GU J Sci, Part C, 9(4): 608-620 (2021)

Gazi University

Journal of Science

PART C: DESIGN AND TECHNOLOGY

http://dergipark.gov.tr/gujsc

Measuring the Security Effectiveness of Machine Learning Methods Used Against Cyber Attacks in Web Applications

Mesut TOĞAÇAR[1,*]

[1]*Fırat University, Vocational School of Technical Sciences, Department of Computer Technologies, 23119, Center/ELAZIĞ*

Article Info:

Research article
Received: 10.06.2021
Revision: 09.09.2021
Accepted: 08.11.2021

Highlights

- *Decision Support.*
- *Web Applications.*
- *AWUGP.*

Keywords

Web Security
Machine Learning
Cyber Attack
Cyber Security
Artificial Intelligence

Graphical/Tabular Abstract

In this study, an approach is proposed to measure the security of web applications. Five machine learning methods are used in the proposed approach.

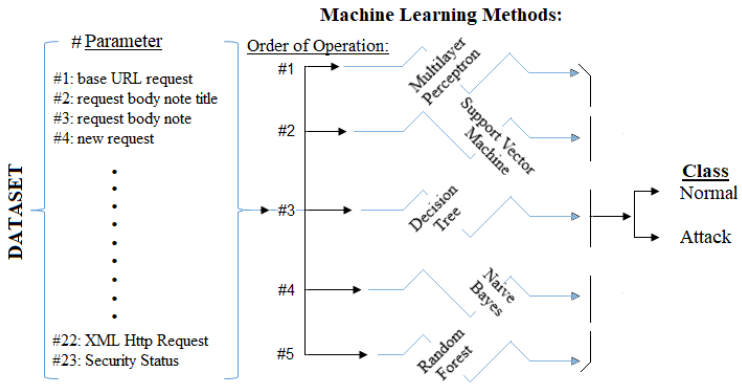


Figure A. The overall design of the proposed approach.

Purpose: The goal of this paper is to successfully detect malicious applications that cause security vulnerabilities in web applications. To do this, it is aimed to process the performance test using machine learning methods and determine the method that performs the best.

Theory and Methods: The proposed model consists of an approach that measures the effectiveness of artificial intelligence based methods of cyber attacks in web applications. The purpose of this approach is to detect attacks in web applications quickly and with the best performance. In this study, machine learning methods, a sub-branch of artificial intelligence, were used in an experimental analysis to determine whether web applications are secure or not.

Results: Multilayer Perceptron, Support Vector Machines, Decision Trees, Naive Bayesian and Random Forest methods were used in the experimental analysis of the study. The overall accuracy obtained by the machine learning methods are 74%, 74%, 100%, 69.5% and 100% respectively. The experimental analysis has shown that the machine learning methods are effective in detecting cyber attacks.

Conclusion: In this study, the parameters of 1000 websites were analyzed using machine learning methods to determine whether the web applications are secure or not. It was found that the Decision Tree and Random Forest methods provided more successful results in web security than other machine learning methods. The overall accuracy of the two successful methods was 100%.

*Corresponding author, e-mail: mtogacar@firat.edu.tr
DOI:10.29109/gujsc.950639

GU J Sci, Part C, 9(4): 608-620 (2021)

Gazi Üniversitesi

Fen Bilimleri Dergisi

PART C: TASARIM VE TEKNOLOJİ

http://dergipark.gov.tr/gujsc

Measuring the Security Effectiveness of Machine Learning Methods Used Against Cyber Attacks in Web Applications

Mesut TOĞAÇAR[1,*]

[1]Fırat Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 23119, Merkez/ELAZIĞ

Makale Bilgisi

Araştırma makalesi
Başvuru: 10.06.2021
Düzeltme:09.09.2021
Kabul: 08.11.2021

Keywords

Web Security
Machine Learning
Cyber Attack
Cyber Security
Artificial Intelligence

Anahtar Kelimeler

Web Güvenliği
Makine Öğrenme
Siber Saldırı
Siber Güvenlik
Yapay Zekâ

Abstract

The rapid advancement of technological developments in the global world, the people who closely follow and share these developments have become the focus of cybercriminals. People realize their basic needs, requests, messages or works through smart devices using the internet infrastructure. While performing these actions, users can inevitably leave an open door through web applications. As a result, custom information can be easily shared with others. Recently, there has been a sharp increase in the number of activities performed on websites. One of the reasons for this increase - and the most important - is the pandemic that is having a global impact. Cybercriminals want to use such situations as an opportunity to enrich themselves financially. They look for vulnerabilities in websites that are in high demand by people and want to access their user and card data. In this study, an approach is proposed to measure the performance of machine learning methods with respect to the vulnerabilities of various websites. The dataset used in the study consists of the parameter properties of 1000 websites. Multilayer Perceptron, Support Vector Machines, Decision Trees, Naive Bayesian and Random Forest methods were used in the experimental analysis of the study. The overall accuracy obtained by the machine learning methods are 74%, 74%, 100%, 69.5% and 100% respectively. The experimental analysis has shown that the machine learning methods are effective in detecting cyber attacks.

Siber Saldırılara Karşı Kullanılan Makine Öğrenme Yöntemlerinin Web Uygulamalarında Güvenlik Etkinliğinin Ölçümü

Öz

Küresel dünyadaki teknolojik gelişmelerin son zamanlarda hızlı ilerlemesi, kitlelerin hızlı bir şekilde bu gelişmeleri yakından takip etmesi ve paylaşımlarda bulunması siber suçluların odak noktası haline gelmiştir. İnsanlar temel ihtiyaçlarını, isteklerini, paylaşımlarını veya çalışmalarını akıllı cihazlar üzerinden internet alt yapısını kullanarak gerçekleştirmektedirler. Bu eylemleri kullanıcılar gerçekleştirirken web uygulamalar üzerinden ister istemez bir açık kapı bırakabilmektedir. Neticesinde kullanıcıya özel tanımlanmış bilgiler başkalarının eline kolayca geçebilmektedir. Son zamanlarda web siteleri üzerinden gerçekleştirilen faaliyetlerde ciddi artış olmuştur. Bu artışın sebeplerinden biri ve en önemlisi ise dünya genelinde etkisini göstermiş olan pandemi sürecidir. Siber suçlular bu gibi durumları fırsata çevirmek ve maddi kazanç sağlamak isterler. İnsanların yoğun talepte bulunduğu web sitelerine yönelik açıklar ararlar ve onların kullanıcı bilgilerine, kart bilgilerine erişmek isterler. Bu çalışma çeşitli web sitelerinin güvenlik açıklarına karşı makine öğrenme yöntemlerinin performansını ölçen bir yaklaşım önermektedir. Çalışmada kullanılan veri kümesi 1000 adet web sitesinin parametre özelliklerinden oluşmaktadır. Çalışmanın deneysel analizlerinde; Çok Katmanlı Algılayıcı, Destek Vektör Makineleri, Karar Ağaçları, Naif Bayes, Rastgele Orman yöntemleri kullanıldı. Makine öğrenme yöntemlerinden elde edilen genel doğruluk başarıları sırasıyla; %74, %74, %100, %69,5 ve %100'dü. Deneysel analizler siber saldırılarının tespitinde makine öğrenme yöntemlerinin etkin olduğunu göstermiştir.

