	8000 (%80) – 2000 (%20)
3-Kat eğitim-test örnek	6670 (%66.7) – 3330 (%33.3)

Tablo 5.4'te belirtilen model parametrelerine göre 10-kat, 5-kat ve 3-kat değerli derin öğrenme modeli oluşturulmuştur. Bu modelin her katta elde ettiği doğruluk ve ortalama doğruluk değerleri Tablo 5.5'te belirtilmiştir.

Tablo 5.5. 10-Kat, 5-Kat ve 3-Kat Çapraz Sınıflandırma Doğruluk Oranları

Kat	**10-Kat Çapraz Doğrulama Doğruluk Değerleri(%)**	**Kat**	**5-Kat Çapraz Doğrulama Doğruluk Değerleri(%)**	**Kat**	**3-Kat Çapraz Doğrulama Doğruluk Değerleri(%)**
1	% 97.50	**1**	% 97.65	**1**	% 97.27
2	% 97.90	**2**	% 97.70	**2**	% 97.06
3	% 97.20	**3**	% 97.65	**3**	% 97.21
4	% 97.60	**4**	% 98.10		
5	% 98.10	**5**	% 96.30		
6	% 97.20				
7	% 97.60				
8	% 98.80				
9	% 96.90				
10	% 96.30				
Ortalama	**% 97.51**		**% 97.48**		**% 97.18**

5.3. Uygulama-II

Chiew ve arkadaşları HEFS adını verdikleri makine öğrenmeye dayalı kimlik avı tespit sistemi önermişlerdir. Bu çalışmada özellik seçim algoritması ile birlikte 48 özellik 10 özelliğe indirgenmiştir. Tüm özelliklerin yalnızca %20,8'i kullanılarak %94,6 oranında başarı elde edilmiştir [29]. Chiew ve arkadaşlarının kullanmış olduğu 10 özellik Tablo 5.6'da belirtilmiştir.

Tablo 5.6. Özellik Seçim Algoritması İle Seçilen 10 Özellik [29]

FrequentDomainNameMismatch Sık kullanılan alan adı karşılaştırma	**PctNullSelfRedirectHyperlinks** Boş ve kendine yönlendirme vb. bağlantıların yüzdesi
PctExtNullSelfRedirectHyperlinksRT Bazı köprülerin yüzdeleri	**PctExtResourceUrlsRT** HTML kaynak kodundaki farklı URL'ler
NumNumericChars Sayısal karakter sayısı	**ExtMetaScriptLinkRT** Etiketlerdeki bağlantılar
PctExtHyperlinks Dış köprü sayısı	**SubmitInfoToEmail** Epostaya bilgi gönderme
NumDash Etki alanındaki "-" simgesi	**NumSensitiveWords** Hassas kelimelerin sayısı

Tablo 5.6'da belirtilen özellik seçim algoritmaları ile 10'a indirgenmiş olan 5000 meşru 5000 kimlik avı web sitesine ait özelliğin sınıflandırılması için Tablo 5.7'de belirtilen hiper-parametreler ile derin öğrenme modeli oluşturulmuştur.

Tablo 5.7. Model Parametreleri ve Veri Seti Yüzdelik Oranları

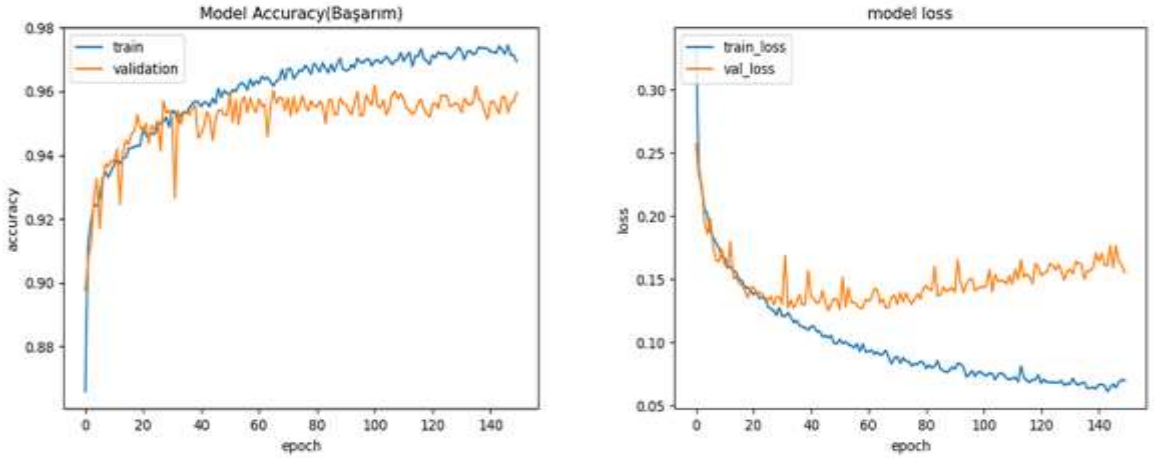
Parametreler	**Değerler**
Katman sayısı	5
Katmanlardaki nöron sayısı	10-90-60-30-2
Adım sayısı (Epoch)	150
Aktivasyon fonksiyonu	Relu – Softmax(Çıkış)
Optimizayon foksiyonu	Adam
Kayıp fonksiyonu	Sparse categorical crossentropy
Batch size	32
Veri seti toplam örnek	10000 (%100)
Eğitim-test örnek	7000 (%70) – 3000 (%30)
3-Kat eğitim-test örnek	6670 (%66.7) – 3330 (%33.3)

