SİTELER ARASI BETİK ÇALIŞTIRMA SALDIRILARI İÇİN DERİN ÖĞRENME TABANLI TESPİT SİSTEMİ

YÜKSEK LİSANS TEZİ

Selim ÇELİK

Danışman
Dr. Öğr. Üyesi Ahmet Haşim YURTTAKAL

BİLGİSAYAR ANABİLİM DALI

Ocak 2024

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

SİTELER ARASI BETİK ÇALIŞTIRMA SALDIRILARI İÇİN DERİN ÖĞRENME TABANLI TESPİT SİSTEMİ

Selim ÇELİK

Danışman
Dr. Öğr. Üyesi Ahmet Haşim YURTTAKAL

BİLGİSAYAR ANABİLİM DALI

Ocak 2024

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

08 / 02 / 2024

İmza

Selim ÇELİK

ÖZET

Yüksek Lisans Tezi

SİTELER ARASI BETİK ÇALIŞTIRMA SALDIRILARI İÇİN DERİN ÖĞRENME TABANLI TESPİT SİSTEMİ

Selim ÇELİK

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ahmet Haşim YURTTAKAL

Günümüz dünyasında yapılan işlerin birçoğu sanal ortamlara aktarılmakta, eskiden el ile yapılan işler web uygulamaları vasıtasıyla yapılmaktadır. Web uygulamalarına aktarılan işler başlarda yerel ağlarda çalıştırılırken artan mobilite, uzaktan çalışma konseptleri, her yerden ulaşılabilme ihtiyaçlarıyla birlikte uygulamalar internete açık hale gelmiş, kullanıcıların erişimi kolaylaşırken saldırganlar içinde yeni hedefler ortaya çıkmıştır. Ayrıca Sıfır Güven(Zero Trust) anlayışıyla birlikte artık yerel ağ-genel ağ konseptleri terk edilerek uygulamaya erişilebilen her alan aynı güvenlik düzeyinde kabul edilmeye başlanmıştır. Bu durum güvenli ve kontrol altındaki ağlarda çalışacağı düşünülerek güvenlik konseptleri dikkate alınmadan geliştirilen web uygulamalarını açık tehdit haline getirmiştir. Bu tür sistemlere yapılan saldırılar imza tabanlı çalışan IPS-IDS sistemleri ile engellenmeye çalışılmakla beraber, saldırganlarda bu tür sistemleri atlatmak için yöntemlerini değiştirmekte, imza veritabanlarının güncelleme hızı yeni yöntemlerin geliştirilme hızının gerisinde kalmaktadır. Bu çalışma ile derin öğrenme sistemlerini kullanarak yüksek başarımlı bir XSS saldırısı tespit sisteminin geliştirilmesi amaçlanmıştır. Çalışmada yapay zeka alanındaki yeni trendlerden biri olan evrişimli sinir ağları kullanılmıştır. Veri seti OWASP ve PortSwigger sayfalarından toplanan örneklemleri içermektedir. Veri setinden zararlı kodlar ile birlikte zararsız kodlarda yer almaktadır. Geliştirilen sistem %99 oranında doğruluğa ulaşmış olup, mevcut sistemlere entegrasyonu açısından elverişli olduğu gösterilmiştir.

Anahtar Kelimeler: XSS, Siteler arası Betik Çalıştırma, Derin Öğrenme, Evrişimli Sinir Ağları

ABSTRACT

M.Sc. Thesis

DEEP LEARNİNG-BASED DETECTION SYSTEM FOR CROSS-SITE SCRIPTING ATTACKS

Selim ÇELİK

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Computer

Supervisor: Asst. Prof. Ahmet Haşim YURTTAKAL

In modern world, many task are being transferred to virtual environments, and works which were once done manually are now being carried out through web applications. Initally, task transferred to web applications were run on local networks. However,with increasing mobility, concepts of remote work, and the need for accessibility from anywhere, applications have become accessible on the internet, making it easier for users to access while also presenting new targets for attackers. Furthermore, the concepts of local network and public network have been abandoned with the zero trust approach, and every area accessible to the application is now considered to have the same level of security. This situation has turned web applications developed without considering security concepts into open threats, assuming they will operate in secure and controlled networks. Attacks on such systems are attempted to be thwarted using signature-based IPS -IDS systems; however, attackers are constantly changing their methods to circumvent such systems. The updating speed of signature databases fall behind the development speed of new methods. The aim of this study is to develop a high-performance XSS attack detection system using deep learning systems. In the study, convolutional neural networks (CNNs), one of the new trends in the field of artificial intelligence, have been utilized. The dataset contains samples collected from OWASP and PortSwigger pages, including both malicious and benign examples. The developed system has achieved a 99% accuracy rate, demonstrating its suitability for integration into existing systems.

Keywords: XSS, Cross-Site Scripting, Deep Learning, Evolutionary Neural Networks

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Dr. Öğr. Üyesi Ahmet Haşim YURTTAKAL, araştırma ve yazım süresince yardımlarını esirgemeyen eşim Hilal ÇELİK'e her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma, arkadaşlarıma teşekkür ederim.

Bu araştırma boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Selim ÇELİK

Afyonkarahisar 2024

İÇİNDEKİLER DİZİNİ

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

Kısaltmalar

XSS	Cross-Site Scripting
OWASP	Open Worldwide Application Security Project
ID	İdentification
HTML	HyperText Markup Language
HTTP	Hyper-Text Transfer Protocol
JS	JavaScript
DOM	Document Object Model
ISR	Intelligence, Surveillance and Reconnaissance
SVM	Support Vector Machines
KNN	k-nearest neighbors
URL	Uniform Resource Locator
PCA	Principal component analysis
SQL	Structured Query Language
URI	Uniform Resource Identifier
GPU	Graphics processing unit
AJAX	Asynchronous JavaScript and XM
API	Application Programming Interface

ÇİZELGELER DİZİNİ

Sayfa

RESİMLER DİZİNİ

Sayfa

1. GİRİŞ

1.1 Motivasyon

Cross-site scripting (XSS), iyi niyetli ve güvenilen ağ sayfalarına yapılan sokuşturma saldırılarıdır(*Cross Site Scripting (XSS) | OWASP Foundation*, t.y.).Bu saldırı, kötü niyetli kişilerin kurbanlar tarafından görüntülenecek sayfalara zararlı kodlar yerleştirilmesine izin vermektedir. Yerleştirilen kodun başarılı şekilde yürütülmesi sonucunda sayfanın davranışı tamamen değişmektedir. Bu yöntem ile kullanıcının kişisel verileri saldırganlar tarafından çalınabilir (Shrivastava vd. 2016). XSS saldırılarında JavaScript tercih edilmesinin sebebi, günümüzde dinamik web sayfalarının oluşturulmasını sağlayan en önemli teknolojilerden biri olmasıdır (Swaswati Goswami vd. 2017). Bu tip saldırılara karşı sunucu tarafında alınan önlemler geliştiricilerin her zaman güvenli yazılım geliştirme deneyimleri bulunmaması sebebiyle artık faydalı görülmemektedir (Rodríguez vd. 2020). Ağ güvenlik duvarı, Intrusion Detection System (IDS) ve şifreleme gibi geleneksel güvenlik mekanizmaları ise ağların korumakla beraber web uygulamalarını hedef alan saldırılara bir çözüm üretmemektedir. Saldırganların hedeflerini ağlardan çok, uygulamalara çevirmesi sebebiyle güvensiz yazılmış uygulamalar büyük risk teşkil etmektedir (Fonseca vd. 2007). Dinamik ve statik kod analizi ile bu iki yöntemin birleştirildiği hibrit modeller XSS saldırılarına karşı önleme yöntemleri olarak sunulmaktadır. Dinamik analiz yazılım geliştirme yöntemlerine ek yük getirirken, statik analiz XSS zafiyetlerini tespit etme konusunda hassas değildir (Rodríguez vd. 2020). Yapılan çalışmalar web uygulama güvenliği alanında geleneksel metotların hala modern metotlara karşı tercih edilen yöntemler olduğunu göstermektedir (Rodríguez vd. 2020).

Geçtiğimiz yüzyılda ortaya çıkan internet tabanlı bilgi sistemleri başlarda ağ tabanlı oluşturulmuştur. Bu durum saldırganların önceliğini ağ sistemlerine yöneltmesine sebep olmuş, buna karşın bu alandaki gelişmelerde belirli bir olgunluğu ulaşmıştır. Günümüzde ise ağ merkezli sistemlerde uygulama merkezli sistemlere geçiş ile birlikte saldırganların ağlara ulaşımı zorlaşmış bu durum uygulamaları hedef haline getirmiştir (Curphey ve Arawo 2006).

Yazılım geliştirme ile ağ ve sunucu yönetimindeki geleneksel yöntemlerin kullanımı her ne kadar uygulanması gereken birinci yöntem olsalar da uygulamaların güvenliğinin sağlanması sadece bu yöntemlerle başarılabilir bir hedef olmaktan çıkmıştır. Hızla gelişen uygulamalar, kullanıcı tarafında çalıştırılan betiklerin uygulama kullanımı kolaylaştırması ile birlikte bu betikler XSS saldırılarının da önünü açan en önemli etmenlerdir. Bu betiklerin kullanımını azaltmak bir savunma yöntemi olmaktan çıkmış, savunmanın istemci-sunucu arasındaki haberleşme sırasında zararlı verinin tespiti, ayıklanması uygulamanın gerçek kullanıcıların deneyimlerinin zarar görmemesi en önemli hedef haline gelmiştir. Bu sebeple çalışmamızda derin öğrenme yöntemlerine dayanan bir XSS saldırı tespit yöntemi önerilmiştir. Evrişimli sinir ağları kullanılan çalışma, zararlı ve zararsız kod örnekleri bulunan veri setini %99 başarı oranıyla sınıflandırmıştır.

1.2 Tezin Organizasyonu

Çalışmamızın ikinci bölümü alana katkı sağlayan literatürün taramasını barındırmaktadır. Kuramsal temeller kısmında derin öğrenme, evrişimli sinir ağları gibi kullanılan yöntemlerin ve kavramların tanımları yapılmıştır. Ayrıca XSS saldırı yönteminin temel yapısı, tipleride bu bölümde tanımlanmıştır. Dördüncü bölüm, Materyal ve Metod kısımda kullanılan veri setinin nitelikleri, sınıflandırma için kullanılan yöntem ve yöntemin performansı yer almaktadır. Son olarak tartışma ve sonuç kısmında çalışma sonucu ve öneriler yer almaktadır.

2. LİTERATÜR BİLGİLERİ

Johns (2006) SessioSafe adını verdiği araçta XSS oturum çalma saldırılarına karşı bağışık oturumlar oluşturmak için üç adet sunucu taraflı yöntem kullanılmıştır. Web uygulamalarının güvenli tutmak için kullanılan bu yöntemler; gecikmiş yükleme, tek kullanımlık URL'ler ve alt alan adı değiştirmedir. Bunlar Oturum ID'lerinin iletimini engelliyor, oturumların tekrar oluşturulmasını engellenmekte ve zafiyetlerin şüpheli sayfalar üzerindeki etkisini kısıtlıyor. Bu teknik birçok web uygulama güvenliği prosedürünün temeli olan girdi-çıktı kontrolünün yerine geçmez. Araç sunucu tarafında kaynak kodda değişiklik gerektirmeyen XSS saldırılarını karşı savunma sağlayan şeffaf bir kalkan gibidir.

Kirda vd. (2006) tarafından geliştirilen Noxes, kişisel güvenlik duvarlarına da ilham olan önde gelen istemci tarafı XSS çözümlerindendir. Genel olarak HTTP isteklerini belirlenen güvenlik duvarı kurallarına göre kabul veya reddeder. Üç yöntem ile kural oluşturulabilir. El ile oluşturma yöntemi kural listesine kuralları elle girmek için kullanılır. Güvenlik duvarı istem yöntemi politikalar belirlenir ve bağlantı isteklerinin bu politikalara uygunluğu kontrol edilir. Snapshot yönteminde kurallar otomatik oluşturulur.

Kals vd. (2006) tarafından geliştirilen SecuBat, karakutu yöntemine dayanan web uygulamalarını tarayarak sömürülebilir XSS zafiyetlerini tespit eden açık kaynak bir web uygulama zafiyet tarayıcıdır. Bu tarayıcı üç ana bileşenden oluşur. İlki hedef web siteleri tarayan crawler bileşenidir. İkincisi ayarlanan saldırıyı tespit edilen sitelere uygulayan saldırı bileşenidir. Yazarlar SQL Injection, basit yansıtılmış XSS, şifrelenmiş yansıtılmış XSS, form yönlendirmeli XSS olmak üzere dört saldırı bileşeni eklemiştir. Sonuncu olarak web sitelerinde dönen cevapları tarayarak saldırının başarısını gösteren analiz bileşenidir.