1. GİRİŞ (INTRODUCTION)

Son yıllarda yapay zekâ kavramı çeşitli alanlarda adını söz ettirmeyi başarmış ve alt dalları olan makine öğrenmesi ile derin öğrenme yaklaşımları birçok çalışmada uygulanmıştır [1]. Teknolojik gelişmelerin hızla ilerlemesi bu alandaki uygulamaların insanlar tarafından ilgi görmesi, kötü niyetli kullanıcılarında ilgi görülen alanlara yönelmesine sebep olmuştur [2]. İnsanlar birçok işini interneti kullanarak daha hızlı bir şekilde gerçekleştirmektedirler. Son zamanlarda Kovid-19 salgınının etkisiyle insanların internette geçirdikleri sürelerde belirgin bir artış gözlemlenmiştir [3,4]. Bu durum ister istemez web sitelerinde zafiyet doğurabilmektedir ve kötü kullanıcılar tarafından insanların kişisel bilgilerine, üyelik ve kredi kartı bilgilerine vb. verilerine, kullandıkları çeşitli yöntemlerle erişebilmektedirler [5].

Web uygulamaları ağlar üzerinde bilgi aktarımı sağladığı için genellikle güvenlik açıkları içerebilmektedir. Bu durum kötü amaçlı yazılımcılar veya kullanıcılar tarafından rahatlıkla hedef haline gelebilmektedir [6]. Web uygulamaların saldırılara hedef olunmasının bir diğer sebebi maddi kazançtır. Her ne kadar güvenlik duvarları, saldırı tespit sistemleri etkin bir şekilde kullanılsa da maalesef olumsuz sonuçlar ortaya çıkabilmektedir [7]. Sahte web uygulamaları, spam içeren web bağlantıları, kısa mesaj gönderileri, sahte e-posta bağlantıları vb. durumları sıklıkla kullanırlar [8]. Kötü amaçlı kullanıcılar web saldırılarını gerçekleştirirken web uygulamaların üç bölümüne odaklanarak bu işlemleri gerçekleştirirler. Bunlar; ağ-sunucu bağlantıları, web tasarımı ve kodlama yapısı, kullanıcı boyutu. Bu üç kısımdan en az birinde buldukları açık ile kişisel saldırıları veya siber saldırıları gerçekleştirmektedirler [9–11].

1.1. Literatür İncelemesi (Literature Review)

Web uygulamalarındaki siber saldırıların tespitinde birçok yapay zekâ tabanlı çalışma literatürde yer almıştır. Bu çalışmalardan bazıları incelenirse; Yao Pan ve ark. [12] çalışmasında web uygulamalarındaki saldırıları tespit edebilmek için hem makine öğrenme yöntemlerini hem de derin öğrenme modelini ayrı ayrı kullanmıştır. Önerdikleri yaklaşımda, en iyi başarıyı otokodlayıcı model ile elde etmişlerdir ve %91,8 oranında bir başarı sağlamışlardır. Tianlong Liu ve ark. [13] web saldırılarının tespitinde kullandıkları veri kümesi için yük sınıflandırma ağı makine öğrenme yöntemini kullanmışlar. Onlar çalışmasında güvenilir web sitelerini güvenli olmayan web sitelerinin parametrelerinden ayırt etmişler. Onların yük sınıflandırma ağı ile elde ettikleri genel doğruluk başarısı %99,84'tü. Rafal Kozik ve ark. [14] çalışmasında web uygulamalarına gerçekleştirilen siber saldırıların tespitini gerçekleştirdiler. Onlar çalışmasında Naif Bayes, Adaboost ve J48 makine öğrenme yöntemlerini kullandılar. Deneysel analizlerde en iyi performansı J48 yöntemi ile sağladılar ve %95,97 oranında bir genel doğruluk başarısı elde ettiler. Dhika Rizki Anbiya ve ark. [15] çalışmasında PHP yazılım dilinde derlenmiş web uygulamaların güvenlik açıklarına yönelik bir analiz gerçekleştirmişlerdir. Onlar kullandıkları veri kümesinde verimli özellikleri Genişlik İlk Arama (GİA) algoritmasıyla çıkardılar. Ardından Naif Bayes, Destek Vektör Makinesi (DVM) ve Karar Ağacı yöntemlerini kullanarak sınıflandırma işlemini gerçekleştirdiler. Makine öğrenme yöntemleri arasında en iyi performansı Naif Bayes yöntemi verdi ve Naif Bayes yöntemi ile elde edilen geri çağırma performansı %92'di.

Bu çalışmalar web tabanlı siber saldırılarının tespitinde başarılı sonuçlar vermiştir. Makine öğrenme yöntemlerinin derin öğrenme modellerine göre daha etkin olduğu gözlemlenmiştir. Fakat bu çalışmaların bazısı tek bir yöntem kullanarak analizleri gerçekleştirmiştir, bazısı birkaç yöntemi belirli aralıklar ile kullanarak analizleri gerçekleştirmiştir. Bu durumda her bir yöntemin ayrı ayrı gerçekleştirilmesi saldırıların tespitinde zaman kaybına yol açabilmektedir. Önerilen yaklaşım uçtan uca bir mimari ile tasarlandığı için analiz sürecinde kullanılan makine öğrenme yöntemleri kesintiye uğramadan tek bir yaklaşım gibi hareket edebilmektedir. Ayrıca önerilen yaklaşımda kullanılan makine öğrenme yöntemleri literatürde sıklıkla tercih edilen, farklı veri kümelerinde başarılı sonuçlar elde etmiş yöntemlerden oluşmaktadır.

Bu makalenin amacı, web uygulamalarında güvenlik açığına yol açan kötü amaçlı uygulamaların tespitini başarılı bir şekilde gerçekleştirmektir. Bunun için makine öğrenme yöntemlerini kullanarak performans testinden geçirilmesi ve en iyi performansı veren yöntemin tespit edilmesi amaçlanmıştır. Bu çalışmada diğer bölümler ise şu şekildedir; veri kümesi ile ilgili bilgiler Bölüm 2'de verilmiştir. Çalışmanın analizinde kullanılan makine öğrenme yöntemleri hakkında bilgiler ve önerilen yaklaşım Bölüm 3'te verilmiştir.

