
GAZİ ÜNİVERSİTESİ
Gazi
1926

ZARARLI WEB SAYFALARININ TESPİTİ VE SINIFLANDIRILMASI İÇİN YENİ BİR SİSTEM ÖNERİSİ

CANSU KADI

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2018

Cansu KADI tarafından hazırlanan "ZARARLI WEB SAYFALARININ TESPİTİ VE SINIFLANDIRILMASI İÇİN YENİ BİR SİSTEM ÖNERİSİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği (Mühendislik Fakültesi) Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Uraz YAVANOĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi)

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

Başkan: Doç. Dr. Ahmet Burak CAN

Bilgisayar Mühendisliği Anabilim Dalı, Hacettepe Üniversitesi

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

Üye: Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

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

Tez Savunma Tarihi: 18/05/2018

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

……

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

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

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

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

Cansu KADI

18/05/2018

ZARARLI WEB SAYFALARININ TESPİTİ VE SINIFLANDIRILMASI İÇİN YENİ BİR SİSTEM ÖNERİSİ

(Yüksek Lisans Tezi)

Cansu KADI

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2018

ÖZET

Gün geçtikçe değişen ve büyüyen web teknolojileri, kullanıcıların bunlara olan ilgisinin artmasını fırsat bilen saldırganlar tarafından hedef haline gelmiştir. Kullanıcılar web sayfaları ile alışveriş, bankacılık, rezervasyon, fatura ödeme gibi etkileşim gerektiren önemli işlerini yapabildikleri gibi yalnızca bilgi de edinebilmektedir. Tüm bu işlemlerde başka kimselerin eline geçmesi istenmeyen hassas bilgiler paylaşılabilmektedir. Yalnızca bu web sayfalarına girilmesi halinde bile bu bilgilere ulaşılabilmek mümkün olmaktadır. Bu saldırılardan korunmak için web sayfasının güvenliğinden emin olmak gerekmektedir. Kurum ve kuruluşlarca kullanılan siber tehdit istihbaratı ile zararlı web sayfası tespitinin önemi daha da artmakla beraber bu amaçla yapılan çalışmalar da artarak geliştirilmiştir. Bu çalışmada da zararlı web sayfalarını tespit etmek için yeni bir sistem önerilmiştir. Zararlı ve zararsız web sayfalarının sözcüksel ve popülerlik özelliklerini, HTML özelliklerini ve JavaScript özelliklerini içeren bir veri seti oluşturulmuştur. Veri setindeki özellikler, incelenen çalışmalardan seçilen ve şüpheli olabileceği değerlendirilip önerilen özelliklerden seçilmiştir. k-En Yakın Komşu (k-NN), Destek Vektör Makineleri (DVM) ve Yapay Sinir Ağları (YSA) yöntemleri ile sınıflandırma işlemleri yapılmıştır. Yapılan testlerde YSA kullanımıyla %98,71 doğruluk oranına ulaşılmıştır. Veri seti üzerinde hesaplanan en düşük yanlış pozitif oranı ise DVM sınıflandırıcısı kullanılan testlerde %0,006 olarak ölçülmüştür. Elde edilen bu oranlar incelenen çalışmalardan daha yüksek olup, seçilen özellik sayısı da daha azdır. Bu özelliği ile işlem karmaşıklığı azaltılmıştır.

Bilim Kodu : 92403

Anahtar Kelimeler : Zararlı zararsız web sayfaları, dinamik ve statik yaklaşımlar, URL sözcüksel özellikleri, HTML özellikleri, JavaScript özellikleri

Sayfa Adedi : 80

Danışman : Dr. Öğr. Üyesi Uraz YAVANOĞLU

A NEW METHOD FOR DETECTION AND CLASSIFICATION OF MALICIOUS WEB PAGES

(M. Sc. Thesis)

Cansu KADI

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2018

ABSTRACT

Web technologies which are changing and increasing day by day become a target by attackers who take advantage of web users growing attention. Users can do not only important things that require interaction such as shopping, banking, reservation, paying bills but also get information. In all these transactions sensitive/critical information that is not intended to be passed on to other people can be shared. It is possible to access this information even if only these web pages are visited. To ensure the security of the web page is needed to prevent these attacks. The importance of identifying malicious web pages with cyber threat intelligence used by institutions and organizations has increased. The works carried out for this purpose has also been improved. In this study, a new system for detecting malicious web pages have been proposed. The systems used for this purpose and the work done have been examined and a detection approach based on machine learning has been developed. A dataset containing lexical and popular features, HTML features and JavaScript features of harmful and harmless web pages has been created. The features in the data set are selected from the studies examined and from the features that are considered to be suspicious.The classification step has been done with k-Nearest Neighbor, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) methods. With the use of ANN, accuracy rate of 98.71% has been achieved in the tests. The lowest false positive rate calculated on the dataset was measured as %0,006 in tests using SVM classifier. These rates are higher than the examined studies and the number of selected features is lower.

Science Code : 92403

Key Words : Malicious and benign web pages, static and dynamic approaches, URL lexical features, HTML features, JavaScript features

Page Number : 80

Supervisor : Assist. Prof. Dr. Uraz YAVANOĞLU

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren, kıymetli tecrübelerinden faydalandığım danışmanım Dr. Öğr. Üyesi Uraz YAVANOĞLU'na, tecrübeleriyle bana yol gösteren Prof. Dr. Şeref SAĞIROĞLU'na, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan annem Kezban KADI, babam Abdullah KADI ve kardeşim Ahmet KADI'ya ve bu süreçte bana destek olan arkadaşlarıma teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
CDN	İçerik Dağıtım Ağı **(**Content Delivery Network**)**
CSFU	Cross Side File Uploading
ÇKA	Çok Katmanlı Algılayıcılar
DN	Doğru Negatif Oranı
DNS	Alan Adı Sistemi **(**Domain Name Server**)**
DP	Doğru Pozitif Oranı
DVM	Destek Vektör Makineleri (Support Vector Machines)
HTML	Hypertext Markup Language
KNN	k-En Yakın Komşu (k- Nearest Neighbor)
LM	Levenberg-Marquardt
MR	Mantıksal Regresyon
URL	Uniform Resource Locator
XSS	Siteler Arası Betik Çalıştırma **(**Cross Site Scripting**)**
YN	Yanlış Negatif Oranı
YP	Yanlış Pozitif Oranı
YSA	Yapay Sinir Ağları

1. GİRİŞ

Son yıllarda, hızla gelişen ve büyüyen web teknolojileri, kullanıcıların bunlara olan ilgisinin artmasını fırsat bilen saldırganlar tarafından hedef haline gelmiştir. Yaklaşık 3,5 milyar internet kullanıcısının var olduğu düşünüldüğünde akılcı bir hedef olduğu düşünülebilir [1].

2016 yılı için bakıldığında dünyada yaklaşık 1 milyar web sitesi bulunmaktadır [2]. Google SafeBrowsing'e göre bu web sitelerinin yaklaşık %7'si oltalama ve zararlı içerik bulunduran web sayfalarıdır [3]. Web sayfası kullanıcıları web sayfalarını kullanarak, alışveriş, bankacılık, rezervasyon, fatura ödeme, bilgi edinme gibi etkileşim gerektiren önemli işlerini yapabildikleri gibi hiçbir işlem yapmadan web sayfalarında yalnızca gezinmektedirler. Tüm bu işlemlerde başka kimselerin eline geçmesini istemeyecekleri kredi kartı bilgileri, adres bilgileri, finansal bilgiler, kişisel bilgiler gibi hassas verileri paylaşabilirler. Herhangi bir paylaşım olmadan da yalnızca bu web sayfalarına girilmesi halinde bile bilgisayarda yer alan bu gibi hassas bilgilere ulaşabilmek ve zafiyetlerden faydalanarak sistemi ele geçirmek, bozmak mümkündür. Bunu engellemek için girilen veya girilecek web sayfasının güvenliğinden emin olmak gereklidir.

Bir web sayfasının zararlı içerik bulundurup bulundurmadığının tespiti her daim güncel bir konudur ve her yıl bu konuda yayınlanan makaleler bulunmaktadır. Güncel tutulmasındaki amaç, geçerli kara listelerin oluşturulması, anti virüs yazılımlarının veri tabanlarının güncel tutulması, web sayfalarında güvenli bir şekilde dolaşmanın sağlanması ve her gün yenilenen tehditlere karşı hazırlıklı olmaktır. Bu konuyla ilgili hâlihazırda hem akademik hem de ticari çalışmalar yapılmaktadır. Akademik çalışmalar, ticari çalışmaları destekleyici olabilecekleri gibi teorik yaklaşımlar da sunabilmektedir.

Kara liste, kötü amaçlı web sayfası URL'lerinin, IP adreslerinin veya anahtar kelime bilgilerinin listesidir [4]. Zararlı web sayfalarının yer aldığı kara listeler herkese açık olarak yayınlanabildiği gibi herkes tarafından bilinmeyen listeler kara liste üreten firmalarca yüksek fiyatlarla satılabilmektedir. Bu listeler oluşturulurken farklı metotlar kullanılmaktadır. Elle analiz edilerek oluşturma yöntemi gibi insan gücüne dayalı yöntemler tercih edilebildiği gibi makine öğrenmesi gibi hızlı ve otomatik tekniklerle

listeler oluşturmak tercih edilmektedir. Bu şekilde oluşturulan listelerde yanlış pozitif oranının düşük olması büyük önem taşımaktadır.

Bu çalışmalar ile siber tehditleri önceden bilmek ve önlem almak mümkün olmaktadır. Bu kavram günümüzde daha çok siber tehdit istihbaratı olarak bilinmektedir. En basit haliyle siber tehdit istihbaratı mevcut veri noktalarına göre kuruluşlara yönelik tehditleri anlama sürecidir [5]. Siber tehdit istihbaratı ile kurum ve kuruluşlar, siber tehditleri önceden sezip saldırganların nasıl hareket edebileceklerini ve bunların nasıl engellenebileceğine veya hafifletebileceğine ilişkin tedbirler almaktadır. Bu yönüyle mahiyeti bilinmeyen web sayfalarının sınıflandırılması siber tehdit istihbaratı üretilmesi aşamasında etkili ve önemli bir çalışmadır.

Bu çalışmada, zararlı web sayfalarını tespit etmek amaçlanmıştır. Bu amaçla, zararlı ve zararsız web sayfalarının yer aldığı, en az özellikle (incelenen çalışmalardan daha az) oluşturulan veri seti üzerinde K-En Yakın Komşu (k-Nearest Neighbour), Destek Vektör Makineleri (Support Vector Machines) algoritmaları ile Yapay Sinir Ağları (Artificial Neural Networks) kullanılarak web sayfalarını yüksek doğruluk oranı ve düşük yanlış pozitif oranı ile zararlı ve zararsız olarak sınıflandıran bir sistem tasarlanmıştır.

İkinci bölümde, zararlı web sayfası tespit yöntemleri ile ilgili yapılan çalışmalar incelenmiş olup, zararlı web sayfalarının neler içerebileceği ve nasıl sonuçlar doğurabileceği anlatılmıştır. Üçüncü bölümde, önerilen çalışma kapsamında kullanılan makine öğrenmesi teknikleri ve yapay sinir ağlarıyla ile ilgili temel bilgiler verilmiştir. Dördüncü bölümde, önerilen sistemin genel yapısı ve bu yapıyı oluşturan modüller yer almaktadır. Veri setinin oluşturulması, yapılan testler ve elde edilen sonuçlar bu bölümde sunulmuştur.

2. ZARARLI WEB SAYFASI TESPİT SİSTEMLERİ

2.1. Literatür Çalışması

Bu bölümde zararlı web sayfalarının tespiti ile ilgili literatürde yapılan çalışmalar incelerek bir yaklaşım oluşturmak amaçlanmıştır. Yapılan çalışmalardaki eksiklikler ve zorluklar belirlenerek önerilen yaklaşımda bu sorunlara nasıl çözümler üretilebileceğine yönelik bakış açısı geliştirme kapsamında faydalı olacağı düşünülmüştür. Ayrıca dünyada yapılan çalışmaların düzeyleri ve kapsamları hakkında fikir sahibi olabilmek açısından çalışmanın ilk ve en önemli adımı olarak kabul edilmektedir.

Kazemian ve arkadaşları [6], zararlı web sayfalarının tespiti için daha önceden belirlenmiş kara listede var olup olmadığını belirleyen geleneksel yöntemler yerine K-En Yakın Komşu, Destek Vektör Makineleri, Naive Bayes Sınıflandırıcı gibi gözetimli, K-Means ve İlişki Yayılımı gibi gözetimsiz makine öğrenmesi tekniklerini kullanmıştır. Zararlı web sayfalarından korunmanın en yaygın yolu, bilinen zararlı URL'lerin ve IP adreslerinin manuel olarak raporlanması, bal küpleri ve özel analiz teknikleri kullanan güvenilirliği onaylanmış kuruluşların raporları, kullanıcıların görüşleri ve gönüllüler aracılığıyla toplanan kara listelere dayanır. Pratikte, ilk defa karşılaşılan web sayfaları zararlı olsa bile kara listeler güncel olmadığında tespit edilememektedir. Ayrıca bazı web siteleri gizleme sistemleri kullanarak veya zararlı web sitelerinin bulunduğu sunucuları sürekli değiştirerek kara listelerde yer almamayı başarabilmektedir. Web sayfalarının sayısının artması ve içeriklerinin sürekli değişmesi ölçeklenebilirlik problemlerine neden olurken, ağ gezginlerinin iç ağlardaki web sayfalarına ulaşabilmek için yetki gerektiren işlemlere ihtiyaç duyması burada yer alan zararlı web sayfalarının tespitini zorlaştırmaktadır. Bahsedilen problemlerden yola çıkan yazarlar, makine öğrenmesi teknikleri ile zararlı web sayfası tespiti yapabilmek amacıyla öncelikle, Alexa'dan zararsız web sayfaları, PhishTank'tan zararlı web sayfaların toplayarak bir veri seti oluşturmuştur. Her iki sınıf için web sayfaları iki çeşit gösterime sahiptir. Bir tanesi sadece HTML kodları içerirken diğeri sadece İngilizce karakterler içermektedir. Özelliklerin çıkarılmasında web sayfalarının anlamsal özellikler, URL özellikleri, sayfa bağlantıları ve görsel özellikler olmak üzere dört özelliğinden faydalanılmıştır. Anlamsal özellikler, bir web sayfasının çok yüksek boyutlu bir uzayda vektörel olarak gösteriminden elde edilmektedir. Zararlı web

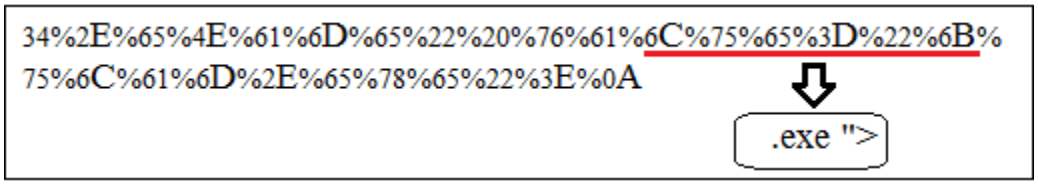
sayfalarının çoğunun URL'lerinde şüpheli karakterler yer alması URL özelliklerinin kullanılmasında önemli rol oynamaktadır. Ayrıca, bazı URL'lerin yanlış hecelenmesi, sözcüklerin yanlış yazılması, yanlış harfler kullanılması gibi metinsel özellikler de şüpheli olarak kabul edilmektedir. Zararlı web sayfaları kendileri gibi zararlı web sayfalarına bağlantılar içerebilmektedir. Bu bağlantılar da çıkarılarak modellemede kullanılmıştır. Görsel özellikler, web sayfasının PhantomJS yazılımı kullanılarak .png formatında görüntüsünün alınmasıyla elde edilmektedir. Bu görüntüleri sınıflandırma modelinin anlayabileceği formata dönüştürmek için Hızlandırılmış Kaba Özellikler yöntemi kullanılmıştır. Görsel özelliklerdeki temel yaklaşım, zararlı web sayfalarının daha az girdili ve daha basit görünüme sahip olmasıdır. Buna karşılık güvenli web sayfaları daha iyi bir tasarıma sahiptir. Yazarlar, geliştirdikleri Chrome tarayıcı eklentisini kullanarak elde ettikleri özellikleri, sınıflandırıcı ile kullanıp hızlı tahmin zamanı ve yüksek doğruluk oranı elde etmiştir. Bahse konu eklenti, sınıflandırıcının karar vermesinde gözetimli modelleri, web sayfalarının görüntülerinin demetlerini elde etmek için de gözetimsiz modelleri kullanır. Görsel özelliklerin çıkarılması, bilgisayar kaynaklarını daha fazla tüketmekte ve hesaplama zamanı uzun sürmektedir. Sonuç olarak, tüm gözetimli öğrenme teknikleri %89'un üzerinde doğruluk oranına ulaşmıştır. Gözetimsiz tekniklerde sınıflandırıcı zararlı ve zararsız olmak üzere iki sınıfın varlığından haberdar olmamasına rağmen bu kadar yüksek olmasa da yakın doğruluk oranı elde edilmiştir [6].

Phakoontod ve Limthanmaphon [7], yaptıkları çalışmada zararlı web sayfalarının tespiti için statik özelliklerin sınıflandırmasını kullanmıştır. Özellikleri açık (öne çıkan) özellikler, tekrar edilen özellikler ve çok yönlü özellikler olmak üzere üç sınıfa ayırmışlardır.

- Açık Özellikler: Bu özelliklerin yer aldığı web sayfaları yüksek ihtimalle zararlıdır. Genel olarak 5 başlıkta incelenirler.

1) Çalıştırılabilir dosyalar: .exe, .ini, .dll, .tmp gibi uzantıları olan çalıştırılabilir dosyalar içeren web sayfaları,
2) Gizli çalıştırılabilir uzak kod: Zararlı kodların çalıştırılmasını sağlayan belirli bir yolu içeren web sayfaları,
3) Etiket-bağlantı eşleşmemesi: Bağlantının gösterilenden farklı bir adrese yönlendirildiği web sayfaları,

4) Kodlanmış JavaScript: Bu kodların yer aldığı sayfaya erişim sağlandığında kodu çözülen ve çalıştırılan web sayfaları,
5) Kodlanmış dosya uzantıları: Çalıştırılabilir dosya uzantılarının kodlanarak farklı bir görünüme büründüğü web sayfaları (Şekil 2.1)
6) zararlı olarak sınıflandırılır.



Şekil 2.1. Kodlanmış .exe uzantısı içeren web sayfası [7]

- Tekrarlanan Özellikler: Bir web sayfası içinde birden fazla bulunan özelliklerdir. Genel olarak 3 başlıkta incelenirler.

1) Bağlantı sayısı: Web sayfası üzerinde bulunan bağlantıların sayısıdır. 1000'den fazla gömülü bağlantının bulunduğu zararlı web sayfaları olduğu bilinmektedir.
2) *image* etiketi sayısı: Web sayfası HTML kodlarında yer alan *image* etiketinin sayısıdır.
3) *script* etiketi sayısı: Web sayfası HTML kodlarında yer alan *script* etiketinin sayısıdır.

- Çok Yönlü Özellikler: Çeşitli zararlı özelliklerin toplandığı gruptur. Genel olarak 12 ana başlıkta incelenirler.

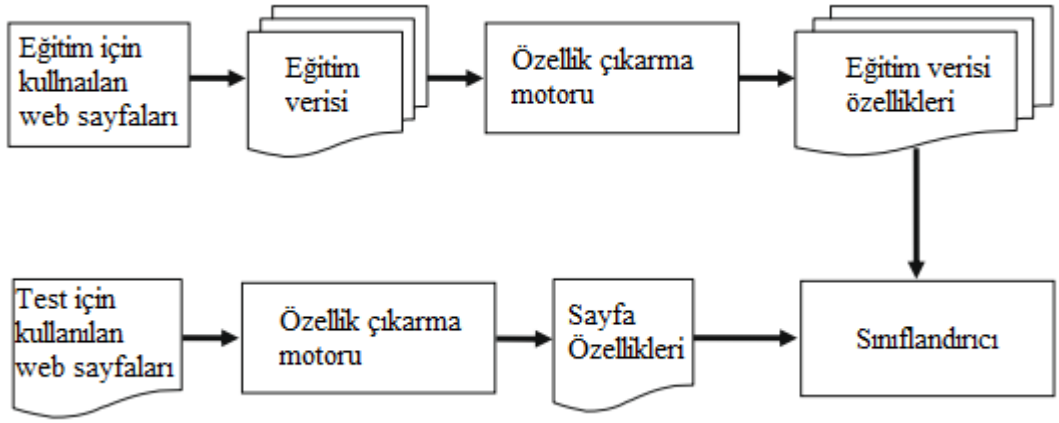
1) *"embed"* etiketi özelliği: Başka sayfaya yönlendiren "*embed*" etiketi ile zararlı içerik web sayfasına eklenir. Kullanımı aşağıdaki gibidir.
2) *<embed src="http://malicious.com/animation/fake.swf">*
3) "*object*" etiketi özelliği: *object* etiketine gömülen zararlı kod otomatik olarak çalıştırılır.
4) *"iframe"* etiketi özelliği: *iframe* etiketi ile zararlı web sayfaları açılır.
5) Pencere açılma özelliği: JavaScript'te yer alan *window.open()* fonksiyonu ile pencere açılarak zararlı kodlar çalıştırılır. Ayrıca zararlı web sayfalarına yönlendirme amacıyla da kullanılır.
6) *"document.location"* özelliği: *document.location* fonksiyonu ile zararlı web sayfasına yönlendirme yapılır.

7) *"document.cookie"* özelliği: Bu fonksiyon ile sayfaya erişildiğinde çerezler çalınır.
8) *"alert"* özelliği: Bu özellik ile uyarı ekranı açılır ve zararlı kodlar çalıştırılır.
9) Gizli kod özelliği: Bu özellik bağlantıları gizler. Bu gizli bağlantılar genel olarak zararlı sayfalara yönlendirmek için kullanılır.
10) JavaScript etiketi özelliği: Zararlı bağlantı bu etiketler arasına eklenir.
11) *"window.location"* özelliği: Zararlı sayfaya yönlendirme yapar.
12) *"Link Rel stylesheet"* özelliği: Zararlı içerik bir stil sayfasına (stylesheet) eklenir. Zararlı sayfaya yönlendirme yapılır.

Yazarlar, yukarıda sıralanan bu özellikleri kullanarak Mozilla Firefox tarayıcısı üzerinde çalışan Greasemonkey adlı bir eklenti geliştirerek istatistiksel yöntemler kullanarak zararlı web sayfalarını tespit etmiştir. İstatistiksel özelliklerin belirlenmesinde her bir grup için ağırlıklandırma yapılmıştır. Sonuç olarak, yapılan testlerde %97,9 doğruluk oranı, %2,76 yanlış pozitif ve %1,42 yanlış negatif oranı elde edilmiştir. Zararlı bir web sayfasının tespit edilme süresi 2,49 saniye olarak ölçülmüştür [7].

Hou ve arkadaşları [8], yaptıkları bir çalışmada önerdikleri yöntemle kolayca gizlenerek ve değişerek zararlı web sayfalarının tespitini zorlaştıran dinamik HTML kodlarını tespit etmiştir. Anti-virüs yazılımlarının genel olarak imza tabanlı yaklaşımları kullanması, bu durumda kolayca değişen ve gizlenen içeriklerin tespitinde yetersiz kalması probleminden yola çıkan yazarlar, Şekil 2.2'deki gibi bir sistem önermiştir [8].

Yapılan testler için StopBadware sitesinden toplanan 176 tane zararlı örnek ve 965 tane zararsız örnekten oluşan bir veri seti kullanılmıştır. Karar Ağacı, Naive Bayes, Destek Vektör Makineleri ve AdaBoost Karar Ağacı sınıflandırıcıları kullanılarak yapılan testlerde %96,14 oranla en iyi sonucun AdaBoost Karar Ağacı sınıflandırıcısına ait olduğu tespit edilmiştir [8].



Şekil 2.2. Hou ve arkadaşlarının önerdiği yaklaşım [8]

Kai ve arkadaşları [9], zararlı web sayfalarının tespiti için WSProxy adını verdikleri bir sistem geliştirmiştir. Bu sistemin kullandığı temel teknolojiler: statik analiz, dinamik JavaScript analizi ve güvenlik politikalarıdır.

- Statik Analiz: Bu kısımda web sayfasının kaynak kodları incelenir. Statik IFrame/Frame analizi ve statik JavaScript analizi aşamalarını içerir. Statik IFrame/Frame analizi ile sayfa içerisinde zararlı içeriğe sahip görünen ya da görünmeyen (yüksekliği ve genişliği 0 olarak ayarlanmış ya da konumu görünür alanın dışında olabilir.) IFrame'ler tespit edilir.
  Statik JavaScript analizi ile web sayfası üzerindeki her bir JavaScript kodu tespit edilir. Daha sonra web sayfası güvenliği için tehdit oluşturup oluşturmadıkları analiz edilir. Zararlı JavaScript kodlarının tespiti için en sık kullanılan teknik düzenli eşleştirme tekniğidir.
- Dinamik JavaScript Analizi: Statik analiz ile tespit edilemeyen gizlenmiş zararlı kodların tespiti için geliştirilmiştir. Bu gizlenme işlemi genellikle orijinal kodun kodlanmasıyla yapılır. Oluşturulan bu kod daha sonra JavaScript'in *eval()* fonksiyonu ile çözülür. Gizleme metotlarının çok çeşitli olması bu durumun otomatik olarak analiz edilmesini zorlaştırır. Gizlenmiş kodlar, çalıştırıldığında JavaScript kodunun bazı özelliklerinin ortaya çıkmasıyla ile tespit edilebilmiştir.
- Güvenlik Politikaları: Önerilen sistemin saldırılardan korunma kısmının temelini oluşturur. Bu politikalar ile gizli *iframe* saldırıları, *iframe* oluşturma, XSS, SQL Enjeksiyonu, çerez okuma ve yazma saldırıları gibi bazı saldırıların engellenebildiği görülmüştür.