Johns vd. (2008) XSSDS adını verdikleri pasif ve sunucu tarafında çalışan bir teknik geliştirmiştir. Teknik http istekleri ve cevapları arasında sapmayı ölçmektedir. Öncelikle kalıcı olmayan XSS saldırılarını, http isteklerini ve sunucu tarafından üretilen http cevaplarını analiz ederek tespit etmektedir. Daha sonra kalıcı XSS saldırılarını

uygulamadaki tüm scriptlerin kaydını tutup değişimleri takip ederek tespit etmektedir. Yazarlar uygulamada kullanılan tüm betiklerin kaydını tutarak eğitilen eğitim temelli bir kalıcı XSS saldırı tespit yöntemi geliştirmiştir. Kayıtlarda bulunmayan bir betik tespit edildiğinde XSS saldırısı olarak raporlanmaktadır.

Bisht ve Venkatakrishnan (2008) XSS-GUARD adını verdikleri çözüm ile bir web uygulamasının HTML web isteklerini oluşturmak için kullandığı betikleri keşfederek XSS saldırılarına karşı savunma sağlıyor. XSS-GUARD aynı zamanda betikleri keşfetme ve http cevabı ile alakalı olmayan betikleri ayıklama içinde bir metod sunuyor. Her http isteği için gerekli betikleri öğrenmek için web sayfasının bir kopyasını oluşturuyor. Gerçek web sitesi ve kopya web sitesi arasındaki her fark sokuşturulmuş bir betik olma ihtimalini gösteriyor.

Wurzinger vd. (2009) tarafından geliştirilen yöntemde web uygulamasındaki tüm betikler şifrelenerek sözdizimsel olarak geçersiz belirleyicilere dönüştürülür. Betik tespit etme sistemine gönderilen kodlar eğer sonradan eklenen bir script tespit edilmezse ters vekil sunucu tarafından şifresi çözülerek istemci tarafına gönderilir. Eğer şifrelenen betikler haricinde bir javascript içeriği tespit edilirse vekil sunucu istemciye HTML cevabı vermek yerine bir uyarı gönderir.

Louw ve Venkatakrishnan (2009) tarafından gelişitirilen BLUEPRINT, sunucu taraflı çalışan, XSS saldırılarını engellemek için farkı tespit etmek amacıyla çıktı HTML belgenin kabul edilebilir girdiler ve kullanıcı girdilerinden oluşan iki kopyasını tarayıcıya ileten bir çözümdür. Bu yaklaşım ağdaki güvensiz içeriklerin tespitinde tarayıcı bağımlılıklarını azaltır. Bu teknik ile güvensiz HTML içerikler için sunucu tarafında betik içeriklerin bulunmadığında emin olmak için bir ayrıştırma ağacı oluşturulur. Bu ayrıştırma ağacı web tarayıcının belge oluşturucusuna gönderilerek javascript yorumlayıcının çalışmadığından emin olunur.

Shahriar ve Zulkernine (2011) tarafından geliştirilen S^2XS^2, boundary injection adı verilen dinamik olarak üretilen içerikleri en kapsüle etmeye ve veriyi doğrulamak için politikalar üretmeye dayanan bir yöntem izliyor. Boundary injection fikri HTML

etiketlerini, javascript içerikleri ve tahmin edilen diğer içerikleri inceler. Bu tahmini geçerli içerikler http cevaplarının üretim aşamasından itibaren incelenerek XSS saldırıları keşfedilir. JSP programlar için çalışan bu yöntem için birde prototip araç geliştirilmiştir.

Shahriar and Zulkernine (2011) rastgele tokenler ve faydalı javascript kodu karakteristiği taşıyan yorum ifadeleri eklemek üzerine dayanan sunucu taraflı bir kod sokuşturma tekniği geliştirmiştir. Bir http cevabı oluşturulduğunda, hatalı javascript yorumları veya yorum ifadelerinin hiç olmaması sokuşturulmuş bir kod olduğunu gösterir. Ayrıca başta üretilen geçerli yorumlarda kopyalarının olup olmadığına ilişkin kontrol edilir. Yorumların birden fazla olması veya benzer kodlar arasındaki farklar javascript konunun sokuşturulmuş olduğunu gösterir. Yazarlar ayrıca javascript yorumlarını yerleştiren bir prototip geliştirmiş ve sunucu tarafında tespit işlemlerini yapan bir detektör kurmuştur.

Gundy ve Chen (2012), XSS zafiyetlerini önlemek için Komut Seti Rastgeleleştirme(Instruction Set Randomization- ISR) ismi verilen yöntemi kullanarak uçtan uca çalışan, tarayıcıların iyi niyetli ve zararlı içerikleri ayırt etmesini sağlayan bir mekanizma geliştirmiştir. Noncespaces isimli mekanizme HTML etiketlerini tarayıcıya göndermeden önce karıştırarak, sokuşturulmuş betikleri ayırt etmeye çalışır. Karıştırma işlemi iki amaçla yapılır: Zararlı içerik bu yöntem ile tespit edilir ve tarayıcı bu içeriğin çalışmasını kısıtlar. İkinci olarak zararlı içeriğin DOM ağacını bozması engellenir. Çünkü karıştırılmış HTML etiketleri saldırgan tarafında kolay tahmin edilemez. Böylece sokuşturmak istenen içerik yanlış yerlere yerleştirilir ve ayıklama hatası oluşması sağlanır.

Kaur vd. (2018) kurbanın tarayıcısında çalıştırılmadan önce saldırı vektörünü tespit eden makine öğrenmesi tabanlı bir model geliştirmiştir. Kör XSS ve Saklanmış XSS saldırılarını tespit etmek için Doğrusal Destek Vektör sınıflandırma algoritması kullanılmıştır. Yazarlar özellik üretme aşamasında saldırganın siteye eklediği javascript olaylarını ve betiklerini durdurmayı başarmıştır. Mutilliade adlı test sitesi kullanılarak yapılan deneylerde doğrusal ayrılabilir veri kümesi kullanılmıştır. Çalışmada %95.4 lük bir başarı oranı görülmüştür.

Sharma vd. (2020) göre özellik kümesi çıkarımı web tabanlı saldırı tespitinde çok önemlidir. Yazarlar makine öğrenmesi yöntemleri ile birlikte uygulandığında sonuçlarda önemli gelişmeler yol açan bir özellik seti çıkarma yaklaşımı öne sürüyor. Deneyler CSIC HTTP 2010 veri seti ve Weka kullanılarak gerçekleştirilmiş. Çalışmanın birinci aşaması veri setinin ön elemeden geçirilmesi, ikinci aşaması ise veri setinden GET,POST, DROP,DELETE gibi anahtar kelimeleri içeren özelliklerin çıkarılmasına yoğunlaşıyor. Son bölümde veri J48,OneR ve Naive Bayes yöntemleri ile Weka'da işleniyor. Çalışma sonucunda J48 algoritması en başarılı sonuçları üretmiştir.

Wang vd. (2015b), sosyal medya platformlarındaki XSS wormlarını tespit eden bir yaklaşım geliştirmiştir. Zararlı ve zararsız web sayfaları birbirinden farklı olup, betik kullanım sıklığı ölçülerek birbirinden ayrılabilir. Araştırmacılar modellerinde ADTree karar ağacı ve AdaBoost.M1 algoritmalarını kullanmıştır. ADTree diğer karar ağaçlarına göre gösterdiği yüksek hassasiyetten, AdaBoost.M1 ise güçlü bir sınıflandırıcı üretmesinden dolayı tercih edilmiştir. Özellik çıkarma ve veritabanı oluşturma sınıflandırma modeli üretmede kritik rol oynadığından, web sayfalarından özellikleri yakalayan otomatik bir yakalayıcı geliştirilmiştir. Zararlı ve zararsız örnekler DMOZ ve XXSed verisetlerinden toplanmıştır. Araştırmacılar web sayfalarından anahtar kelime özellikleri, Javascript özellikleri, HTML etiket özellikleri ve URL özellikleri olmak üzere 4 grup özellik çıkarmıştır. Çalışmada sınıflandırma modeli oluşturma ve değerlendirme için Weka aracı, değerlendirme için 10-fold cross validation metodu kullanılmıştır. Fakat önerilen teknik %4.20 gibi yüksek bir false-positiflik oranına sahip olup, sığ bir tespit oranı sunmaktadır.

Kascheev ve Olenchikova (2020) XSS saldırılarını tespit etmek için eğitim örneklerini %70 ini kullanan test örneklerini %30 unu üreten cross-validation metodunu kullanan gözetimli makine öğrenmesi algoritmelarını öne sürmüştür. Modellerin performansı doğru yanıt, hassasiyet, bütünlük ve F-puanına göre değerlendirilmiştir. Veriseti 40.000 zararlı, 200.000 zararsız kod örneği içermektedir. Öncelikle tüm istekler Unicode karakterlere dönüştürülüp, sorgu cümlelerini çıkarmak için düzenli ifadeler kullanılmıştır. Normal olmayan çok fazla parametre içeren sorgular veri setinden çıkarılmıştır. Kaynak veri Word2Vec olarak ifade edilmiştir. Önişlem olarak t-SEN

kullanılmıştır. Karar ağacı, Naive Bayes sınıflandırıcı, Logistic regresyon ve support vector machine olmak üzere 4 makine öğrenmesi algoritması karşılaştırılmıştır ve karar ağacı en iyi performansı göstermiştir. Çalışmada doğruluk, hassasiyet, geri çağrıma ve f-puanı metrik olarak kullanılmıştır.

Banerjee vd. (2020) SVM, KNN, Random Forest ve Logistic Regression olmak üzere dört makine öğrenmesi algoritmasının XSS ataklarının tespitinde kullanımını incelemiştir. Veri kümesindeki doğru ve yanlış değerler logistic regression modeli ile işaretlenmiştir. Deney python ve scikit kütüphanesi kullanılarak URL ve Javascript özellikli 24 nitelik içeren veri kümesi üzerinde yapılmıştır. Dört sınıflandırıcıdan Random Forest sınıflandırıcı yüksek hassasiyet ve düşük false-pozitif oranı ile umut vaad etmektedir.

Rathore vd. (2017) Facebook ve Twitter gibi sosyal medya sitelerindeki XSS saldırılarını tespit eden XSSClassifier isimli sınıflandırıcıyı öne sürmüştür. Yaklaşımları saldırı tespiti için makine öğrenmesi sınıflandırıcılarının kullanımını içermektedir. Öncelikle özellikler URL'lerden, sosyal medya sitelerinden ve HTML içeriklerden çıkarılır. Değerlendirme aşamasında tespit modeli sosyal medya sitesi özellikleri ile beslendiğinde yüksek hassasiyet gösterdiği görülmüştür. Tespit modelinin performans ölçümü için 10 makine öğrenmesi sınıflandırıcı kullanılırken Random forest ve ADtree %97.2 doğruluk ve 0.87 false-pozitiflik oranıyla gerçek zamanlı olarak en iyi performansı göstermiştir.

Khan vd. (2017) XSS saldırı tespiti için SVM, KNN, J48 ve Naive Bayes olmak üzere dört makine öğrenmesi sınıflandırıcı kullanmıştır. Veri seti eğitim ve test olmak üzere ayrıldığında, J48 %99.22 lik doğruluk göstermiştir. Çalışmada sunucu ve tarayıcı arasında güvenli iletişimi sağlayan, araya giren bir eklenti geliştirilmiştir. HTTP GET istekleri sunucuya eklenti tarafından gönderilir. Eğer Javascript kodu içerisinde zararlı bir aktivite tespit edilirse, web sayfası tarayıcıya gönderilmeden önce engellenir. Yaklaşım tarayıcı ve platform bağımsızdır ve gerçek zamanlı olarak saldırıları tespit edebilmektedir. Ayrıca doğası gereği düşük çalışma gereksinimlerine ihtiyaç duymaktadır. Sınıflandırıcıya verilen özellikler statik analizle çıkarılmıştır. Tespit modelinde web sayfalarının kaynak kodlarını token listelerine dönüştürmek için bir yacc

ayıklacısı kullanılmıştır. Son olarak saldırganlar tarafından zararlı aktiviteleri gizlemek amacıyla karıştırma uygulanan kodları tespit etmek için lexical analiz kullanılmıştır.

Wang vd. (2015) sosyal medya sitelerindeki XSS zafiyetlerini tespit etmek için n-gram modelini kullanan bir modelin geliştirilmiş sürümünü öne sürmüştür. Özellik çıkarım amacıyla korele edilmiş bir veri seti oluşturulmuştur. Özellikler URL, Javascript ve HTML etiketleri ile ilgili anahtar kelimelere dayanılarak çıkarılmıştır ve sınıflandırıcı ADTree kullanılarak geliştirilmiştir. Tespit modeli iki defa itere edilmiş, önerilen model ikinci iterasyonda zararlı örneklerin tespiti için kullanılmıştır.