Deneysel analizler ve sonuçları Bölüm 4'te verilmiştir. Sırasıyla Bölüm 5 ve Bölüm 6'da Tartışma ve Sonuçlar yer almıştır.

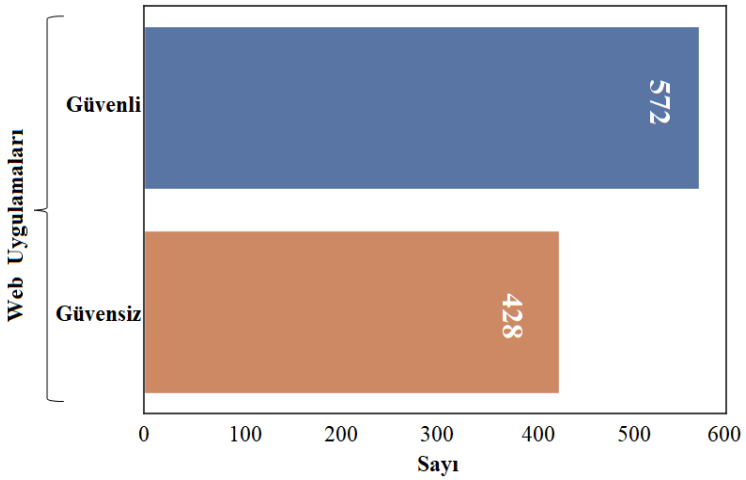
2. VERİ KÜMESİ (DATASET)

Veri kümesi, 1000 web uygulaması içeriğini barındıran, metin tabanlı ve "csv" uzantılı tek dosyadan oluşmaktadır. Veri kümesi, her bir web sitesinin 23 özellik parametresini içerir ve bu parametrelerin etiket grupları {metin}, {sayı}, {doğru, yanlış} değerlerinden birine sahiptir. Veri kümesinin parametre özellikleri ile detaylı bilgileri Tablo 1'de verildi. Veriler, ikili sınıflandırma modeli için tasarlanmıştır ve "is Safe" (güvenlik durumu) parametre özelliğine göre sınıflandırma gerçekleştirilmektedir. Dolayısıyla, web uygulamaların iki güvenlik durumu söz konusudur; eğer web uygulaması "doğru" etiketine sahip ise güvenilir, "yanlış" etiketine sahipse güvenilir değildir. Veri kümesindeki "is Safe" (güvenlik durumu) etiket değeri belirlenirken Açık Web Uygulaması Güvenlik Projesi (AWUGP) kriterleri göz önüne alınmıştır ve bir web uygulamasında AWUGP kriterlerine uymayan en az 10 parametre değeri varsa ilgili web uygulamasının güvenlik durumu güvensiz / yanlış olarak belirlenmiştir. Veri kümesi, 2021 yılında GitHub web sitesi üzerinden erişime sunulmuştur [16].

Tablo 1. *Veri kümesini oluşturan web uygulamalarının parametreleri ve etiket değeri*

Özellik numarası	Veri kümesindeki orijinal parametreler	Orijinal parametrelerin Türkçe karşılıkları	Etiket Değer Türü
1	request base Url	Temel URL isteği	{metin}
2	request body note title	Gövde notu başlığı iste	{metin}
3	request body note desc	Gövde notu talep et	{metin}
4	request fresh	Yeni talep	{metin}
5	request headers host	Üstbilgi barındırıcısı iste	{metin}
6	request headers user-agent	Başlıkları isteme kullanıcı aracısı	{doğru, yanlış}
7	request headers content type	Başlık içerik türü isteme	{metin}
8	request headers org id	Üstbilgiler kuruluş kimliği iste	{metin}
9	request headers user session id	Üstbilgi isteme kullanıcı oturum kimliği	{metin}
10	request headers accept	Başlık isteme kabul etme	{metin}
11	request headers content-length	Başlık içerik uzunluğu isteği	{sayı}
12	request headers user name	Başlık kullanıcı adı isteği	{metin}
13	request headers user role	Üst bilgi istemede kullanıcı rolü	{metin}
14	request hostname	Ana bilgisayar adı isteği	{metin}
15	request IP	İnternet protokol isteği	{İP adres}
16	request original URL	Orijinal URL isteği	{metin}
17	request path	İstek yolu	{metin}
18	request protocol	İstek protokolü	{http, https}
19	request secure	Güvenli istek	{doğru, yanlış}
20	request stale	Eski istek	{doğru, yanlış}
21	request subdomains	Alt alan adı isteme	{metin}
22	request XHR (XML Http Request)	XHR isteği (XML Http İsteği)	{doğru, yanlış}
23	is Safe	Güvenlik durumu	{doğru, yanlış}

Bu çalışmada veri kümesinin %20'si test verisi, %80'i eğitim verisi olarak ayrıldı. Deneysel analizlerde kullanılan web uygulamaların istatistiksel bilgileri Şekil 1'de gösterildi.

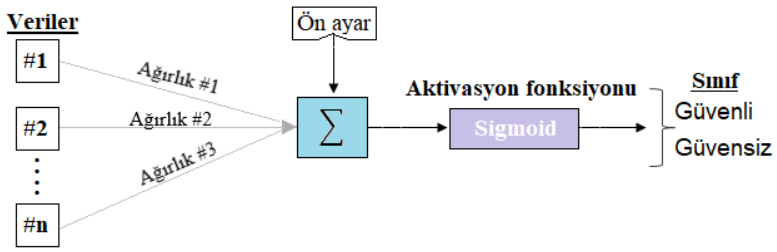


Şekil 1. *Veri kümesindeki web uygulamalarının güvenlik durumu istatistiksel bilgileri*

3. YÖNTEMLER VE ÖNERİLEN YAKLAŞIM (METHODS AND PROPOSED APPROACH)

3.1. Çok Katmanlı Algılayıcı (Multilayer Perceptron)

Çok katmanlı algılayıcı (ÇKA), yapay sinir ağlarını temel almış, girdi verilerini işleyerek çıktı katmanına aktaran ve sınıflandırma işlemini gerçekleştiren makine öğrenme yöntemidir. ÇKA, yapay sinir ağlarını temel almış, girdi verilerini işleyerek çıktı katmanına aktaran ve sınıflandırma işlemini gerçekleştiren makine öğrenme yöntemidir. ÇKA, denetimli bir öğrenme yaklaşımı ile geliştirilmiş bir yöntemdir ve bu yöntemde en önemli faktör eşik değeridir. Eşik değerinin belirlenmesi veri kümesinin sayısı ve içeriği ile doğrudan bağlantılıdır ve bu değer değişkendir. ÇKA yönteminde iterasyon sayısı artırılarak öğrenme olayı artırılabilir. Girdi verileri ÇKA yönteminde işlenirken başlangıç ağırlık değerleri rastgele verilir ve öğrenme olayı ile ağırlık parametreleri sınıflandırma süreci tamamlayana kadar güncellenir. ÇKA yönteminin çıkışına doğru sigmoid aktivasyon fonksiyonu kullanılır. Bu fonksiyon sayesinde ikili bir sınıflandırma sürecinde çıktı değeri eşik değerinin üzerinde ise sınıf türü 1(bir) olarak çıkış yapar veya eşik değerinin altında bir değer ise sınıf türü 0 (sıfır) olarak çıkış yapar. Bu yöntemde öğrenme gerçekleştirilirken ileri doğru ve geriye doğru adımlar ile ağırlıklar güncellenir [17,18]. Basit bir ÇKA yönteminin tasarımı Şekil 2'de gösterildi. Bu çalışmanın deneysel analizinde kullanılan ÇKA yöntemi, Python yazılımında Sklearn kütüphanesindeki kod parametreleri kullanılarak derlendi. Analizler gerçekleştirilirken varsayılan parametre değerleri kullanıldı.