Sonuç olarak, 120 potansiyel zararlı web sayfasıyla yapılan testlerde, yanlış pozitif oranının %5'in, yanlış negatif oranının da %10'un altında olduğu tespit edilmiştir [9].

Yue ve arkadaşları [10], çoğu zararlı web sayfası tespiti yöntemlerinin yalnız bir saldırı tipini tespit etmesinden yola çıkarak, web sayfalarının çeşitli özelliklerinin makine öğrenmesi teknikleri ile kullanılmasıyla etkin bir sınıflandırıcı oluşturmuştur. Bu sınıflandırıcıyla tüm popüler tehditlerin ve zararlı web sayfalarının tespit edilebildiği ifade edilmiştir.

Çalışmada 30 ayırt edici özellik üç ana başlıkta toplanmıştır.

- HTML Özellikleri: Web sayfasının HTML kaynak kodlarından elde edilen özelliklerdir. Çalışma için gizli element sayısı, *iframe* etiketi sayısı, *embed* etiketi sayısı, *object* etiketi sayısı, URL'lerin sayısı, *meta* etiketinin görünmesi, web sayfasındaki karakter sayısı gibi özellikler kullanılmıştır.
- JavaScript Özellikleri: Dinamik web sayfalarının analizinde kullanılan önemli özelliklerdir. Zararlı JavaScript kodlarının çoğu tespit mekanizmalarından kaçınmak için maskelenmiş ve gizlenmiştir. Çalışma için *eval()* fonksiyonu sayısı, şüpheli karakter dizisi sayısı, *setInterval()* fonksiyonu sayısı, *setTimeout()* fonksiyonu sayısı, uzun karakter dizisi sayısı, uzun değişken ismi sayısı, kod içindeki karakter dizilerinin ortalama uzunluğu gibi özellikler kullanılmıştır.
- URL Özellikleri: Bazı durumlarda zararlı web sayfalarının tespiti için URL içeriği incelenir. Çalışmada URL uzunluğu, URL'de IP adresinin görünmesi, URL içindeki nokta sayısı, URL'nin dosya tipi, URL'nin PageRank değeri, URL'nin WHOIS bilgilerinden elde edilen kayıt zamanı ve zaman aşımı süresi, arama motorundaki arama sonuçlarının sayısı gibi bilgiler kullanılmıştır. WHOIS bilgilerinden web sayfası son zamanlarda kayıt edildiyse ve düşük skora sahipse zararlı olarak sınıflandırılır. Bu durum yanlış pozitif oranını artırır [11].

Yazarlar, testler için dört farklı tipte toplam 2 500 URL toplamıştır. Bunların 1 000 tanesi zararsız, 500 tanesi PhishTank'tan alınan oltalama, 500 tanesi WEBSPAM-UK2007'den alınan spam, 500 tanesi de http://www.mwsl.org.cn adresinden alınan zararlı URL'lerdir. K-En Yakın Komşu ve Destek Vektör Makineleri sınıflandırıcıları kullanılarak yapılan testlerde en iyi sonuç k parametresinin 5 olduğu durumda %95,4 ile elde edilmiştir [10].

Tao ve arkadaşları [12], düşük maliyetli ve kolay uygulanabilir kod metni analizine dayalı bir sistem geliştirmiştir. Buna göre sistemin temel unsurları, domain veri tabanı, tespit sunucusu, analiz sunucusu, rapor veri tabanıdır. Domain veri tabanı, zararlı bağlantı içerip içermediği kontrol edilecek domainlerin yer aldığı veri tabanını ifade eder. Tespit sunucusu, bazı bilinen Truva atı kütüphanesine ve zararlı karakterlere dayalı web sayfası kodlarındaki zararlı bağlantıları tespit eder. Analiz sunucusu, web bağlantılarının zararlı olma olasılığını matematiksel analiz modeline göre belirler, yüksek olasılıklı web bağlantılarını rapor eder. Rapor veri tabanı, zararlı olduğu belirlenmiş veya zararlı olduğundan şüphelenilen bağlantılarını depolar, kullanıcılar buradan kolayca sorgu yapabilir [12].

Komiya ve arkadaşları [13], yaptıkları bir çalışmada SQL Enjeksiyonu ve XSS saldırılarını tespit etmeyi amaçlamıştır. k- En Yakın Komşu ve Destek Vektör Makinesi kullanılarak yapılan testlerde DVM ve Gauss çekirdeği kullanılarak gerçekleştirilen sınıflandırma işleminde SQL Enjeksiyonu için %99,16, XSS için %98,95 başarı oranı elde edilmiştir [13].

Choi ve arkadaşları [14], yaptıkları çalışmada zararlı kodlardan özellik çıkarmak için n-gram analizini, web sayfalarının SQL Enjeksiyonu ve XSS gibi zararlı kod içerip içermediğinin tespitinde Destek Vektör Makinelerini kullanmıştır. Yapılan testlerde 225 zararsız, 283 zararlı web sayfası eğitim, 117 normal, 189 zararlı web sayfası da test verisi olarak ayrılmıştır. En iyi başarı oranı %98,04 ile Destek Vektör Makineleri ile elde edilmiştir [14].

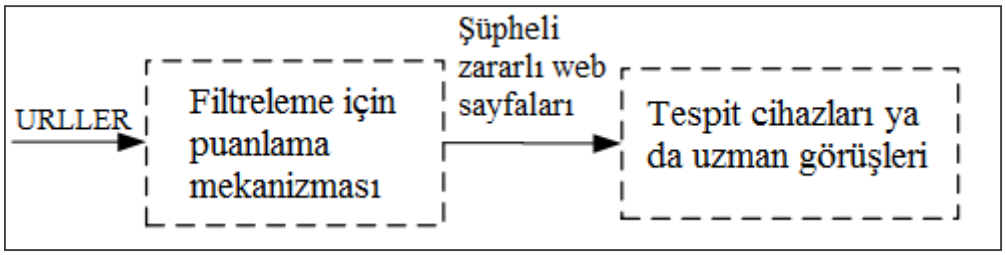
Liu ve arkadaşları [15], yaptıkları bir çalışmada makine öğrenmesi tekniklerini kullanarak şüpheli URL'leri sınıflandırmayı amaçlamışlardır. Bu amaçla bir web sayfasının içeriğine bakılmadan direkt olarak URL özellikleri kullanılarak zararsız (güvenli web sayfaları), zararlı (bilgisayar işlemlerini bozan, hassas bilgileri toplayan ya da özel sistemlere erişim sağlayan web sayfaları) ve oltalama (kullanıcı adı, şifreler, kredi kartı bilgileri gibi hassas verileri ele geçirmek amacıyla tasarlanmış güvenilirmiş gibi görünen web sayfaları) olmak üzere üç sınıfta etiketleme yapılmıştır. Yalnızca URL özelliklerinin kullanılması, çalışma zamanı gecikmesi ve kullanıcıların tarayıcı tabanlı zafiyetlere maruz kalması olasılığı durumlarını ortadan kaldırmıştır.

Özelliklerin çıkarılmasında URL'lerin sözcüksel, popülerlik ve alan tabanlı özelliklerinden faydalanılmıştır. Çoğu yasal olmayan, zararlı web sitelerinin URL adresleri zararsız olanlara göre farklılık göstermektedir. Bu özellikten yola çıkan yazarlar sözcüksel özellikleri incelerken URL'leri alan adı ve yolu olmak üzere iki parçaya ayırmıştır. Buna göre sözcüksel özellikler; alan adının uzunluğu, tüm URL'nin uzunluğu, URL içindeki nokta sayısı, IP adresi içerip içermediği, güvenlik hassasiyeti içeren sözcükler (confirm, account, banking, secure, ebayisapi, login, signin vs.) içerip içermemesi şeklinde incelenmiştir. Elde edilen sonuçlarda oltalama sitelerinin URL'lerinin genellikle daha uzun olduğu, daha çok nokta içerdiği, alan adında ve yolunda daha fazla ve daha uzun simgeler yer aldığı görülmüştür. Zararlı web sayfalarının zararsız olanlara göre daha az popüler olmasından yolan çıkan yazarlar sitenin popülerlik özelliklerini üç özellik içerisinde incelemiştir. Bunlar Google'dan toplanan bu siteye gönderilen bağlantı sayısı, Alexa'dan alınan sitenin gerçek trafik kademesi ve amazon.com'dan erişilebilen 1 000 000 iyi bir üne sahip domain içerisinde olup olmadığı bilgisidir. Zararlı web sayfalarının daha az üne sahip yer sağlayıcılarda (hosting) ya da bölgelerde kayıtlı olduğu gözlemlenmiştir. Buna göre alan tabanlı özellikler; alan adının otonom sistem numarası, IP'nin ait olduğu ülke, kayıt bilgilerinin sayısı, çözümlenmiş IP adreslerinin sayısı, alan adının geçerli bir PTR kaydı içerip içermediği bilgisi şeklinde ayrılmıştır. Benzer çalışmalarda whois kayıtlarından kayıt tarihi, güncelleme tarihi ve sona erme tarihi bilgilerinin kullanılmasının yazarların yaptığı özellik seçme işlemlerinde bir sitenin zararlı olup olmadığının belirlenmesinde etkin rol oynamadığını göstermiştir.

DMOZ, Açık Dizin Projesi'den zararsız 22 190, PhishTank'tan 5 703 oltalama ve DNS-BH projesinden 9 220 zararlı URL toplanmıştır. Toplanan URL'ler etiketlenerek eğitim-test sistemi için k-kat metodu kullanılmıştır. Etkin ve daha bilgi verici öznitelikleri seçmek için Chi-Kare testi ve Weka sınıflandırma yazılımı içindeki sanallaştırma aracı kullanılarak doğruluk oranı artırılmıştır. Özellikleri seçtikten sonra, Karar Ağacı, Destek Vektör Makineleri ve Mantıksal Regresyon (MR) ile eğitim yapılmıştır.

Her bir algoritma için her bir testte 5-kat çapraz doğrulama ile 30 test yapılmıştır. Performansların karşılaştırılmasında t-test kullanılmıştır. En yüksek doğruluk oranı ortalama %97,53'e ulaşan sınıflandırma başarısıyla karar ağacıyla elde edilmiştir. Ayrıca karar ağacının, DVM'ye göre çalışma zamanı gecikmesi oldukça düşük, MR'ye göre doğruluk oranının daha yüksek olduğu tespit edilmiştir [15].

Le ve arkadaşları [16], şüpheli web sayfalarının sayısını azaltarak zararlı web sayfalarının tespitini kolaylaştıran bir puanlama mekanizması geliştirmiştir. Bu mekanizmanın amacı web sayfalarını filtreleyerek bunları zararsız ya da potansiyel zararlı olarak sınıflandırmaktır (Şekil 2.3).



Şekil 2.3. Le ve arkadaşlarının önerdiği puanlama mekanizması [16]

Puanlama mekanizması üç adımla tanımlanmıştır. İlk aşamada, web sayfalarının zararlılık derecelerini puanlayan bir filtreleme yapılır. İkinci olarak, web sayfalarını yüklemeden istatistiksel özellikleri kullanılır. Son aşamada, puanlama algoritması tespit edilen potansiyel zararlı web sayfaları ile yanlış negatif oranı (tespit edilemeyen saldırılar) arasında karşılaştırma yapar. Puanlama sisteminin temel amacı, tespit cihazları ya da uzman görüşlerinin ihtiyaç duyduğu şüpheli sayfaların sayısını azaltmaktır [16].

Puanlama mekanizmasında kullanılacak özellikler üç ana başlıkta incelenmiştir. *frame*, *iframe*, *image*, *source* gibi HTML etiketleri yabancı içerikli özellikler grubuna girer. Özellikle *iframe* en çok bilinen dışarıdan zararlı sayfa yükleme metodudur. Yabancı içerikli özelliklerin çoğu, önceden anlaşmaya varılmış ya da kontrol edilemeyen reklam ve site hit sayaçları gibi üçüncü şahıs içeriklerden meydana gelir. Kod içerikleri, zararlı web sayfalarının tespiti için en çok kullanılan özelliklerdendir. Çoğunlukla zararlı kodları gizlemek ve hedefe ulaştırmak amacıyla kullanılırlar. Zararlı ve zararsız web sayfaları arasında kod boyutu, karakter dizisi boyutu, karakter dağılımı, değişken boyutu gibi özellikler farklılık gösterir. Üçüncü özellik, istismar edilebilen kod içeriklerini kapsar. İstismar edilecek hedefler web tarayıcılar, eklentiler ya da işletim sistemlerinde yer alan zafiyetlerdir [16].

Yazarlar, bu çalışmada bu üç gruptan toplam 52 özellik belirlemiştir. Daha sonra bilgi kazancı ölçme metodu ile tespitte daha yüksek değere sahip özellikleri ayırmıştır. Bilgi kazancı, aşağıdaki formül kullanılarak elde edilir (Eş. 2.1).

$$BK(S,a) = Entropi(S) - \sum_{v \in a} \frac{|S_v|}{|S|} * Entropi(S_v) \quad (2.1)$$

Entropi, belirsizliğin ortaya çıkma olasılığı, S, örnek dizisi, S_v , a özelliğinin v değeri ile ilişkili örneklerini ifade etmektedir. Bilgi kazancı yüksek olan özellikler seçilerek, sonuç olarak potansiyel özellik sayısı 26 ya indirgenmiştir. Bilgi kazancı tekniği hem zararlı hem zararsız web sayfaları içeren eğitim veri seti için uygulanmıştır.

Yapılan testlerde eğitim için 20 000 zararsız web sayfasının her bir özelliği için standart sapma ve ortalamaları hesaplanarak puanlama algoritması için kullanılmıştır. Test verisi için 13 422 zararsız 224 zararlı web sayfası kullanılmıştır. İlk aşamada puanlama mekanizması için ortalama, standart sapma gibi bazı istatiksel değerler hesaplanmıştır. Sonrasında yukarıda bahsedilen üç farklı özellik grubu için test veri seti içindeki her bir örnek için grup puanları hesaplanmıştır. Her bir örneğin yabancı içerik puanı, kod içerik puanı ve istismar edilebilen kod içerik puanı olmak üzere üç tip puanı bulunmaktadır. Son olarak yanlış negatif oranı ile tanımlanmış potansiyel web sayfaları arasındaki ilişkiyi tespit etmek için bir eşik değeri belirlenmiştir. Tüm bu işlemler sonuncunda şüpheli web sayfalarının sayısı %86 oranında azaltılmıştır [16].

Choi ve arkadaşları [17], zararlı web sayfalarını ve saldırı tiplerini tespit etmeyi amaçlamıştır. Bu amaçla makine öğrenmesi teknikleri kullanmışlardır. Sözcüksel özellikler, bağlantı yapıları, web sayfası içerikleri, DNS bilgileri ve ağ trafiği gibi ayırt edici özellikler kullanılmıştır. Bu özelliklerinin çoğunun yeni ve yüksek verimliliğe sahip olduğu gözlemlenmiştir. Bağlantı popülerlik özellikleri; bağlantı popülerliği, diğer web sayfalarından gelen bağlantı sayılarının hesaplanması ile elde edilir. Zararlı web sayfalarında bu oran düşüktür. Bu oranın elde edilmesinde arama motorları kullanılır. Bu çalışmada, Altavista, AllTheWeb, Google, Yahoo! ve Ask gibi popüler arama motorları kullanılmıştır [17].

Bu kapsamda bağlantı popülerliği özellikleri;

- URL'nin popülerliği,
- Alan adının popülerliği,

- Ayırt edici alan adı bağlantı oranı (hedef URL'ye bağlanan alan adlarından tekil alan adı sayısının, toplam alan adı sayısına oranı),
- Maksimum alan adı bağlantı oranı (hedef URL'ye bağlanan alan adlarından yalnız bir alan adından gelen bağlantı sayısının toplam alan adı sayısına oranı)
- Spam, oltalama ve zararlı bağlantı oranı olarak belirlenmiştir.

Web sayfası içeriğine ait özellikler;

- HTML etiketleri sayısı,
- *iframe* etiketi sayısı,
- Boyutu 0 olan *iframelerin* sayısı,
- Satır sayısı,
- Her bir şüpheli JavaScript fonksiyonları sayısı (*eval(),escape(),link(),unescape(), exec(),search()*)
- Toplam şüpheli JavaScript fonksiyonu sayısı olarak belirlenmiştir.

DNS özelikleri, URL'nin alan adıyla ilgili özelliklerdir. Çözümlenmiş IP Adresi sayısı, alan adının sunulduğu isim sunucusu sayısı, isim sunucusu ip sayısı, çözümlenmiş iplerin zararlı otonom sistem numarası oranı, isim sunucusu iplerinin zararlı otonom sistem numarası oranı olarak belirlenmiştir [17].

Ağ özellikleri, yönlendirme sayısı, içerik boyutundan elde edilen bayt sayısı, gerçekte indirilen bayt sayısı, alan adı yüklenme hızı, ortalama indirme hızı olarak belirlenmiştir [17].

Testler için gerçek internet kaynaklarından 40 000 zararsız URL, 32 000 zararlı URL toplanmıştır. Yapılan testlerde zararlı URL'lerin tespitinde %98 başarı oranı ve %93 üzerinde saldırı tiplerini belirleme başarısı elde edilmiştir [17].

Canali ve arkadaşları [18], zararlı web sayfalarını tespit edebilmek amacıyla dinamik olarak analiz edilen web sayfalarının sayısını azaltmak için Prophiler ismini verdikleri bir filtreleme sistemi geliştirmiştir. Temel olarak bir web sayfasının zararlı içerik içerip içermediğine karar verebilmek için sayfanın HTML özellikleri, JavaScript özellikleri ve

URL özellikleri gibi statik özellikleri kullanılmıştır. Web sayfalarından HTML özellikleri elde edilirken yazarlar Neko HTML Parser yazılımını kullanmıştır. Bunun sonucunda 19 tane HTML özelliği çıkarılmıştır. *iframe* etiketi sayısı, *hidden* elementi sayısı, küçük alana sahip element sayısı (*div, iframe, object* için eşik değeri 30 px belirlenmiştir.), *script* elementi sayısı, içerdiği URL sayısı, *embed* ve *object* etiketi sayısı, bilinen zararlı içerik deseni sayısı, sayfadaki karakter sayısı, kodlanmış içeriğin yüzdesi, şüpheli içerik içeren elementlerin sayısı, beklenen yerinde olmayan element sayısı (*title* etiketi içinde ya da *HTML* etiketi dışında yer alan *iframe* elementi gibi) gibi özellikler kullanılmıştır.

JavaScript özellikleri, HTML özellikleri gibi web sayfasının hem istatistiksel hem de metinsel özellikleri içerir. 25 tane JavaScript özelliği belirlenmiştir. Örnek olarak, *eval()* fonksiyonu sayısı, *setTimeout()* ve *setInterval()* fonksiyonları sayısı, uzun karakter dizileri sayısı (bu çalışma için eşik değeri 40 karakter olarak belirlenmiştir.), atanmış kelimelerin tüm kelimelere oranı (atanmış kelimeler, *var, for, while* gibi kodlamada kullanılan kelimeler), kod içindeki boşluk oranı, kullanılan karakter dizilerinin ortalama uzunluğu, zararlı amaçlı kullanılabilecek etiket ismi içeren karakter dizisi sayısı gibi özellikler verilebilir.

URL ve sunucu özellikleri, zararlı web sayfalarının genellikle güvenilir olmayan sunucularda bulunması ve WHOIS kayıtlarının olmaması ya da az olması bilgisinden yola çıkılarak oluşturulmuştur. 33 tane özellik belirlenmiştir. Şüpheli URL desenlerinin sayısı, URL içinde IP adresinin bulunması, URL içinde alt alan adı bulunması, WHOIS kayıt tarihi gibi özellikler kullanılmıştır. Veri setinden tespit modelleri çıkarmak için WEKA makine öğrenmesi platformu kullanılmıştır. Naive Bayes, rastgele orman, karar ağacı ve mantıksal yayılım sınıflandırıcıları test edilmiştir.

Büyük boyutlu testler için sistem 60 günlük bir periyotta 18 939 908 etiketlenmemiş web sayfası içeren bir veri seti ile çalıştırılmıştır. Sistem bu sayfaların %14,3 ‘ünü zararlı olarak sınıflandırmıştır. Bu büyük boyutlu veri setini %85,7 oranında küçülterek analize hazır hale getirmiştir [18].

Lee ve arkadaşları [19], Google’ın PageRank algoritmasından yola çıkarak, siber olay kaynaklarının sayısal olarak ifade edilebildiği ResourceRank algoritmasını geliştirmiştir. Bu algoritmanın kapsamı, kötü amaçlı siber olaylarda kullanılan IP adresi, kötücül kod,

alan adı ve URL bilgilerinin yanında alan adının kaydında kullanılan e-posta adresi ve kötücül kodların dinamik analizinden elde edilen bilgilerdir. Bu bilgiler çok miktarda olup ilişkilendirilerek hem ham veri hem de ilişkili veriler kaydedilir. ResourceRank değerinin hesaplanması, mevcut çalışmaların değiştirilmesi ve çevresel faktörlerden kaynaklanan farklılıkların düzenlenmesiyle başlar. ResourceRank değeri hesaplanırken veri tabanı kullanılır. Böylece sıklıkla ve kolaylıkla güncellenebilir. Bu haliyle uzun aralıklarla güncellenen Google PageRank algoritmasından daha avantajlı olduğu sonucuna ulaşılmıştır [19].

Invernizzi ve Comparetti [20], web sayfalarını inceleyerek zararlı aktivite içerenleri belirleyen bir arama motoru altyapısı kullanmıştır. Temelinde rastgele web tarama yaklaşımını kötü niyetli URL'ler için kılavuzlu bir arama ile tamamlayan bir sistem amaçlanmıştır. Başlangıç olarak bilinen zararlı web sayfalarından ortak olan karakteristik benzerlikler çıkarılmıştır. Çıkarılan veriler aynı özellikleri içeren dolayısıyla zararlı olabilecek diğer sayfaların hızlıca belirlenebilmesinde kullanılmıştır. EVILSEED adı verilen araç, geniş ölçekli veri setlerinde geçerli olup, rastgele tarama ve elle hazırlanmış kötücül sorguların döndürdüğü sonuçlar ile karşılaştırıldığında çok daha yüksek oranda kötü amaçlı web sayfası içeren bir dizi aday web sayfası getirmiştir [20].

Chiba ve arkadaşları [21], zararlı web sayfalarının tespiti için IP adreslerinin karakteristik özelliklerini kullanmıştır. Yapılan deneysel gözlemlerde IP adreslerinin DNS ve URL gibi metriklerden daha kararlı olduğu sonucuna ulaşılmıştır. URL ile alan adlarını oluşturan karakter dizeleri son derece değişken iken IP adresleri için bu değişkenlik daha azdır (IPv4 adresleri için adres alanı 4 baytlık dizilere eşlenir.). Hem zararlı hem zararsız web sayfalarının IP adresleri çözülerek elde edilen IP adreslerinden özellik vektörleri çıkarılıp makine öğrenmesi teknikleriyle (Naive Bayes, yapay sinir ağları, destek vektör makineleri) eğitim modeli oluşturulmuştur. Yapılan testlerde, yalnızca IP adreslerinden alınan özelliklerin mevcut yaklaşımların kısıtlamasını telafi etmeye olanak tanıyan farklı göstergeleri olduğu tespit edilmiştir. Sistemin bilinmeyen kötü amaçlı web sitelerini düşük hatalarla tespit edebileceği öngörülmüştür [21].

Stringhini ve arkadaşları [22], zararlı web sayfalarının tespiti için kötü niyetli bir web sayfasının belirli özelliklerine bakmak yerine büyük ve farklı web tarayıcıların bu sayfalara nasıl ulaştığını incelemiştir. Web kullanıcıların web sayfalarıyla olan etkileşimini

kaydetmek için bir dizi web tarayıcısı ve kullanıcıların nihai hedeflerine ulaşabilmek için geçtikleri yönlendirmeler kullanılmıştır. Daha sonra belirli bir web sayfasına yönlendiren farklı yönlendirme zincirleri toplanıp, son olarak elde edilen yönlendirme grafiğinin özellikleri analiz edilmiştir. Analiz edilen bu özellikler zararlı web sayfalarının tespitinde kullanılmıştır. SPIDERWEB adı verilen sistemde tarayıcı açıklıklarını kullanan web sayfalarından ziyade sosyal mühendislik yöntemlerini kullanan aldatmaca sayfalarının tespitinin mümkün olduğu görülmüştür [22].