Laghrissi vd. (2021) LSTM sınıflandırma tabanlı üç adet IDS'in karşılaştırmasını yapmıştır. LSTM, LSTM-PCA ve LSTM-MI adı verilen IDS'lerden PCA tabanlı olan model en iyi sonucu üretmiştir. KD99 veri seti, Google Colab'da üç kategoriye (test verisi, eğitim verisi, doğrulama verisi) ayrılarak kullanılmıştır. Değerlendirme için F-1 puanı doğruluk, duyarlılık ve hassasiyet kullanılmıştır.

Hoang (2021) SQLi,XSS,Path traversal ve CMDi saldırılarını karşı web iz kayıtlarına dayanan bir makine öğrenmesi modeli önermiştir. 20000 payload içeren HTTP Param veri seti eğitim seti olarak, 11067 payload test seti olarak kullanılmıştır. Tespit aşamasında n-gram modeli kullanan tespit yöntemine girilmek üzere öncelikle URI'lar web iz kayıtlarından çıkarılmıştır. İkinci aşamada URI'lar TF-IDF metodu ile vektör gösterime dönüştürülmüştür. Son olarak özelliklerin sayısını azaltmak için PCA metodu kullanılmıştır. Gerçek zamanlı modellerde en iyi sonucu verdiği için eğitim amacıyla CART karar ağacı modeli uygulanmıştır. Tespit amacıyla eğitim aşamasında üretilen sınıflandırıcı URI vektörlerini normal ve saldırı sonuçlarını üretmek üzere kullanılmıştır. Deneyler %98,52 tespit oranıyla sonuçlanmıştır.

Fang vd. (2019) XSS saldırı tespiti için pekiştirmeli öğrenme kullanarak RLXSS'i geliştirdi. Yazarlar karşı saldırılar ile başa çıkacak ve önceki modeller tarafından tanımlanan zararlı örnekler ile beslenen rakip ve yeniden öğrenen modeller fikrini ortaya attı. Optimizasyon için karşı modelden toplana karşı örnekler, yeniden eğitim modeline verildi. Çalışmada hassas kelimelerin değiştirilmesi, kodlamalı karıştırma, konumsal

morfolojik dönüşüm ve özel karakterler ekleme olmak üzere dört yeni saldırı savunma yöntemi önerildi.

Literatürde yapılan çalışmalar incelendiğinde, siteler arası betik saldırı tespit sistemlerinde klasik makine öğrenmesi yöntemlerinin sıklıkla kullanıldığı görülmektedir. Sınıflandırma problemlerinde yüksek başarı oranı yakalayan derin öğrenme yöntemlerinin kullanılması büyük önem arz etmektedir.

3. MATARYEL ve METOD

3.1 Yapay Zekâ

Yapay zeka basitçe tanımlanması zor bir terim olmakla birlikte, 1955 yılında alanın öncülerinden John McCarthy tarafından yapay zekanın amacı akıllıymış gibi davranan makinalar geliştirmek olarak açıklanmıştır (Ertel 2018). Yapay zekanın direkt bir tanımı ise şuanda insanların iyi olduğu şeyleri makinaların yapmasıdır (Rich 1983). Terimi tanımlamaktaki güçlük zeka (intelligence) kelimesinden ileri gelmektedir. Sözlükte deneyimlerden öğrenme, anlama ya da bilgiyi almak ve unutmamak olarak tanımlanmaktadır. Zekanın makinalara uygulanması her ne kadar mümkün olsa da sözlük anlamıyla uygulanması mümkün değildir. Zekâ kavramının tanımında yaşanan güçlük, makinaların zeki olup olmadığını anlama noktasında da baş göstermektedir. Bu problemi çözmek için birçok yöntem önerilmiş olmakla birlikte bu yöntemlerden en bilineni geliştiricisi Alan Turing'den adını alan Turing Testi'dir. Turing Testi Alan Turing'in imitation game adını verdiği bir oyuna dayanır. Örnek vermek gerekirse; bir gözlemci, zeki olduğu gösterilmek istenen makine ve bir insan ile mülakata sokulur. Makine gerçek insanı taklit etmeye çalışır. Gözlemci eğer hangisini makine olduğunu anlayamazsa makine testi geçmiş kabul edilir (Fetzer 1990).

2016 yılında AlphaGo'nun dünya şampiyonunu yenmesiyle birlikte alana olan ilgi hızla arttı. Yapay zekanın gelişimi insanlığa büyük ekonomik getiriler sağlamakla beraber yaşamın pek çok alanında da getiriler sağladı. Toplumsal gelişim, iş gücü tasarrufu ve yeni iş kolları bunların en belirgin örnekleridir (Zhang ve Lu 2021).

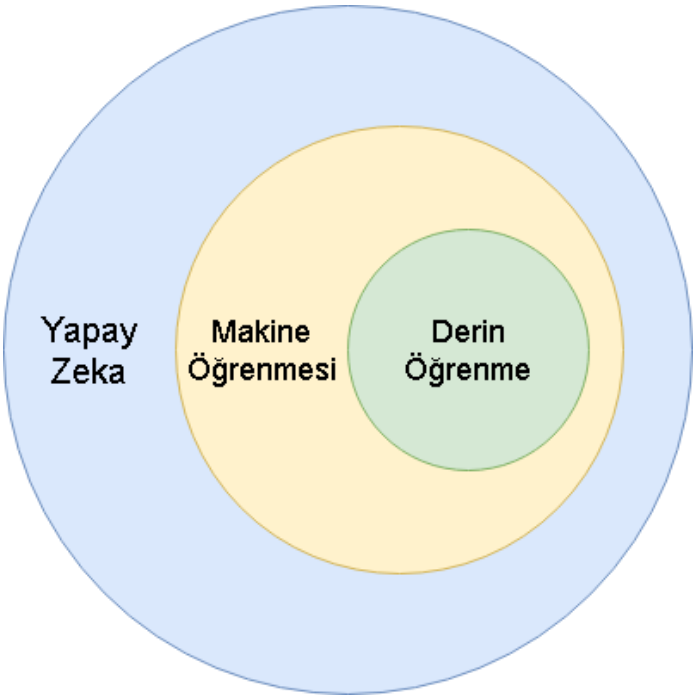
Bu hızlı gelişime ayak uyduran ülkeler uluslararası rekabetin yeni alanlarından biri olan yapak zekâ için politikalar geliştirmeye başladı. Ülkelerin yanı sıra pek çok teknoloji devi şirket te yapay zekayı ürünlerine entegre etti ve alan ile ilgili direkt ürünler çıkardı (Zhang ve Lu 2021).

İlk çıkışından itibaren sürekli gelişimini sürdüren yapay zekâ, bilgisayar işlem güçlerinin sınırlarına pek çok defa takılarak uygulamalı alanda duraksamalar yaşamıştır. Bu

duraksamalar 1997 yılında IBM Deep Blue gibi atılımlarla aşılmış alanın gelişimi tekrar hızlanmıştır. Son yıllarda artan GPU hesaplama kapasiteleri ve özel olarak yapay zeka için geliştirilen hesaplama birimleri ile beraber gelişim hız tekrar gözlemlenir şekilde artmıştır (Zhang 2019).

3.2 Makine Öğrenmesi

Makine öğrenmesi, verilen girdilerden istenen çıktıları bilinen anlamda programlanmadan veren hesaplama süreçleridir. Makine öğrenmesi algoritmaları tekrarlamalar, bir anlamda deneyimler sonucu kendini geliştir, mimarisini değiştirir ve sonuçta istenen görevleri yapmakta daha iyi konuma gelir. Girdi verileri ve istenen çıktıların verildiği bu gelişim ve değişim sürecine eğitim adı verilir. Eğitim sonucunda algoritma eğitim sürecindekinde farklı veriler verildiğinde de istenen sonuçları çıkarabilmelidir. Bu eğitim kısmı makine öğrenmesindeki öğrenme kısmıdır. Başarılı bir makine öğrenmesi algoritması sadece eğitim sürecinde verilen verilerle kendini geliştirmesinin yanında, aynı insanlarda olduğu gibi çalışma sırasında aldığı verileri de gelişim için kullanabilir. Bu şekilde çalışan bir algoritma yaşam döngüsü boyunca ürettiği hatalı çıktıları düzelterek kendi başarımını maksimize edebilir (El Naqa ve Murphy 2015).



Resim 3.2 Derin öğrenme ve yapay zekâ arasındaki hiyerarşik ilişki

Yukarıda öğrenme süreci olarak bahsedilen aşamanın uygulanması için 3 adet yöntem bulunur: gözetimli öğrenme, gözetimsiz öğrenme, takviyeli öğrenme. Birinci yöntem olan gözetimli öğrenmede, her eğitim verisi bilinen bir sınıflandırmaya tabi tutulur. Böylece algoritma verilen girdilerde her ne kadar çok fazla değişken bulunsa da eğitim sürecinde çıkardığı temel benzerlikleri kullanarak sınıflandırma yapabilir. Burada önemli olan nokta daha önce hiç karşılaşılmamış verilerinde doğru sınıflandırılabilmesidir. Gözetimsiz öğrenme olarak adlandırılan ikinci yöntemde, eğitim sürecinde algoritmaya verinin istenen çıktılara ulaşacağı yöntem hakkında hiçbir bilgi verilmez. Algoritma iterasyonlar sonucu en uygun yöntemi kendisi bulmalıdır. Her tekrar sonucu parametrelerde yapılacak ayarlamalar sonucu algoritma istenen çıktıya kendi bulduğu yöntem ile erişir. Üçüncü yöntem olan takviyeli öğrenmede eğitim verilerini bir kısmı etiketli bir kısmı etiketsizdir. Etiketli olan veri etiketsiz olan verinin öğrenilmesi aşamasında yardımcı olarak kullanılır. Doğadaki pek çok süreç ve insan öğrenmesi de aslen bu şekilde gerçekleşmektedir (El Naqa ve Murphy 2015).

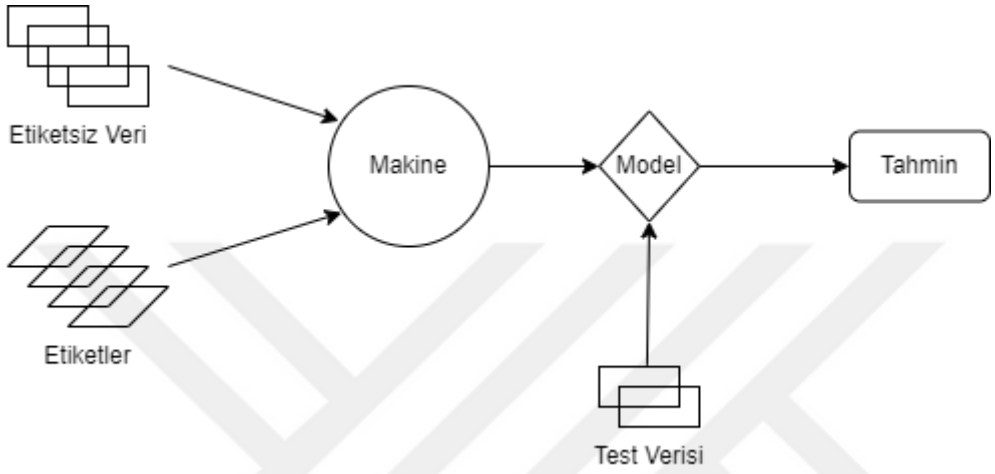
3.2.1 Denetimli Öğrenme

Denetimli öğrenme girdi değişkenlerinin çıktı değişkenleri ile eşleştirildiği bir ver setinin öğrenilmesi ve buradan yola çıkarak daha önce görülmemiş girdi verileri için çıktıların tahmin edilmesine dayanır. Cunningham vd. (2008) Denetimli öğrenme sınıflandırma problemleri için en çok tercih edilen yöntemdir. Bu yöntemde özellikleri verilen nesnenin etiketlerini tahmin eden bir tahminci geliştirilmesi gerekmektedir. Daha sonra algoritma girdi verileri ile birlikte düzeltilmiş çıktıları alır ve gerçek çıktılar ile düzeltilmiş çıktıları karşılaştırarak hatalı bulur. Modeli bu sonuçlara göre değiştirir (Nasteski 2017).