Şekil 2. *ÇKA yönteminin işleyişini gösteren genel tasarım*

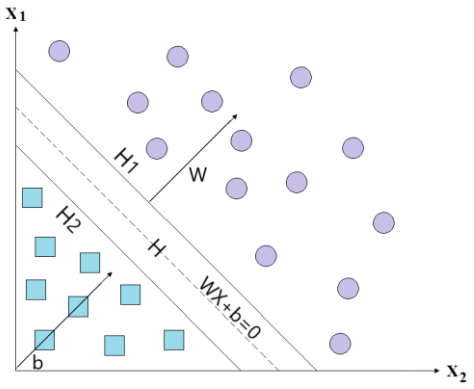
3.2. Destek Vektör Makineleri (Support Vector Machines)

DVM, birçok uygulama alanlarında tercih edilen, başarılı sonuçlar verebilen sınıflandırma ve regresyon işlemlerinde kullanılan bir makine öğrenme yöntemidir. DVM yöntemi çeşitli veri tiplerine göre (ikili, çoklu, büyük veri vs.) algoritma yapılarında farklı optimizasyon algoritmaları kullanarak başarılı sonuçlar elde etmişlerdir. DVM sınıflandırma sürecinde ikili veya çok sınıf türü içerisindeki marjı en üst düzeyde ayırmayı amaçlar ve bunun içinde karar fonksiyonlarını kullanır. Karar fonksiyonları, karar sınır aralığının belirlenmesini sağlayan doğrudan eğitim verilerinden faydalanarak boyutlu bir alan çıkartan algoritmalardır. Diğer bir deyişle, eğitim veri kümesindeki ortalama karesel hatayı en aza indirmek yerine genelleme hatası üzerindeki bir sınırın en aza indirilmesini sağlar.

Veri kümesindeki örnek sayısını n olarak belirtelim. Her bir örnek giriş vektörü (X_i) ve sınıf türü etiketinden (Y_i) oluşmaktadır. Bu ilişki Denklem 1'de gösterildiği gibi ifade edilir. İki boyutlu bir girdi verisinin durumu doğrusal olarak ayrılabilir ve bunu gerçekleştirmek için hiper düzleme ihtiyaç vardır. Karar düzeyi ya da karar fonksiyonu hiper düzlem için kullanılan matematiksel denkleme göre hesaplanır. Bu hesaplama işlemi Denklem 2'ye göre gerçekleştirilir. Bu denklemde W değişkeni, hiper düzlemi tanımlar ve bu düzlemi uygun değer şeklinde ayırır. Ayrıca, b değişkeni ön ayar olarak tanımlanır [19]. Hiper düzlem belirlenirken iki adet H_1 ve H_2 alanları oluşur ve H_1 ile H_2 arasında hiç bir örnek girdisi yer almaz [19,20]. Bu durum Şekil 3'te gösterilmiştir. Bu çalışmada DVM yöntemi Python dilinde Sklearn kütüphanesi kullanılarak derlendi. DVM yönteminde çekirdek, radyal temelli fonksiyon seçilerek sınıflandırma gerçekleştirildi ve diğer parametre değerleri Sklearn kütüphanesinde varsayılan değerler değiştirilmeden kullanıldı.

$$(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n) \quad (1)$$

$$W^T X_i + b = 0 \quad (2)$$



Şekil 3. *DVM yöntemi ile ikili sınıflandırma süreci*

3.3. Karar Ağacı (Decision Tree)

Karar ağaçları veri madenciliğinde sıklıkla kullanılan girdi verilerini sınıflandırma işlemi için genellikle tercih edilen makine öğrenme yöntemidir. Bu yöntem, kök-düğüm-yaprak-çocuk bağıntısı ile girdi verisinin özelliklerine ve özellik değerlerine göre sorgular gerçekleştirip sınıflandırma sürecini tamamlar. Basit bir ifade ile girdi değerlerine kök-düğüm-yaprak gibi birimlerde sorgulanan sorular ile "evet-hayır" gibi cevaplar alınır ve bu cevaplar ile ilgili verinin sınıflandırılması ağaç yapısında dallanarak belirlenir. Bazen girdi verisi özelliklerine göre yapraklara ulaşırken, her bir yaprak şartlı olasılıksal değerlere göre sınıflandırma sürecini gerçekleştirir [21,22]. Karar ağaçları için ölçeklendirme işleminde entropi ve Gini indeksleri kullanılarak gerçekleştirilir. Burada E örnek verisi içerisinde m sınıf türüne sahip bir küme şu şekilde ifade edilir; P_i $(i = 1,2, \dots, m)$. Burada P_i değişkeni, E veri kümesi içerisinde i. sınıfa ait öğelerin değerini temsil eder. Olasılıksal değerlerin entropisinin hesaplanmasında kullanılan matematiksel formül

Denklem 3'te verildi. Entropi değerinin hesaplanmasında Denklem 4'daki formül kullanılır ve Gini indeks değerinin hesaplanmasında Denklem 5'teki matematiksel formül kullanılır [21]. Bu çalışmanın deneysel analizinde Karar ağaçları yöntemi Python yazılım dilinde Sklearn kütüphanesi kullanılarak derlendi. Karar ağaçları için maksimum derinlik oranı 33 olarak belirlendi ve diğer parametreler için varsayılan değerler tercih edildi.