Sha ve arkadaşları [23], GuideTracker adını verdikleri sistemde başlangıç olarak, bilinen zararlı web sitelerini ve bunların ağ geçidi loglarına kaydedilen ziyaret bağlantılarına bakarak onlara sıklıkla erişen mağdur kullanıcıları kullanmıştır. Mağdur kullanıcıların sistemin potansiyel zararlı web sayfalarına ulaşabilmesi için bir yol gösterici olabileceği düşüncesinden yola çıkılmıştır. Kullanıcılara karşılık gelen ziyaret bağlantıları ile şüpheli URL'lerin kapsamı daraltılarak çok büyük miktarda içeriğin analiz edilmesi engellenmiştir. Sistem, kara listeleri takip ederek ziyaret bağlantıları ile kurbanlar hakkında bilgi edinip, bu kurbanların ortak erişimlerinden daha fazla şüpheli URL'lerin nerede bulunacağını sisteme öğretir. Bu özelliği ile kurbanların değişim ve erişim alışkanlıklarına uyum sağlamaktadır. Yapılan testlerde başlangıç web sayfalarından üç kat daha fazla zararlı URL tespit edilebilmiştir [23].

Manek ve arkadaşları [24], web sayfasının URL tabanlı özelliklerine, sunucu bilgilerine ve web sayfasının içeriğine dayalı özelliklerini kullanarak zararlı web sayfalarını tespit etmeyi amaçlamıştır. Destek vektör makineleri, Naive Bayes ve mantıksal yayılım sınıflandırıcıları ile oltalama ve kötücül yazılım dağıtımı amaçlı oluşturulmuş web sayfaları yüksek doğruluk oranıyla tespit edilmiştir [24].

Eshete ve arkadaşları [25], zararlı web sayfalarını daha kesin olarak analiz ve tespit etmek için evrime duyarlı ve öğrenme tabanlı, genetik algoritmaları kullanan bir yaklaşım olan EINSPECT'i geliştirmiştir. EINSPECT başlangıç popülasyonu olarak URL, HTML, JavaScript ve web sayfasının itibar verilerinin ayırt edici özelliklerine dayalı olarak eğitilmiş aday modelini seçer. Daha sonra, web sayfası içeriklerinin analizini ve tespitini kapsayan özelliklerin ve öğrenme algoritmalarının en iyi etkileşimine otomatik olarak karar vermek için tespit modelini sürekli ayarlar. Genetik algoritma yönlendirmeli tespit

modeli arama ve optimizasyonu ile zararlı web sayfalarının tespitinde özellikle yanlış negatif oranında iyileştirme olduğu görülmüştür [25].

Shibahara ve arkadaşları [26], yalnızca yüksek etkileşimli bal küpleri tarafından belirlenen web sayfalarının ayırt edici sonuçlarını kullanarak web sayfalarının kötü amaçlı olup olmadığını değerlendiren bir yöntem önermiştir. Bu yöntem, saldırganın genellikle temiz bir web sayfasına, o web sayfasını ziyaret eden kullanıcıdan habersiz zararlı yazılım yükleyerek, web tarayıcısındaki ya da işletim sistemindeki açıklıkları sömürecek zararlı kod bulaştırdığı saldırılardan (drive-by download) kaynaklanan yeniden yönlendirme zincirlerinin yapısal benzerliği ile kötü niyetliliği değerlendirir. Web sayfalarındaki yeniden yönlendirmelerde düğümleri URL, kenarları yönlendirmeler olan ağaç yapıları bulunur. Yeniden yönlendirmelerin benzerliği, alt ağaçlar arasındaki uyuşma derecesine göre hesaplanır. Önerilen yöntemde sınıflandırıcı düzenli aralıklarla yüksek etkileşimli bal küpleri tarafından belirlenen ayırt edici sonuçlara göre otomatik olarak hazırlanmış eğitim verileri ile yeniden eğitilir. Bu sayede yüksek doğruluk oranı korunmaktadır [26].

Nunan ve arkadaşları [27], makine öğrenmesi teknikleriyle web sayfalarındaki JavaScript kodlarını çalıştıran web sayfalarında oltalama, çerez ve oturum bilgilerini çalma, zararlı işlemler yapma amacıyla yazılmış kod parçacıklarıyla yapılan XSS saldırılarını tespit etmeyi amaçlamıştır. Özelliklerin çıkarılmasında URL özelliklerinden ve web sayfası içeriğinden faydalanılmıştır. Sınıflandırıcı olarak Naive Bayes ve destek vektör makineleri yöntemleri ile yapılan testlerde %99 yüksek doğruluk oranı elde edilmiştir [27].

Jodavi ve arkadaşları [28], kaçak indirme saldırılarının tespitinde karşılaşılan yanlış alarmları ortadan kaldırmak için DbdHunter adını verdikleri grup tabanlı bir anomali tespiti yaklaşımı geliştirmiştir. Bu yaklaşım, birden fazla taban sınıflandırıcıdan oluşan bir grubun algılama performansının bunların herhangi birinden daha iyi olma eğiliminin gözlemlenmesine dayanır. DbDHunter, web sayfalarındaki JavaScript kodunun davranış desenlerindeki önemli değişikliklere dayanarak kötü niyetli web sayfalarını tespit eder ve böylece daha önceden görülmemiş kaçak indirme saldırılarını tespit edebilir. Ayrıca, tek sınıflı sınıflandırıcılardan bir başlangıç grubu oluşturur ve web sayfalarını zararlı veya zararsız olarak sınıflandırmak için grup içinde optimuma en yakın bir alt grubu bulmak için SwarmSnips adlı bir ikili parçacık yığını optimizasyonu algoritmasını uygular.

Böylece yanlış alarm oranı büyük oranda azaltılmış olur. Yapılan testlerde %97 doğruluk oranı ile %1,3 yanlış pozitif oranı elde edilmiştir [28].

Liang ve arkadaşları [29], başkalaşıp şekil değiştirerek geleneksel statik imza tabanlı yaklaşımlarda kolayca kaçabilen kodlanmış zararlı web sayfalarını tespit edebilmek için bilgi entropisine dayalı bir yaklaşım önermiştir. Bilgi entropisi, olasılık teorisine ve rastgele bir değişkenin değerindeki belirsizlik miktarını veya rastgele bir işlemin sonucunu nicelleştiren istatistiğe dayanan bir teoridir. Yapılan analizlerde *base64* veya *escape* ile kodlanmış zararlı web sayfalarında karakter dizilerinin [a-z], [A-Z], [0-9] karakterleri ile "%" işaretinden oluştuğu görülmüştür. Seçilen karakterler sayfanın kaynak kodunda sayılıp sıklığına göre sıralanır. Elde edilen değerler kötü amaçlı web sayfalarını tespit etmede istatistiksel olarak belirlenen eşik değerleriyle karşılaştırılır. Kodlamadan sonra bazı karakterlerin sıklığının daha fazla olması beklenmektedir. Karakter frekansı ne kadar yüksekse, web sayfası o kadar şüpheli kabul edilmektedir [29].

Dewan ve Kumaraguru [30], Facebook'ta gözlemledikleri 17 önemli haber başlıklarında (terör olayları, doğal afetleri vs.) yapılan yaklaşık 4,5 milyon paylaşımın %4'ünün zararlı URL'ler içerdiğini tespit etmiştir. Facebook'un kendi tespit yöntemleri aşan bu zararlı içerikler genellikle üçüncü parti ve web uygulamaları kaynaklı olup, meşru içeriğin yarısından çoğu mobil uygulamalar kaynaklı olduğu belirlenmiştir. Yazarlar, zararlı içeriğin gerçek zamanlı otomatik olarak tespiti için varlık profili, metin içeriği, üst bilgi (metadata) ve URL özelliklerine dayalı kapsamlı bir özellik seti önermiştir. Özellik seti birden fazla makina öğrenmesi tekniğinde kullanılarak %86,9'luk bir doğruluk oranına ulaşılmıştır. Yapılan testlerde gerçek zamanlı tespit için tarayıcı üzerine kurulan bir eklenti kullanılmıştır [30].

Gabriel ve arkadaşları [31], zararlı URL'lerin tespit eden sistemlerin gerçek zamanlı akış üzerinde çalışması gerektiğini ve her bir URL'nin işlenme hızının 300-400 milisaniye/URL'den daha az olması gerektiğini düşünmektedir. Yapılan gözlemler sonucunda, zararlı URL'lerin ömürleri kısa olmakla birlikte birkaç saat içinde ortaya çıkıp bulundukları bölgelerdeki internet sağlayıcılar tarafından kapatıldığı sonucuna ulaşılmıştır. Yazarlar önerdikleri sistemde, ağ trafiğindeki URL'leri analiz eden ve yeni kötü amaçlı içeriğe uyum sağlamak üzere algılama modellerini ayarlayabilen bir sistem sunmayı amaçlamaktadır. Doğru sınıflandırılan her URL, yeni algılama modellerinin omurgası

olarak işlev gören yeni bir veri kümesinin parçası olarak yeniden kullanılır. Sistem ayrıca, zararlı URL'lerde özelliklerin eksikliğini tanımlamak için farklı kümeleme teknikleri kullanmaktadır ve bu şekilde bu tür tehditler için algılamayı geliştirmenin bir yolunu oluşturmaktadır. Eğitim algoritması için Tek Taraflı Sınıf (OSC) algılayıcısı kullanılmıştır. Sistem gerçek zamanlı çalışacağı için doğruluk oranı düşük olsa bile yanlış pozitif oranının mümkün olduğu kadar düşük olma prensibi izlenmiştir. Bu yüzden yanlış pozitif oranı 0 olacak şekilde eğitim yapılmıştır. 40 gün (1000 saat) boyunca toplanan 2 057 091 zararsız ve 562 424 zararlı URL ile yapılan testlerde doğruluk oranı %82,9, yanlış pozitif oranı %0,7 olarak elde edilmiştir [31]

Thakur ve arkadaşları [32], RIPPER algoritması kullanarak büyük veriden zararlı URL tespit etmeyi amaçlamıştır. Kaçak indirme saldırıları, spam, oltalama gibi suç unsuru bulunduran zararlı URL'ler büyük veri içindedir. Yazarlar çalışmada kullanılmak üzere test ve eğitim veri setlerini oluşturmak için "Wiktionary_en_2012-07-21.hdt" isimli büyük veri tabanını kullanmıştır. İki eğitim ve test veri setlerinin oluşturulması için 25 adet URL özelliği çıkarılmıştır. Eğitim veri setleri sırasıyla 6 000 ve 12 000 URL içerirken test veri seti 1050 URL'den oluşmaktadır. Oluşturulan eğitim veri setleri WEKA sınıflandırma uygulaması içinde yer alan ve kural tabanlı sınıflandırma modeli oluşturan JRIP (RIPPER) algoritmasını eğitmek için kullanılmıştır. 1050 URL ile yapılan testlerde 6 000 URL ile eğitilen model için doğruluk oranı %82 iken, 12 000 URL ile eğitilen model için doğruluk oranı %83 olarak ölçülmüştür. Buna göre daha büyük bir veri seti kullanıldığında daha optimize sonuçlar elde edilebileceği sonucuna varılmıştır [32].

Kumar ve arkadaşları [33], zararlı web sayfalarının tespiti için makine öğrenmesi teknikleri ile kara liste ve beyaz listeleri de kullanan çok katmanlı filtreleme modeli kullanmıştır. Yazarlar tespit sisteminin doğruluğunu artırmak için üç farklı sınıflandırıcıdan oluşan çok katmanlı bir model önermiştir. İlk aşamada bir web sayfasının zararlı olup olmadığına karar vermek için kara listede veya beyaz listede yer alıp almadığına bakılır. İkinci katmanda web sayfasının zararlı olma veya olmama olasılığının hesaplandığı Naive Bayes filtresi kullanılır. Modelin üçüncü katman filtresi CART (Classification And Regression Trees) Karar Ağacı filtresidir. Modelin eğitilmesi ve eşik değerinin eğitilmesi olmak üzere iki adımdan oluşur. Modelin son katmanını DVM filtresi oluşturur. Test için "http://www.mwsl.org.cn/" web sitesinden indirilen zararlı web sayfaları, http://www.dir001.com/ web sayfasından toplanan zararsız web sayfaları kullanılmıştır.

Her bir sınıf için 10 000 URL'den Python dili kullanılarak özellikler çıkarılmıştır. Özellikler, alan adının toplam uzunluğu, en uzun alan adı parçasının uzunluğu, alan adındaki nokta sayısı, özel karakter içeren alan adları (#, $, @, ~, _, -), ardışık 4 sayıdan fazla sayı içeren alan adları, birincil alan adındaki anlamlı katsayılar, en üst 5 seviye alan adları (com, en, net, org, cc) olmak üzere 7 tanedir. Yapılan testlerde yalnızca DVM sınıflandırıcısında %76,8, Karar Ağacında %79,35, Naive Bayes'de %77,3, üçünün de kullandığı önerilen yaklaşımda %79,55 doğruluk oranı elde edilmiştir [33].

Desai ve arkadaşları [34], kullanıcılar ve zararlı web sayfaları arasında aracılık görevi görerek kullanıcıların bu gibi sitelerden zarar görme riskini azaltan bir Chrome eklentisi geliştirmeyi hedeflemiştir. Bu amaçla, aracı eğitmek ve her yeni içeriği belirli kategorilere ayırabilmek için makine öğrenmesi kullanılmıştır. Öncelikle "UCI-Machine Learning" sitesinden 11055 adet oltalama ve zararsız olarak etiketlenmiş veri ve her bir veri için 30 farklı özellik içeren "Oltalama Web Sayfaları" veri seti üzerinden özellik seçme yöntemleri ile özellik sayısı 22'ye indirilmiştir. Seçilen özellikler arasında URL uzunluğu, Google Index değeri gibi özellikler bulunmaktadır. Oltalama sitelerinde şüpheli kısımları gizlemek için genellikle uzun URL'ler kullanılmaktadır. Özellikle URL karakteri sayısı 52'den fazla olanlar oltalama sitesi olarak düşünülmektedir. Google Index değeri, web sayfasının Google üzerinde indekslenip indekslenmediğini belirtir. Çoğu oltalama web sayfası kısa ömürlü olduğu için Google'da indekslenmez. Özelliklerin seçilmesinden sonra KNN, DVM ve Rastgele Orman sınıflandırıcılarıyla yapılan testlerde en yüksek doğruluk oranı Rastgele Orman sınıflandırıcısında %96,11 olarak elde edilmiştir [34].

Vanhoenshoven ve arkadaşları [35], zararlı web sayfaların tespitinde kullanılan kara liste yöntemlerinin yeni ve kara listede yer almayan zararlı web sayfalarının tespitinde yetersiz kalmasından yola çıkarak makine öğrenmesi teknikleri kullanarak zararlı web sayfalarını tespit etmeyi amaçlamıştır. Bu amaçla Naive Bayes, DVM, Çok Katmanlı Algılayıcılar, Karar Ağacı, Rastgele Orman ve KNN gibi bilinen makine öğrenmesi teknikleri kullanılarak karşılaştırmalar yapılmıştır. 2,4 milyon URL ile 3,2 milyon özellikten oluşan veri seti ile yapılan testlerde çoğu sınıflandırıcı kabul edilebilir oranlarda doğruluk oranlarına ulaşsa da Rastgele Orman ve Çok Katmanlı Algılayıcı yöntemlerinde daha yüksek doğruluk oranlarına (%97,69 ve %97,28) ulaşılmıştır. Çok sayıda örnek ve özellik olması özellikler arasındaki bağımlılığın hesaplanmasını zorlaştırmaktadır. Yaklaşık 3

milyon özelliğin yalnızca 67 tanesinin gerçek değeri vardır. İşlem karmaşıklığını azaltmak için farklı özellik seçme metotları ile farklı özellik setleri kullanılmıştır. A özellik setinde ikili (0, 1) ve gerçek değerli özellikler yer almakla birlikte mutlak bir korelasyon katsayısı 0,2 olan rastgele bir değerden daha yüksek olan özellikler seçilir. B özellik setinde yalnızca ikili özellikler bulunmakta ve gerçek değerlerin olmayışının dezavantajlı olmaması için daha yüksek bir mutlak Pearson katsayısına sahip tüm özellikleri seçmek için 0,1'lik daha düşük bir eşik değeri kullanılır. C özellik seti ise gerçek değerli özelliklerden oluşturulmuştur. Bu özellik setindeki özellikler sıfırdan farklı değerlere sahip olması yönüyle URL hakkında daha açıklayıcı bilgiler içermektedir. Yapılan testlerde en yüksek doğruluk oranları A özellik seti, en düşük doğruluk oranları da B özellik seti kullanıldığında elde edilmiştir [35].

Zararlı web sayfalarının tespiti için önerilen yaklaşım ile benzer yöntemler içeren çalışmalarda kullanılan yöntemler, veriler ve elde edilen başarı oranları ile önerilen çalışmanın karşılaştırılması Çizelge 2.1'de gösterilmiştir.

2.2. Web Sayfası Tabanlı Tehditler

Günümüzde web uygulamaları ve istemciler (kullanıcılar) sürekli olarak etkileşim halindedir. Bu etkileşimde istemciler telefon numaraları, kredi kartı numaraları, ev ya da iş adresleri, finansal bilgiler, seyahat bilgileri gibi saldırganların hedefi olabilecek kişisel ve hassas bilgileri paylaşmaktadır. Saldırganlar bu amaçla var olan bir web sayfasının zafiyetlerini kullanarak ya da tamamen kötü niyetli olarak bilgi edinme amaçlı oluşturdukları web sayfaları ile hedeflerine ulaşmaya çalışır. Zararlı web sayfalarında genel olarak aşağıdaki yöntemler tercih edilir.

2.2.1. Virüsler

Bilgisayar virüsleri, kendi kopyalarını çalıştırılabilir diğer kodlara ve belgelere yerleştirilerek yayılan ve kendi kendine çoğalabilen kötücül yazılımlardır. Çoğu bilgisayar virüsünün, önemli dosyaları silmek, konakladığı sistemi çalışamaz hale getirmek gibi yıkıcı etkileri bulunmaktadır. Bir dosyanın açılması, bir e-postanın okunması, sosyal ağlar, metin mesajları veya virüs bulaşmış bir programı çalıştırması ile kullanıcılar farkında olmadan virüsleri yayabilir [36].

Web sayfalarında kullanılan PHP, JavaScript gibi web programlama dilleriyle yazılan virüsler mevcuttur. Bu yüzden bu dillerin desteği olan zararsız gibi gözüken web sayfaları aslında virüs barındırabilir. Hatta saldırganlar web sayfasının "Yorumlar" bölümüne bu şekilde zararlı kodlar içeren yorumlar bırakarak web sayfası yöneticisi farkında olmadan bu kodları web sayfalarına ekleyebilir [37].

2.2.2. Truva atları

Meşru yazılım gibi görünen kötücül yazılımlardır. Bir Truva atı faydalı bir yazılımla gelebileceği gibi kullanıcıları faydalı bir işleve sahip olduğuna ikna edip, bizzat kullanıcı tarafından çalıştırılarak da etkinleşebilirler.

Genel olarak, bulaştığı makinenin uzaktan kontrolünü sistem yöneticisinin farkına varmadan saldırgana iletirler. Kişisel bilgisayarlardaki şifreleri çalan, hedef kullanıcıyı belirli bir web sitesine veya internet kaynağına yönlendiren, başka bir kötücül yazılım veya reklam yazılımı indirip kuran, tuş basımları, ekran görüntüleri ve diğer kullanıcı faaliyetlerini toplayan ve saldırgana ileten türleri de bulunmaktadır [36].

2.2.3. Casus yazılımlar

Casus yazılım, kullanıcılara ait önemli bilgilerin ve kullanıcının yaptığı işlemlerin, kullanıcının bilgisi olmadan toplanmasının ve bu bilgilerin kötü niyetli kişilere gönderilmesini sağlayan yazılım olarak tanımlanır [36]. Casus yazılımların amacı virüsler gibi sisteme bulaştıktan sonra kendini kopyalayarak çoğalma amacına taşımaz, amacı kurban seçilen sistem üzerinde saklanarak şifreler, kredi kartı numaraları, klavye tuşlarının

Çizelge 2.1. Benzer çalışmaların karşılaştırılması

Kaynak	Makine Öğrenmesi Yöntemleri	Örnek sayısı	Öznitelik Sayısı	Faydalanılan Özellikler	Veri Kaynağı	Sınıflar	Başarı Oranı
[6]	KNN, DVM, Naive Bayes, K-Means,	100 000 web sayfası		Anlamsal, URL, sayfa bağlantıları ve görsel özellikler	Alexa, PhishTank	Zararlı, Zararsız	%89
[7]	Statik özelliklerin sınıflandırımı	100 zararlı, 100 zararsız URL	19	Öne çıkan özellikler, Tekrar edilen ve çok yönlü özellikler	Farklı kaynaklardan 1000 web sayfası	Zararlı, Zararsız	%97,9
[8]	Karar Ağacı, Naive Bayes, DVM ve AdaBoost Karar Ağacı sınıflandırıcıları	176 zararlı, 965 zararsız URL	12	JavaScript fonksiyonları, HTML özellikleri, ActiveX nesneleri gibi gelişmiş özellikler	StopBadware, Diğer	Zararlı, Zararsız	%96,14
[10]	KNN ve DVM	2 500 URL	30	HTML, JavaScript ve URL özellikleri	PhishTank, WEBSPAM-UK2007, www.mwsl.org	Zararlı, Zararsız	%95,4
[15]	Destek Vektör Makinaları	22 190 zararsız, 14 923 zararlı URL,	21	URL'lerin sözcüksel, popülerlik ve alan tabanlı özellikleri	DMOZ, PhishTank, DNS-BH	Zararlı, Zararsız, Oltalama	%97,53
[13]	KNN ve DVM	Eğitim için 695 test için 358	-	Web sayfası kodları	-		%99,16
[14]	Destek Vektör Makinaları	Eğitim için 225 normal, 283 zararlı, test için 117 normal, 189 zararlı	-	Web sayfası kodları	-	Zararlı, Zararsız,	%98,04
Önerilen Sistem	KNN, DVM, YSA	3302 URL	20	HTML, JavaScript, URL sözcüksel ve popülerlik özellikleri	DMOZ, PhishTank, openphish.com, Alexa	Zararlı, Zararsız,	%98,71

izlenmesi, tarayıcı alışkanlıklarının takip edilmesi, e-posta adreslerinin toplanması gibi gizli bilgilere erişmektir [38].

Web sayfalarında kullanıcılardan bilgi toplama amacıyla bulunan casus yazılımlar olabilmektedir. Kimi zaman sayfayı ziyaret eder etmez, kimi zaman bir bağlantıya tıklanıldığında, kimi zaman da kullanıcı şaşırtılması gibi yanlış bir şey yapılmadan da yüklenmesi mümkündür.

2.2.4. Reklam yazılımları

Kullanıcının bedava ya da paylaşımlı bir yazılımı indirmesiyle habersizce kurulurlar [39]. Reklamların, uygulamanın maliyetini azaltma, kullanıcılara düşük fiyat sunma gibi yararlı amaçları olsa da kötücül olarak nitelendirilen reklam yazılımlarının amacı, asıl uygulamanın değil de başka firmalarca sağlanan reklamları göstermektir [36]. Uygulamayla gelen bu reklamların gösterilmesi ve tıklanılmasıyla kazanç sağlanır. Genel itibariyle zararlı gibi görünmeseler de istek dışında gösterilmesi rahatsız edici olabilir, zararlı kodlarla kötücül bir karakter kazandırılabilir.

Zararlı reklam yazılımları, konum ve tarayıcı geçmişi hakkında bilgi toplayarak hedeflenmiş reklamlar sunmak için izleme araçlarını kullanır. Ayrıca yakın bir tarihte antivirüs yazılımlarını devre dışı bırakan zararlı reklam yazılımları tespit edilmiştir [38].

2.2.5. Oltalama

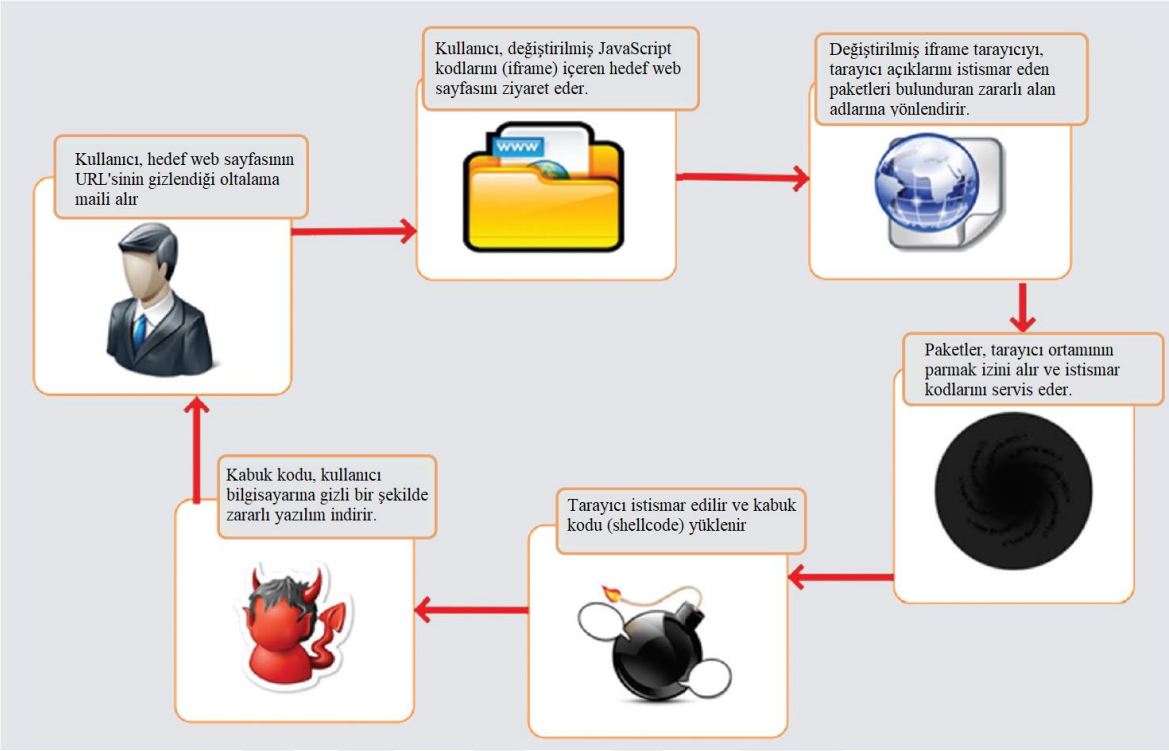
Kullanıcı adı, şifreler, kredi kartı bilgileri gibi hassas verileri ele geçirmek amacıyla tasarlanmış güvenilirmiş gibi görünen (güvenilir markalar, bankalar, e-ticaret siteleri vs gibi isimler içerir) web sayfaları tanımlanabilir. Kullanıcı bu web sayfalarını çoğunlukla güvenilir gibi görünen bir e-posta aracılığıyla erişir. Oltalama saldırılarının hedefine ulaşabilmesi için genellikle kullanıcı tarafından bir form gönderme ya da hesap bilgilerini doğrulama gibi etkileşim olması gerekmektedir. Sosyal mühendisliğin bir uygulama alanı olan bu tür sistemlerde kullanıcı, istenilen gizli bilgileri göndererek, bu bilgilerin kötü niyetli üçüncü şahısların eline geçmesine ve akabinde oluşabilecek zararlara maruz kalınmasına olanak sağlamaktadır [36].