Modelin belirtilen hiper-parametrelerle eğitimi sonucu bir sınıflandırma işlemi yapılmıştır. Bu sınıflandırma işlemine ait son başarım ve en iyi başarım Tablo 5.8'de belirtilmiştir.

Tablo 5.8. Veri Seti Uygulama Sonuçları

Son Başarım	**En İyi Başarım**	**Adım Sayısı (Epoch)**
%95.97	%96.17	150

Tablo 5.8'de belirtilen değerlere göre model eğitim işlemine ait başarım ve hata oran grafiği Şekil 5.8'de, başarım ve hata oran çıktıları Şekil 5.9'da gösterilmiştir.



Şekil 5.8. Model Başarım(Solda) ve Hata Oranları(Sağda) Grafiği

```
Epoch 140/150
 - 1s - loss: 0.0633 - acc: 0.9734 - val_loss: 0.1707 - val_acc: 0.9513
Epoch 141/150
 - 1s - loss: 0.0668 - acc: 0.9724 - val_loss: 0.1643 - val_acc: 0.9567
Epoch 142/150
 - 1s - loss: 0.0669 - acc: 0.9724 - val_loss: 0.1704 - val_acc: 0.9570
Epoch 143/150
 - 1s - loss: 0.0662 - acc: 0.9720 - val_loss: 0.1620 - val_acc: 0.9550
Epoch 144/150
 - 1s - loss: 0.0612 - acc: 0.9743 - val_loss: 0.1606 - val_acc: 0.9530
Epoch 145/150
 - 1s - loss: 0.0645 - acc: 0.9730 - val_loss: 0.1768 - val_acc: 0.9583
Epoch 146/150
 - 1s - loss: 0.0672 - acc: 0.9717 - val_loss: 0.1591 - val_acc: 0.9583
Epoch 147/150
 - 1s - loss: 0.0644 - acc: 0.9746 - val_loss: 0.1766 - val_acc: 0.9533
Epoch 148/150
 - 1s - loss: 0.0687 - acc: 0.9714 - val_loss: 0.1641 - val_acc: 0.9563
Epoch 149/150
 - 1s - loss: 0.0706 - acc: 0.9714 - val_loss: 0.1616 - val_acc: 0.9567
Epoch 150/150
 - 1s - loss: 0.0699 - acc: 0.9694 - val_loss: 0.1553 - val_acc: 0.9597
```

Şekil 5.9. Model Başarım ve Hata Oranı Çıktıları

Modelin eğitim ve test aşamasında veri dağılımdan kaynaklı bazı sapmalar olabilmektedir. Bu sapmaları minimum seviyeye indirmek için k-katlamalı çapraz doğrulama yöntemi uygulanmıştır. Bu işlem için Tablo 5.7'de belirtilen parametreler kullanılmıştır. 3-Kat çapraz doğrulama işlemi yapılmış olup sonuçlar Tablo 5.9'da belirtilmiştir.