$$(P_i)_{i=1}^{m} \tag{3}$$

$$-\sum_{i=1}^{m} P_i \log(P_i) \tag{4}$$

$$1-\sum_{i=1}^{m} {P_i}^2 \tag{5}$$

3.4. Naif Bayes (Naive Bayes)

Naif Bayes, Bayes teoremine dayalı geliştirilen ve girdi verilerini olasılıksal tabanlı süreçlerden geçirerek sınıflandırma işlemini gerçekleştiren makine öğrenme yöntemidir. Bu yöntemin çalışma prensibinde bir girdi örneği için her sınıf türü için olasılık değerleri üretir ve olasılık değeri en yüksek olan sınıf türüne girdi örneği aktarılır. Bazen girdi örneklerinin olasık değeri sıfır olabilir ve bu durum sıfır frekans olarak adlandırılır. Bu durumda girdi görüntüsü herhangi bir sınıfa aktarımı gerçekleştirilemez. Bu sebeple olasılık değeri sıfır olan örnek türleri Naif Bayes yönteminde düzeltme algoritmaları (örneğin; laplace dönüşümü) kullanılarak sıfırdan farklı bir olasılık değeri alır [23]. Naif Bayes yöntemi gerçek zamanlı sınıflandırma, metin tabanlı sınıflandırma, çok sınıflı tahmin ve spam filtreleme gibi alanlarda sıklıkla tercih edilir. Olasılık değerlerinin hesaplanmasında kullanılan Bayes formülü Denklem 6'da verilmiştir. Bu denklemde olasılık değerleri P ile temsil edilir ve A ile B değişkenleri örnek girdileri temsil eder. Denklem 6'da değişken durumları ile ilgili açıklamalar şu şekildedir;

- $P(A|B)$: B durumunun gerçekleşmesi sırasında A durumunun gerçekleşme olasılığı,
- $P(B|A)$: A durumunun gerçekleşmesi sırasında B durumunun gerçekleşme olasılığı,
- $P(A)$: A durumunun gerçekleşme olasılığı,
- $P(B)$: B durumunun gerçekleşme olasılığı [24],

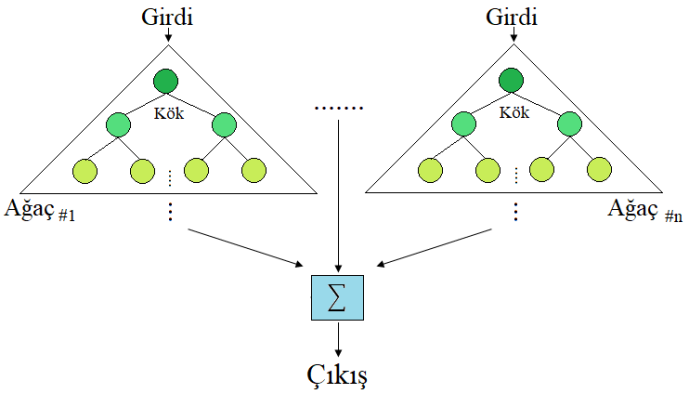
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{6}$$

Bu çalışmanın deneysel analizinde kullanılan Naif Bayes yöntemi, Python yazılımında Sklearn kütüphanesindeki kod parametreleri kullanılarak derlendi. Bu yöntem için tercih edilen parametreler varsayılan değerlerden oluşmaktadır.

3.5. Rastgele Orman (Random Forest)

Rastgele Orman, denetimli yaklaşımla sınıflandırma işlemini gerçekleştirebilen ve birden fazla karar ağacını bir araya getirerek (orman oluşumu) daha istikrarlı sonuçlar üretmeyi amaçlayan makine öğrenme yöntemidir. Bu yöntem ile hem regresyon hem de sınıflandırma işlemi gerçekleştirilebilmektedir. Rastgele Orman yöntemi ile eğitim esnasında aşırı uyum gibi problemlerin önüne geçilir. Sınıflandırma sürecinde ağaç gurupları arasında olasılık değeri yüksek olan ağaç gurubu tarafından işlenerek, girdi verisi ilgili sınıfa aktarılır [25,26]. Rastgele Orman yönteminin işleyişini gösteren tasarım Şekil 4'te gösterildi. Bu çalışmanın deneysel analizinde Rastgele Orman yöntemi, Python yazılımında Sklearn kütüphanesindeki kod

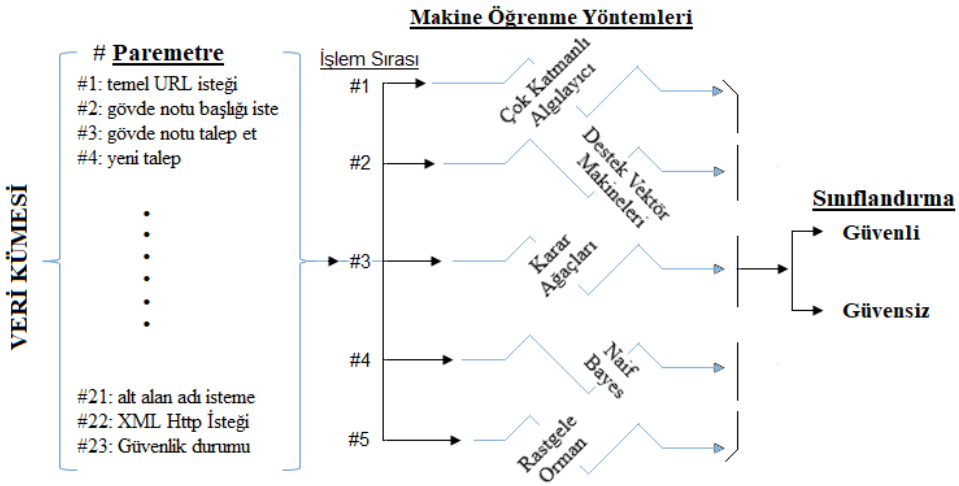
parametreleri kullanılarak derlendi. Bu yöntem için tercih edilen parametreler varsayılan değerlerden oluştu.



Şekil 4. *Rastgele Orman yöntemi ile sınıflandırma süreci*

3.6. Önerilen Yaklaşım (Proposed Approach)

Önerilen model, web uygulamalarında siber saldırıların yapay zekâ tabanlı yöntemlerinin etkinliğini ölçen bir yaklaşımdan oluşmaktadır. Bu yaklaşımdaki amaç web uygulamalarındaki saldırılarını en iyi performans ile hızlı bir şekilde tespit edebilmesidir. İnternet ağında birçok kullanıcının işlemleri özel veriler girildikten sonra gerçekleşmektedir. Kullanıcıların bilgilerini sızdıran bir web uygulamasına gün içerisinde birçok kişi girebilmektedir. Web uygulamalarındaki güvenlik açıklarının tespiti anlık ve başarı oranı yüksek bir şekilde gerçekleşmesi gerekir. Bu tür durumlarda teknolojik gelişmeler yapay zekâ tabanlı sistemleri ön plana çıkarmaktadır. Bu çalışmada yapay zekâ yaklaşımının alt dalı olan makine öğrenme yöntemlerini deneysel analizlerde kullanarak web uygulamalarının güvenli olup olmadığını tespiti gerçekleştirilmiştir. Önerilen yaklaşımın genel tasarımı Şekil 5'te gösterildi.