Son zamanlarda kullanıcıları aldatmaya yönelik yeni sosyal mühendislik yöntemleri ortaya çıkmıştır. Örneğin, çevrimiçi bankacılık sitesi için içinde hesap bilgileri olan anket doldurma, rezervasyon amaçlı müşterilerin kredi kartı bilgilerini doğrulamak için sorma gibi [40]. "Anti-Phishing Working Group" tarafından 2016 yılının mart ayında yayımlanan "Anti-Phishing" etkinliği raporunda tespit edilen tekil 123 555 oltalama web sitesi olduğu, bu sayının 2015 yılının Ekim ayında 48 141 olduğu ifade edilmiştir. Bu artış, bu gibi zararlı web sayfalarından korunma yöntemleri ile ilgili araştırmalar yapılmasını sağlamıştır [41].

Bir güvenlik firması olan "Webroot" tarafından Aralık 2016'da yayımlanan bir raporda oltalama web sayfalarının %84'ünün 24 saat içinde erişime kapandığı hatta bazılarının yalnızca 15 dakika erişilebilir olduğu ifade edilmiştir. Ayrıca çoğu oltalama web sayfası meşru alan adları içinde gizlenmiştir. Bu durum güvenlik firmalarının, oltalama web sayfalarına karşı saldırı stratejisi geliştirmesini ve bu verileri inceleyip analiz etmesini zorlaştırmaktadır [42].

2.2.6. Kaçak indirme saldırıları

Bu saldırılarda her şeyden habersiz kullanıcının tarayıcısındaki veya tarayıcısındaki ActiveX yorumlayıcısı, multimedya eklentileri, Flash Player gibi elemanlardaki ya da işletim sistemindeki açıklıkları istismar edebilen virüs, casus/zararlı yazılımlar yüklenir [43]. Özellikle web sitelerinde gezinirken herhangi bir program site tarafından indirilip kurulması durumunda tarayıcılar pencere açarak bu durumu belirten bir güvenlik uyarısı göstermektedir. Güvenilir web siteleri dışında zararlı/casus yazılım içeren sitelerde bu tip güvenlik uyarılarında kullanıcıları yanıltma yoluna gidilmektedir [36]. Genel yapısıyla bir kaçak indirme saldırısı Şekil 2.4'teki gibi gerçekleşmektedir.



Şekil 2.4. Kaçak indirme saldırıları [44]

Şekil 2.4'te ifade edilen kullanıcının zararlı web sayfalarına yönlendirilmesi yalnızca *iframe* gibi HTML elemanı ile değil, HTTP yönlendirme protokolleri ve JavaScript fonksiyonları ile de gerçekleştirilebilir [43].

Kaçak indirme saldırılarında, kullanıcıları istismar paketlerini içeren web sayfasına yönlendirme ya da doğrudan bu paketlerin kurulabileceği şekilde web sayfası içeriklerini değiştirme işlemleri için XSS, SQL Enjeksiyonu, CSFU gibi web zafiyetlerini istismar etme, web sayfasının sunulduğu sunucuları istismar etme ve içerik dağıtım ağlarına (CDN) bulaşma gibi yöntemler kullanılmaktadır [44].

2.3. Metotların Analizi

Web sayfalarının ve uygulamalarının hızla artması saldırganlar için ana hedef haline gelmelerine neden olmuştur. Hiçbir şeyden haberi olmayan kullanıcılar sadece bu zararlı sayfaları ziyaret ederek kurban haline gelmektedir. Saldırganlar zararlı yazılımı yaymak yerine web sayfasına zararlı kod yükleyerek ya da gömerek web ortamını daha kolayca istismar edebilmektedirler. Google Araştırma Merkezi'ne göre, web sayfalarının %10'undan fazlasının zararlı kod içermektedir [45]. Bu yüzden web ortamını kullanan

kullanıcıları bu tehditlerden korumak için zararlı web sayfalarının tespiti oldukça önem kazanmıştır.

Zararlı web sayfalarının tespitinde farklı yaklaşımlar kullanılmaktadır. Temel olarak, dinamik ve statik yaklaşımlar olmak üzere ikiye ayrılmıştır.

Dinamik yaklaşımlar, bir web sayfasının zararlı olup olmadığını o web sayfasını ziyaret ederek karar veren istemci bal küplerini kullanır. Yüksek etkileşimli bal küpleri analizi, takip edilen bir alan içinde geleneksel tarayıcılar kullanılarak yapılır ve başarılı bir saldırının emareleri tespit edilir. Bu emareler, dosya sisteminde, kayıt defterinde ya da çalışan süreçlerde değişiklikler olabilir. Düşük etkileşimli bal küplerinde analiz, saldırı göstergesini tespit etmek için izlenen bir web sayfasının ziyaret edildiği anda çalışmaya başlayan benzetilmiş tarayıcılara dayanır [18]. Bir eklentideki zafiyet içeren bir metodun başlatılması bu göstergelere örnek verilebilir.

Hem yüksek hem düşük etkileşimli sistemler, tüm web sayfası içeriğinin gerçek zamanlı olarak tamamen çalıştırılarak hareketlerinin izlenmesiyle çalışır [4]. Bir web sayfasının tamamen çalıştırılması, sayfanın kendisi, ona bağlantı veren tüm kaynakların getirilmesini ve JavaScript kodu gibi dinamik içeriklerin yorumlanmasını içerir. Bu yaklaşımlar, yüksek tespit oranı ile düşük yanlış pozitif oranına sahiptir. Bu sistemlerin en büyük dezavantajı ise gerek gerçek gerekse benzetilmiş tarayıcılar ile web sayfası içeriklerini çalıştırma ve getirme işlemleri için zaman gerektirmesidir. Bu süre, analiz edilen web sayfasının karmaşıklığına bağlı olarak birkaç saniye ile birkaç dakika arasında değişmektedir [18].

Günümüz bal küpü sistemlerinin düşük işleme hızı ve yüksek donanım gereksinimleri statik yaklaşımların motivasyonu olmuştur. Statik yaklaşımlar, bir web sayfasının zararlı içerik bulundurup bulundurmadığını web sayfasının metinsel içerikleri, HTML ve JavaScript kodu özellikleri, ilişkili URL'nin karakteristik özellikleri gibi statik özelliklerini kullanarak tespit eder [18]. Genelde, statik yaklaşımlar, büyük ölçekli verileri tespit etmek için daha uygundur, çünkü statik yaklaşımlar dinamik olanlardan daha kısa sürede çok daha fazla web sayfası test edebilir ve bu haliyle doğruluk oranı dinamik yaklaşımlardan daha düşüktür [4].

Çeşitli sistemler zararlı web sayfalarını tespit etmek için JavaScript kodlarının statik analizi yapmaya odaklanmıştır. Kodlardan elde edilen ve yaygın olarak kullanılan özellikler; yönlendirmeler (*location.href* özelliğinin atanması gibi), gizleme için kullanılan fonksiyonların varlığı, *eval()* fonksiyonun çağrılması, normal olamayacak şekilde uzun satırlar, kabuk koduna benzer karakter dizilerinin varlığıdır.

Benzer çalışmalarda web sayfalarının HTML yapısından elde edilen özellikleri kullanılmıştır. Sayfadaki *iframe* etiketlerinin boyutu, görünürlüğü, harici kaynakların kullanımını gösteren *script* etiketinin sayısı gibi özellikler örnek verilebilir.

Bir web sayfasının URL özellikleri ve sunucu bilgileri kullanılarak oltalama ve dolandırma gibi zararlı aktivitelerin tespiti yapılabilmektedir.

Araştırmalar statik yaklaşımların her zaman düşük tespit oranlarını gösterdiğini kanıtlasa da büyük ölçekli bir veri analizi için daha uygun oldukları için, makine öğrenmesine dayalı tespit yaklaşımlarına yönelik araştırmalar artmaktadır [4].

Dinamik ve statik yaklaşımların yanında kara liste yaklaşımı adı verilen sistemle daha önce tespit edilmiş olan kötü amaçlı sayfalar doğru bir şekilde tespit edebilir. Kara liste yaklaşımı, Google Safe Browsing, PhishTank, Malware Domain List gibi projelerde yaygın olarak kullanılmaktadır. Kara listelerin geç güncellenmesi bu yaklaşımın etkili bir şekilde kullanımı sınırlandırmaktadır [4].

2.3.4. Güvenlik duvarları

Güvenlik duvarlarında temel prensip, iç ağı dışarıdaki ağdan izole etmektir. Güvenlik duvarları ilk aşamada dışarıdaki ağdan gelen istekleri inceler ve kaynağın güvenilir olup olmadığına kanaat getirir. İkinci aşama gelen isteğin güvenlik duvarı tarafından belirlenen politikalara uyup uymadığının kontrol edilmesidir. Eşleşme durumunda iç ağ veri alışverişini başlatır. Diğer durumda istek reddedilir [46].

2.3.5. İmza tabanlı tespit

Bilinen saldırıların imzalarına ya da kurallarına dayalı tespit yapar. Kaynak verilere göre tüm bilinen zararlı kodları tespit edebilir. İlgili zararlı kodların tüm olası varyasyonlarını kapsayan bir imza oluşturmak oldukça zorlu bir iştir. Var olan imza ile başarılı eşleşme durumunda tespit tamamlanır. Bilinen zararlı kodlarda tespit edicilerin yanlış alarm olasılıkları oldukça düşüktür [46]. Aynı şekilde bilinmeyen bir saldırının tespitinde yanlış pozitif oranı yüksektir [47].

2.3.6. Bal küpü verilerine dayalı tespit

Bal küpleri aldatma amacıyla özel olarak ayrılmış cihazlardır. Temelinde saldırganın bilgisi olmadan el ile ya da otomatik saldırılar ve saldırgan hakkında bilgi toplama amacı yatar. Saldırganın dikkatini çekebilmek için bilinçli olarak sistemde açıklıklar bulundurulur [48]. Otomatikleştirilmiş davranışlardan zararlı yazılımları toplayan bal küpleri en yaygın kullanılan tespit yöntemlerinden biridir [12].

Bal küpleri düşük etkileşimli, yüksek etkileşimli ve hibrit olmak üzere üç çeşittir. Düşük etkileşimliler, benzeştirilmiş tarayıcı ve asgari işletim sistemi özelliklerini kullanır. Yüksek etkileşimliler, gerçek tarayıcı ve tam işletim sistemi özelliklerini kullanır. Hibrit sistemler iki sistemin iyi yanlarını kullanır.

HoneyC gibi düşük etkileşimli sistemler ön tanımlı imzalara karşı etkileşim esnasında işlemlerin izlerini izlemede sınırlı özelliklere sahiptir. Bunun sonucu olarak kaynak imzaların statik oluşundan dolayı sıfırıncı gün saldırılarının tespiti yapılamaz [11].

Capture-HC, MITRE HoneyClient, HoneyMonkey gibi yüksek etkileşimli istemci bal küpleri etkin ve verimli bir yöntem olmasına karşın ölçeklenebilir değildir. Bu sistemlerde dosya sistemindeki değişiklikler, kayıt defteri bilgileri ya da iş paketlerinin oluşturulması gibi web sayfasının sunulması sırasında sistemde gerçekleşen anormal değişikler izlenir. Bu teknik ile bilinmeyen saldırıların tespitinde verimli sonuçlar elde edilir. Kaynak ve zaman tüketimi açısından verimsizdir. Ayrıca saldırganlar hedef olarak genellikle açıklıkları bulunan işletim sisteminin, tarayıcıların ve eklentilerin belirli açıklıklarını seçer. Bu açıklıkların istismar edilebilmesi için korunmasız unsurlar veya bu unsurların

birleşiminin kullanıcının sisteminde olması gerekir. Dolayısıyla tüm korunmasız unsurların bir arada bulunmadığı sistemlerde zararlı web sayfalarının tespiti için yüksek etkileşimli istemci bal küplerinin kullanımı elverişsizdir [47].

2.3.7. Makine öğrenmesine dayalı tespit

Zararlı web sayfalarından alınan eğitim verilerinin karakteristik özellikleri ile bir model oluşturulur. Oluşturulan bu model test verileri ile test edilir. Daha sonra gelen yeni bir örneğin modele uyup uymamasına göre sınıflandırma yapılır. Makine öğrenmesinde sınıflandırma ve özellik çıkarma için farklı teknikler kullanılır [6-8]. Makine öğrenmesi tekniklerinin etkili kullanımında özelliklerin verimli kullanılması gerekir. Zararlı web sayfalarının tespitinde genel olarak, metin içeriği, HTML, URL özellikleri, JavaScript fonksiyonları ve elemanları, ActiveX elemanları, alan adı ve popülerlik gibi özellikler kullanılır [11].

3. MAKİNE ÖĞRENMESİ TEKNİKLERİ VE YAPAY SİNİR AĞLARI

Makine öğrenmesi teknikleri, temelde sınıfı bilinmeyen bir örneğin önceden oluşturulmuş bir modele göre sınıfının belirlenmesine dayanır. Bu yüzden doğru modelin oluşturulması büyük önem taşır. Modelin oluşturulmasında var olan veriler ile istatistiksel ve matematiksel yaklaşımlar kullanılır. Böylece sisteme öğrenme yetisi kazandırmak amaçlanır.

Temelde, gözetimli ve gözetimsiz olmak üzere iki farklı öğrenme yöntemi vardır. Gözetimli öğrenmede, model sınıfları belli olan verilerle eğitilir. Örneğin, pozitif ve negatif sınıfları olan verilerle gözetimli olarak eğitilen model ile yeni gelen örneğin sınıfının pozitif mi negatif mi olduğuna karar verilir. Gözetimsiz öğrenmede, sınıfı belli olmayan verilerle yapılan öğrenmedir. Algoritmalar, verilerin birbirine olan benzerlikleriyle hareket eder. Örnekler, örnek uzayda birer nokta olarak ifade edilir ve birbirine benzeyen örnekler kümelenir. Çalışmada, sınıfları belli olan bir veri seti üzerinden sınıflandırma modeli oluşturulması uygun görülmüştür.

Literatürde zararlı web sayfalarının tespitiyle ilgili KNN, Bayes, DVM, YSA, Karar Ağacı gibi algoritmalar ve yaklaşımlar mevcuttur. Bu çalışmada KNN, DVM ve YSA tabanlı sistemler çalışıldığı için bu bölümde yalnızca bu algoritmalar hakkında bilgi verilmiştir.

3.2. K-Nearest Neighbor

K-Nearest Neighbor sınıflandırma yöntemi, birbirine yakın olan nesnelerin aynı sınıfa ait olduğunu ileri sürer. K-en yakın komşu algoritması, veri madenciliği, bilgi güvenliğinin sağlanmasında saldırı tespit sistemlerinde, genetik ve biyoinformatiğin birçok alanında, örüntü tanıma sistemleri gibi birçok benzeri sistemde kullanılmaktadır. Temel olarak aşağıdaki adımlara sahiptir.

- Tüm örnekler n-boyutlu Öklid uzayında bir nokta olarak ifade edilir.
- Yeni örnek geldiğinde de bu uzayda gösterilir.
- Yeni gelen örneğin tüm örneklere olan Öklid uzaklığı (Eş. 3.1) hesaplanır.

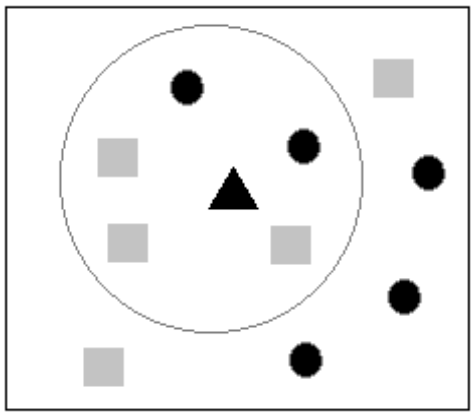
$$d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3.1)$$

Burada x örneğinin y örneğine olan uzaklığı ifade edilmiştir. n, örnek uzayının boyutunu gösterir.

- Kendisine en yakın k örnek ağırlıklandırılarak en yüksek ağırlığa sahip olanlar belirlenir. Ağırlıklandırma faktörü Eş. 3.2'teki gibi hesaplanır.

$$W(y) = \frac{1}{d^2} \quad (3.2)$$

Şekil 3.1'de üçgen örneğin hangi sınıfa ait olduğu belirlenen k değerine göre yakınlık incelemesi yapıldığında kare sınıfına ait olduğu görülmüştür. Buradaki k değeri 5 olarak belirlenmiştir. 5 değeri üçgen örneğe en yakın 5 öğeyi ifade eder [49].



Şekil 3.1. Üçgen örneğin sınıflandırılması

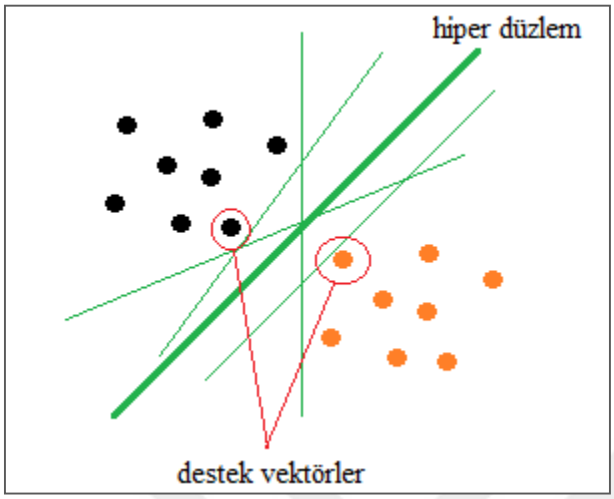
En uygun k değerinin belirlenmesi deneme yanılma ile olur. Hangi k değerinde başarı oranı yüksekse o değer tercih edilir.

3.3. Destek Vektör Makineleri

Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırma temeline dayanır. Bu çizginin (hiper düzlem) çizileceği yer ise iki grubu en bağımsız şekilde bölebilen en uzak olan yer olmalıdır. DVM bu çizginin nasıl çizileceğini belirleyen bir yöntem kullanır [50].

Fazla sayıda veri için çalışması verimsiz olup, çalışma hızı düşüktür. Gürültülü ve eksik verilerde performansı düşüktür [51].

Destek vektörler denilen eğitim setine en yakın noktalara, maksimum mesafedeki karar yüzeyini bularak iki sınıf arasında desen tanımlamayı gerçekleştirir (Şekil 3.2).



Şekil 3.2. Destek Vektör Makineleri sınıflandırıcısında terimlerin gösterimi

$$w^T x_i + b \geq 1 \qquad y_i = 1 \tag{3.3}$$

$$w^T x_i + b \leq -1 \qquad y_i = -1 \tag{3.4}$$

N boyutlu bir $S = \{(x_i + y_i)\}_{i=1}^{N}$ eğitim verisi için i=1, 2, … , N için $x_i \in$ R ve $y_i \in \{-1,+1\}$ dir. S uzayındaki örnekler -1 ve +1 olmak üzere iki sınıftan birine aittir (Eş. 3.3, Eş. 3.4). Şekil 3.2'deki gibi bir hiper düzlemin iki sınıfında tüm örneklerini birbirinden en iyi şekilde ayıran (çizilen hiper düzlemin iki sınıfın örneklerine de en uzak mesafede olması) bir denklemle ifade edilmelidir.

$w \in R^n$ iken aşağıdaki durumları sağlayacak şekilde S kümesi doğrusal olarak ayrılabilir.

$$i = 1,2, \ldots . N \rightarrow y_i(w.x_i + b) \geq 1 \tag{3.5}$$

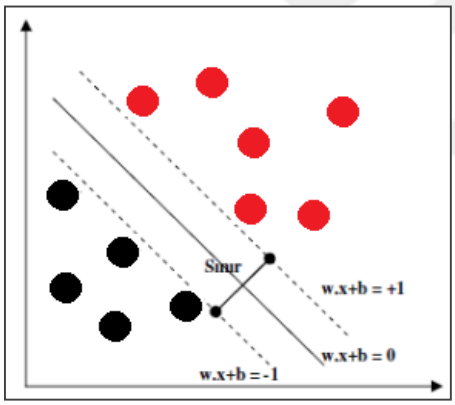
Eşitlik 3.5, (w,b) çiftinin w.x + b = 0 eşitliğine ait ayırt edici hiper düzlemini ifade etmektedir.

$$d_i = \frac{(w.x_i+b)}{\|w\|} \tag{3.6}$$

Burada d_i ayırt edici hiper düzleminden x_i noktasına olan uzaklığı ifade etmektedir. $\|w\|$, w 'nin uzunluğunu göstermek için kullanılır (Eş. 3.6). Böylece bütün $x_i \in$ S'ler için (eğitim örnekleri için) aşağıdaki eşitsizlik sağlanır (Eş. 3.7).

$$\frac{1}{\|w\|} \leq y_i d_i \tag{3.7}$$

S kümesi için en uygun ayırt edici hiper düzlem, S'e ait en yakın noktaya en uzak olan hiper düzlemdir. En uygun ayırt edici hiper düzlemi ve S'e ait en yakın nokta arasındaki mesafe $1/\|w\|$ niceliğinin en büyük olduğu değer S için ayırt edici hiper düzlemi olarak kabul edilebilir. $1/\|w\|$ değerini en büyük yapan değer, $\|w\|$'nin en küçük değerine bağlı olduğu için bu problemin çözümü S'nin en uygun ayırt edici hiper düzlemine eşittir (Şekil 3.3).



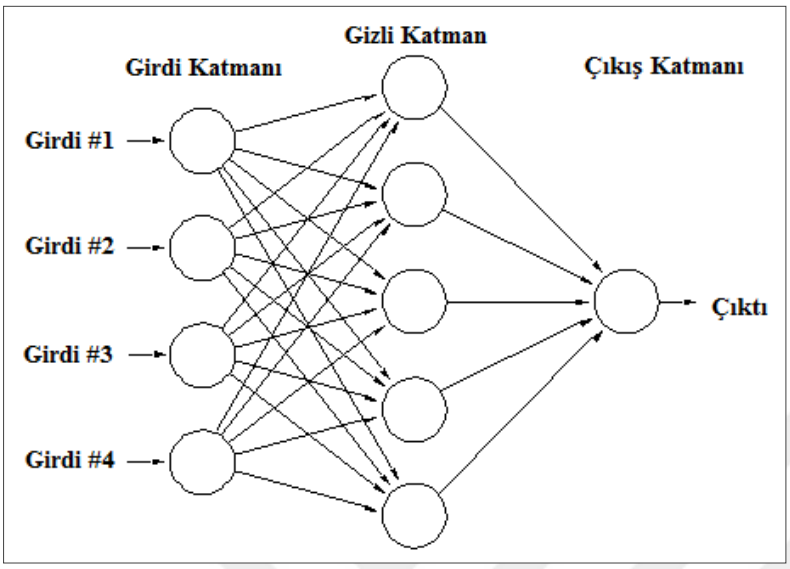
Şekil 3.3. Hiper düzlemin belirlenmesi

Destek Vektör Makineleri genellikle iki sınıflı problemlerde iyi sonuçlar verse de çok sınıflı problemlerde de kullanılabilir.

3.4. Yapay Sinir Ağları

Yapay sinir ağlarının (YSA) ana yapısını insandaki nöronların işleyişi oluşturur. YSA'nın pratik kullanımı genelde, çok farklı yapıda ve formlarda bulunabilen öznitelik verilerini hızlı bir şekilde tanımlama ve algılama üzerinedir. Girişler ve istenen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir [52] (Şekil 3.4). Her

türlü uygulamaya eklenebilir, problem olmadan çalışır. Ağın karmaşıklığı performansını etkileyen unsurlardan biridir.