Denetimli öğrenmede görevler ikiye ayrılır: sınıflandırma ve tekrarlama. Sınıflandırmada etiketler ayrıkken tekrarlamada etiketler süreklidir. Algoritma eğitim sürecinde verilen x diyeceğimiz veriler arasında ayrımı yapar. Bu aşamada denetimli öğrenme algoritması tahmini modeli oluşturur. Eğitim sürecinden sonra son model test verisindeki yeni bir x verisinin etiketlerini tahmin etmeye çalışır. Çıktı olan y'nin tipine bağlı olarak:

- Eğer y kategorik çıktı değerlerine sahipse (yani tam sayı ise), y yi tahmin etme işlemi sınıflandırma olarak,
- Eğer y ondalıklı değerler sahipse, y'yi tahmin etme işlemi tekrarlama,

olarak adlandırılır (Nasteski 2017).



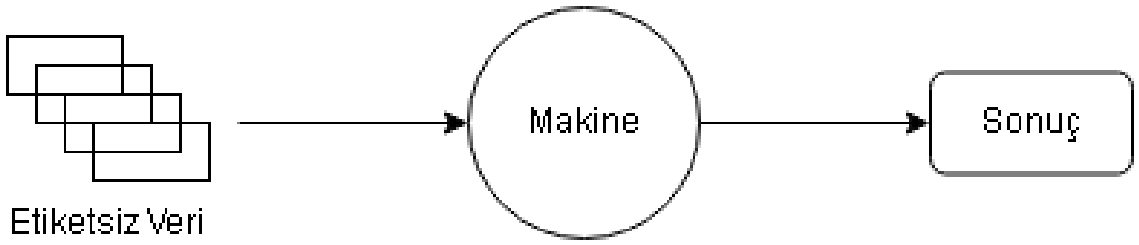
Resim 3.2.1 Denetimli öğrenme akış şeması

Denetimli öğrenme yönteminde algoritma olarak karar ağaçları, doğrusal regresyon, Naive Bayes, Lojistik regresyon gibi çeşitli algoritmalar kullanılır. Bu algoritmalar veri bilimciler tarafından sürekli geliştirilmektedir. Denetimli öğrenme yöntemi, etiketli verilerin model optimizasyonu için daha belirgin kriterler sunması sebebiyle denetimsiz öğrenme yöntemlerine göre daha başarılı bulunur. Bu sebeple makine öğrenmesi alanında önemli yöntemlerden biri olarak kabul edilir (Nasteski 2017).

3.2.2 Denetimsiz Öğrenme

Denetimsiz öğrenme, verilen x veri seti için denetimli bir hedef çıktı olmadan veya herhangi bir geri besleme verisi olmadan çıktının tahmin edilmesi yöntemidir. Geri besleme olmadan makinanın öğrenmesi mümkün değil gibi gözükse de, temiz olmayan verinin içinden bir örüntü çıkarmak mümkündür. Denetimsiz öğrenmenin en bilinen iki yöntemi kümeleme ve boyut indirgemedir (Ghahramani 2004).

Denetimsiz öğrenmede aslen yapılan iş verinin olasılıksal modelinin öğrenilmesidir. Bir denetim veya geri besleme olmadığında dahi verilen yeni bir x verisi için olasılıksal dağılım önceden verilen veriler üzerinden çıkarılabilir. Bu sebeple denetimsiz öğrenme borsa veya hava durumu tahmini için uygun bir yöntemdir (Ghahramani 2004).



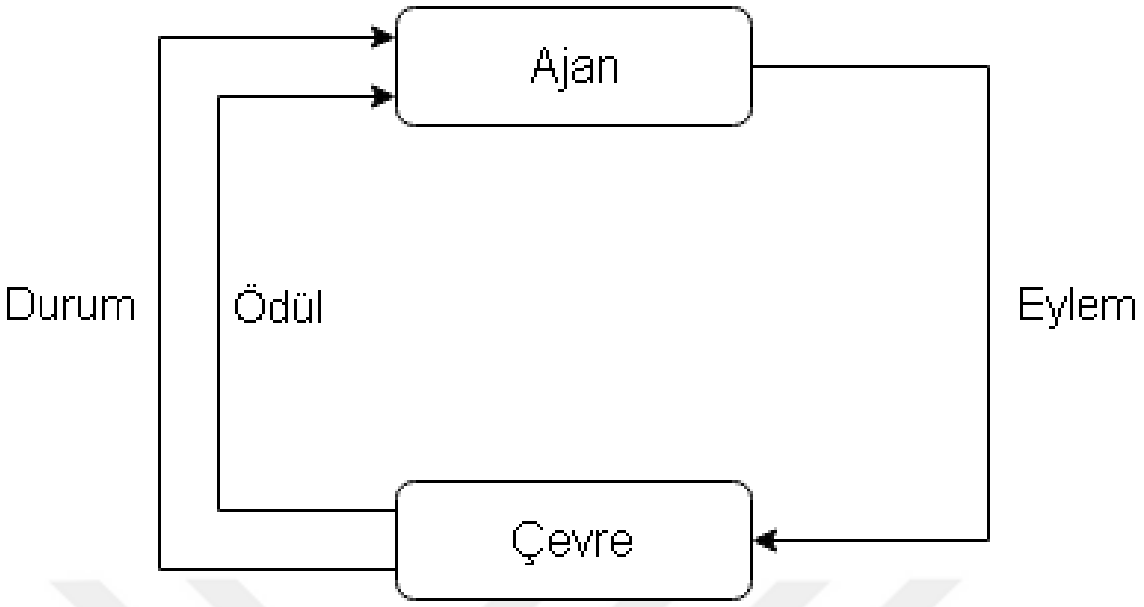
Resim 3.2.2 Denetimsiz öğrenme akış şeması

Bu tür bir model aykırı durumların tespiti veya izleme işlerinde kullanılabilir. X in bir tesiste kurulan yeni bir sensör, P(x) inde normal çalışan bir tesisten alınan verilerden üretilen model olduğunu düşünürsek, P(x) yeni sensörden alınan verileri değerlendirmek için kullanılabilir. Eğer elde edilen olasılık normal olmayan şekilde düşükse burada ya modelin zayıf olduğu yada tesisin normal işlemediği düşünülür (Ghahramani 2004).

3.2.3 Takviyeli Öğrenme

Takviyeli öğrenme denetimli ve denetimsiz öğrenmenin arasından kalan bir yöntemdir. Aslında birçok takviyeli öğrenme yöntemide aslında denetimli veya denetimsiz öğrenme yöntemlerinin ek bilgilerle genişletilmesine dayanmaktadır (Zhu ve Goldberg 2009).
Takviyeli öğrenmede temelde iki yöntem uygulanmaktadır: takviyeli sınıflandırma ve kısıtlı kümeleme (Zhu ve Goldberg 2009).

Takviyeli sınıflandırmada, sınıflandırma etiketli ve etiketsiz veriler kullanılarak yapılır. Denetimli sınıflandırmaya dayanan bu teknikte temel olarak etiketsiz verinin etiketli veriden daha fazla olduğu varsayılır. Bu teknikle eğitilen sınıflandırıcıların sadece etiketli veriler ile eğitilen denetimli sınıflandırıcılardan daha iyi olması amaçlanır (Zhu ve Goldberg 2009).



Resim 3.2.3 Takviyeli öğrenme akış şeması

Kısıtlı kümeleme, denetimsiz sınıflandırıcıların geliştirilmiş halidir. Eğitim materyali etiketsiz verilerin yanında etiketli olan bir miktar denetimli bilgide içerir. Burada amaç etiketsiz verilerden yapılan kümelemeden daha iyi bir kümeleme yapmaktadır (Zhu ve Goldberg 2009).

Takviyeli öğrenmede ayrıca, etiketli ve etiketsiz verilerle regresyon, etiketli veri ile boyut indirgeme gibi çeşitli yöntemlerde kullanılmaktadır (Zhu ve Goldberg 2009).

Takviyeli öğrenme daha iyi bilgisayar algoritmaları oluşturmak konusundaki başarısı ve makine ve insan öğrenmesi açısından oluşturduğu çıktılar sebebiyle ilgi gören bir alandır (Zhu ve Goldberg 2009).

3.3 Derin Öğrenme

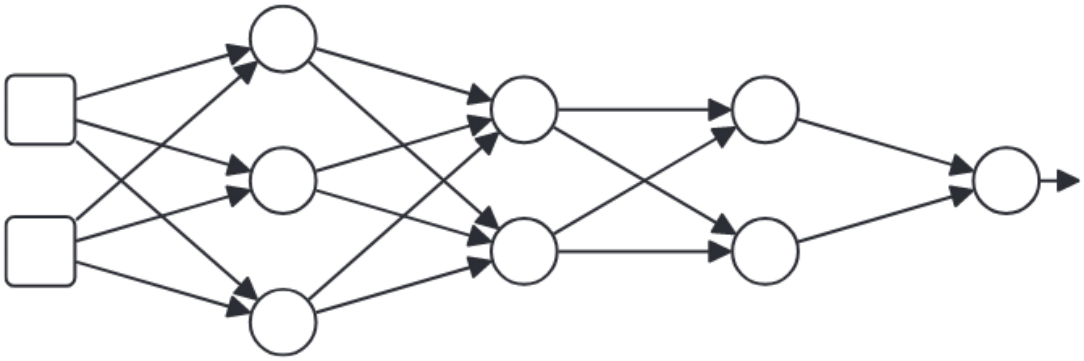
Derin öğrenme veri temelli doğru kararlar vermek için büyük sinir ağları oluşturmaya dayanan bir yapay zeka alt dalıdır. Derin öğrenme veri setinin büyük ve karmaşık olduğu durumlar için uygun yöntemdir. Örneğin Facebook online yazışmalardaki metinleri analiz etmek için; Google, Baidu ve Microsoft ise görsel arama ve metin çevirileri için derin öğrenmeyi kullanmaktadır. Günlük hayatımızda da izlerini gördüğümüz derin

öğrenme telefonlarımızdaki sesi metne dönüştürme, yüz tanıma gibi uygulamalarda da kullanılmaktadır. Ayrıca son zamanlarda standart haline gelen akıllı araçlarda konumlama, haritalama, sürüş yardımcıları gibi özellikler ile kendi kendini süren araçlardaki neredeyse tüm uygulamalar derin öğrenme yöntemlerini kullanmaktadır. Bunların yanında sağlık alanında medikal görüntülerin işlenmesi ve hastalıkların tespiti de derin öğrenmenin uygulama alanlarındandır (Kelleher 2019).

DeepMind taraından geliştirilen AlphaGo 2016 ve 2017 yılında Go oyununda gösterdiği başarılarla derin öğrenmenin en bilinen uygulamalarından biri oldu. AlphaGo dan önce bir bilgisayarın go oyununda insanı yenmesinin zor olduğu, bunun için yapılması gereken programlama faaliyetinin go'nun satrançtan çok daha fazla olasılık içermesi sebebiyle çok uzun süreler alacağı düşünülmekteydi. Fakat derin öğrenmenin gelişmesi ile birlikte bilgisayarın go oyununda amatör seviyeden dünya şampiyonu seviyesine çıkması sadece yedi yıl sürdü (Kelleher 2019).

Derin öğrenmede önemli yeri olan fonksiyonlar girdilerin çıktılara haritalanmasıdır. Bu tanımda makine öğrenmesinin temel amacı olan verilerden fonksiyonları öğrenme dikkate alınmıştır. Fonksiyonlar basit aritmetik ifadeler olabileceği gibi, eğer-değilse ifadelerinden oluşan uzun bir zincir veya çok daha karmaşık gösterimler olabilir. Fonksiyonları ifade etme yöntemlerinden biri sinir ağlarıdır. En başta da ifade edildiği gibi derin öğrenme, makinesi öğrenmesinin derin sinir ağları üzerine yoğunlaşan bir alt alanıdır. Derin öğrenme algoritmalarının veri setlerinden çıkardığı desenler sinir ağları olarak gösterilen fonksiyonlardır. Resim 3.2.1 yapay sinir ağlarının bir gösterimidir. Sol taraftaki kareler girdileri gösteren hafıza alanlarıdır. Resimdeki her bir daire nöronları göstermektedir. Her nöron bir fonksiyonu gerçekler yani aldığı girdileri bir çıktı değerine eşleştirir. Oklar ise her nöronun çıktısının hangi yol ile diğer nöronlara iletildiğini gösterir (Kelleher 2019).

Sinir ağları bir fonksiyonu öğrenmek için böl-parçala-yönet stratejisini kullanır. Daha açık ifade ile her nöron basit bir fonksiyonu öğrenir ve sinir ağını oluşturan fonksiyon bu basit fonksiyonların birleştirilmesi ile oluşur (Kelleher 2019).