Tablo 5.9. 3-Kat Çapraz Sınıflandırma Doğruluk Oranları

Kat	**3-Kat Çapraz Doğrulama Doğruluk Değerleri(%)**
1	% 95.26
2	% 95.83
3	% 95.17
Ortalama	**% 95.42**

6. SONUÇLAR

Bu çalışmada kimlik avı web sitelerinin tespiti üzerine bir yöntem sunulmuştur. Önerilen yöntemde 48 özelliğe sahip 10000 adet web site bulunan veri seti kullanılmıştır. Veri setinde yıllara göre değişme durumu olan harici özelliklerin alınmaması sadece dâhili özelliklerden oluşması önemli rol oynamaktadır. Bu veri setine derin öğrenme yönteminin uygulanması ile model başarıyla eğitilmiş olup yüksek doğruluk sonuçları elde edilmiştir. Model için çeşitli parametreler uygulanıp bir dizi testler yapılmıştır. Bu testler sonucunda en yüksek başarı değerinin elde edildiği 150 adım değeri için son başarım % 97.70 ve en yüksek başarım % 97.77 olarak gözlemlenmiştir. Modelin eğitim ve test aşamasında veri dağılımdan kaynaklı bazı olası sapmaları minimum seviyeye indirmek için k-katlamalı çapraz doğrulama yöntemi kullanılmıştır. 10-kat çapraz doğrulama için en yüksek başarım % 98.80 ortalama başarım % 97.51 olarak gözlemlenmiştir. 5-kat çapraz doğrulama için en yüksek başarım % 98.10 ortalama başarım % 97.48 olarak gözlemlenmiştir. 3-kat çapraz doğrulama için en yüksek başarım % 97.27 ortalama başarım % 97.18 olarak gözlemlenmiştir. Aynı veri setini kullanarak yapılan farklı bir çalışmada birden fazla makine öğrenmesi yöntemi denenmiştir. En iyi başarımı gösteren Random Forest sınıflandırma algoritması ile % 96.17 oranında doğruluk değeri elde edildiği belirtilmiştir [29]. Bu çalışmaların doğruluk değerleri kıyaslandığında derin öğrenme yöntemi ile elde edilen sonuçların daha başarılı olduğu anlaşılmaktadır. Ayrıca aynı çalışmada özellik seçim algoritması ile birlikte 48 özellik 10 özelliğe indirgenerek daha az sayıda özellikle yüksek başarı elde edilmek istenmiştir. Tüm özelliklerin yalnızca %20,8'i kullanılarak %94,6 oranında başarı elde edilmiştir [29]. Bu çalışmada ise aynı 10 özellik kullanılarak derin öğrenme yöntemi ile bir model oluşturulup eğitilmiştir. Test sonuçlarına göre Son başarım %95.97 en iyi başarım %96.17 oranında gözlemlenmiştir. 3-kat çapraz doğrulama işlemi sonucu ortalama %95.42 oranında başarım elde edilmiştir. 10 özelliğe indirgeme sonucu elde edilen başarım değerleri kıyaslandığında derin öğrenme yöntemi ile elde edilen sonuçların daha başarılı olduğu ortaya çıkmaktadır.

ÖNERİLER

Gelecek çalışmalarda mevcut uygulamanın hayata geçirilerek anlık olarak her girilen web site incelenip veri setine eklenilebilir. Derin öğrenmede başarı, veri seti boyutu ile orantılı olduğundan mevcut verilerin arttırılmasıyla başarının daha da arttırılacağı ve tutarlı olacağı düşünülebilir.

KAYNAKLAR

[1] Rao, R. S.; Pais, A. R. (2018). Detection of phishing websites using an efficient feature-based machine learning framework, Neural Computing and Applications.

[2] Dhamija, R.; Tygar, J. D.; Hearst, M. (2006). Why phishing works. In Proceedings of the SIGCHI conference on Human Factors in computing systems, ACM, 581-590.

[3] Phishing Activity Trends Report, Anti Phishing Working Group (APWG), http://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf, 2nd Quarters 2019.

[4] Feng, F.; Zhou, Q.; Shen, Z.; Yang, X.; Han, L.; Wang, J. (2018). The application of a novel neural network in the detection of phishing websites, Journal of Ambient Intelligence and Humanized Computing.

[5] Kumaraguru, P.; Cranshaw, J.; Acquisti, A.; Cranor, L.; Hong, J.; Blair, M.A.; Pham, T. (2009). School of phish: a real-world evaluation of anti-phishing training, In Symposium on Usable Privacy and Security, 1–12.

[6] Lungu, I.; Tabusca, A. (2010). Optimizing anti-phishing solutions based on user awareness, education and the use of the latest web security solutions, Informatica Economica, 14(2), 27.