Şekil 3.4. Yapay sinir ağının yapısı

Bilgiler ağa girdi katmanından gönderilir. Gizli katmanlarda işlenerek oradan çıktı katmanına iletilir. Bilginin işlenmesi, girdi verilerinin ağırlıklandırılmasını ifade eder. Doğru çıktılar üretilebilmesi için doğru ağırlıklandırma gereklidir. Doğru ağırlıkların hesaplanarak bulunması ağın eğitilmesi işlemidir. Ağırlıklandırma başlangıçta rastgele yapılsa da eğitim süresince doğru sonuçlar üretilinceye kadar ağırlıklar sürekli değiştirilir. Daha sonra test verileri ağa gönderilir ve cevaplar beklenir. Doğru cevaplar üretilmişse ağ eğitilmiş olarak kabul edilir. Belirlenen ağırlıkların her birinin ne anlama geldiği bilinmemektedir. Bu yönüyle kara kutu olarak anılır [53].

Gerçek hayatta birçok problem doğrusal olmayan yapıdadır. Çok katmanlı algılayıcılar (ÇKA) doğrusal olmayan problemlerin çözümünde en sık kullanılan YSA modelidir [54]. ÇKA, girdi katmanından gelen verileri işleyen normalde bir adet ara katmanın, girdi/çıktı arasındaki ilişkinin doğrusal olmadığı ve karmaşıklığın arttığı durumlarda birden fazla kullanıldığı ağlardır.

Çok katmanlı algılayıcı bazı zor ve farklı problemleri başarılı bir şekilde çözmek için uygulanır. Oldukça popüler olan hatanın geriye yayılması mantığına dayalıdır. Hatanın geriye doğru yayılması mantığı, ağın ürettiği çıktılar ile üretmesi beklenen çıktılar

arasındaki farkın yani hatanın ağın ağırlıklarına dağıtılarak zamanla en aza indirgenmesidir [55]. Bir ÇKA aşağıda verilen üç önemli özelliğe sahiptir.

1) Ağdaki her nöron modeli çıkışta doğrusallık içermez. Burada önemli bir nokta, doğrusal olmama özelliğinin Rosenblatt'ın algılayıcısında kullanılan keskin-geçişli fonksiyona göre yumuşak geçişli olmasıdır.
2) Ağ, çıkış ya da girişe ait olmayan bir ya da daha fazla saklı nörona sahiptir. Bu saklı nöronlar ağın karmaşık işleri öğrenmesini sağlar.
3) Ağda, her nöron birbiriyle bağlıdır. Bağlantılardaki bir değişiklik, sinaptik bağlantılarda ve ağırlıklarda değişikliğe neden olur [55].

Yukarıda bahsedilen çok katmanlı ağlarda katmanlardaki girdi elemanları ya bir sonraki katmana ya da dış dünyaya gönderilir. Bu elemanların çıktıları geri doğru tekrar girdi olarak kullanılmaz. Bu yapıya ileri beslemeli ağlar denir. Bunun tersine çıkış katmanından veya ara katmanlardan gelen çıktılar geriye doğru önceki katmanlara ya da girdi katmanına geri beslenir.

Çok katmanlı ağlarda giriş ve çıkış sayıları, uygulanacak olan probleme göre belirlenirken, ara katman sayısı ile ara katman nöron sayıları "deneme yanılma" yolu ile bulunur. YSA'larda performansı etkileyen diğer bir önemli etken ise öğrenme algoritmalarıdır. Öğrenme algoritmaları danışmanlı, danışmansız ve takviyeli olmak üzere üç farklı gruba ayrılabilir. Geri Yayılım (BP), Levenberg-Marquardt (LM), Esnek Yayılım (RP), Delta-Bar-Delta (DBD) ve Hızlı Yayılım (QP) gibi algoritmalar danışmanlı öğrenme algoritmalarına, Adaptif Rezonans Teorisi (ART) ve Kendi Kendini Düzenleyen Haritalar (SOM) gibi yapılarda kullanılan algoritmalar ise danışmansız öğrenme algoritmalarına, Genetik Algoritmalar (GA) ise takviyeli öğrenme algoritmalarına örnek olarak verilebilir [55]. Bu tez çalışmasında literatürde pek çok çalışmada başarıyla uygulanan MLP YSA modelini eğitmek için LM, GD (Gradient descent backpropagation), GDA (Gradient descent with adaptive learning rate backpropagation), GDX (Gradient descent with momentum and adaptive learning rate backpropagation), BR (Bayesian regularization backpropagation) gibi öğrenme algoritmaları kullanılmıştır.
Bu algoritmalar aşağıda kısaca açıklanmıştır.

3.4.4. Levenberg-Marquardt Algoritması

LM algoritması, maksimum komşuluk fikri üzerine kurulmuş bir en az kareler hesaplama metodudur [55]. Bu algoritma, Gauss-Newton ve En Dik İniş (Steepest Descent) algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını ortadan kaldırır. Genel olarak bu metot, yavaş yakınsama probleminden etkilenmez.

LM algoritmasında E(w)'nin bir amaç hata fonksiyonu olduğu düşünülürse, m adet çıkış nöronu için hata terimi $e_i^2(w)$ [55]:

$$E(w)=\sum_{i=1}^{m} e_i^2(w)=\|f(w)\|^2 \tag{3.8}$$

olarak verilir.

Eş. 3.8'de w ağırlıkları ifade edilirken,

$$e_i^2(w) \equiv (y_i - yd_i)^2 \tag{3.9}$$

Burada, amaç fonksiyonu f(.) ve onun Jakobiyeni J'nin bir noktada w bilindiği farz edilir.

LM öğrenme algoritmasında hedef, parametre vektörü *w*'nin, E(*w*)'yi minimum yapacak şekilde optimize edilmesidir. LM algoritmasının kullanılmasıyla yeni vektör w_{k+1}, farz edilen vektör w_k'dan Eş. 3.9 yardımıyla aşağıdaki gibi hesaplanabilir (Eş. 3.10).

$$w_{k+1} = w_k + \delta w_k \tag{3.10}$$

burada δw_k ifadesi

$$(J_k^T J_k + \lambda I)\delta w_k = -J_k^T f(w_k) \tag{3.11}$$

eşitliğinden faydalanılarak hesaplanır. Eş. 3.11'de;
J_k**:** f'nin w_k değerlendirilmiş Jakopyeni,
λ**:** Marquardt parametresi,

I: birim veya tanımlama matrisidir.

Levenberg-Marquardt algoritmasında hesaplama akışı aşağıdaki şekilde özetlenebilir [55].

1) $E(w_k)$ 'yı hesapla
2) küçük bir λ değeri ile başla mesela $(\lambda = 0.01)$
3) δw_k için Eş. 3.5'i çöz ve $E(w_k + \delta w_k)$ değerini hesapla,
4) şayet $E(w_k + \delta w_k) \geq E(w_k)$ λ'yı 10 kat artır ve (iii)'e git
5) şayet $E(w_k + \delta w_k) < E(w_k)$ λ'yı 10 kat azalt, $w_k : w_k \leftarrow w_k + \delta w_k$'yı güncelleştir ve 3.adıma git.

Hedef çıkışı hesaplamak için YSA ağırlıklarının LM öğrenme algoritması kullanılarak eğitilmesi, ağırlık dizisi w_0'a bir başlangıç değerinin atanması ile başlar ve hataların karelerinin toplamı e_i^2'nin hesaplanmasıyla devam eder. Her e_i^2 terimi, hedef çıkış (y) ile gerçek çıkış (y_d) arasındaki farkın karesini ifade eder. Bütün veri seti için e_i^2 hata terimlerinin tamamının elde edilmesiyle, ağırlık dizileri (i)'den (v)'ye kadar olan hesaplama akışı içerisinde, daha önce de açıklandığı gibi LM öğrenme algoritması adımlarının uygulanmasıyla adapte edilir.

3.4.5. Geri yayılım algoritması

Bu algoritma özetle, ağırlıklar ve eşik değerleri için aynı şekilde bir hesaplama yöntemi tanımlamaktadır.

$$x_{k+1} = x_k - \alpha_k g_k \tag{3.12}$$

ile tanımlanır. Burada x_k mevcut andaki ağırlık değerini, x_{k+1}düzeltilmiş ağırlık değerlerini, α_k öğrenme katsayısını, g_k eğimin mevcut değerini ifade eder. Standart geri yayılım algoritması eğimin azaltılması için 2 farklı yöntem sunmaktadır. İlk yöntem olan artırmalı şekilde eğitimlerin hesaplanarak ağırlıkların güncelleştirilmesi işlemi her bir girişin ağa uygulanmasıyla gerçekleştirilir, diğer yöntem olan grup şeklinde ise ağırlıkların güncelleştirilmesi için tüm eğitim kümesinin ağa uygulanması gerekmektedir.

3.4.6. Momentumlu geri yayılım algoritması

Uygulamalarda en yaygın ve en sık kullanılan yöntem standart geri yayılım algoritmasıdır. Bu algoritmanın işlem yeteneği rahat anlaşılabilir ve matematiksel olarak ifade edilmesi kolaydır. Bu nedenle en çok tercih edilen öğrenme algoritmasıdır. Bu algoritma, hataları çıkıştan girişe doğru azaltmaya çalışmasından dolayı geri yayılım olarak adlandırılmıştır. Geri yayılım algoritması ÇKA'ları eğitmede en çok kullanılan temel bir öğrenme algoritmasıdır [55]. Ağırlıkları değiştirme işlemi veya öğrenme işlemi aşağıdaki adımlarda verilmiştir.

1) Başlangıç ağırlıklarını rastgele seç
2) Öğrenmeye başla
3) Giriş kümesini girişe uygula
4) İşlemci elemanlarının üzerinden çıkışı hesapla
5) Hayatı Bul ve Hata kabul edilemez ise Eğim (Gradient) azaltma ile ağırlıkları düzenle ve (3) işlemine git.
6) Test işlemine başla
7) Test kümesini yapay sinir ağının girişine uygula
8) İşlemci elemanlarının üzerinden çıkışı hesapla
9) Test kümesi tamamlandıysa dur, yoksa (7) nolu işlemine git.

Eğitim ve test işlemleri aynı şekilde yapılmaktadır. Bu algoritma ile i ve j elemanları arasındaki ağırlıklardaki değişim $\Delta w_{ji}(t)$ Eş. 3.13'teki gibi verilir.

$$\Delta w_{ji}(t+1) = \eta \delta_j x_i + \alpha \Delta w_{ji}(t) \tag{3.13}$$

η : Öğrenme katsayısı

α : Momentum Katsayısı

δ_j : Ara veya çıkış katmanındaki herhangi bir j nöronuna ait faktördür.

Çıkış katmanı için ilgili denklem Eş. 3.14'te verilmiştir.

$$\delta_j = \frac{\partial f}{\partial net_j}(y_j^{(t)} - y_j) \tag{3.14}$$

Bu denklemde $net_j = \sum x_j w_{ji}$ ile ifade edilirken, $y_j^{(t)}$ ise j işlemci elemanının hedef çıkışıdır. Ara katmanlardaki nöronlar için bu faktör Eş. 3.15'de verilmiştir.

$$\delta_j = \frac{\partial f}{\partial net_j}\sum w_{qi}\,\delta_q \tag{3.15}$$

Ara katmanlarda nöronlar için herhangi bir hedef çıkış bulunmadığından Eş. 3.14 yerine Eş. 3.15 kullanılır. Bu algoritma temel olarak istenilen çıkış ile yapay sinir ağının hesaplanan çıkışı arasında hatanın ağırlıklara bağlı olarak düşürülmesini hedeflemektedir [55].

3.4.7. Esnek yayılım algoritması

ÇKA yapılarında genel olarak ara katman transfer fonksiyonları sigmoid tercih edilmektedir. Sonsuz aralıkta olabilen giriş değerleri sınırlı bir aralığa sıkıştırıldığı için bu fonksiyonlar sıkıştırıcı fonksiyonlar olarak anılmaktadır. Sigmoid transfer fonksiyonları büyük giriş değerlerini sıfıra yakınsayacak şekilde işlem görürler. Bu sebeple bias ve ağırlık değerleri henüz istenilen düzeye erişmeden yaşanan eğim değerinin çok yavaş değişmesi olasılığı öğrenme sürecinde sorunlara neden olmaktadır. Bu öğrenme algoritmasının amacı kısmi türevlerin olumsuz sonuçlara neden olmasını engellemektir. Ağır değerini güncelleştirmek için sadece türev değerlerine ait işaretler kullanılır. Bu öğrenme algoritması ile diğer algoritmalar arasındaki en belirgin farklılık türev değerinin öneminin olmamasıdır. Bu özellik esnek yayılım algoritmasına hızlı çözüm yeteneği kazandırmaktadır [55].

$$\sum w_{ji}(k) = \begin{cases} -A_{ji}(k)\ ise\ B(k) > 0 \\ +A_{ji}(k)\ ise\ B(k) < 0 \\ 0, ise\ değişim\ yok \end{cases} \quad (3.16)$$

ile hesaplanır. Burada $B(k) = \frac{\partial E}{\partial w_{ji}}(k)$ hesaplanır. $A_{ji}(k)$ ise,

$$A_{ji}(k) = \begin{cases} \eta A_{ji}(k-1)\ ise\ B(k-1)B(k) > 0 \\ \mu A_{ji}(k-1)\ ise\ B(k-1)B(k) < 0 \\ A_{ji}(k-1)\ ise\ değişim\ yok \end{cases} \quad (3.17)$$

Eşitlik $B(k-1) = \frac{\partial E}{\partial w_{ji}}(k-1)$, $\eta\ ve\ \mu$ sırasıyla $0 < \mu < 1 < \eta$ artma ve azalma faktörleridir. Bu hesaplama süreçleri içerisinde türev değeri sıfır ise güncelleme değeri sabit kalır. Ağırlıklar salınım yaptığında, ağırlık değişimi azalacağı gibi birkaç iterasyon boyunca değişim aynı yönde gerçekleşirse ağırlık değişimi artar. Bu algoritma sayesinde hafıza eski değerler saklanmadığından sistem performansını artırmaktadır [55].

4. ZARARLI WEB SAYFASI TESPİTİ

Bu tez çalışmasının amacı, incelenen çalışmalar doğrultusunda belirlenen ve çıkartılan özniteliklerin DVM, KNN ve YSA teknikleriyle sınıflandırılarak zararlı web sayfalarını en yüksek doğruluk oranı ve en düşük yanlış pozitif oranı elde ederek tespit etmektir. Seçilen sınıflandırma tekniklerinin kullanıldığı çalışmalar literatürde mevcutken, oluşturulan veri seti ve uygulanan öznitelik seçme yöntemleri, incelenen çalışmalardan farklıdır.

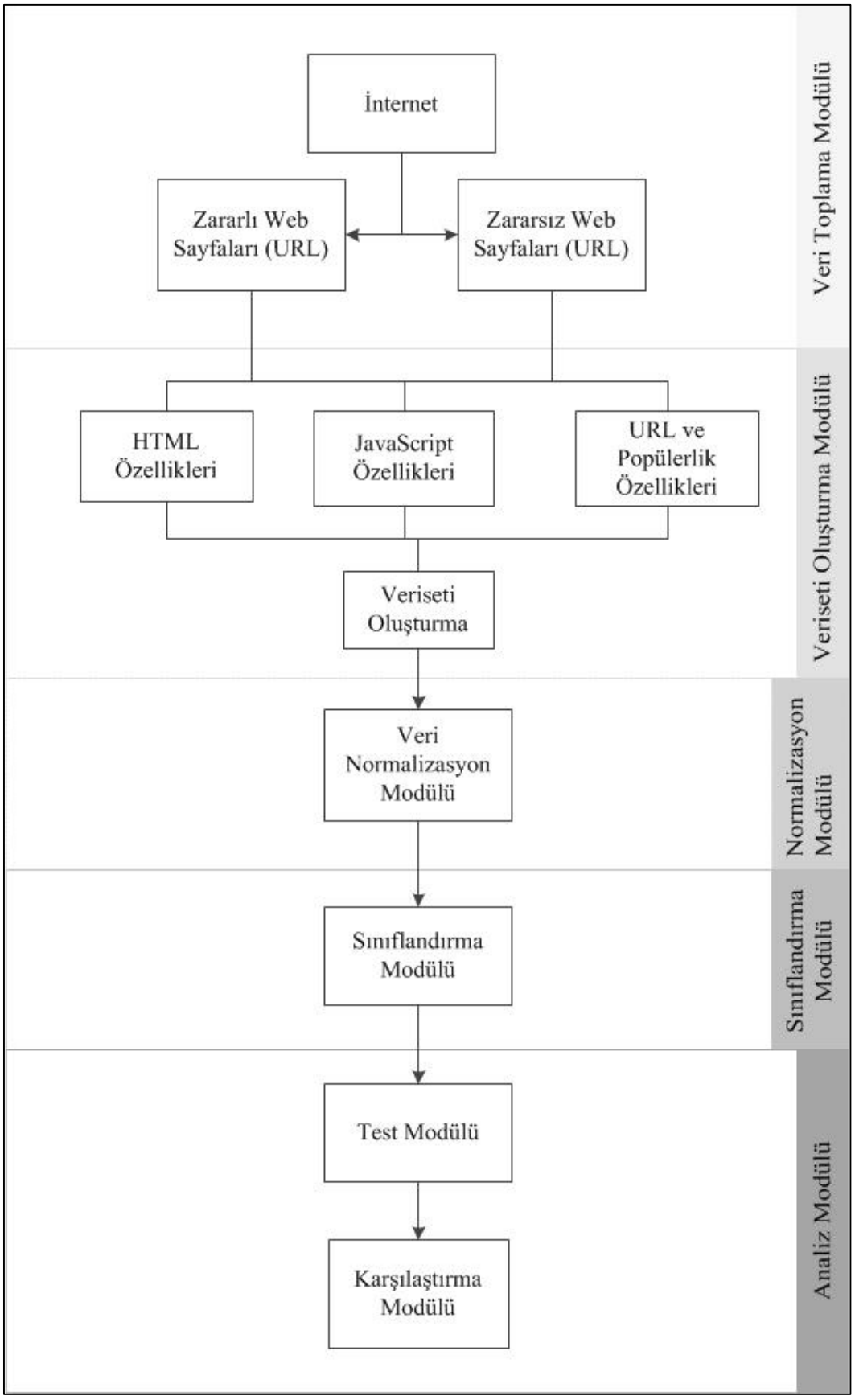
Çalışmada zararlı web sayfalarının tespiti için izlenen akış ana hatlarıyla Şekil 4.1'de gösterilmiştir.

Bu bölümde önerilen modelin blok diyagramında da gösterilen temel modüllerine yer verilecektir.

4.2. Veri Toplama Modülü

Makina öğrenmesi teknikleriyle geliştirilen zararlı web sayfalarının tespitinde karar vermeyi etkileyen en önemli hususlardan biri uygun bir veri seti oluşturmaktır. Bu işlemin ilk adımı özellikleri çıkarılacak web sayfalarına ulaşmayı gerektirmektedir.

İnternet üzerinde çok sayıda web sayfası bulunmaktadır. Zararlı web sayfaları, web sayfalarının zararlı olduğunu tespit eden ve sınıflandırıp arşivleyen PhishTank, openphish.com, malc0de.com, clean-mx realtime database, Google Safe Browsing gibi açık kaynaklarda yer aldıkları gibi güvenlik duvarlarının, anti-malware yazılımlarının engellediği ve zararlı kabul ettiği ücretli listelerde de bulunabilmektedir. Bahse konu kaynaklardan ulaşılan web sayfaları kara listelere işlendiğinden birçoğuna erişim tarayıcılar tarafından engellenmektedir. Zararlı web sayfalarının türleri oltalama, zararlı kod içerme, reklam yazılımları gibi farklı tiplerde olacak şekilde seçilmiştir. Böylece farklı türlerden zararlı web sayfalarının tespit edilmesi öngörülmektedir.



Şekil 4.1. Sistemin blok diyagramı

Zararsız web sayfalarının toplanmasında kullanılacak kaynakların güvenilir olması önem taşımaktadır. Toplanan web sayfaların gerçekten zararsız olduğundan emin olmak için incelenen çalışmalarda en çok Alexa'nın en çok ziyaret edilen 1 milyon web sitesi listesi ile DMOZ açık kaynak projesinde arşivlenen web sitelerinin kullanıldığı görülmüştür. Zararlı web sayfalarının toplanma şeklinden anti-malware ve güvenlik duvarları tarafından engellenmeyen web sayfalarının zararsız olduğu sonucuna varılabilir. Fakat bu durum her koşulda geçerli değildir.

Bu amaçla, çalışmada kullanılmak üzere işlenecek web sayfası seti, PhishTank, openphish.com, malc0de.com, clean-mx realtime database arşivlerinden ve Fortinet güvenlik duvarı yazılımından toplanan zararlı, Alexa, DMOZ açık kaynak projesi ve diğer açık kaynaklardan toplanan zararsız web sayfalarından oluşturulmuştur. İşlenmek üzere yaklaşık 15 000 URL toplanmıştır.

4.3. Veri Seti Oluşturma Modülü

Bu modülde oluşturulacak özellikler literatürde kabul görmüş ve kullanılmş özelliklerden seçilmiş olup, en yüksek doğruluk oranını elde edebilmek için farklı özellik setleri ile testler yapılmıştır.

Sınıflandırma için kullanılacak özniteliklerin çıkarılmasında web sayfasından elde edilebilen özellikler kullanılmıştır. Bu özelliklerin elde edilmesi ve kullanılması yukarıda bahsedilen statik yaklaşımı kapsar. Çalışmada kullanılan bu özellikler üç ana başlıkta toplanmıştır.

URL ve popülerlik özellikleri, URL'nin sözcüksel ve popülerliği ile ilgili özellikleri içerir. URL uzunluğu, URL içindeki belirli karakterlerin sayısı gibi özellikler sözcüksel; whois bilgileri sayısı, Alexa sıralaması gibi özellikler popülerliği ifade eden özellikler olarak kabul edilmiştir.

JavaScript ile ilgili özellikler, web sayfasında kullanılan şüpheli JavaScript fonksiyonlarıyla ilgili özelliklerdir. *exec()* fonksiyonu, *document.cookie* değişkeni, *escape()* fonksiyonu sayısı gibi özellikler bu grupta incelenmiştir.

HTML ile ilgili özellikler web sayfasının içerdiği şüpheli html etiketleri ile ilgili özellikleri içerir. *"script"* etiketi sayısı, *"iframe"* etiketi sayısı gibi özellikler bu grupta incelenmiştir.

Toplanan URL'lerden özniteliklerin elde edilmesinde URL'leri ve kaynak kodlarını inceleyen ve parçalayan, Alexa global sıralama gibi popülerliğini sorgulayan ve WHOIS sorgusu yaparak site kayıt bilgilerine ulaşan Python [56] programlama dili ile geliştirilen parçalayıcı yazılımından faydalanılmıştır. Python programlama dilinin tercih edilmesinin sebebi tüm platformlarda uyumlu çalışması, kolayca ulaşılabilen ve kullanılabilen çok sayıda ilgili kütüphaneye sahip olması ve örnek olarak seçilen web sayfalarında öznitelik çıkarma işleminin daha hızlı gerçekleştirilmesidir.

Toplanan zararlı ve zararsız web sayfaları "webpages.txt" isimli bir dosyaya kaydedilir. Bu dosyadan okunan URL'ler "test.py" adlı Python kodu ile işlenir. İşlemin ilk basamağı dosyadan okunan verilerin bir diziye atılmasıdır. Daha sonra bu dizinin elemanları çoklu işlem özelliği ile işlenmiştir. Çoklu işlem sisteminde işlenecek liste için iş parçacıkları oluşturulmuştur. Bu sayede verilen listenin daha hızlı bir şekilde işlenmesi ve özelliklerin çıkarılması sağlanmıştır (Şekil 4.2).

```
def main():

    file = 'webpages.txt'
    f = open(file, "r")
    lines = f.readlines()
    f.close()

    jobs = []
    start = 0
    finish = len(lines) / 10
    for i in range(0, 10):
        if i == 10:
            p = multiprocessing.Process(target=featureextraction, args=(lines[start:],))
        else:
            p = multiprocessing.Process(target=featureextraction, args=(lines[start:finish],))

        print str(i+1) + '. process basladi.'
        jobs.append(p)
        p.start()
        start = finish
        finish += len(lines) / 10

    for index, p in enumerate(jobs):
        p.join()
        print str(index) + '. process bitti.'
```

Dosya Okuma

İş Parçacıklarını Başlatma

Şekil 4.2. Dosya okuma ve iş parçacığı oluşturma işlemleri

Çalışmada zararlı web sayfalarının tarayıcılarca engellenmesi HTML kodlarına erişimi kısıtlayan bir etken olmuştur. Bu web sayfalarının açılıp arkasında çalışan kodlara erişebilmek için Python'un "selenium" ve "webdriver" kütüphaneleri kullanılmıştır. Özellik çıkarmada kullanılan bazı web sayfalarının açılır açılmaz bir dosya indirmeye çalıştığının gözlenmesi üzerine bu tarz web sayfalarının tespitine yönelik bir ekleme yapılmıştır. Bunun ayırt edici bir özellik olarak kullanılabileceği düşünülse de bu çalışmada tercih edilmemiştir. Çıkarılan özelliklerin doğruluğunu artırmak ve eksikliğini azaltmak amacıyla yalnızca erişime açık web sayfaları seçilmiştir. Tüm koşullar sağlandıktan sonra web sayfası içeriğinden, URL, popülerlik ve alan adına ait özelliklerinden 20 tane özellik çıkartılmıştır (Şekil 4.3).