Resim 3.3 Sinir ağlarının temsili gösterimi

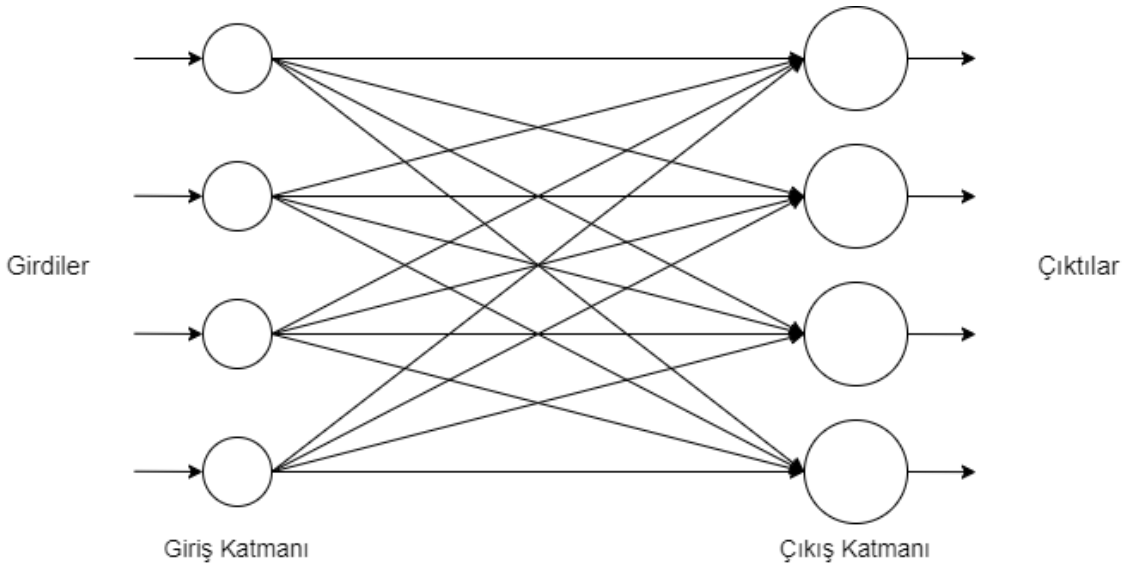
3.4 Yapay Sinir Ağları

Yapay sinir ağları, iki veya daha fazla yapay sinirin birleştirilmesi ile oluşan ağlardır.(Suzuki, 2011) Yapay sinirler, ağlardaki her bir işlem birimine verilen addır. Bu işlem birimleri giriş, çıkış ve gizli olmak üzere üçe ayrılır. Giriş birimleri gerçek dünyadan verileri alır. Çıkış birimleri ise sistem tarafından üretilen sonuçları gerçekler. Gizli birimler ise giriş ve çıkış arasındaki, sistemin dışından gözlemlenemeyen birimlerdir (Wu ve Feng 2018).

Bağımsız yapay sinirlerin birbirine bağlanma şekillerine topoloji denir. Yapay sinirlerin sayısına bağlı olmakla birlikte birbirine bağlanış şekillerine göre pek çok farklı topoloji oluşturulabilmekle birlikte temel olarak topolojiler ileri beslemeli ve yinelemeli olarak ikiye ayrılabilir (Suzuki 2011).

3.4.1 İleri Beslemeli Yapay Sinir Ağları

İleri beslemeli yapay sinir ağları, çıktılardan alınan geri dönüşün ağ üzerinden girdilere aktarılmadığı ağlardır (Sazli 2006). Buradan da anlaşılacağı üzere bu ağlarda veri tek yönlü olarak giriş katmanı, gizli katman ve çıkış katmanı yolunu izler. Doğru seçilmiş bir mimari ile oluşturulan ileri beslemeli yapay sinir ağları pek çok örüntü tanıma görevi için kullanılabilir (Yegnanarayana 2009).



Resim 3.4.1 İleri beslemeli yapay sinir ağları (tek katmanlı)

3.4.1.1 Doğrusal Sinir Ağları

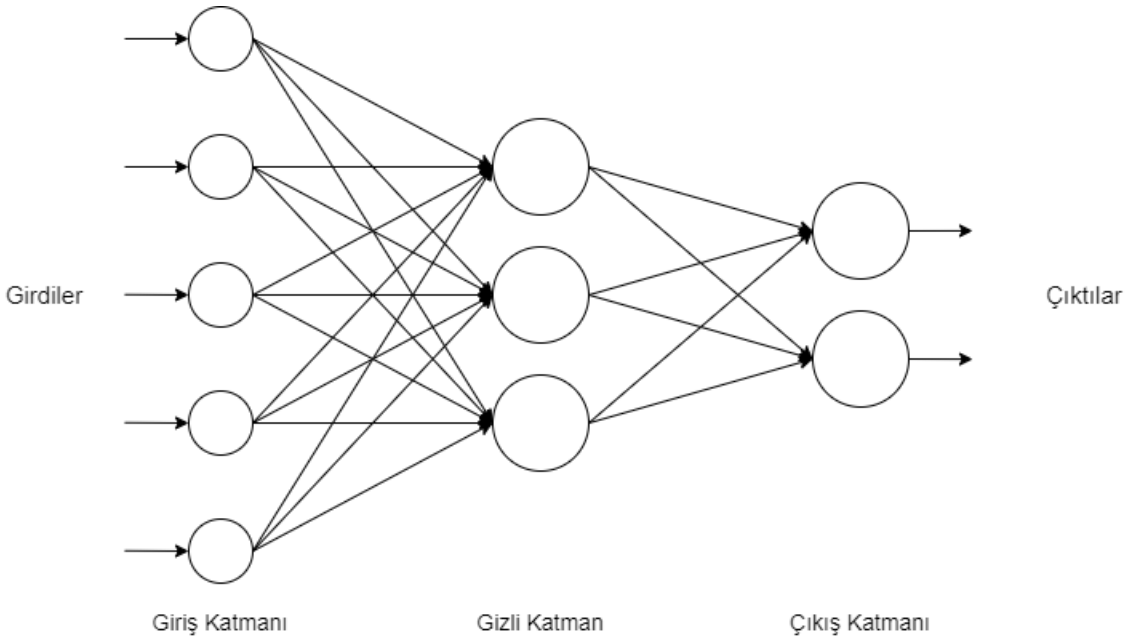
Doğrusal sinir ağları, öğrenme, genelleştirme ve kendini düzenleme gibi alanlarda çalışma yapılabilecek en basit ağ türüdür. Tek katmalı çıkış düğümlerinden oluşan bu ağlarda girdiler direkt olarak çıkış düğümlerini besler (Baldi ve Hornik 1995).

3.4.1.2 Tek Katmanlı Algılayıcılar

Tek katmanlı algılayıcılar sadece girdi ve çıktıdan oluşan ağlardır. Doğrusal fonksiyonun sonucu 1 veya -1 olarak çıkar ve sınıflandırma buna göre yapılır (Öztürk ve Şahin 2018).

3.4.1.3 Çok Katmanlı Algılayıcılar

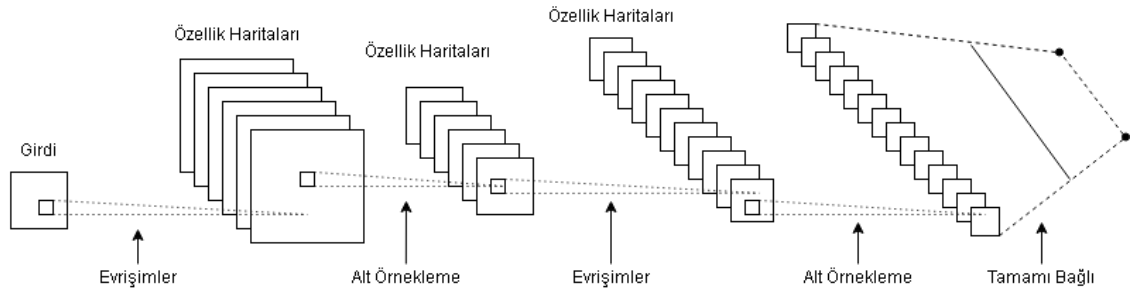
Çok katmanlı algılayıcılar bir giriş vektörü ve çıkış vektörü arasında birbirine doğrusal olmayan şekilde bağlı nöronlar ve düğümlerden oluşur. Düğümler birbirine aktivasyon fonksiyonu tarafından değiştirilmiş giriş değerlerinin toplamından oluşan çıkış ve bağıl değerlerle bağlıdır. Bu algılayıcılar birçok doğrusal olmayan fonksiyonun birleşimi şeklinde çalışarak doğrusal olmayan fonksiyonlara en üst düzeyde yakınsamayı sağlar (Gardner ve Dorling 1998).



Resim 3.4.1.3 Çok katmanlı algılayıcı

3.4.1.4 Evrişimli Sinir Ağları

Canlılardaki görme sisteminden esinlenilen evrişimli sinir ağları, öznitelikleri kendisi kernel adı verilen filtreler sayesinde kendisi öğrenen ağ türüdür. Bu ağ tipi görüntü tanıma, ses tanıma, doğal dil işleme gibi alanlarda iyi performans gösterir. GPU performanslarının artışıyla beraber en büyük atılımı gösteren yöntemlerden birisi olmuştur (Gu vd. 2018).

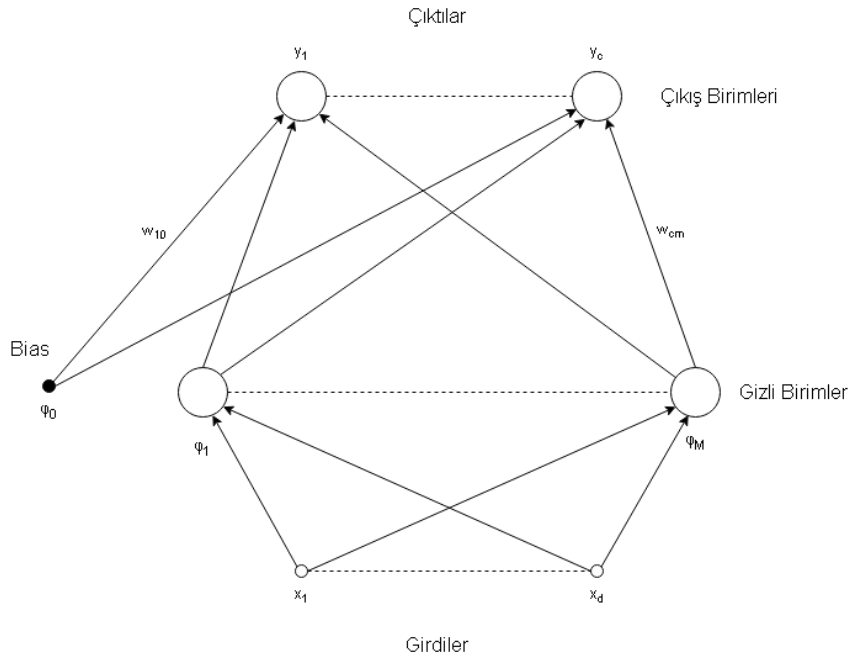


Resim 3.4.1.4 Evrişimli sinir ağı

Klasik yapay sinir ağlarından farklı olarak parametre sayısını düşürmesi sebebiyle daha büyük modellerin işlenebilmesinden dolayı büyük avantajlar sağlamaktadır. Özniteliklere olan bağımlılığın az olması örnek verilecek olursa bir yüz tanıma uygulamasında yüzün verilen imajın neresinde olduğunun önemli değildir. Evrişimli sinir ağları için önemli nokta yüzün pozisyondan bağımsız olarak tespit edilebilmiş olmasıdır. Diğer bir önemli özelliği ise soyut öz niteliklerin katmanlar derinleştikçe daha iyi tespit edilmesidir. Örneğin bir görüntü tanıma uygulamasında ilk katmandan geçişte sadece çekil hatları tespit edilirken ikinci geçişte nesneler, üçüncü katmanda insan yüzleri tespit edilebilir (Albawi vd. 2017).

3.4.1.5 Yarıçapsal Tabanlı Fonksiyon Ağları

Aktivasyon fonksiyonu olarak yarıçap tabanlı fonksiyonları kullanan ağlardır. (Ghosh Nag, 2001) Yarıçap tabanlı fonksiyonlar yaklaşım teoreminin temellerini oluşturur. Bu ağ tipi çok katmanlı algılayıcılara bir alternatif olarak daha basit bir yapı ve daha hızlı bir eğitim süreci sunar. Bu sebeple sistem tanıma, tahmin, yakınsama, sınıflandırma, kontrol, öznitelik çıkartma gibi görevlerde sıkça kullanılır. Yarıçap tabanlı fonksiyon ağlarının temeli çok boyutlu uzaydaki veri kümelerine tam bir interpolasyon uygulama üzerine dayanır (Wu vd. 2012).