Şekil 5. *Önerilen yaklaşımın genel tasarımı*

4. DENEYSEL ANALİZ (EXPERIMENTAL ANALYSIS)

Deneysel analizler için Python yazılım dili kullanıldı ve açık erişimli kaynak kodlar [27] Jupyter Notebook arayüzü kullanılarak Google Colab sunucusu üzerinde derlendi [28]. Analizlerin sonuçlarını

değerlendirilirken ölçek olarak karmaşıklık matrisi kullanıldı. Karmaşıklık matrisinin hesaplanmasında kullanılan metrikler ise şunlardır; duyarlılık (Duy), özgüllük (Özg), hassasiyet (Has), f-skoru (f-skr) ve doğruluk (Dğr). Metriklerin hesaplanmasında Denklem 7 ile Denklem 11 arasındaki formüller kullanıldı ve formüllerde kullanılan değişkenler şunlardır; doğru pozitif (DP), doğru negatif (DN), yanlış pozitif (YP), yanlış negatif (YN)'tir [29,30].

$$\text{Duy} = \frac{\text{DP}}{\text{DP+YN}} \tag{7}$$

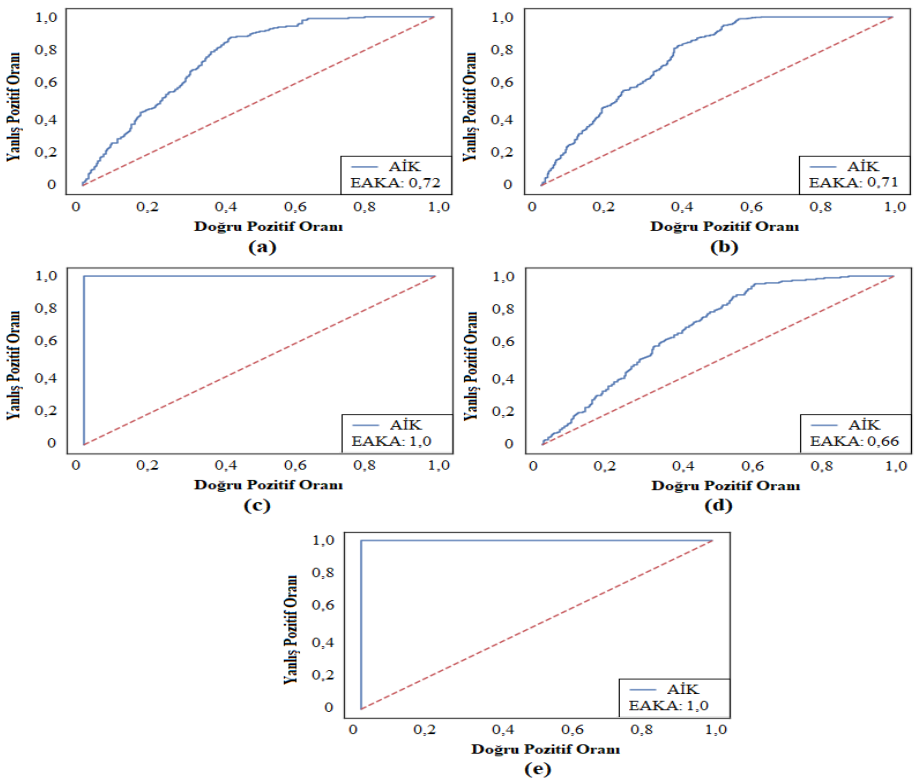
$$\text{Özg} = \frac{\text{DN}}{\text{DN+YP}} \tag{8}$$

$$\text{Has} = \frac{\text{DP}}{\text{DP+YP}} \tag{9}$$

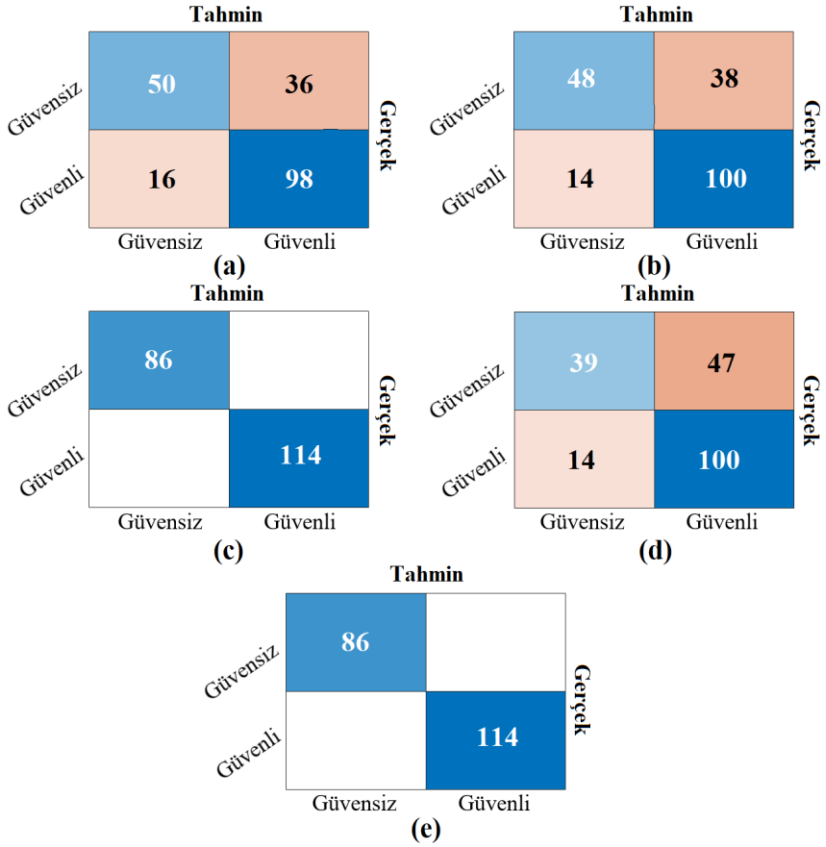
$$\text{f-skr} = \frac{2\text{xDP}}{2\text{xDP+YP+YN}} \tag{10}$$

$$\text{Dğr} = \frac{\text{DP+DN}}{\text{DP+DN+YP+YN}} \tag{11}$$

Çalışmanın analizlerinde veri kümesinin %80'i eğitim verisi olarak ayrıldı. Makine öğrenme yöntemlerinin analizlerinde elde edilen genel doğruluk başarıları şu şekildedir; ÇKA yöntemi ile %74, DVM yöntemi ile %74, Karar Ağacı yöntemi ile %100, Naif Bayes yöntemi ile %69,5 ve Rastgele Orman yöntemi ile %100'dü. Analizlerin makine öğrenme yöntemlerinden elde edilen alıcı işletim karakteristiği (AİK) ve eğri altında kalan alan (EAKA) grafikleri Şekil 6'da gösterildi. Makine öğrenme yöntemlerinden elde edilen karmaşıklık matrisleri Şekil 7'de gösterildi ve analiz sonuçları ile ilgili detaylı bilgiler Tablo 2'de verildi. Analiz sonuçları Karar Ağacı yönteminin ve bu yönteme dayalı oluşturulmuş Rastgele Orman yönteminin en iyi performansı verdiği gözlemlendi.