HTML ve JavaScript özelliklerinin çıkarılmasında Python'un "requests" ve "BeautifulSoup" kütüphaneleri kullanılmıştır [57]. Bu kütüphaneler ile parametre olarak verilen URL'nin HTML içeriği kolayca elde edilmiştir. Bu içerik içinde çıkarılmak istenen özellikler aranır ve saydırılır.

Alan adına ait özelliklerin çıkarılmasında Python'un "tldextract" [58] ve "pythonwhois" [59] kütüphaneleri kullanılmıştır. "tldextract" kütüphanesi ile parametre olarak verilen URL'in üst seviye alan adı ve domain adı parçalara ayrılır. Bu kütüphane yardımıyla parçalan URL'in alan adı "registered_domain" özelliğiyle çıkarılır. URL'den çıkarılan alan adı, whois bilgilerine ulaşmak için kullanılacak olan "pythonwhois" modülünde parametre olarak kullanılır. Bu sayede ulaşılan alan adının whois bilgileri sayısı rahatça saydırılmıştır. Aynı şekilde alan adı sunucusu sayısına da whois bilgilerinden ulaşılmaktadır.

Popülerlik ile ilgili özellikler Google PageRank ve Alexa global sıralamasından oluşmaktadır. "Google PageRank" değeri Google firmasının PageRank güncellemesini kaldırmasıyla geçerliliğini kaybettiği için başlangıçta kullanılıyor olsa da sonrasında özellik setinden kaldırılmıştır.

İkinci bir popülerlik ile ilgili özelliklerden biri olan URL'nin Alexa'daki global sıralamasının elde edilmesinde Şekil 4.4'teki fonksiyon kullanılmıştır.

```
def featureextraction(lines):
    for line in lines:
        try:
            res = requests.get(line,timeout=10)
        except Exception as hata:
            print "request "+str(hata)
            continue
        if res.status_code == 200:

            try:
                driver = webdriver.Firefox()
                driver.get(line)
                html = bs(driver.page_source)
                driver.close()
            except Exception as e:
                print "hata driver" + str(e)
                driver.close()
                continue

            try:
                if is_downloadable(line):
                    downloadable = 'downloadable'
                else:
                    downloadable = 'undownloadable'

                html =str(html)
                hrefcount=html.count('href=')
                scsay=html.count('<script')
                embsay=html.count('<embed')
                ifrsay=html.count('<iframe')
                hiddsay=html.count('display: none')
                winlocsay=html.count('window.location')
                cookiesay=html.count('document.cookie')
                srcsay=html.count('src=')
                dotcount=line.count('.')
                charactercount=len(line)
                etcount=line.count('@')
                evalcount=html.count('eval(')
                docwritecount=html.count('document.write(')
                escount=html.count('escape(')
                execcount=html.count('exec(')
                linkcount=html.count('window.open(')

                ext=tldextract.extract(line)
                tt=pythonwhois.get_whois(ext.registered_domain)

                whoiscount=str(len(tt))

                nameservers=str(len(tt["nameservers"]))
                a = ['account', 'login', 'email','confirm','secure','payment']
                if any(x in line for x in a):
                    specialwordsvarmi="1"
                else:
                    specialwordsvarmi="0"
```

Web sayfasının erişilebilir olduğunun kontrolü

Web sayfasının tarayıcıda açılmasının sağlanması

Açılan içeriğin indirilebilir olup olmadığının kontrolü

HTML ve JavaScript özelliklerinin çıkarılması

URL ve popülerlik özelliklerinin çıkarılması

Şekil 4.3. Veri seti özelliklerinin çıkarılması

```
def get_rank(domain_to_query):
    result = {'Global':''}
    url = "http://www.alexa.com/siteinfo/" + domain_to_query.strip()
    page = requests.get(url).text
    soup = bs(page,"html.parser")
    for span in soup.find_all('span'):
        if span.has_attr("class"):
            if "globleRank" in span["class"]:
                for strong in span.find_all("strong"):
                    if strong.has_attr("class"):
                        if "metrics-data" in strong["class"]:
                            result['Global'] = strong.text.strip()
    return result
```

Şekil 4.4. Alexa global sıralamasının elde edilmesi

Bu fonksiyonda web sayfalarından bilgi edinmek (web scraping) için popüler bir kütüphane olan "BeautifulSoup" kütüphanesi kullanılmıştır. Bu kütüphanenin fonksiyonları yardımıyla HTML sayfalarındaki veriler parçalanır ve bilgiler çıkarılır. Temelinde bu işlem, "http://alexa.com/siteinfo" sayfasına parametre olarak verilen bir web sayfası sorgusundan global sıralama değerini bulmak olarak ifade edilebilmektedir.

Veri setinin amacı, web sayfalarının kaynaklarından elde edilen bilgilere göre o web sayfasının zararlı olup olmadığına karar verilmesini sağlamaktır. Veri setinde kullanılan özellikler temel 3 ana başlıkta incelenmiştir.

4.3.4. HTML özellikleri

HTML ile ilgili özellikler, web sayfasının içerdiği şüpheli HTML etiketleri ile ilgili özellikleri içerir. İncelenen çalışmalarda HTML özellikleri, zararlı web sayfalarının tespitinde sıklıkla kullanılmıştır. Bu çalışmada yer alan HTML özelliklerinin kullanıldığı diğer çalışmalar ile karşılaştırılması Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. HTML özelliklerinin karşılaştırılması

Özellik / Kaynak	script	embed	iframe	src	Başarı Oranı (%)	Yanlış Pozitif Oranı (%)
[7]	√	√	√	-	97,9	0,0142
[10]	-	√	√	-	95,4	4,5
[17]	-	-	√	-	98	0,011
[18]	√	√	√	-	85,14*	5,46
Önerilen Yaklaşım	√	√	√	√	98,57	0,006

*: Bu çalışma en az %85,14 başarı oranı sağlayan bir filtreleme çalışmasıdır.

Buna göre önerilen yaklaşımda, diğer çalışmalarda kullanılan HTML özelliklerinin yanı sıra farklı olarak dışarıdan kaynak kullanımını mümkün kılan *src* elementi özelliği yer almaktadır. Seçilen HTML özellikleri kombinasyonunun aynı şekilde kullanıldığı başka bir çalışma bulunmamaktadır.

script etiketi sayısı

Web sayfasının HTML ile ilgili özelliklerindendir. Sayfanın kaynak kodunda geçen *script* elementi sayısını ifade eder. Bu etiket ile web sayfasında içeriden veya *src* elementi ile harici olarak kaynak gösterilip kod çalıştırılabilir. Bu kodlar faydalı amaçlarla kullanılabileceği gibi bazı zararlı fonksiyonlarla zararlı hale getirilebilmektedir [7, 18].

embed etiketi sayısı

Web sayfasının HTML ile ilgili özelliklerindendir. Web sayfasına kaynağı harici veya dahili olan ses, görüntü, video gibi dosyaları ekler. Bu etiket ile zararlı içeriğe sahip web sayfalarının kaynakları yüklenebilir. Bir web sayfasında bulunması şüpheli olarak kabul edilebilir [7, 10, 18].

iframe etiketi sayısı

Web sayfasının HTML özelliklerinden olan *iframe* etiketi ile web sayfalarında harici başka web sayfaları bir çerçeve içinde görüntülenebilir. Çerçeve boyutu, etiketin genişlik ve yükseklik değerleriyle belirlenir. Zararlı web sayfalarının kaynak olduğu *iframe*

etiketlerinde genellikle boyut 0 olarak verilir ya da görünmez olarak ayarlanır. Böylece ziyaret edilen web sayfasında aslında zararlı web sayfalarının var olduğu görülmemiş olur. Bu etiket incelenen çalışmaların birçoğunda kullanılmıştır [7, 9, 10, 16-18].

display:none özelliği sayısı

Web sayfasının HTML ile ilgili özellikleri kısmında incelenmesine rağmen bu yazım biçimi CSS (Cascading Style Sheets) olup HTML elemanlarının görünümlerini düzenleyen bir dildir. Bahse konu özellik web sayfasındaki bir nesnenin görünürlük özelliğini ifade eder. Bir nesnenin görünürlüğünün "none" olarak ayarlanması genellikle şüpheli bir durumdur, ziyaretçinin göremediği zararlı içerikler olabileceği anlamına gelir. Gizli nesne sayısının fazla olması da aynı şekilde şüpheli olarak görülebilir.

href elementi sayısı

Web sayfasının HTML ile ilgili özellikleri kısmında incelenmiş olup web sayfasında bağlantı yapmak için kullanılır. İçeriğinden farklı bir bağlantıya yönlendirilen web sayfalarında şüpheli olarak kabul edilebilir.

src elementi sayısı

Web sayfasının HTML ile ilgili özelliklerindendir. Özelliği olduğu nesnenin kaynağını ifade eder. Bu kaynak, ses, video ye ya URL olabilir. Kaynakların zararlı dosya veya web sayfası içerme durumu olabileceği için şüpheli görülebilir.

4.3.5. JavaScript özellikleri

Web sayfasında kullanılan şüpheli JavaScript fonksiyonlarıyla ilgili özelliklerdir. İncelenen çalışmalarda JavaScript özellikleri, JavaScript diline özgü bazı fonksiyonların zararlı web sayfalarının içeriğinde tespit edilmesinden elde edilmiştir. Bu çalışmada yer alan JavaScript özelliklerinin kullanıldığı diğer çalışmalar ile karşılaştırılması Çizelge 4.2'de gösterilmektedir.

Çizelge 4.2. JavaScript özelliklerinin karşılaştırılması

Özellik / Çalışma	window.location	document.cookie	document.write	eval	escape	exec	Başarı Oranı (%)
[7]	√	√	-	-	-	-	97,9
[10]	-	-	-	√	-	-	95,4
[17]	-	-	-	√	√	√	98
[18]	-	-	-	√	-	-	85,14*
Önerilen Yaklaşım	√	√	√	√	√	√	98,57

*: Bu çalışma en az %85,14 başarı oranı sağlayan bir filtreleme çalışmasıdır.

Buna göre önerilen çalışmada incelenen diğer çalışmalardan farklı olarak document.write fonksiyonu sayısı özelliği bulunmaktadır. Seçilen JavaScript özellikleri kombinasyonunun aynı şekilde kullanıldığı başka bir çalışma bulunmamaktadır.

window.location fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. Yer aldığı web sayfasından başka bir web sayfasına yönlendirme yapmayı sağlar. Yönlendirdiği web sayfası zararlı olabileceği için şüpheli olarak kabul edilmiştir [7].

document.cookie değişkeni sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. Yer aldığı web sayfasına girildiğinde kullanıcının çerez bilgilerini kullanır. Çerezler, ziyaret edilen web sayfaları tarafından oluşturulur ve ziyaretçinin tekrar geldiğinde profil bilgilerini, şifrelerini hatırlaması için kaydeder. Çerezler içinde hassas veriler olabileceği için zararlı web sayfalarının çerezleri ele geçirme amacı muhtemeldir. Dolayısıyla bu web sayfalarında *document.cookie* nesnesi bulunması şüpheli kabul edilmiştir [7].

eval fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. Bu fonksiyon, içine aldığı karakter dizisi parametresi JavaScript dilinde yorumlanabilecek anlamlı bir ifade içeriyorsa o komutu çalıştırır (Şekil 4.5).

```
eval(execute(NVDIF(,1D+VB+41+*2,1D+4*+4 & *+V[+VI1YM1[*/Z>2RN1Q & )2?=/8Y2?B.;[1=H2+Q2S &
;0N@.@3,OO/8+/ZU0?O0N & C3(Y.@I13I1Q62S;0N@.: & Z2[L1G:2?P1=C2[Y1Q)+Q & L2?C2S5/VT-.T2X-
.6Q2D & E.JR-BJ3(U1G>2?R.;[29 & B3))3AW0IG1OT1[*3)63( & K.OA1F.2D<1ZS0,F-BJ1P &
U1[*2D<2??,;Q3=(/[A29 & B3))3AW0IG3;D1ZU1<(+* & =25I1L12S;2DH,7J2C,2D & ?2DD,7J2VP13O3AT+*R1G
& /1GD+*=+*A,7S1ET1GD24 & F1G82X01G>2DO+5D+4*2> & 2/QY+VN+4*+4*25P1UY,< &
+2IH2S5/VC+5++4*+QF1= & H2+Q/BR3))1G-.<*+4*1F & (3.41G/.F(1F(2DK.;X3( & 21[,1=F18K++R3(Y2X0+5
& A2>0/QY+VN24:1G8/VH2[ & Z1[,1=F05W+QF1=H2+Q/B & R3))1G-.<*2[Z1[,1=F05 & W ))
```

Şekil 4.5. Eval fonksiyonu kullanımı

Bu fonksiyon çalıştırıldığında Şekil 4.5'te gizlenmiş aslında Şekil 4.6'daki komut çalıştırılır. Bu komut şüpheli bir "powershell" aktivitesi gerçekleştirmektedir.

```
WScript.CreateObject("WScript.Shell").Run "mshta
vbscript:Close(Execute(""CreateObject(""""WScript.Shell"""").Run""""powershell.exe -w 1 -exec Bypass -nologo -
noprofile -c iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String((get-content
C:\ProgramData\SYSTEM32SDK\ProjectConfManagerNT.ini))));"""",0 ""))",0
```

Şekil 4.6. Eval fonksiyonun çalıştırdığı komut

Bu özelliğiyle istismar edilmeye uygun bir fonksiyondur ve kötü niyetli kişilerce sıklıkla kullanılmaktadır [9, 10, 12, 17, 18].

document.write fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. İçine aldığı HTML ve JavaScript parametrelerini ekranda gösterir. Bu haliyle zararlı kod çalıştırılmasına imkân verebilir. Tam olarak şüpheli bir fonksiyon olmasa da kesin olarak zararsız kabul edilemez.

escape/unescape fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. İçine aldığı karakter dizisini kodlar ya da kodlanmış karakter dizisini çözer. Bu özelliği ile istismar edilmeye açıktır [17, 29].

window.open fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. Parametre olarak aldığı URL adresini yeni bir tarayıcı penceresinde açar. Çoğu film dizi izleme sitelerinde kendiliğinden açılan bahis, oyun ve reklam sayfaları bu ve bunun gibi kullanılabilen şüpheli fonksiyonlardan kaynaklanmaktadır.

exec fonksiyonu sayısı

Web sayfasının JavaScript ile ilgili özelliklerindendir. Bu fonksiyon, içine aldığı karakter dizisinin önceden belirlenen düzenli bir ifadeyle eşleşip eşleşmediğini gösterir. Zararlı web sayfaları tarafından istismar edilmeye uygun bir fonksiyondur [17].

4.3.6. URL ve popülerlik özellikleri

URL özellikleri, URL adresinin sözcüksel ve alan adıyla ilgili özelliklerini kapsar. Alan adıyla ilgili özellikler web sayfasının güvenilirliğiyle ilgili önemli bilgiler vermektedir. Bu kapsamda alan adı kaydedilirken verilen bilgilerin sayısı ve alan adının barındığı sunucu sayısı özellikleri çıkarılmıştır. Zararlı web sayfaların çok sık ziyaret edilmeyeceği düşüncesinden yola çıkılarak popülerlik özellikleri incelenmiştir. Bununla ilgili olarak web sayfasının trafiğine göre popülerlik çıkarımı yapabilen Google PageRank ve Alexa hizmetinden faydalanılmıştır. Çalışmanın sonraki zamanlarında Google PageRank değeri kullanımı getirilen kısıtlamalar yüzünden kaldırılmıştır. Bu çalışmada yer alan URL ve popülerlik özelliklerinin kullanıldığı diğer çalışmaların karşılaştırılması Çizelge 4.3'te gösterilmektedir.

Çizelge 4.3. URL ve popülerlik özelliklerinin karşılaştırılması

Özellik / Çalışma	URL nokta sayısı	URL karakter sayısı	Alexa global sıra	Whois bilgileri sayısı	Alan adı sunucu sayısı	Başarı Oranı (%)
[10]	√	√	-	-	-	95,4
[15]	√	-	√	√	-	97,5
[17]	-	-	-	-	√	98
[18]	-	-	-	√	-	85,14*
Önerilen Yaklaşım	√	√	√	√	√	98,57

*: Bu çalışma en az %85,14 başarı oranı sağlayan bir filtreleme çalışmasıdır.

Seçilen URL ve popülerlik özellikleri kombinasyonunun aynı şekilde kullanıldığı başka bir çalışma bulunmamaktadır.

URL içindeki nokta sayısı

Web sayfasının URL ile ilgili özelliklerindendir. Zararlı web sayfalarının URL'lerindeki nokta sayısı fazladır. Özellikle oltalama sitelerinin URL'lerinin daha çok nokta içerdiği, alan adında ve yolunda daha fazla ve daha uzun simgeler içerdiği bilinmektedir [15].

URL karakter uzunluğu

Web sayfasının URL ile ilgili özelliklerindendir. Özellikle oltalama sitelerinde ya da çok sayıda yönlendirme ve dış bağlantı içeren web sayfalarında URL içinde yer alan karakter sayıları fazladır [10].

URL içinde özel kelimeler bulunması

Web sayfasının URL ile ilgili özelliklerindendir. Çoğu oltalama sitelerinin URL'leri içinde yer alan "account, secure, payment, banking" gibi özel kelimelerin yer alması kullanıcıya güvenli bir site izlenimi vererek kullanıcıyı aldatma amacı taşımaktadır [15]. Şekil 4.7'de veri seti içinde yer alan ve özel karakterler içiren zararlı bir web sayfası gösterilmiştir.

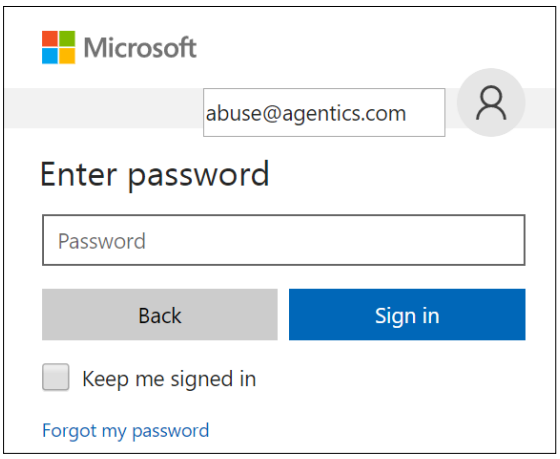
Şekil 4.7. URL içinde özel kelimelerin yer alması

URL içindeki "@" sayısı

Web sayfasının URL ile ilgili özelliklerindendir. URL içinde yer alan "@" işareti genellikle içinde bir e-posta adresi barındırdığını gösterir (Şekil 4.8).

Şekil 4.8. URL içinde "@" karakteri bulunması

URL içinde yer alan e-posta adresi ile kullanıcı sahte bir e-posta yazma ekranına yönlendirilir (Şekil 4.9). Bu yaklaşım oltalama sitelerinde sıklıkla görülmektedir.



Şekil 4.9. Sahte e-posta ekranı

Whois bilgileri sayısı

Web sayfasının popülerlik ile ilgili özelliklerindendir. Whois bilgileri, bir web sayfasının alan adı kaydedilirken kullanılan alan adının hak sahibinin; adı, soyadı, e-posta adresi, telefonu, adresi, alan adının; kayıt tarihi, güncelleme tarihi, bitiş tarihi, alan adının bitmesine kalan gün, sunulduğu firma, kötüye kullanımda bildirilmesi gereken e-posta adresi ve telefonu, sunulduğu alan adı sunucuları bilgilerini içerir. Zararlı web sayfalarının Whois bilgilerinin bu kadar detaylı olarak kaydedilmesi beklenmez. Dolayısıyla zararlı web sayfalarında Whois bilgileri sayısının az olduğu ya da bu özelliğin kullanımının diğer özelliklerle birlikte web sayfasının güvenilirliği hakkında fikir verebileceği düşünülmüştür [18].

Alexa global sıralaması

Web sayfasının popülerlik ile ilgili özelliklerindendir. Alexa, dünyadaki tüm web sitelerinin trafikleri ile ilgili bilgileri toplayan bir internet şirketidir. Alexa'da web siteleri trafiklerine göre popülerlikleri sıralanır. Zararlı web sayfalarının trafiğinin az olacağı düşünülürse, sıralamasının yüksek olması gerektiği düşünülmektedir. Çıkan verilere bakıldığında zararlı web sayfalarının çoğunda Alexa üzerinde sıralama bilgisini hesaplayabilecek bir veri bulunamadığı görülmüştür. Ayrıca düşünüldüğünün aksine zararsız olduğu bilinen web sayfalarının sıralamalarının da yüksek olabildiği sonucuna ulaşılmıştır. Bu durumda, karar vermede diğer özellikler gibi bu özelliğin de tek başına yeterli olmayacağı düşünülmüştür.

Alan adı sunucusu sayısı

Web sayfasının popülerlik ile ilgili özelliklerindendir. Eğer web sayfası trafiği yoğun olan çok popüler bir web sayfası değil ise alan adı sunucusu sayısının fazla olması şüpheli olarak düşünülebilir [17].

İncelenen çalışmalarda bahsedilen bu sınıflara ait özelliklerden farklı sayılarda kullanılmıştır. Çizelge 4.4'te web sayfalarını değerlendirmek için kullanılan özellikler açısından, bu alandaki diğer ilgili çalışmalarla önerilen yaklaşım karşılaştırılmaktadır.

Çizelge 4.4. Üç farklı özellik sınıfına bölünmüş özelliklerin karşılaştırılması

Özellikler	Özellik Sayıları					
	Önerilen	[7]	[10]	[15]	[17]	[18]
HTML	6	9	10	13	5	19
JavaScript	7	9	12	0	7	25
URL ve Popülerlik	5	0	8	8	18	33
Diğer	2	4	-	-	-	-
Toplam	20	22	30	21	30	77

Çizelge 4.4'te kullanılan özellik sayıları ve Çizelge 4.3'te doğruluk oranlarına bakarak özellik sayısının doğrudan başarı oranını etkilemediği sonucuna ulaşabiliriz. Fakat web sayfasının zararlı olup olmadığı ile ilgili bilgi veren önemli özelliklerin belirlenmesi karar verme mekanizmasını doğrudan etkiler. Bunlara ek olarak veri setinin diğer özellikleri aşağıda verilmiştir.

- İşlenen yaklaşık 15 000 URL'den erişilemeyenler, bilgi toplanamayanlar ve yinelenenler çıkarıldığında 3302 adet örnek kalmıştır. Bu 3302 adet örneğin, 1728 tanesi zararlı, geri kalan 1574 tanesi zararsız web sayfalarını içerir.
- Veri setindeki örnekler zararlı (1), zararsız (0) olmak üzere iki sınıflıdır.
- Veri seti örnek web sayfalarından çıkarılan gerçek verilerden oluşmaktadır.
- Eksik veri bulunmamaktadır.
- Yinelenen değerler kaldırılmıştır.

Tüm örnek veriler için yukarıda bahsedilen tüm öznitelik verileri sağlanmıştır.

Veri setleri oluşturulurken farklı özelliklerin ve örnek sayılarının doğruluk oranına olan etkilerinin tespiti için farklı veri setleri ile testler yapılmıştır. Elde edilen sonuçlara göre veri setlerinde de birtakım değişiklikler yapılmıştır. Örneğin, "Google PageRank" değeri Google firmasının PageRank güncellemesini kaldırmasıyla geçerliliğini kaybettiği için başlangıçta kullanılıyor olsa da sonrasında özellik setinden kaldırılmıştır. Çıkarılan fakat kullanılmayan özellikler arasında URL içinde IP adresi bulunması, URL içindeki "_", "/", "-" karakteri sayıları, *hidden* elementi sayısı, *link* fonksiyonu sayısı gibi özellikler bulunmaktadır.

Özellik çıkarma yazılımının çıktılarının (çıkarılan özelliklerin) yer aldığı örnek veri seti EK-1'de sunulmuştur.

Oluşturulan bu veri setinin benzer çalışmalarda kullanılabilmesi ve geliştirilmesi için web ortamında yayınlanması planlanmaktadır.

4.4. Normalizasyon Modülü

Elde edilen özellik verileri belirli bir aralığa dağılmış olup birbirinden çok farklı değerleri kapsamaktadır. Birbirinden çok farklı olan bu değerleri belirli bir aralığa indirgemek ve hesaplamayı kolaylaştırmak amacıyla her bir özellik için normalizasyon işlemi gerçekleştirilmiştir. Literatürde çok sayıda normalizasyon yöntemi yer almaktadır. Bu çalışmada kullanılmak üzere min-max normalizasyon yöntemi uygulanmıştır. Bu yöntemde veriler içindeki en büyük ve en küçük değerler belirlenerek 0-1 aralığında göstermek amaçlanmaktadır. İşlem için kullanılan eşitlik Eş. 4.1'de ifade edilmiştir [60].

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

Burada X_{min} veriler içindeki en küçük değeri, X_{max} en büyük değeri, X_{normal} ise 0-1 arasına normalleştirilen değeri ifade etmektedir. Bu işlem her bir özellik için yapılmıştır. Böylece veri setindeki tüm değerler 0-1 arasında olacak şekilde indirgenerek genelleştirilmiş bir yaklaşım kullanılmıştır (EK-2).

4.5. Sınıflandırma Modülü

Sınıflandırma modülü, normalleştirilen veriler doğrultusunda bir sınıflandırma modeli oluşturma işlemini içerir. Bu sınıflandırma modeli, yukarıda bahsedilen makine öğrenmesi yöntemleri kullanılarak oluşturulmuştur. Oluşturulan bu model daha sonra analiz modülünde test edilmektedir.