[7] Tseng, S. S.; Chen, K. Y.; Lee, T. J.; Weng, J. F. (2011). Automatic content generation for anti-phishing education game. In 2011 International Conference on Electrical and Control Engineering (pp. 6390-6394). IEEE.

[8] Cao, Y.; Han, W.; Le, Y. (2008). Anti-phishing based on automated individual white-list. In Proceedings of the 4th ACM workshop on Digital identity management (pp. 51-60). ACM.

[9] Jain, A. K.; Gupta, B. B. (2016). A novel approach to protect against phishing attacks at client side using autoupdated white-list, EURASIP Journal on Information Security.

[10] Khonji, M.; Iraqi, Y.; Jones, A. (2013). Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4), 2091-2121.

[11] Sharifi, M.; Siadati, S. H. (2008). A phishing sites blacklist generator, In 2008 IEEE/ACS international conference on computer systems and applications, 840–843.

[12] Dhamija, R.; Tygar, J. D. (2005). The battle against phishing: Dynamic security skins. In Proceedings of the 2005 symposium on Usable privacy and security (pp. 77-88). ACM.

[13] Wenyin, L.; Huang, G.; Xiaoyue, L.; Min, Z.; Deng, X. (2005). Detection of phishing webpages based on visual similarity. In Special interest tracks and posters of the 14th international conference on World Wide Web (pp. 1060-1061). ACM.

[14] Medvet, E.; Kirda, E.; Kruegel, C. (2008). Visual-similarity-based phishing detection. In Proceedings of the 4th international conference on Security and privacy in communication netowrks (p. 22). ACM.

[15] Jain, A. K.; Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches, Security and Communication Networks, 1–20.

[16] Jain, A. K.; Gupta, B. B. (2017). Two-level authentication approach to protect from phishing attacks in real time, J Ambient Intell Hum Comput, 1–14.

[17] Huh, J. H.; Kim, H. (2011). Phishing detection with popular search engines: Simple and effective. In International Symposium on Foundations and Practice of Security (pp. 194-207). Springer, Berlin, Heidelberg.

[18] Chang, E. H.; Chiew, K. L.; Tiong, W. K. (2013). Phishing detection via identification of website identity. In 2013 International Conference on IT Convergence and Security (ICITCS) (pp. 1-4). IEEE.

[19] Chiew, K. L.; Chang, E. H.; Tiong, W. K. (2015). Utilisation of website logo for phishing detection. Computers & Security, 54, 16-26.

[20] Varshney, G.; Misra, M.; Atrey, P. K. (2016). A phish detector using lightweight search features. Computers & Security, 62, 213-228.

[21] Tan, C. L.; Chiew, K. L.; Wong, K. (2016). PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. Decision Support Systems, 88, 18-27.

[22] Pan, Y.; Ding, X. (2006). Anomaly based web phishing page detection. In 2006 22nd Annual Computer Security Applications Conference (ACSAC'06) (pp. 381-392). IEEE.

[23] Miyamoto, D.; Hazeyama, H.; Kadobayashi, Y. (2008). An evaluation of machine learning-based methods for detection of phishing sites. In International Conference on Neural Information Processing (pp. 539-546). Springer, Berlin, Heidelberg.

[24] Mohammad, R. M.; Thabtah, F.; McCluskey, L. (2012). An assessment of features related to phishing websites using an automated technique. In 2012 International Conference for Internet Technology and Secured Transactions (pp. 492-497). IEEE.

[25] Mohammad, R. M.; Thabtah, F.; McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25(2), 443-458.

[26] Hadi, W.; Aburub, F.; Alhawari, S. (2016). A new fast associative classfication algorithm for detecting phishing websites, Elsevier Science Publishers B. V., 48, 729-734.

[27] Hanbay, D.; Kaytan, M. (2017). Efective classication of phishing web pages based on new rules by using extreme learning machines, Anatolian Journal of Computer Sciences, 2, 15–36.

[28] Jain, A. K.; Gupta, B. B. (2018). Towards detection of phishing websites on client side using machine learning based approach, Telecommunication Systems, 68, 687—700.

[29] Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.C. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system, Elsevier Inc, 484, 153-166.