Şekil 6. *Makine öğrenme yöntemlerinde elde edilen AİK grafikleri;*
a) ÇKA yöntemi, b) DVM yöntemi, c) Karar Ağacı, d) Naif Bayes, e) Rastgele Orman



Şekil 7. *Makine öğrenme yöntemlerinde elde edilen karmaşıklık matrisleri; a) ÇKA yöntemi, b) DVM yöntemi, c) Karar Ağacı, d) Naif Bayes, e) Rastgele Orman*

Tablo 2. *Deneysel analizlerden elde edilen karmaşıklık matrislerinin metrik sonuçları (%)*

Yöntem	Sınıf	Duy	Özg	Has	f-skr	Dğr
ÇKA	Güvensiz	58,14	85,96	75,76	65,79	74
	Güvenli	85,96	58,14	73,13	79,03	
DVM	Güvensiz	55,81	87,72	77,42	64,86	74
	Güvenli	87,72	55,81	72,46	79,37	
Karar Ağacı	Güvensiz	100	100	100	100	100
	Güvenli	100	100	100	100	
Naif Bayes	Güvensiz	45,35	87,72	73,58	56,11	69,50
	Güvenli	87,72	45,35	68,03	76,62	
Rastgele Orman	Güvensiz	100	100	100	100	100
	Güvenli	100	100	100	100	

5. TARTIŞMA (DISCUSSION)

Önerilen yaklaşımda en iyi performansları Karar Ağacı ve Rastgele Orman yöntemleri verdi. İki yöntemin mimari yapısı benzer özellikler gösterdiği için elde edilen başarıya da bu durum yansımıştır. Bu çalışmada

gerçekleştirilen analizlerde beş makine öğrenme yöntemi kullanıldı. Önerilen yaklaşımda ÇKA, DVM ve Naif Bayes yöntemleri istenilen başarıyı veremedi. Bu durumun sebepleri arasında;

- Makine öğrenme yöntemlerinin veri kümesi türüne ve büyüklüğüne bağlı olması,
- Makine öğrenme yöntemlerinin çekirdeğini doğrudan etkileyen parametre seçimleri,
- Makine öğrenme yöntemlerinin mimari yapısı,
- Kullanım amacı (sınıflandırma, regresyon, kümeleme vb.) ve gösterdiği performans [23-26].

Bu tür olası sebepler öngörüldüğünden çoklu makine öğrenme yaklaşımı kullanarak analizler gerçekleşti ve neticesinde iki makine öğrenme yöntemi (karar ağacı, rastgele orman) ile istenilen başarı sağlanmış oldu. Önerilen yaklaşımda dezavantaj olarak gördüğüm nokta, web uygulamalarında çıkartılan gereksiz parametrelerin özellik seçim yöntemi kullanarak elenmemesidir. Çünkü gereksiz özelliklerin önerilen yaklaşımın başarı performansını düşürmektedir ve en önemlisi zaman kaybına neden olmaktadır. Aynı veri kümesini kullanarak analizleri gerçekleştirilen çalışmalar Tablo 3'te verildi.

Tablo 3. *Aynı veri kümesini kullanan çalışmaların karşılaştırılması*

Çalışma	Yıl	Kullanılan Yöntemler	Başarılı Yöntemler	Dğr
Neha Hemane [31]	2021	Makine öğrenme yöntemleri	DVM (gauss fonksiyonlu)	%84
Bu çalışma	2021	ÇKA, DVM, Karar Ağacı, Naif Bayes, Rastgele Orman	Karar Ağacı, Rastgele Orman	%100

Neha Hemane [31]'nin analizinde makine öğrenme yöntemlerinin parametre ve fonksiyon değerleri ile performans ölçülmüştür. Dolayısıyla bu seçimler önerdiği yaklaşımın performansını istenilen düzeyde elde edilmesini sağlamıştır ve %84 oranında doğruluk başarısını DVM yönteminin gauss fonksiyonu parametresini seçerek elde etmiştir. Bu çalışmanın deneysel analizinde kullanılan DVM yöntemi için radyal temelli fonksiyon tercih edildi ve %74 oranında genel doğruluk başarısı elde edildi. Neha Hemane [31]'nin deneysel analizinde DVM yöntemi için tercih edilmiş çekirdek fonksiyonu gauss'tur. İki çalışmada DVM yöntemiyle elde edilmiş performans farkları doğrudan çekirdek fonksiyon türleri (gauss, radyal temelli) ile alakalıdır.

6. SONUÇ (CONCLUSION)

Son zamanlarda web uygulamaları popülerliğini artırmıştır. Dijital dönüşüm içerisinde olan birçok küçük, orta ve büyük işletmeler, kurumlar, kuruluşlar web uygulamaları üzerinde kullanıcılar ile iletişimlerini sağlamaktadır. Kötü amaçlı kullanıcılar tarafından web uygulamalarındaki işlemler, kullanıcı hesapları, şifreler, kart bilgileri vb. veriler ele geçirilmek istenmektedir ve bunun için internet üzerinden birçok sahte yollara başvurabilmektedirler. Sonuç olarak, kişisel / kamu verilerin ele geçirilmesinde web uygulamaların güvenliği önemlidir. Anlık birçok insanın web uygulamalarını kullandığı ve bu etkileşimin güvenliğinin de yapay zekâ tabanlı bir savunma mekanizması ile başarılı sonuçlar verileceği öngörülmektedir. Bu çalışmada, 1000 adet web sitesine ait parametreler makine öğrenme yöntemleri tarafından analiz edilerek web uygulamaların güvenli olup olmadığının tespit ölçümü gerçekleştirildi. Karar Ağacı ve Rastgele Orman yöntemlerinin web güvenlik konusunda diğer makine öğrenme yöntemlerine göre daha başarılı sonuçlar verdiği gözlemlendi. İki başarılı yöntemlerden elde edilen genel doğruluk oranı %100'dü.

Gelecek çalışmada, farklı yöntem ve yaklaşım içeren yapay zekâ destekli derin öğrenme modelleri üzerinde analizler gerçekleştirilecektir.

KAYNAKLAR (REFERENCES)

[1] Yin Z., Liu W., Chawla S., Adversarial Attack, Defense, and Applications with Deep Learning Frameworks, (2019) 1–25. doi:10.1007/978-3-030-13057-2_1.