4.6. Analiz Modülü

Bu modül, oluşturulan sınıflandırma modüllerinin doğruluk oranlarını test etme ve sonuçlarını karşılaştırarak en başarılı modeli belirleme adımlarını içerir. Yapılan testler sırasıyla KNN, DVM sınıflandırıcıları ile YSA modelini kapsamaktadır.

Veri setindeki tüm örneklerin hem test hem de eğitim verisi olarak kullanılmasını sağlayan böylece performansı artırmayı amaçlayan, k-kat çapraz doğrulama yönteminin başarı oranını nasıl etkilediğini görmek için testlerde kullanılmasının uygun olacağı düşünülmüştür.

K-kat çapraz doğrulama yönteminde, veri seti k adet parçaya bölünür. Böylece k tane test yapılır. Her seferinde k adet parçadan bir tanesi test geri kalan k-1 tanesi de eğitim verisi olarak kullanılır [61].

Burada amaç veri setini verimli kullanmaktır. Sınıflandırma modeli, k kere oluşturulup denendiği için modelin başarı oranının hesaplanması zaman alır. Uygulamada k-kat çapraz doğrulama için k değeri 10 olarak belirlenmiştir.

Yapılan testlerdeki başarı oranının belirlenmesinde yalnızca doğru sınıflandırılan örnek sayısı değil, sınıflandırma sonucunda tahmin edilen sınıflar ile başlangıçta ayrılan test verisi sınıfları karşılaştırılarak doğru pozitif, yanlış pozitif, doğru negatif ve yanlış negatif oranları gibi performans belirleyici değerlerde kullanılmıştır.

Doğru pozitif (DP) oranı, gerçekte pozitif olup, sınıflandırıcı tarafından da pozitif olarak sınıflandırılmış örnek sayısının toplam test örneği sayısına oranıdır. Yanlış pozitif (YP) oranı, gerçekte negatif olup, sınıflandırıcı tarafından pozitif olarak sınıflandırılan örnek

sayısının toplam test örneği sayısına oranıdır. Doğru negatif (DN) oranı, gerçekte negatif olup, sınıflandırıcı tarafından da negatif olarak sınıflandırılan örnek sayısının toplam test örneği sayısına oranıdır. Yanlış negatif (YN) oranı, gerçekte pozitif olup, sınıflandırıcı tarafından negatif olarak sınıflandırılan örnek sayısının toplam test örneği sayısına oranıdır. Bu çalışmada pozitif sınıflandırma, bir web sayfasının zararlı olarak sınıflandırılmasını, negatif sınıflandırma, web sayfasının zararsız olarak sınıflandırılmasını ifade etmektedir. Tüm bu değerler hesaplanarak karışıklık matrisi adı verilen tabloya yerleştirilmiştir (Çizelge 4.5).

Çizelge 4.5. Karışıklık matrisi

		Tahmini Sınıf	
		(+)	(-)
Gerçek Sınıf	(+)	Doğru Pozitif	Yanlış Negatif
	(-)	Yanlış Pozitif	Doğru Negatif

Modelleri oluşturma ve test etme işlemleri MATLAB R2015a platformu üzerinde gerçekleştirilmiş olup EK-3'te gösterilen kod bloğu kullanılmıştır.

4.6.4. KNN ile yapılan testler

Önerilen yaklaşımın belirlenmesiyle ilgili yapılan ilk testte KNN sınıflandırma algoritmasının Öklid uzaklığı kullanılarak farklı k değerleri için doğruluk oranları elde edilmiştir. En uygun k değeri deneme yanılma yöntemiyle bulunmuştur. Buna göre tek sayılardan seçilen üç farklı k değeri için yapılan testlerde başarı oranının en yüksek olduğu k değerinin 3 olduğu belirlenmiştir. Üç farklı k değeri için elde edilen sonuçlar Çizelge 4.6'da gösterilmiştir.

Çizelge 4.6. Üç farklı k değeri için elde edilen sonuçlar

k Değeri	Çalışma zamanı (sn)	Doğruluk Oranı (%)
3	0,27	98,42
5	0,26	98,18
7	0,29	97,84

Buna göre k değeri arttıkça doğruluk oranının düştüğü gözlenmiştir. Çalışma zamanları arasında gözle görülür bir farka rastlanılmamıştır.
Ulaşılan en yüksek doğruluk oranı için oluşturulan karmaşıklık matrisinde değerler yerine koyulduğunda Çizelge 4.7 elde edilmiştir.

Çizelge 4.7. KNN en yüksek doğruluk oranı için elde edilen karışıklık matrisi (%)

		Tahmini Sınıf	
		(+)	(-)
Gerçek Sınıf	(+)	Doğru Pozitif 0,46	Yanlış Negatif 0,008
	(-)	Yanlış Pozitif 0,008	Doğru Negatif 0,51

Bir sınıflandırma işleminde yanlış pozitif oranının düşük olması önemli bir husustur. Bu çalışma için aslında zararsız olan bir web sayfasının zararlı olarak sınıflandırılması anlamına gelir. Yanlış alarm olarak düşünülebilir. Çıkarılan %0,008'lik yanlış pozitif oranı bu sınıflandırma modelinden elde edilen en düşük orandır.

4.6.5. Destek vektör makineleri ile yapılan testler

Önerilen yaklaşımın belirlenmesiyle ilgili yapılan ikinci testte Destek Vektör Makineleri sınıflandırıcısı modeli test edilmiştir. Elde edilen sonuçlar Çizelge 4.8'de gösterilmiştir.

Çizelge 4.8. DVM sınıflandırıcısı testi sonuçları

Çalışma zamanı (sn)	Doğruluk Oranı (%)
4,2	98,57

Buna göre elde edilen doğruluk oranı KNN sınıflandırıcısında daha yüksek olmasına rağmen çalışma zamanı olarak daha uzun sürmüştür. Daha fazla olması durumunda işlem karmaşıklığının artacağı düşünülürse bu sürenin daha fazla veri olması durumunda daha da uzun süreceği öngörülmektedir.

Sınıflandırıcı performansının daha rahat görülmesi amacıyla yanlış pozitif, doğru pozitif, yanlış negatif ve doğru negatif oranları çıkarılmıştır. Bunun için test için ayrılan verilerinin sınıfları ile sınıflandırıcının tahmin ettiği sınıflar karşılaştırılmıştır. Buna göre elde edilen sonuçlar Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9. DVM sınıflandırıcısı ile elde edilen karışıklık matrisi (%)

		Tahmini Sınıf	
		(+)	(-)
Gerçek Sınıf	(+)	Doğru Pozitif 0,46	Yanlış Negatif 0,007
	(-)	Yanlış Pozitif 0,006	Doğru Negatif 0,51

Buna göre sınıflandırıcı performansında önemli rolü olan yanlış pozitif oranı %0,006 olarak hesaplanmıştır. Bu değer ilk testte kullanılan KNN algoritmasının yanlış pozitif oranından daha düşüktür. Bu durumda sınıflandırıcının hem doğruluk oranı KNN sınıflandırıcısından yüksek hem de yanlış alarm oranı daha düşük ölçülmüştür.

Yapılan iki testte en yüksek doğruluk ve en düşük yanlış alarm oranının DVM sınıflandırıcısı ile sağlandığı sonucuna ulaşılmıştır. Fakat çalışma performansı olarak KNN sınıflandırıcısı daha hızlı sonuç vermiştir. Bu durumda hangi sınıflandırıcının seçileceği gereksinimler doğrultusunda belirlenmelidir.

4.6.6. Yapay sinir ağlarıyla yapılan testler

Önerilen yaklaşımın belirlenmesiyle ilgili yapılan üçüncü testte Yapay Sinir Ağları algoritması kullanılmıştır. MATLAB ortamının sağladığı Yapay Sinir Ağı Aracı (Neural Network Tool) kullanılarak uygulanan testlerde Çok Katmanlı Algılayıcı Yapay Sinir Ağı yapısı seçilmiştir. 5 farklı öğrenme algoritması ile YSA sınıflandırma performansı test edilmiştir.

YSA performansında etkili olan nöron sayısı, geçiş fonksiyonları, eğitim modeli parametreleri değiştirilerek en başarılı sonucu sağlayan parametre kombinasyonuna deneme yanılma yöntemiyle ulaşılmaya çalışılmıştır (Çizelge 4.10).

Çizelge 4.10. YSA parametrelerinin karşılaştırılması

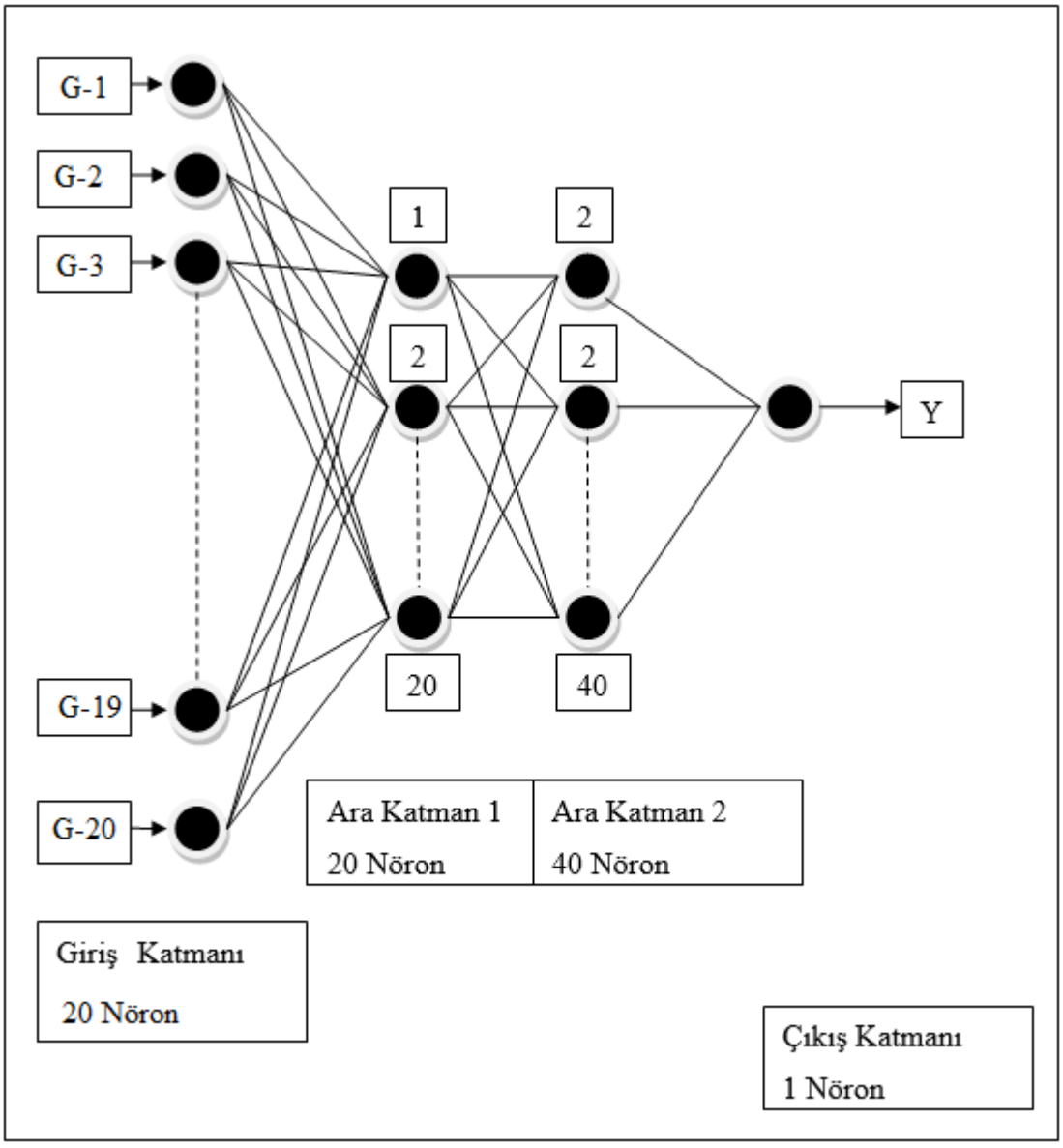
YSA	AKS	HBKNS	Geçiş Fonksiyonları	Eğitim Modeli	Ortalama Hata Oranı	Epok	Doğruluk Oranı (%)
1	2	20,40,1	tansig, logsig	LM	$1{,}38 \times 10^{-10}$	576	98,14
2	2	35,50,1	logsig, tansig	LM	$1{,}14 \times 10^{-7}$	343	98
3	2	50,30,1	tansig, logsig	GD	0,0566	2308	95
4	2	20,40,1	tansig, logsig	GD	0,0398	3526	97,57
5	3	30,40, 20, 1	logsig, logsig, tansig	GDA	0,0448	2160	97
6	2	20,40,1	logsig, tansig	GDA	0,0389	9962	97,85
7	3	20,40, 20,1	tansig, tansig,tansig	GDM	0,0423	3448	96,71
8	2	20,40,1	logsig, tansig	GDM	0,0596	994	95,57
9	2	20,40,1	tansig, logsig	GDX	0,0311	1037	98,71
10	3	50,30,20,1	tansig, logsig	GDX	0,0167	3280	98,57

AKS: Ara Katman Sayısı, HBKNS: Her Bir Katmandaki Nöron Sayısı

YSA modelinin eğitilmesi için yapılan testler sonucunda 1. ara katmanda tanjant hiperbolik transfer fonksiyonuna sahip 20 nöronlu yapı ile 2. ara katmanda sigmoid transfer fonksiyonuna sahip 40 nöronlu yapı kullanılmıştır. Sistem sıfır hata oranına ulaşmak için 1037 epok (iterasyon sayısı) seviyesine kadar eğitime alınmıştır. Epok sayısı arttıkça hata oranının sıfıra yaklaştığı görülmüştür. Ortalama hata oranı 0,0311 ulaştıktan sonra eğitim

işlemi sonlandırılmış ve tespit edilen bu parametrelerle sistemin eğitilmesi gerçekleştirilmiştir.

Sistemin eğitilmesinden sonra ağ üzerinde yapılan testlerde elde edilen sonuçlar, olması gereken değerlerle karşılaştırılarak en yüksek doğruluk oranı hesaplanmıştır. Buna göre Şekil 4.10'daki gibi bir ağ yapısı ile ara katmanlarda sırasıyla tanjant hiperbolik ve sigmoid transfer fonksiyonları ile GDX öğrenme algoritması kullandığında en yüksek doğruluk oranı olan %98,71 elde edilmiştir. En yüksek doğruluk oranı için hesaplanan yanlış pozitif oranı %0, 0114'tür. Doğruluk oranı makine öğrenmesi yöntemlerinden daha yüksek olsa da yanlış pozitif oranı daha yüksektir.



Şekil 4.10. Kullanılan YSA yapısı

Seçilen YSA için elde edilen en yüksek doğruluk için yapılan hesaplanan karışıklık matrisi Çizelge 4.11'de gösterilmiştir.

Çizelge 4.11. YSA ile elde edilen karışıklık matrisi (%)

		Tahmini Sınıf	
		(+)	(-)
Gerçek Sınıf	(+)	Doğru Pozitif 0,53	Yanlış Negatif 0,001
	(-)	Yanlış Pozitif 0,011	Doğru Negatif 0,43

Testlerde kullanılan makine öğrenmesi teknikleri ve YSA ile elde edilen en iyi sonuçlar ile benzer yöntemlerin kullanıldığı çalışmaların sonuçlarının karşılaştırıldığı çizelge Çizelge 4.12'de gösterilmiştir.

Çizelge 4.12. Sonuçların karşılaştırılması

	Doğruluk Oranı (%)	Yanlış Pozitif Oranı (%)
KNN	98,42	0,008
DVM	98,57	0,006
YSA	98,71	0,011
[7]	97,9	0,014
[8]	96,14	0,21
[10]	95,4	4,5
[14]	98,04	0,015
[15]	97,53	0,2
[17]	98	0,011
[34]	96,11	0,01
[31]	82,9	0,7

Çizelge 4.12'den de anlaşılacağı gibi bu çalışmada;

- Zararlı web sayfası tespiti için kullanılan makine öğrenmesi yaklaşımlarının hepsinin %98'in üzerinde doğruluk ve %0,01'in altında yanlış pozitif oranı ile çalıştığı,
- Bu durumda incelenen çalışmalardan daha yüksek doğruluk oranları sağlandığı sonuçlarına ulaşılmıştır.

5. SONUÇ VE ÖNERİLER

Bu çalışmada, zararlı web sayfaları tespit etmek için yeni bir sistem önerilmiştir. Bu kapsamda, önerilen sistemin belirlenmesinde öncelikle literatürde yapılan çalışmalar ve güncel yöntemler incelenmiştir. İncelenen çalışmalar doğrultusunda literatürde;

- Zararlı web sayfalarını tespit eden geleneksel yöntemlerin çoğunun imza tabanlı yaklaşımlar kullanması ve bu tekniklerin bilinmeyen zararlı web sayfalarının tespitinde yetersiz kalması,
- Çalışmalarda yanlış pozitif oranlarının yüksek olması,
- Web sayfası içeriklerinin değişmesi ve web sayfası sayılarının artması ile içeriklerin işlenmesinin uzun sürmesi,
- Tespit yöntemlerinin saldırı tipine göre değişebilmesi, bazı tespit yöntemlerinin yalnızca tek bir zararlı web sayfası türünü tespit edebilmesi (yalnızca oltalama web sayfaları gibi),
- Örnek veri seti kaynaklarının kısıtlı olması ve standart bir veri seti bulunmaması problemleri belirlenmiştir.

Bu problemlerin ortadan kaldırılması amacıyla önerilen sistemde öncelikle zararlı ve zararsız web sayfalarının özelliklerinin yer aldığı bir veri seti oluşturulmuştur. Bu özellikler, temel olarak, URL ve popülerlik özellikleri, JavaScript özellikleri ve HTML özellikleri olarak üç ana başlıkta toplanmıştır. URL ile ilgili özellikler, URL'nin sözcüksel özelliklerini, popülerlik ile ilgili özellikler web sayfasının popülerliği ile ilgili bilgileri içerir. URL içindeki nokta sayısı, Alexa sıralaması örnek verilebilir. JavaScript özellikleri, sayfa içinde yer alan ve kötü niyetli kullanımı mümkün kılan JavaScript fonksiyonlarını ve değişkenlerini içerir. HTML özellikleri, web sayfasının kaynak kodlarında yine kullanımı şüpheli olabilecek etiketleri ve değerleri içerir. Tüm bu özellikler kullanılarak Destek Vektör Makineleri, K-En Yakın Komşu ve Yapay Sinir Ağları sınıflandırıcıları ile sınıflandırma modelleri oluşturularak testler yapılmıştır.

Bu çalışmada, incelenen yayınlarda kullanılan veya bu doğrultuda ilk kez önerilen özelliklerden oluşan hibrit bir özellik seti kullanılmıştır. Özellik seti 20 farklı özellikle literatürdeki çalışmalarda kullanılan özelliklerden daha az olup, özellik sayısının az olması

web sayfalarının gerçek zamanlı işlem karmaşıklığını azaltan önemli bir parametredir. Oluşturulan bu veri seti önerilen sistemin en önemli adımı olup bu adımda yapılan tüm değişiklikler doğruluk oranını doğrudan etkilemiştir. Dolayısıyla farklı özelliklerin ve kombinasyonlarının yer aldığı farklı veri setleri oluşturularak çok sayıda test gerçekleştirilmiştir.

Veri seti ile farklı makine öğrenmesi teknikleri kullanılarak yapılan testlerde en yüksek doğruluk oranı YSA kullanılarak %98,71 olarak hesaplanmıştır. Bu oranı etkileyen farklı parametreler olmakla beraber bu parametreler deneme yanılma yöntemiyle test edilerek bulunmuştur.

Önerilen sistemin motivasyonu olan başarısını etkileyen önemli etkenlerden biri olarak belirlenen en düşük yanlış pozitif oranı DVM kullanılan testlerde %0,006 olarak ölçülmüştür. Bu oran literatürde incelenen çalışmalardan oldukça düşüktür. İncelenen çalışmalarda bu orana en yakın değer %0,011'dir [17].

Bu çalışmanın sonucunda elde edilen çıktılar ve sonuçlar aşağıda maddeler halinde verilmiştir.

- Literatürdeki çalışmalardan daha yüksek doğruluk ve daha düşük yanlış pozitif oranı sağlanmıştır.
- Başarı oranı en az özellikle elde edilmiştir.
- Özgün bir özellik seti oluşturulmuştur.
- Zararlı web sayfası tespitinde kullanılabilecek yeni özellikler önerilmiştir.
- Farklı tiplerde zararlı web sayfaları tespit edilmiştir.

Ayrıca bu çalışmanın;

- Literatüre, makine öğrenmesinde kullanılabilecek farklı zararlı web sayfası özellikleri kombinasyonları ve özellik seti ile farklı bir bakış açısı kazandırabileceği,
- Zararlı web sayfalarının arşivlendiği ve anti virüs yazılımları gibi bu sayfalara karşı önlemler alan uyarıcı uygulamaların kullandığı kara listeleri oluşturmak amacıyla kullanılabileceği,

- Özgün veri setinin karşılaştırmalı veri kümeleri çalışmalarında kullanılabileceği,
- Çok miktarda URL'den veri elde etmek için çok sayıda URL işleme ve parçalama kütüphanesine sahip Python programlama dilinin kullanılmasının performanslı ve hızlı olmasının benzer problemlerinin çözümünde etkili olabileceği değerlendirilmektedir.

Önerilen çalışmada karşılaşılan en büyük zorluk veri seti için zararlı web sayfaları toplanması olmuştur. Kara listelerde yer alan web sayfalarının çoğu ağ geçidi kısıtlamalarından veya zararlı web sayfalarının erişime kapanmasından dolayı erişilebilir olmadığından URL özellikleri çıkarılsa da HTML ve JavaScript özelliklerinin toplanması mümkün olmamıştır. Eksik veri olmaması için erişilebilir zararlı web sayfaları bulunmaya çalışılmıştır. Web sayfalarına manuel olarak ulaşılması çok sayıda web sayfasına erişimi zorlaştırmıştır. Veri seti oluşturmada karşılaşılan zorluklar ve literatürdeki ortak veri seti problemi oluşturulan veri setinin benzer çalışmalarda kullanılabilmesi için web ortamında yayımlanması fikrini ortaya çıkarmıştır.

Ayrıca bu çalışmada en iyi sonuca ulaşabilmek için Temel Bileşen Analizi, Fisher Ayırt Edici Oranı, Ayrık Kosinüs Dönüşümü gibi öznitelik seçme yöntemleri üzerine çalışmalar yapılmış fakat iyi sonuçlar alınamamıştır. Gelecek çalışmalarda farklı ve yeni metotlar kullanılarak başarı oranını artırmaya yönelik yeni çalışmalar yapılması mümkündür.

KAYNAKLAR

1. İnternet: Stevens, John. Internet Stats & Facts for 2017. *hostingfacts*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fhostingfacts.com%2Finternet-facts-stats-2016%2F&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

2. İnternet: Total number of Websites. *internet live stats*. URL:http://www.webcitation.org/query?url=http%3A%2F%2Fwww.internetlivestats.com%2Ftotal-number-of-websites%2F&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

3. İnternet: Safe Browsing: malware and phishing. *Google*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.google.com%2Ftransparencyreport%2Fsafebrowsing%2F&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

4. Wang, Z., Feng, X., Niu, Y., and Su, J. (2017). TSMWD: A High-speed Malicious Web Page Detection System Based on Two-Step Classifiers. *Networking and Network Applications*, 170-175.

5. İnternet: Matt Bromiley. Threat Intelligence: What It Is, and How to Use It Effectively. *SANS Institute*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.google.com%2Ftransparencyreport%2Fsafebrowsing%2F&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

6. Kazemian, H.B., Ahmed, S. (2015). Comparisons of machine learning techniques for detecting malicious webpages. *Expert Systems with Applications*, 42(3), 1166-1177.

7. Phakoontod, C., Limthanmaphon, B. (2012). Malicious Web Page Detection Based on Feature Classification, *Computing and Convergence Technology*, 66-71.

8. Hou, Y.T., Chang, Y., Chen, T., Laih, C.S., and Chen, C.M. (2010). Malicious web content detection by machine learning, *Expert Systems with Applications*, 37(1), 55-60.

9. Kai, F., Jianhuan, S., and Hao, C. (2011). WSProxy: Detecting and Fighting Malicious Websites. *Business Computing and Global Informatization*, 649-652.

10. Yue, T., Sun, J., and Chen, H. (2013). Fine-Grained Mining and Classification of Malicious Web Pages. *Digital Manufacturing & Automation*, 616-619.

11. Eshete, B., Villafiorita, A., and Weldemariam, K. (2011). Malicious Website Detection: *Effectiveness and Efficiency Issues*, *First SysSec Workshop*, 123-126.

12. Tao, J., Jun, L., and Rong, X. (2012). Research on Malicious Links Detection System Based On Script Text Analysis. *Advanced Communication Technology*, 439-442.

13. Komiya, R., Paik, I., and Hisada, M. (2011). Classification of malicious web code by machine learning. *Awareness Science and Technology*,406-411.

14. Choi, J., Kim, H. (2011). Efficient Malicious Code Detection Using N-Gram Analysis and SVM. *Network-Based Information Systems*, 618-621.