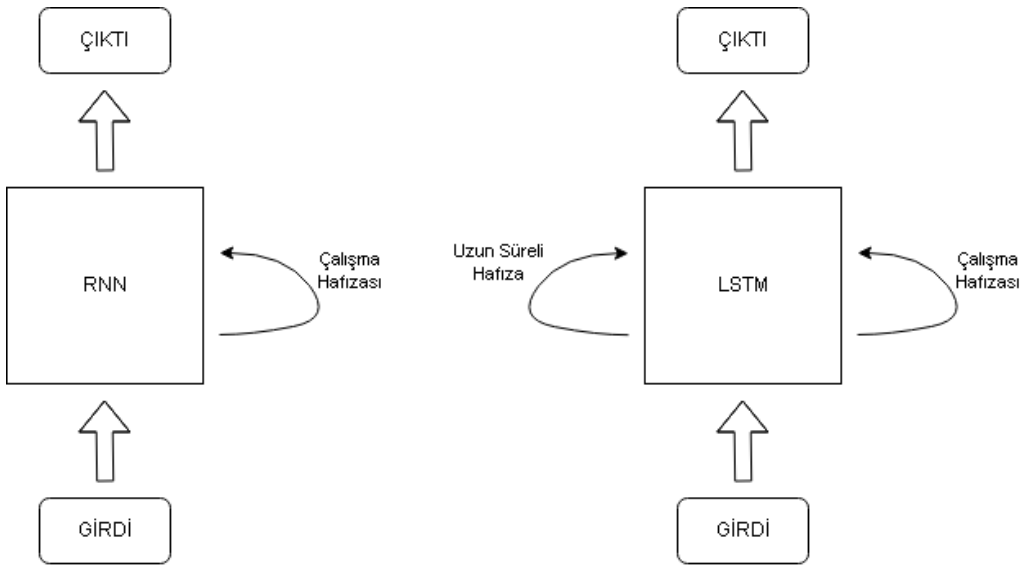
Resim 3.4.1.5 Yarıçapsal tabanlı fonksiyon ağı

3.4.2 Yinelemeli Yapay Sinir Ağları

Sıra tahmini için kullanılan sıralı veriler üzerinde kullanılan yinelemeli yapay sinir ağları adından da anlaşılacağı üzere yinelemeli bağlantılardan oluşan ağlardır. Gizli katmanlarını hafıza gibi kullanan bu ağ tipinde gizli katmanlarının durumu bir önceki durumuna göre sürekli değiştirilir. Bu ağlar bir girdiyi bir çıktıya o anki durumda eşler ve bir sonraki adımı tahmin etmeye çalışır (Salehinejad vd. 2018). Yüz takibi, rüzgar tribünü güç üretim tahmini, ekonomik tahminler, elektrik yük tahmini, taşkın tahmini gibi birçok uygulaması bulunur (Medsker 2000).

Bu ağ tipinin göze çarpan özelliği her düğümün diğer tüm düğümlerden girdi almasıdır. Ayrıca düğüm kendisinden de geri besleme alabilir (Medsker 2000).

Yinelemeli sinir ağları yöntemlerinden biri olan uzun kısa süreli bellek tekniği yinelemeli yapay sinir ağlarının en başarılı türlerindendir. Bu teknik ağın gizli katmanında bulunan yapay nöronları hafıza hücresi adı verilen hesaplama üniteleri ile değiştirir. Bu özelliği ile hafıza ve anlık girdileri verimli bir şekilde birleştirerek yüksek tahmin kapasitesi ile veri yapısını dinamik olarak algılar (Chen vd. 2015).



Resim 3.4.2 Klasik yinelemeli sinir ağları ile uzun kısa süreli bellek ağları

3.5 Derin Sinir Ağları

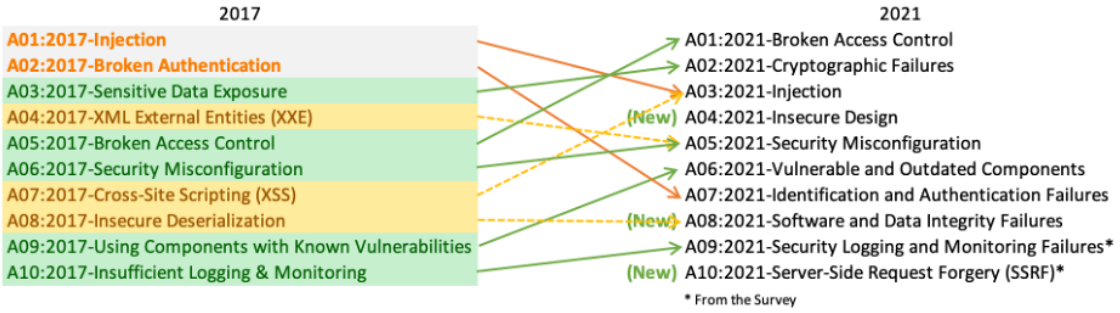
Derin sinir ağları nöron olarak adlandırılan basit hesaplama birimlerinin paralel çalışarak oluşturdukları hesaplama modelleridir. Derin sinir ağları her biri ayrı girdi ve çıktılardan oluşan basit sinir ağlarının üst üste eklenmesiyle oluşur ve eklene katman sayısı arttıkça ağ derinleşir (Cichy ve Kaiser 2019).

Büyük ölçekli veri setlerinin ortaya çıkması ile denir sinir ağları bilgisayarlı görme, bilgi elde etme ve dil işleme gibi alanlarda başarılar sergilemeye başlamıştır. Bu başarım büyük ölçekli fakat elde etmesi zor ve pahalı olan düzgün etiketlenmiş veriler sayesinde oluşmaktadır. Amazon tarafından sunulan Mechanical Turk gibi hizmetlerle bu etiketleme yapılabiliyor olsa da bu yöntem oldukça pahalıya mal olmakta üstelik her zaman doğru etiketleme yapılamamaktadır. Bu tür yanlış etiketlenmiş veriler gürültü olarak adlandırılmaktadır. Bu tür verinin gerçek dünyadaki veri kümelerinde görülme oranı %8 ile %38.5 arasında raporlanmaktadır (Song vd. 2023).

3.6 Web Uygulama Güvenliği

Günümüzde web uygulamaları, statik birer sayfa olmaktan çıkıp, dağıtık uygulamalardan oluşan ve internet üzerinde bilgi aktarımı konusunda önemli roller üstlenen bir kavram olmuştur. AJAX gibi teknolojilerin gelişmesiyle birlikte etkileşimli uygulamaların sayısı artmış, kullanıcı deneyimi yükseltilmiştir (Xiaowei ve Xue 2011).

Daha önceleri bir bilgi sisteminin en önemli kavramlarından biri ağ güvenliği iken, bulut teknolojileri gibi gelişmeler ile ağ merkezli sistemlerden uygulama merkezli sistemlere geçilmesi ile birlikte web uygulama güvenliğinin önemi hızla artmıştır. Gelişmelerin hızlanması uygulamalardaki zafiyetlerin tespitinde otomatize yöntemlerin kullanımını zorunlu hale getirmiştir (Curphey ve Arawo 2006).



Resim 3.6 2017 ve 2021 yıllarında hazırlanan OWASP Top Ten listeleri

Bu bölümde OWASP tarafından hazırlana Top Ten listesinde yer alan bazı web uygulama saldırıları incelenmiştir. (OWASP Top Ten | OWASP Foundation, t.y.)

3.6.1 Siteler Arası Betik Çalıştırma

Siteler arası betik sahteciliği, kullanıcı girdilerinin düzgün şekilde temizlenmemesi sonucu ortaya çıkan zafiyetlerdir. Kullanıcı girişlerinin doğrulanmaması sonucunda, kötü niyetli kişiler bu giriş alanlarından gönderdikleri zararlı kod parçalarını güvenilen web sayfalarına yerleştirebilir. Sonucunda tarayıcı çerezlerinin ve oturum bilgilerinin çalınması, hesap bilgilerini çalınması, web sayfası içeriğinin değiştirilmesi, servis dışı bırakma gibi pek çok zararlı faaliyetle karşı karşıya kalınır (Hydara vd. 2015).

Siteler arası betik çalıştırma saldırıları temel olarak istemci taraflı ve sunucu taraflı olarak ikiye ayrılabilir. Bu alt kategoriler yöntem olarak kalıcı XSS, kalıcı olmayan XSS ve DOM tabanlı XSS olmak üzere üç temel tip saldırı barındırır (Gupta ve Gupta 2017).

Kalıcı tip saldırılarda ilk saldırı gerçekleştikten sonra zararlı betik script tagları arasına girerek tüm kullanıcıları etkiler hali gelir. Bu saldırılardan korunmanın bir yöntemi tarayıcı tarafında javascript çalıştırmayı engellemek olsada, bu durum uygulamaların kullanımı imkânsız hale getirir. Bu sebeple saldırıların engellenmesinde daha derinlemesine analiz gerektiren yöntemler kullanılmalıdır (Rodríguez vd. 2020).

Kalıcı olmayan tip saldırılar, oturum çerezlerinin çalınması ve bu yolla kullanıcın taklit edilmesine dayanır. Çalınan oturum çerezleri sayesinde saldırgan kullanıcının gerçekleştirmeye yetkisi olan tüm işlevleri kullanabilir. Saldırı özellikle web sayfasının

oluşturulmasında kullanıcı tarafından yapılan girişlerin kullanıldığı uygulamalarda işe yarar. Saldırının çalışması için özellikle kullanıcıyı yanıltarak bir bağlantıya tıklamak gibi aksiyonlar aldırmak gereklidir (Rodríguez vd. 2020).

DOM tabanlı saldırılar genel olarak javascript betiklerinin fazla şekilde kullanıldığı uygulamalarda ortaya çıkmaktadır (Stock vd. 2014). DOM(Document Object Model) HTML elementlerine erişimi API'ler ile sağlayan ağaca benzer bir yapıdır. Bu ağaç üzerinde gerçekleşen manipülasyonlar sunucu tarafında bir etki oluşturmaz ve diğer kullanıcıları etkilemezken kurban istemci tarafında sayfa ile alakalı önemli değişikliklere sebep olabilmektedir (Pan ve Mao 2017).

3.6.2 Erişim Kontrol Hataları

Erişim kontrol hataları, uygulama tasarımı veya kodlama sırasında yapılan hatalar sonucu, saldırganların yetkisi olmayan işlemleri yapabilir hale gelmesidir. Genellikle ele geçirilmiş az yetkili hesapların yetki yükseltme gibi dikey olarak ilerlenen saldırı yöntemleri daha yüksek yetkili işlemleri çalıştırabilmesi üzerine dayanır. Ayrıca erişim kontrolünün düzgün planlanmadığı durumlarda keşif yöntemleri ile uygulama ara yüzünde görünmeyen http metotlarının çalıştırılması gibi stratejilerde saldırılarda kullanılır.

3.6.3 Sunucu Taraflı İstek Sahteciliği

Bu saldırı tipinde zafiyetli uygulama saldırganın isteklerini yerel ağa yönlendirerek normal şartlarda dışarıya kapalı olan servisleri saldırgana açık hale getirir. Saldırı sunucu tarafında çalışacak komutları e-posta kabul eden servislerde eposta göndermek veya kullanıcı girdilerinden alınan komutların temizlenmemesi gibi şekillerle çalıştırılmasıdır. Saldırının pek çok katmandan geçerek ulaşması sebebiyle zafiyetli katmanı tespit etmek oldukça güçtür (Jabiyev vd. 2021).

3.6.4 SQL Enjeksiyonu

Günümüzde uygulamaların gittikçe çevrimiçi kullanımının artması ile birlikte veritabanlarına duyulan ihtiyaç artmaktadır. Sıklıkla tercih edilen ilişkisel veri tabanlarını

programlamak için kullanılan SQL dili, esnekliği ve gücü sayesinde hiçbir bilgisi olmayan kullanıcılarından kullanabilmesine uygundur. Bunun yanında daha karmaşık sorgular yazarak yüksek performans elde etmekte mümkündür. Tüm bu esneklik ve güç dilin kullanıldığı uygulamaları ayrıca bir saldırı hedefi haline getirmektedir (Sadeghian vd. 2013).

SQL enjeksiyonu veritabanı kullanılan uygulamaları hedef alan bir saldırı türüdür. Çoğunlukla saldırgan girdi kontrolü düzgün yapılmayan girdi alanlarına SQL cümleleri yazarak girdileri içeren sql cümlelerini bozar ve bu sorgu veritabanı sunucusuna veya motoruna iletilir. SQL cümleleri http tipine bağlı olarak HTML form alanları veya URL'e yerleştirilebilir. OWASP 2017 listesinde ilk sırada yer alan sokuşturma saldırıları 2021 yılında hazırlanan liste ile üçüncü sıraya kaymıştır. Her ne kadar bir düşüş görülse de SQL enjeksiyon saldırıları artık sadece web uygulamalarını değil, masaüstü ve mobil uygulamaları da hedef almaktadır (Mukherjee vd. 2015) .