[2] Jang-Jaccard J., Nepal S., A survey of emerging threats in cybersecurity, J Comput Syst Sci, (2014) 80:973–93. doi:https://doi.org/10.1016/j.jcss.2014.02.005.

[3] Nguyen M.H., Gruber J., Fuchs J., Marler W., Hunsaker A., Hargittai E., Changes in Digital Communication During the COVID-19 Global Pandemic: Implications for Digital Inequality and Future Research, Soc Media + Soc, (2020) 6:2056305120948255. doi:10.1177/2056305120948255.

[4] Dunton G.F., Do B., Wang S.D., Early effects of the COVID-19 pandemic on physical activity and sedentary behavior in children living in the U.S., BMC Public Health, (2020) 20:1351. doi:10.1186/s12889-020-09429-3.

[5] Buchanan R., What We Know about Identity Theft and Fraud Victims from Research-and Practice-Based Evidence center for victim Research Report, (2019) 34.

[6] Hashizume K., Rosado D.G., Fernández-Medina E., Fernandez E.B., An analysis of security issues for cloud computing, J Internet Serv Appl, (2013) 4:5. doi:10.1186/1869-0238-4-5.

[7] Marashdih A.W., Zaaba Z.F., Suwais K., Mohd N.A., Web application security: An investigation on static analysis with other algorithms to detect cross site scripting, Procedia Comput Sci, (2019) 161:1173–81. doi:10.1016/j.procs.2019.11.230.

[8] Ferrara E., The history of digital spam, Commun ACM, (2019) 62:82–91. doi:10.1145/3299768.

[9] Ingle D., Attacks on Web Based Software and Modelling Defence Mechanisms, Int J UbiComp, (2012) 3:11–30. doi:10.5121/iju.2012.3302.

[10] Bhagwani H., Log based Dynamic Intrusion Detection of Web Applications. Master of Technology, (2019).

[11] Liu Y., Wang Z., Tian S., Security Against Network Attacks on Web Application System BT - Cyber Security, In: Yun X, Wen W, Lang B, Yan H, Ding L, Li J, ve ark., editors., Singapore: Springer Singapore, (2019) 145–52.

[12] Pan Y., Sun F., Teng Z., White J., Schmidt D.C., Staples J., ve ark., Detecting web attacks with end-to-end deep learning, J Internet Serv Appl, (2019) 10:16. doi:10.1186/s13174-019-0115-x.

[13] Liu T., Qi Y., Shi L., Yan J., Locate-then-DetecT: Real-time web attack detection via attention-based deep neural networks. IJCAI Int Jt Conf Artif Intell, (2019) 4725–31. doi:10.24963/ijcai.2019/656.

[14] Kozik R., Choraś M., Renk R., Holubowicz W., Kozik R., Choraś M., ve ark., A Proposal of Algorithm for Web Applications Cyber Attack Detection, (2016) 1–8.

[15] Anbiya D.R., Purwarianti A., Asnar Y., Vulnerability Detection in PHP Web Application Using Lexical Analysis Approach with Machine Learning 5th Int. Conf. Data Softw. Eng., (2018) 1–6. doi:10.1109/ICODSE.2018.8705809.

[16] Hemane N., Cyber Security: Machine Learning Model to protects web and mobile applications from runtime attacks /(Dataset). Github, (2021) https://github.com/nehahemane/Cyber_Security (Erişim tarihi: 6 Haziran 2021).

[17] Thomas P., Suhner M-C., A New Multilayer Perceptron Pruning Algorithm for Classification and Regression Applications, Neural Process Lett, (2015) 42:437–58. doi:10.1007/s11063-014-9366-5.

[18] Castro W., Oblitas J., Santa-Cruz R., Avila-George H., Multilayer perceptron architecture optimization using parallel computing techniques, PLoS One, (2017) 12:e0189369. doi:10.1371/journal.pone.0189369.

[19] Cervantes J., Garcia-Lamont F., Rodríguez-Mazahua L., Lopez A., A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing, (2020) 408:189–215. doi:https://doi.org/10.1016/j.neucom.2019.10.118.

[20] Ma Y., Zhang Q., Li D., Tian Y., Linex Support Vector Machine for Large-Scale Classification, IEEE Access, (2019) 7:70319–31. doi:10.1109/access.2019.2919185.

[21] Kingsford C., Salzberg S.L., What are decision trees? Nat Biotechnol, (2008) 26:1011–3. doi:10.1038/nbt0908-1011.

[22] Gadekallu T.R., Khare N., Bhattacharya S., Singh S., Maddikunta P.K.R., Srivastava G., Deep neural networks to predict diabetic retinopathy, J Ambient Intell Humaniz Comput, (2020) doi:10.1007/s12652-020-01963-7.

[23] Xu S., Bayesian Naïve Bayes classifiers to text classification, J Inf Sci, (2016) 44:48–59. doi:10.1177/0165551516677946.

[24] Goh J.O.S., Hung H-Y., Su Y-S., Chapter Seven - A conceptual consideration of the free energy principle in cognitive maps: How cognitive maps help reduce surprise. In: Federmeier KDBT-P of L and M, 69, Academic Press, (2018) 205–40. doi:https://doi.org/10.1016/bs.plm.2018.09.005.

[25] Zhang H., Zhou J., Jahed Armaghani D., Tahir M.M., Pham B.T., Huynh V. V., A Combination of Feature Selection and Random Forest Techniques to Solve a Problem Related to Blast-Induced Ground Vibration, Appl Sci, (2020) 10. doi:10.3390/app10030869.

[26] Wang P., Hu J., A hybrid model for EEG-based gender recognition, Cogn Neurodyn, (2019) 13:541–54. doi:10.1007/s11571-019-09543-y.

[27] Amudaakindele K., Telecommunication Churn Prediction, Github, (2020) https://github.com/amudaakindele/Telecommunication-Churn-Prediction/blob/master/Telecom_churn.ipynb (Erişim tarihi: 9 Haziran 2021).

[28] Carneiro T., NóBrega RVM D., Nepomuceno T., Bian G., Albuquerque VHC D., Filho PPR., Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, IEEE Access, (2018), 6:61677–85. doi:10.1109/access.2018.2874767.

[29] Hasnain M., Pasha M.F., Ghani I., Imran M., Alzahrani M.Y., Budiarto R., Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking, IEEE Access, (2020) 8:90847–61. doi:10.1109/access.2020.2994222.

[30] Demir F., Ismael A.M., Sengur A., Classification of Lung Sounds With CNN Model Using Parallel Pooling Structure, IEEE Access, (2020) 8:105376–83. doi:10.1109/access.2020.3000111.

[31] Hemane N. Cyber Security analysis results, Github, (2021) https://github.com/nehahemane/Cyber_Security/blob/main/Cyber_Security.ipynb (Erişim tarihi: 10 Haziran 2021).