15. Liu, H., Pan, X., and Qu, Z. (2009). Learning based Malicious Web Sites Detection using Suspicious URLs. *Software Engineering*,1-3.

16. Le, V.L., Welch, I., Gao, X., and Komisarczuk, P. (2011). Identification of Potential Malicious Web Pages. *Australasian Information Security Conference*, 33-40.

17. Choi, H., Zhu, B.B., and Lee, H. (2011). Detecting Malicious Web Links and Identifying Their Attack Types. *USENIX Conference on Web Application Development*, 1-12.

18. Canali, D., Cova, M., Vigna, G., and Kruegel, C. (2011). Prophiler: a fast filter for the large-scale detection of malicious web pages. *World wide web*, 197-206.

19. Lee, S., Cho, H., Kim, B., Shin, Y., and Lee, T. (2015). A Methodology for Calculating the Impact of Malicious Resources by PageRank. *Information Science and Security*, 1-2.

20. Invernizzi, L., Comparetti, P.M. (2012). EVILSEED: A Guided Approach to Finding Malicious Web Pages. *Security and Privacy*, 428-442.

21. Chiba, D., Tobe, K., Mori, T., and Goto, S. (2012). Detecting Malicious Websites by Learning IP Address Features. *Applications and the Internet*, 29-39.

22. Stringhini, G., Kruegel, C., and Vigna, G. (2013). Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. *Computer and Communications Security*, 133-144.

23. Sha, H., Liu, Q., Zhou, Z., and Zheng, C. (2014). GuidedTracker: Track the Victims with Access Logs to Finding Malicious Web Pages. *Communication and Information System Security*, 564-569.

24. Manek, A.S., V, S., Shenoy, P.D., Mohan, M.C., K R, V., and Patnaik, L. (2014). DeMalFier:Detection of Malicious Web Pages using an Effective ClassiFier. *Data Science & Engineering*, 83-88.

25. Eshete, B., Villafiorita, A., Weldemariam, K., and Zulkernine, M. (2013). EINSPECT: Evolution-Guided Analysis andDetection of Malicious Web Pages. *Computer Software and Applications*, 375-380.

26. Shibahara, T., Yagi, T., Akiyama, M., Takata, Y., and Yada, T. (2015). Detecting Malicious Web Pages based onStructural Similarity of Redirection Chains, *Computer and Communications Security*, 1671-1673.

27. Nunan, A.E., Souto, E., Santos, E. M., and Feitosa, E. (2012). Automatic Classification of Cross-Site Scripting in Web Pages Using Document-based and URL-based Features. *Computers and Communications*, 702-707.

28. Jodavi, M., Abadi, M., and Parhizkar, E. (2015). DbDHunter: An Ensemble-based Anomaly Detection Approach to Detect Drive-by Download Attacks. *Computer and Knowledge Engineering*, 273-278.

29. Liang, S., Ma, Y., Huang, Y., Guo, J., and Jia, C. (2016). The Scheme of Detecting Encoded Malicious Web Pages Based on Information Entropy. *Innovative Mobile and Internet Services in Ubiquitous Computing*, 310-312.

30. Dewan, P., Kumaraguru, P. (2015). Towards Automatic Real Time Identification of Malicious Posts on Facebook. *Privacy, Security and Trust*, 85-92.

31. Gabriel, A.D., Gavrilut, D.T. (2016). Detecting malicious URLs. A semi-supervised machine learning system approach. *Symbolic and Numeric Algorithms for Scientific Computing*, 233-239.

32. Thakur, S., Meenakshi, E., and Priya, A. (2017). Detection of Malicious Urls in Big Data Using Ripper Algorithm. *Recent Trends in Electronics Information & Communication Technology*, 1296-1301.

33. Kumar, R., Zhang, X., Tariq, H.A., and Khan, R.U. (2017). Malicious Url Detection Using Multi-Layer Filtering Model. *Wavelet Active Media Technology and Information Processing*, 97-100.

34. Desai, A., Jatakia, J., Naik, R., and Raul, N. (2017). Malicious Web Content Detection Using Machine Learning. *Recent Trends in Electronics Information & Communication Technology*, 1432-1436.

35. Vanhoenshoven, F., Napoles, G., Falcon, R., Vanhoof, K., and Köppen, M. (2016). Detecting Malicious URLs using Machine Learning Techniques. *Computational Intelligence*, 1-8.

36. Sağıroğlu Ş., Canbek G. (2007). *Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri* (Birinci Baskı). Ankara: Grafiker, 170-215.

37. İnternet: 9 Types Of Computer Viruses That You Should Know About – And How To Avoid Them. *youngupstarts*. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.youngupstarts.com%2F2016%2F04%2F14%2F9-types-of-computer-viruses-that-you-should-know-about-and-how-to-avoid-them%2F&date=2018-04-08, Son Erişim Tarihi: 08.04.2018.

38. İnternet: Computer Viruses and Malware Facts & FAQs. *Kaspersky Lab*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fusa.kaspersky.com%2Fresource-center%2Fthreats%2Fcomputer-viruses-and-malware-facts-and-faqs&date=2018-04-08, Son Erişim Tarihi: 08.04.2018.

39. İnternet: Web Threats. *Kaspersky Lab*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fusa.kaspersky.com%2Finternet-security-center%2Fthreats%2Fweb%23.WKnwpm-LTIU&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

40. İnternet: Milletary, J. Technical Trends in Phishing Attacks. *US-CERT* . URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.us-cert.gov%2Fsites%2Fdefault%2Ffiles%2Fpublications%2Fphishing_trends0511.pdf+&date=2018-03-25, Son Erişim Tarihi: 25.03.2018.

41. Ibrahim, D.R., Hadi, A.H. (2017). Phishing Websites Prediction Using Classification Techniques. *New Trends in Computing Sciences*, 133-137. *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, 557-564.

42. Li, J.H., Wang, S.D. (2017). PhishBox: An Approach for Phishing Validation and Detection.

43. Kikuchi, H., Matsumoto, H. (2015). Automated Detection of Drive-by Download Attack. *Innovative Mobile and Internet Services in Ubiquitous Computing*, 511-515.

44. Sood, A.K., Zeadally, S. (2016). Drive-By Download Attacks: A Comparative Study. *IT Professional*, 18(5), 18-25.

45. Zhang, S., Wang, W., Chen, Z., Gu, H., Liu, J., and Wang, C. (2014). A Web Page Malicious Script Detection System. *Cloud Computing and Intelligence Systems*, 394-399.

46. Bhuyan, M.H, Bhattacharyya, D. K., and Kalita, J. K. (2014). Network Anomaly Detection:Methods, Systems and Tools, *Communications Surveys & Tutorials*, 16(1), 303-336.

47. Le, V.L, Welch, I., Gao, X., and Komisarczuk, P. (2012). A Novel Scoring Model to Detect Potential Malicious Web Pages. *Trust, Security and Privacy in Computing and Communications*, 254-263.

48. Ikinci, A., Holz, T., and Freiling, F. (2008). "Monkey-Spider: Detecting Malicious Web Sites". *Sicherheit*, 1-15.

49. Eren, Ö. (2008). *Automated Classification Of Allergen Proteins*. Yüksek Lisans Tezi, Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 18-20 .

50. Guo, G., Li, S., and Chan, K. (2001). Support vector machines for face recognition. *Image and Vision Computing*, 19(9-10), 631-638.

51. Heisele, B., Ho, P., and Poggio, T. (2001). Face Recognition with Support Vector Machines: Global versus Component-based Approach. *Computer Vision*, 688-694.

52. İnternet: Yalçın, H. Yapay Sinir Ağları. *İTÜ*. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fweb.itu.edu.tr%2Fhulyayalcin%2FSignal_Processing_Books%2FYapay_Sinir_Aglari__Yildiz.pdf&date=2018-03-25, Son Erişim Tarihi: 25.03.2018.

53. Öztemel, E. (2012). *Yapay Sinir Ağları* (Üçüncü Baskı), Ankara: Papatya Yayıncılık, 29-57.

54. İnternet: Tavsanoglu, V. Cellular Neural Networks. *YTÜ*. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.yildiz.edu.tr%2F%7Etavsanav%2Flecturesnotes%2FHSA_Not.pdf&date=2018-03-25, Son Erişim Tarihi: 25.03.2018.

55. Sağıroğlu, Ş., Beşdok, E., Erler, M., (2003). *Mühendislikte Yapay Zeka Uygulamaları-1:Yapay Sinir Ağları* (Birinci Baskı), Kayseri: Ufuk Kitabevi, 10-100.

56. İnternet: Python. *Python Software Foundation*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.python.org&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

57. İnternet: beautifulsoup4 4.6.0. *Python Software Foundation*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fpypi.python.org%2Fpypi%2Fbeautifulsoup4&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

58. İnternet: tldextract 2.2.0. *Python Software Foundation*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fpypi.python.org%2Fpypi%2Ftldextract&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

59. İnternet: python-whois 0.6.8. Python Software Foundation. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fpypi.python.org%2Fpypi%2Fpython-whois&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

60. İnternet: Takaki, J.;Petersen, T.; Ericson, G. Normalize Data. *Microsoft*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Fmachine-learning%2Fstudio-module-reference%2Fnormalize-data&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

61. İnternet: Schneider, J. Cross Validation. *Carnegie Melon University*. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.cs.cmu.edu%2F%7Eschneide%2Ftut5%2Fnode42.html&date=2018-03-15, Son Erişim Tarihi: 15.03.2018.

EKLER

EK-1. Çıkarılan özellikler

	Script	embed	iframe	display:none	window.loc	document.cook	src	nokta	url uzunlu	"@" sayısı	eval	doc.write	escape	exec	window.o	whois	alexa	nameserv	href	özelkelime	sınıf
1	11	0	1	3	0	0	0	8	1	24	0	0	0	0	0	0	11 -	2	11	0	1
2	0	0	0	0	0	0	0	0	2	37	0	0	0	0	0	0	10 -	4	0	0	1
3	0	0	0	0	0	0	0	0	2	32	0	0	0	0	0	0	11 -	2	1	0	1
4	0	0	0	0	0	0	0	1	1	29	0	0	0	0	0	0	11 -	2	0	0	1
5	0	0	0	0	0	0	0	0	1	23	0	0	0	0	0	0	11 -	2	0	0	1
6	22	0	0	3	0	0	0	4	2	45	0	0	0	0	0	1	11 1	4	12	0	1
7	1	0	0	0	0	0	0	0	3	174	1	0	0	0	0	0	5 -	2	0	1	1
8	2	0	0	0	1	0	0	0	3	70	1	0	0	0	0	0	7 4,347,728	3	2	0	1
9	0	0	0	0	0	0	0	0	1	146	3	0	0	0	0	0	11 15,610,924	2	0	0	1
10	2	0	0	0	1	0	0	0	4	119	1	0	0	0	0	0	11 3,049,850	2	1	0	1
11	1	0	0	0	0	0	0	0	3	156	1	0	0	0	0	0	11 10,596,545	2	0	1	1
.	5	0	0	1	5	0	0	7	4	80	1	0	0	0	0	0	11 -	2	74	1	1
.	4	0	0	0	0	0	0	10	3	94	1	0	0	0	0	0	10 -	3	1	0	1
.	0	0	0	0	0	0	0	0	5	148	1	0	0	0	0	0	7 -	2	1	1	1
.	1	0	0	0	0	0	0	0	3	161	1	0	0	0	0	0	11 -	2	0	1	1
.	1	0	0	0	0	0	0	8	1	73	1	0	0	0	0	0	9 -	2	8	0	1
.	0	0	0	0	0	0	0	1	2	83	1	0	0	0	0	0	9 -	2	0	0	1
.	2	0	0	0	1	0	0	0	3	71	1	0	0	0	0	0	7 4,347,728	3	2	0	1
.	2	0	0	0	1	0	0	0	4	108	1	0	0	0	0	0	6 10,285,419	2	1	0	1
.	5	0	1	5	0	0	0	7	4	82	1	0	0	0	0	0	11 -	2	74	1	1
.	5	0	1	5	0	0	0	7	7	351	1	0	0	0	0	0	10 -	3	74	0	1
.	5	0	1	5	0	0	0	7	6	317	1	0	0	0	0	0	10 -	3	74	0	1
.	5	0	1	5	0	0	0	7	7	351	1	0	0	0	0	0	10 -	3	74	0	1
1726	25	1	4	0	9	0	0	119	2	20	0	0	0	0	0	6	11 518,567	2	136	0	1
1727	26	1	1	37	4	0	0	47	1	18	0	0	5	0	0	2	11 -	2	130	0	1
1728	25	1	4	0	9	0	0	119	2	20	0	0	0	0	0	6	11 519,141	2	136	0	1
.	0	1	0	0	0	0	0	1	3	28	0	0	0	0	0	0	8 -	2	0	0	0
.	27	1	0	0	0	0	0	46	2	30	0	0	0	1	0	0	11 19,116,298	2	61	0	0
.	3	1	0	0	0	0	0	17	3	33	0	0	0	0	0	0	8 -	2	31	0	0
.	33	1	6	3	0	0	0	67	2	32	0	0	2	1	0	0	11 -	4	70	0	0
.	1	12	0	0	0	0	0	16	3	28	0	0	0	1	0	0	8 -	2	3	0	0
.	12	1	1	0	0	0	0	13	2	31	0	0	2	1	0	0	11 -	4	45	0	0
.	31	8	11	7	0	0	0	69	2	38	0	0	1	0	0	0	11 697,872	2	167	0	0
.	4	1	0	0	0	0	0	15	2	23	0	0	0	0	0	0	11 -	3	22	0	0
.	1	2	0	0	0	0	0	33	2	22	0	0	1	0	0	0	11 8,117,787	2	42	0	0
.	9	1	2	0	1	0	0	23	2	26	0	0	0	0	0	0	11 2,143,485	2	33	0	0
.	3	1	0	0	0	0	0	16	2	33	0	0	0	0	0	0	11 -	2	23	0	0
.	7	1	2	1	0	0	0	11	2	24	0	0	0	0	0	0	11 -	2	15	0	0
.	18	1	1	2	0	0	0	65	2	29	0	0	3	0	0	0	11 5,700,789	4	192	0	0
.	0	1	0	0	0	0	0	1	2	27	0	0	0	0	0	0	11 -	2	1	0	0
.	8	1	1	1	0	0	0	19	3	32	0	0	0	0	0	0	11 91,928	2	21	0	0
.	6	1	1	0	0	0	0	97	2	32	0	0	0	0	0	0	10 3,940,707	3	173	0	0
.	30	5	5	0	0	0	0	75	2	32	0	0	1	1	0	0	11 -	4	370	0	0
.	1	1	0	0	0	0	0	38	2	32	0	0	0	0	0	0	11 -	2	16	0	0
.	16	0	2	1	5	0	0	111	3	25	0	0	2	4	0	1	8 3,967,665	2	139	0	0
.	13	0	0	7	0	0	0	27	2	32	0	0	1	1	1	0	11 1,019,560	2	141	0	0
.	42	0	3	9	0	0	0	57	2	22	0	0	0	0	0	0	10 9,637,795	4	103	0	0
.	0	0	0	0	0	0	0	0	3	45	0	0	0	0	0	0	11 4,572,096	2	2	0	0
3300	10	0	0	0	0	0	0	125	2	23	0	0	0	0	0	1	11 446,452	2	227	0	0
3301	18	0	1	4	0	0	0	14	2	30	0	4	0	0	0	0	10 394,774	2	3652	0	0
3302	59	0	0	14	0	0	0	53	2	23	0	0	0	0	0	0	11 1,145,573	2	97	0	0

Şekil 1.1. Özellikler ve gerçek değerleri

EK-2. Normalleştirilen özellikler

Sayı	embed	iframe	display:no	window.lo	document	src	nokta	url uzunlu	"@" says	eval	doc.write	escape	exec	window.o	whois	alexa	nameserv	href	özelkelim	sınıf
1	5	0	0	0	0	0	0.14286	0.02443	0	0	0	0	0	0.0122	1	0.000000	0.3	0.00281	0	1
2	0	0	0	0	0	0	0	0.00315	0	0	0	0	0	0	1	0.000000	0.1	0	0	1
3	0	0	0	0	0	0	0	0.00315	0	0	0	0	0	0	1	0.000000	0.1	0	0	1
4	0	0	0	0	0	0	0	0.00315	0	0	0	0	0	0	1	0.000000	0.1	0	0	1
5	7	0	0.01911	0	0.01042	0	0.00203	0.42857	0.04098	0	0.02632	0	0	0	0.85714	0.000000	0.1	0.00515	1	1
6	0	0	0	0	0	0	0	0	0.00946	0	0	0	0	0	0	0.000000	0.1	0	0	1
7	3	0	0.01911	0.00297	0.01042	0	0.04917	0.28571	0.01261	0	0	0	0	0	0.57143	0.149890	0.1	0.08525	0	1
8	3	0	0.00637	0.0089	0	0	0.00271	0	0.02206	0	0	0	0	0	1	0.000000	0.1	0.00258	0	1
9	3	0	0	0.00593	0	0	0.00678	0.28571	0.02916	0	0	0	0	0	1	0.000000	0.1	0.01475	1	1
10	0	0	0	0	0	0	0	0	0.00709	0	0	0	0	0	1	0.000000	0.1	0	0	1
11	0	0	0	0	0	0	0	0	0.00473	0	0	0	0	0	1	0.000000	0.1	0	0	1
.	0	0	0	0	0	0	0	0	0.00236	0	0	0	0	0	1	0.000000	0.1	0	0	1
.	0	0	0	0	0	0	0	0.42857	0.03704	0	0	0	0	0	1	0.601490	0.1	0	1	1
.	7	0	0	0	0.01042	0	0.00068	0.28571	0.01891	0	0.05263	0.01667	0	0.0122	0.57143	0.000954	0.1	0.00258	1	1
.	1	0	0	0	0	0	0	0	0.00946	0	0	0	0	0	1	0.000000	0.1	0	1	1
.	8	0	0	0	0	0	0.00305	0.14286	0.01497	0	0	0	0	0	1	0.000203	0.1	0.00375	0	1
.	3	0	0.01911	0	0	0	0.01017	0.28571	0.03231	0	0	0	0	0.0122	1	0.000000	0.1	0.00445	1	1
.	0	0	0	0	0	0	0	0	0.01024	0	0	0	0	0	1	0.000000	0.1	0	0	1
.	3	0	0.00637	0.0089	0	0	0.00271	0	0.00709	0	0	0	0	0	1	0.000000	0.1	0.00258	0	1
1726	0	0	0	0	0	0	0	0.28571	0.02679	0	0	0	0	0	1	0.000000	0.1	0	0	1
1727	0	0	0.01911	0	0.01042	0	0.00203	0.14286	0.0331	0	0.02632	0	0	0	0.85714	0.000000	0.1	0.00515	1	1
1728	7	0	0.01274	0.00297	0.05208	0	0.03764	0.28571	0.00867	0	0	0.03333	0.07407	0	0.0122	0.57143	0.199535	0.1	0.03255	0	0
.	1	0	0	0.02077	0	0	0.00916	0.14286	0.01418	0	0	0.01667	0.01852	0.03846	0	1	0.051274	0.1	0.03302	0	0
.	4	0	0.01911	0.02671	0	0	0.01933	0.14286	0.0063	0	0	0	0	0	0.85714	0.484688	0.3	0.02412	0	0	0
.	9	0	0	0	0	0	0	0.28571	0.02443	0	0	0	0	0	1	0.229932	0.1	0.00047	0	0	0
.	0	0	0	0	0	0	0.04239	0.14286	0.00709	0	0	0	0	0.0122	1	0.022452	0.1	0.05316	0	0	0
.	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	3	0	0.00637	0.01187	0	0	0.00475	0.14286	0.01261	0	0.10526	0	0	0	0.85714	0.019853	0.1	0.85527	0	0	0
.	5	0	0	0.04154	0	0	0.01797	0.14286	0.00709	0	0	0	0	0	1	0.057611	0.1	0.02272	0	0	0
.	5	0	0	0.00297	0.01042	0.03571	0.01221	0	0.01103	0	0	0.03333	0	0	1	0.124152	0.3	0.02459	0	0	0
.	0	0	0	0	0	0	0.00102	0	0.00788	0	0	0	0	0	1	0.000000	0.1	0	0	0	0
.	1	0	0.00637	0.00297	0	0	0.01085	0.28571	0.01182	0	0	0	0	0	0.28571	0.276229	0.3	0.04098	0	0	0
.	5	0	0.03185	0.00593	0.01042	0.03571	0.02984	0.14286	0.00394	0	0	0.03333	0	0	1	0.003317	0.1	0.18384	0	0	0
.	0	0	0	0	0	0	0.00475	0.14286	0.00788	0	0	0	0	0	1	0.583358	0.1	0.0171	0	0	0
.	5	0	0	0.16024	0	0	0.01221	0.14286	0.0134	0	0	0	0	0	1	0.027792	0.1	0.03677	0	0	0
.	0	0	0	0	0	0	0	0.28571	0.01812	0	0	0	0	0	0.42857	0.000077	0.1	0.03349	0	0	0
.	1	0	0.03185	0.00593	0	0	0.01221	0.14286	0.01418	0	0	0	0	0	1	0.639907	0.1	0.02272	0	0	0
.	5	0	0.02548	0.0089	0	0	0.02408	0.14286	0.01024	0	0	0	0	0	1	0.003021	0.3	0.04918	0	0	0
.	1	0	0.00637	0.04748	0	0	0.01255	0.14286	0.01024	0	0	0	0	0.09756	0.85714	0.196818	0.1	0.1007	0	0	0
.	7	0	0	0	0	0	0.00475	0.28571	0.00946	0	0	0	0	0	0.71429	0.016360	0.3	0.02693	0	0	0
.	5	0	0.02548	0.01484	0	0	0.0156	0.14286	0.01734	0	0	0	0	0	0.85714	0.185147	0.1	0.01475	0	0	0
.	7	0	0	0	0	0	0.00882	0.14286	0.00552	0	0	0	0	0	1	0.053555	0.1	0.01546	0	0	0
.	8	0	0	0	0	0	0.00983	0.14286	0.00552	0	0	0	0	0	1	0.772430	0.1	0.00609	0	0	0
.	8	0	0.01274	0.00297	0	0	0.02781	0.14286	0.00552	0	0	0.06667	0	0.03659	0.71429	0.036220	0.1	0.04824	0	0	0
3300	3	0	0	0.0089	0.13542	0.60714	0.05663	0.28571	0.02916	0	0.05263	0.38333	0.03704	0.03846	0	0.42857	0.000005	0.3	0.0829	0	0
3301	5	0	0.00637	0	0	0	0.02747	0	0.01261	0	0	0.01667	0	0	0	1	0.363442	0.1	0.03068	0	0
3302	9	0	0.01274	0.02671	0	0	0.03493	0.14286	0.01182	0	0	0	0	0	0	1	0.091730	0.4	0.04473	0	0

Şekil 1.2. Normalizasyon işlemi sonucu elde edilen örnek özellik değerleri

EK-3. Model oluşturma ve test

```
veri=xlsread('veri-3410v.xlsx','veriseti');%veriseti isimli dosyadan verilerin alınması
sinif=xlsread('veri-3410v.xlsx','sinif');%verilerin sinif bilgilerinin alınması

tic;%çalışma zamanı hesaplanacak modülün başlangıcı
indices = crossvalind('Kfold',sinif,10);% k-kat çapraz doğrulama
cp = classperf(sinif);

for i = 1:10%verisetinin 10 parçaya bölünmesi
    test = (indices == i); train = ~test;

      svmStruct = svmtrain(veri(train,:),sinif(train,:));%DVM sınıflandırma modelinin oluşturulması
      tahmin= svmclassify(svmStruct,veri(test,:));%DVM sınıflandırıcı testi
      knns=ClassificationKNN.fit(veri(train,:),sinif(train,:),'NumNeighbors',3);%KNN sınıflandırma modelinin oluşturulması

     tahmin=predict(knns,veri(test,:));%KNN test
     c=confusionmat(sinif(test,:),tahmin);%karışıklık matrisi oluşturma
     %k-fold için confusion matris oluşturma
     tp(i)=c(1);
     fp(i)=c(2);
     fn(i)=c(3);
     tn(i)=c(4);
     tpr=sum(tp)/3302;%dp oranının hesaplanması
     fpr=sum(fp)/3302;%yp
     fnr=sum(fn)/3302;%yn
     tnr=sum(tn)/3302;%dn
     classperf(cp,tahmin,test)%10 test için doğruluk oranlarının hesaplandığ matris

end
cp.ErrorRate

toc;%çalışma zamanı hesaplanacak modülün bitişi
```

Şekil 1.3. MATLAB ortamında sınıflandırma modeli oluşturma ve test işlemleri

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: KADI, Cansu
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 24.07.1991, Trabzon
Medeni hali	: Bekar
Telefon	: 0 (555) 655 60 79
E-mail	: cansukadi@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi /Bilgisayar Mühendisliği	Devam Ediyor
Lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	2013
Lise	Trabzon Kanuni Anadolu Lisesi	2009

İş Deneyimi

Yıl	Yer	Görev
2014-Halen	Arçelik A.Ş.	Ar-Ge Mühendisi

Yabancı Dil

İngilizce

Yayınlar

1. Kadı C., Yavanoğlu U. (2017). Malicious Web Page Detection With Machine Learning Techniques. *International Conference on Advanced Technology & Science*, 233-237.

Hobiler

Sinema, Spor

GAZİ GELECEKTİR..