3.6.5 Kriptografik Hatalar

2021 OWASP Top Ten listesinde ikinci sırada yer alan bu hatalara kaynak koda yerleştirilmiş parolalar, güvensiz kriptografik algoritmaların kullanımı, yetersiz entropi(kriptografik anahtarların oluşturulması için gereken karmaşıklık) örnek gösterilebilir.(*A02 Cryptographic Failures - OWASP Top 10:2021*, t.y.)

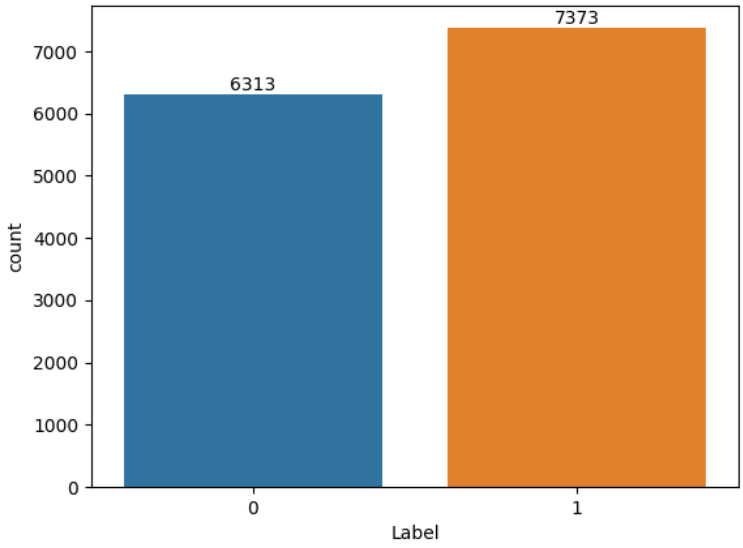
Kaynak koda yerleştirilmiş parolalar sadece parola ile sınırlı kalmayıp, API tokenları gibi başka hassas veriler de olabilir. Normal şartlarda kullanıcı tarafından görülemeyen backend kodları, saldırganlar tarafından sunucuya sızma, versiyon takip sistemleri gibi araçlarda ele geçirildiğinde başka sistemlere de erişimin yolunu açmaktadır (Singh Verma ve Chandavarkar 2019).

Kryptos(gizli) ve graphien(yazmak) kelimelerinin birleşiminden oluşan kriptografi kelimesi bilgi sistemlerinde de gizli kalması gereken bilgileri geri döndürülebilir şekilde saklanması amacıyla kullanılmaktadır. Geri dönüş için yapılacak işlemlerin sırasının ve parametrelerinin bilinmesi gereklidir. Cipher adı verilen parametrelerin veya sıralamanın

tahmin edilebilmesi halinde kriptografik algortima zayıf kabul edilir ve gizli kalması gereken bilgiler çözülebilir hale gelir (Salami vd.).

4. BULGULAR

Siteler arası betik çalıştırma saldırıları, web uygulamalarının güvenliği tehdit eden güvenlik açığıdır. Saldırganlar kullanıcıların tarayıcılarında zararlı kodları çalıştırarak kullanıcıların oturum bilgilerini, çerezleri, kişisel verileri elde edebilir hatta kötü amaçlı web sitelerine yönlendirerek kullanıcıların güvenliğini riske atabilir. Bu nedenle bu tez çalışmasında yapay zekâ tabanlı otomatik ve hızlı siteler arası betik saldırıları tespit sistemi geliştirilmiştir. Sistemin karar verme süreçlerinde yapay zekânın yeni trendlerinden olan derin öğrenme tabanlı evrişimsel sinir ağları önerilmiştir. Önerilen sistem açık kaynak kodlu Python ortamında test edilmiştir. Veri kümesi zararsız kodlarla birlikte zararlı saldırı komutlarını da barındırmaktadır. Veriler portswigger ve owasp sayfalarından toplanan 13686 örneklem içermektedir. Verilerin gruplara dağılımı Resim 4.1'de verilmiştir.



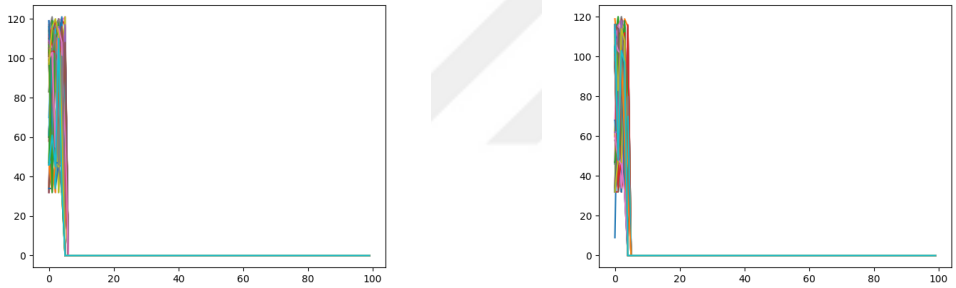
Resim 4.1 Veri kümesinin dağılımı

Resme göre, zararlı kodlar 1 olarak zararsız kodlar 0 olarak etiketlenmiştir. Veri kümesinde zararlı kodlara ait örneklem sayısı 7373 iken, zararsız kodlara ait örneklem sayısı 6313'tür. Çizelge 4.1'de veri kümesinden örnek bazı komutlar verilmiştir.

Çizelge 4.1 Veri kümesinden örnekler

Komut	Etiket
"<tt onmouseover=""alert(1)"">test</tt>"	Zararlı (1)
" <span class=""reference-text"">Steering for the 1995 ""No Hands Across America "" required ""only a few human assists"". "	Zararsız (0)

Geliştirilen sistemin önişlem aşamasında alınan komut cümlelerinin ASCII değerleri işlenerek öncelikle tek boyutlu diziye dönüştürülmekte daha sonra 100x100 boyutunda bir NumPy dizisine dönüştürülmektedir. ASCII değeri 128den küçük olanlar otomatik diziye eklenmiştir. Karakterlerin normalleştirilmesinde sorun çıkmaması için ASCII değeri 8216, 8217, 8220, 8221 olan karakterler, 129,130,131,132 olarak eklenmiştir. Resim 4.2'de önişlemden sonraki veri kümesinden örnek görüntüler verilmiştir.



Resim 4.2 Önişlem aşamasından sonra veri kümesinden örnek görüntüler

Veriler açık kaynak kodlu OpenCV kütüphanesi ile bikübik interpolasyon kullanılarak 100x100 boyutunda resim formatında yeniden boyutlandırılmıştır. Bikübik interpolasyon yeni enterpolasyonlu pikseli oluşturmak için 16 pikselin (4x4) ağırlık ortalamasını almaktadır. Daha sonra veri kümesinin rasgele %80'i eğitim için ayrılırken geri kalan kısmı doğrulama için ayrılmıştır. Buna göre eğitim aşamasından önce gruplarda oluşan örneklem sayıları Çizelge 4.2'de verilmiştir.

Çizelge 4.2 Veri kümesinin gruplara göre örneklem sayıları

Grup	Zararlı	Zararsız
Eğitim	5895	5053
Doğrulama	1478	1260

Sınıflandırma aşamasında 4 adet Evrişimsel Katman, 4 adet Maksimum Havuzlama Katmanı, ve 4 adet Tam Bağlı Katmana sahip bir Evrişimsel Sinir Ağı tasarlanmıştır. Mimari yapıya yapılan işlem gücü, süre ve doğruluk oranları göz önünde bulundurularak karar verilmiştir. Önerilen Evrişimsel Sinir Ağının temel mimarisi Çizelge 4.3'de verilmiştir.

Çizelge 4.3 Önerilen ESA Mimarisi

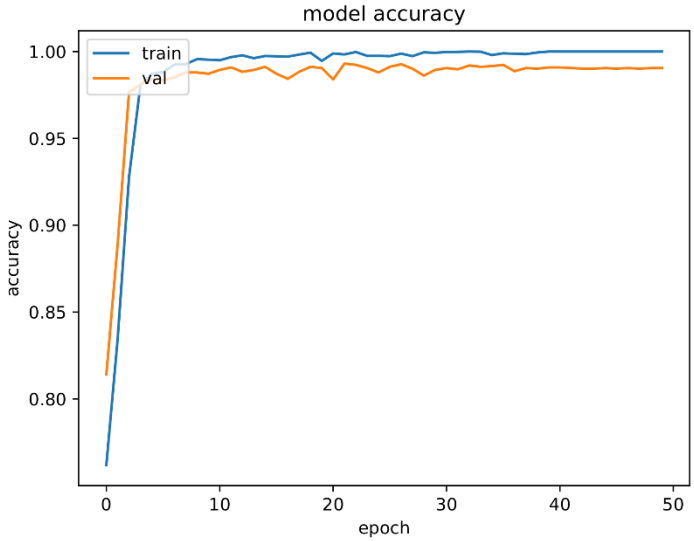
Katman Türü	Boyut	Parametre Sayısı
Evrişim Katmanı 1	(-,98,98,32)	320
Maksimum Havuzlama Katmanı 1	(-,49,49,32)	0
Evrişim Katmanı 2	(-,47,47,64)	18496
Maksimum Havuzlama Katmanı 2	(-,23,23,64)	0
Evrişim Katmanı 3	(-,21,21,128)	73856
Maksimum Havuzlama Katmanı 3	(-,10,10,128)	0
Evrişim Katmanı 4	(-,8,8,256)	295168
Maksimum Havuzlama Katmanı 4	(-,4,4,256)	0
Düzleştirme	(-,4096)	0
Tam Bağlı Katman 1	(-,256)	1048832
Tam Bağlı Katman 2	(-,128)	32896
Tam Bağlı Katman 3	(-,64)	8256
Tam Bağlı Katman 4	(-,1)	65

Önerilen mimarinin eğitilebilir parametre sayısı 1.477.889'dur. Evrişimsel katmanlardaki filtrenin boyutu 3x3, aktivasyon fonksiyonu ReLu'dur. Havuzlama katmanlarındaki filtrenin boyutu 2x2, aktivasyon fonksiyonu ReLu'dur. Tam Bağlı katmanlarda ise sadece son katman Sigmoid iken diğer 3 katmanın aktivasyon fonksiyonu ReLu'dur. Optimize edici olarak ADAM kullanılırken kayıp fonksiyon çapraz entropidir. Mimarinin diğer hiper parametreleri Çizelge 4.4'de verilmiştir.

Çizelge 4.4 Hiper parametreler

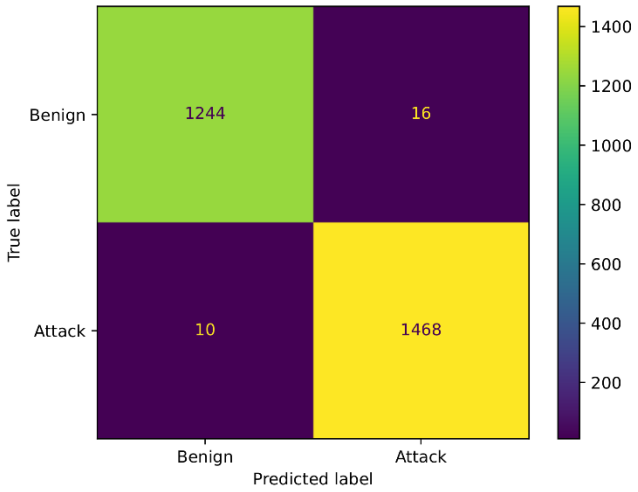
Parametre	**Değer**
Epoch	50
Mini Küme	128
Öğrenme Oranı	0.001
Beta 1	0.9
Beta 2	0.999
Epsilon	10^{-7}

Eğitim sürecinde doğruluk değerinin değişim grafiği Resim 4.3'de verilmiştir. Grafikte eğitime ait doğruluk değerleri mavi renkte verilirken doğrulama grubuna ait doğruluk grafiği turuncu renkte verilmiştir.



Resim 4.3 Doğruluk değerlerinin değişim grafiği

Grafiğe göre eğitim sürecinde doğruluk değerleri uyumlu bir şekilde devam etmiştir. Bu sayede eğitim sürecinde yapay zekâda sıklıkla karşılaşılan ezberleme sorunun olmadığı görülmektedir. Sonuçlara ilişkin değerlendirmeler karışıklık matrisi ile değerlendirilmiştir. Resim 4.4'de sonuçlara ilişkin karışıklık matrisi verilmiştir.