[30] Sağıroğlu, Ş.; Alkan, M. (2018). Siber Güvenlik ve Savunma Farkındalık ve Caydırıcılık, Ankara

[31] Kaytan, M.; Hanbay, D. (2013). Kurumsal Bilgi Güvenliğine Yönelik Tehditler ve Alınması Önerilen Tedbirler, 1. Uluslararası Adli Bilişim ve Güvenlik Sempozyumu (1st International Symposium on Digital Forensics and Security, ISDFS'13), Fırat Üniversitesi, Elazığ, 20-21 Mayıs, s.267-270.

[32] ÖĞÜN, M. N.; Adem, K. (2013). Siber güvenliğin milli güvenlik açısından önemi ve alınabilecek tedbirler. Güvenlik Stratejileri Dergisi, 9(18), 145-181.

[33] Can, Ö.; Akbaş, M. (2014). Kurumsal Ağ ve Sistem Güvenliği Politikalarının Önemi ve Bir Durum Çalışması. TÜBAV Bilim Dergisi, 7(2), 16-31.

[34] Kaytan, M. (2016). Web Tabanlı Oltalama Saldırılarının Makine Öğrenmesi Yöntemleri İle Tespiti, Yüksek Lisans Tezi., T.C. İnönü Üniversitesi Fen Bilimleri Enstitüsü.

[35] Sertkaya, M.E. (2018). Derin Öğrenme Tekniklerinin Biyomedikal İmgeler Üzerine Uygulamaları, Yüksek Lisans Tezi., T.C. Fırat Üniversitesi Fen Bilimleri Enstitüsü.

[36] Makine Öğrenimi Derin Öğrenme ve Yapay Zeka Arasındaki Fark, https://proente.com/makine-ogrenimi-derin-ogrenme-ve-yapay-zeka-arasindaki-fark/, Erişim: 7 Ağustos 2019.

[37] Makine Öğrenimi Nedir, https://www.endustri40.com/makine-ogrenimi-nedir/, Erişim: 5 Mayıs 2019.

[38] Makine Öğrenimi Nedir, https://proente.com/makine-ogrenimi-nedir/, Erişim: 5 Mayıs 2019.

[39] Denetimli Öğrenme - Denetimsiz Öğrenme, https://veribilimcisi.com, Erişim: 5 Mayıs 2019.

[40] Kın, Z.B. (2019). Türk İşaret Dili Alfabesinin Derin Öğrenme Yöntemi İle Sınıflandırılması, Yüksek Lisans Tezi., Başkent Üniversitesi Fen Bilimleri Enstitüsü.

[41] Larsen, J. (1999). Introduction to Artificial Neural Networks. Section for Digital Signal Processing Department of Mathematical Modeling Technical University of Denmark, 1st edn (November 1999).

[42] Öztemel, E. (2012).Yapay Sinir Ağları, 3. Basım, Papatya Yayıncılık

[43] Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler,https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4, Erişim: 3 Eylül 2019

[44] Kayaalp, K.; Süzen, A.A. (2018). Derin Öğrenme ve Türkiye'deki Uygulamaları, IKSAD Yayınevi, ISBN 978-605-7510-53-2.

[45] Tan, C.L. Phishing Dataset for Machine Learning: Feature Evaluation, Mendeley Data, v1, https://doi.org/10.17632/h3cgnj8hft.1, Erişim: 20 Mayıs 2019.

ÖZGEÇMİŞ

Ramazan İNCİR

KİŞİSEL BİLGİLER

Doğum Yeri : Antakya

Doğum Yılı : 1991

Uyruğu : T.C.

Adres : Atatürk Mah. Müderris Osman Efendi Cad. No:66/8 KELKİT / GÜMÜŞHANE

E-posta : ramazan_incir@hotmail.com

EĞİTİM BİLGİLER

Y. Lisans : Fırat Üniversitesi, Sosyal Bilimler Enstitüsü, İş Güvenliği ve İşçi Sağlığı, 2017

Lisans : Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, 2014

Lise : Korgeneral Hulusi Sayın Lisesi, 2009

ARAŞTIRMA DENEYİMİ

- ✓ Programlama Dilleri: .NET, Java, PHP, Python, HTML, CSS, MSSQL, Arduiono

İŞ DENEYİMİ

- ✓ Gümüşhane Üniversitesi Kelkit Aydın Doğan MYO – Öğretim Görevlisi 2017-Halen
- ✓ BMS Proje Yazılım Firması – Yazılım Geliştirici, 2014-2017