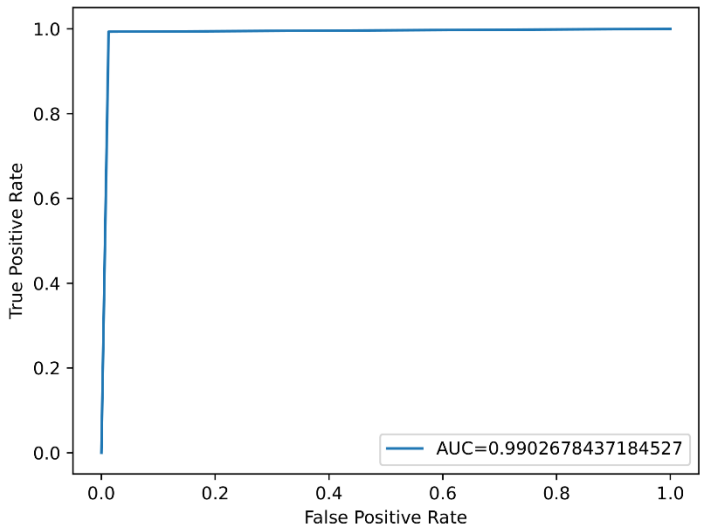
Resim 4.4 Karışıklık matrisi

Elde edilen sonuçlara göre önerilen model ilk karşılaştığı 2736 adet komuttan 2712'sinin zararlı ve zararsız olduğunu ayırt edebilmektedir. Diğer performans metrikleri Çizelge 4.5'de verilmiştir.

Çizelge 4.4 Performans Metrikleri

Metrik	**Değer**
Doğruluk	0.9905
Kesinlik	0.9892
Duyarlılık	0.9932
F1 Skor	0.9912

Önerilen yöntem, %99 doğruluk ve %99 F1 değeriyle oldukça yüksek bir doğruluk oranı elde etmiştir. Önerilen yöntem ayrıca ROC eğrisi ve Eğri altında kalan alan değerleriyle de değerlendirilmiş ve %99 doğruluk oranına ulaşmıştır. Resim 4.5'de ROC eğrisi verilmiştir.



Resim 4.5 ROC Eğrisi

Geliştirilen sistem 8GB RAM, 10.nesil i5 işlemciye sahip kişisel dizüstü bilgisayarda geliştirilmiş olup siteler arası betik çalıştırma saldırılarını engellemede hızlı, etkin ve yüksek doğrulukta çözüm üretmiştir.

5. TARTIŞMA ve SONUÇ

Web uygulamaları genel olarak değerlendirildiğinde büyük ve karmaşık yapılardır. İçerisinde hem ön ara yüze hem de arka ara yüze sahip kod parçaları barındırabilmektedir. Manuel bir şekilde siteler arası betik saldırı taramaları yapmak zor, zaman alıcı, insan iş gücü ve uzmanlığı gerektiren zorlu bir süreçtir. Bu açıdan geliştirilen sistem hızlı ve ölçeklenebilir şekilde zararlı kodları tespit etmektedir.

Web uygulamalarının karmaşıklaşması büyük çaplı uygulamaların kullanımını yaygınlaştırmakta, hataların ortak çözümü kolaylaşmakla birlikte hatalarda çok daha fazla uygulamanın zafiyetlerden etkilenmesine sebep olmaktadır. Tüm güvenlik önlemlerinin geliştirici tarafından alınması yaklaşımı günümüz teknoloji dünyasında bir seçenek olmaktan çıkmıştır. Bu sebeple zafiyetli uygulamalarda, zafiyet henüz sömürülmeden saldırıyı tespit edip engellemek önemli bir konu haline gelmiştir.

Öte yandan web uygulamaları sürekli olarak güncellenmesi gerekebilir. Bu nedenle siteler arası betik çalıştırma saldırı tespiti bir süreç olarak sürekli izlemeyi gerektirmektedir. Geliştirilen sistem, bu süreci otomatikleştirerek güncel tehditleri kolaylıkla yakalayabilmektedir. Ayrıca büyük verileri analiz edebilme yeteneklerine sahiptir. Geliştirilen sistemin en büyük avantajlarından bir tanesi de saldırıların erken tespitini sağlayarak kullanıcıların hassas verilerinin korunmasını sağlamaktadır.

Sınıflandırmada önerilen metin bilgilerinin grafiklere dönüştürülerek evrişimsel sinir ağı ile sınıflandırma yöntemi çalışmanın özgünlüğünü oluşturmaktadır. Ayrıca sığ bir evrişimsel sinir ağı mimarisiyle yüksek doğrulukla sınıflandırması çalışmanın avantajıdır.

Geliştirilen uygulama gerçek zamanlı sistemler üzerinde test edilmemiştir. Bu çalışmanın dezavantajıdır. Gelecek çalışmalarda veri seti artırılarak daha büyük parametreli ağlarla test edilerek gerçek zamanlı sistemlere entegre edilmesi hedeflenmektedir. Ayrıca sınıflandırma yönteminde kelime gömme yöntemleri test edilmesi düşünülmektedir.

Sonuç olarak, önerilen sistem siteler arası betik çalıştırma saldırılarının tespitinde, web uygulamalarının güvenliğini artırmak ve potansiyel saldırılara karşı daha hızlı ve etkili

bir koruma sağlamak için önemlidir. Web uygulamalarının daha güvenli hale getirilmesine yardımcı olur ve kullanıcıların verilerini koruma konusunda katkı sağlamaktadır.

6. KAYNAKLAR

Albawi S, Mohammed T A, Al-Zawi S, 2017, Understanding of a convolutional neural network, 2017 International Conference on Engineering and Technology (ICET), 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Baldi P F, Hornik K, 1995, Learning in linear neural networks: A survey. IEEE Transactions on Neural Networks, 6(4), 837-858. https://doi.org/10.1109/72.392248

Banerjee R, Baksi A, Singh N, Bishnu S, 2020, Detection of XSS in web applications using Machine Learning Classifiers. https://doi.org/10.1109/IEMENTech51367.2020.9270052

Bisht P, Venkatakrishnan V N, 2008, XSS-GUARD: Precise Dynamic Prevention of Cross-Site Scripting Attacks, 23-43, Springer. https://doi.org/10.1007/978-3-540-70542-0_2

Chen K, Zhou Y, Dai F, 2015, A LSTM-based method for stock returns prediction: A case study of China stock market. 2015 IEEE International Conference on Big Data (Big Data), 2823-2824. https://doi.org/10.1109/BigData.2015.7364089

Cichy R M, Kaiser D, 2019, Deep Neural Networks as Scientific Models, Trends in Cognitive Sciences, 23(4), 305-317. https://doi.org/10.1016/j.tics.2019.01.009

Cunningham P, Cord M, Delany S J, 2008, Supervised Learning, 21-49, Springer. https://doi.org/10.1007/978-3-540-75171-7_2

Curphey M, Arawo R, 2006, Web application security assessment tools. IEEE Security Privacy, 4(4), 32-41. https://doi.org/10.1109/MSP.2006.108

El Naqa, I Murphy, M J, 2015, What Is Machine Learning?, 3-11, Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1

Ertel W, 2018, Introduction to Artificial Intelligence, Springer.

Fang Y, Huang C, Xu Y, Li Y, 2019, RLXSS: Optimizing XSS Detection Model to Defend Against Adversarial Attacks Based on Reinforcement Learning. Future Internet, 11, 177. https://doi.org/10.3390/fi11080177

Fetzer J H, 1990, What is Artificial Intelligence?, 3-27, Springer Netherlands. https://doi.org/10.1007/978-94-009-1900-6_1

Fonseca J, Vieira M, Madeira H, 2007, Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks. 13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007), 365-372. https://doi.org/10.1109/PRDC.2007.55

Gardner M W, Dorling S R, 1998, Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences, Atmospheric Environment, 32(14), 2627-2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Ghahramani Z, 2004, Unsupervised Learning, 72-112, Springer. https://doi.org/10.1007/978-3-540-28650-9_5

Ghosh J, Nag A, 2001, An Overview of Radial Basis Function Networks, 1-36, Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-1826-0_1

Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen, T, 2018, Recent advances in convolutional neural networks, Pattern Recognition, 77, 354-377. https://doi.org/10.1016/j.patcog.2017.10.013

Gundy M, Chen H, 2012, Noncespaces: Using randomization to defeat cross-site scripting attacks, Computers Security, 31, 612-628. https://doi.org/10.1016/j.cose.2011.12.004

Gupta S, Gupta B B, 2017, Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art. International Journal of System Assurance Engineering and Management, 8(1), 512-530. https://doi.org/10.1007/s13198-015-0376-0

Hoang X D, 2021, Detecting Common Web Attacks Based on Machine Learning Using Web Log, Advances in Engineering Research and Application, 311-318, Springer International Publishing. https://doi.org/10.1007/978-3-030-64719-3_35

Hydara I, Sultan A B Md, Zulzalil, H, Admodisastro, N, 2015, Current state of research on cross-site scripting (XSS) – A systematic literature review.

Information and Software Technology, 58, 170-186. https://doi.org/10.1016/j.infsof.2014.07.010

Jabiyev B, Mirzaei O, Kharraz A, Kirda E, 2021, Preventing server-side request forgery attacks. Proceedings of the 36th Annual ACM Symposium on Applied Computing, 1626-1635. https://doi.org/10.1145/3412841.3442036

Johns M, 2006, SessionSafe: Implementing XSS Immune Session Handling, 444-460, Springer. https://doi.org/10.1007/11863908_27

Johns M, Engelmann B, Posegga J, 2008, XSSDS: Server-Side Detection of Cross-Site Scripting Attacks, 2008 Annual Computer Security Applications Conference (ACSAC), 335-344. https://doi.org/10.1109/ACSAC.2008.36

Kals S, Kirda E, Kruegel C, Jovanovic N, 2006, SecuBat: A web vulnerability scanner. Proceedings of the 15th international conference on World Wide Web, 247-256. https://doi.org/10.1145/1135777.1135817

Kascheev S, Olenchikova T, 2020, The Detecting Cross-Site Scripting (XSS) Using Machine Learning Methods, 265-270. https://doi.org/10.1109/GloSIC50886.2020.9267866

Kaur G, Malik Y, Samuel H, Jaafar F, 2018, Detecting Blind Cross-Site Scripting Attacks Using Machine Learning. Proceedings of the 2018 International Conference on Signal Processing and Machine Learning, 22-25. https://doi.org/10.1145/3297067.3297096

Kelleher J D, 2019, Deep Learning, MIT Press.

Khan N, Abdullah J, Khan A S, 2017, Defending Malicious Script Attacks Using Machine Learning Classifiers, Wireless Communications and Mobile Computing, 2017, 5360472. https://doi.org/10.1155/2017/5360472

Kirda E, Kruegel C, Vigna G, Jovanovic N, 2006, Noxes: A client-side solution for mitigating cross-site scripting attacks, Proceedings of the 2006 ACM symposium on Applied computing, 330-337. https://doi.org/10.1145/1141277.1141357

Laghrissi F, Douzi S, Douzi K, Hssina B, 2021, Intrusion detection systems using long

short-term memory (LSTM), Journal of Big Data, 8(1), 65. https://doi.org/10.1186/s40537-021-00448-4

Louw M T, Venkatakrishnan V N, 2009, Blueprint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers, 2009 30th IEEE Symposium on Security and Privacy, 331-346. https://doi.org/10.1109/SP.2009.33

Medsker L, 2000, Recurrent Neural Networks: Design and Applications. CRC-Press.

Mukherjee S, Sen P, Bora S, Pradhan C, 2015, SQL Injection: A sample review. 2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7. https://doi.org/10.1109/ICCCNT.2015.7395166

Nasteski V, 2017, An overview of the supervised machine learning methods, HORIZONS.B, 4, 51-62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

Öztürk K, Şahin, M E, 2018, Yapay Sinir Ağları ve Yapay Zekâ'ya Genel Bir Bakış. Takvim-i Vekayi, 6(2), Article 2.

Pan J, Mao X, 2017, Detecting DOM-Sourced Cross-Site Scripting in Browser Extensions. 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), 24-34. https://doi.org/10.1109/ICSME.2017.11

Rathore S, Sharma P, Park J, 2017, XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs, Journal of Information Processing Systems, 13. https://doi.org/10.3745/JIPS.03.0079

Rich E, 1983, Artificial Intelligence. McGraw-Hill.

Rodríguez G E, Torres J G, Flores P, Benavides D E, 2020, Cross-site scripting (XSS) attacks and mitigation: A survey, Computer Networks, 166, 106960. https://doi.org/10.1016/j.comnet.2019.106960

Sadeghian A, Zamani M, Abdullah S M, 2013, A Taxonomy of SQL Injection Attacks. 2013 International Conference on Informatics and Creative Multimedia, 269-273. https://doi.org/10.1109/ICICM.2013.53

Salami Y, Khajehvand V, Zeinali, E (t.y.). Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges.

Salehinejad H, Sankar S, Barfett J, Colak E, Valaee S, 2018, Recent Advances in

Recurrent Neural Networks (arXiv:1801.01078). arXiv. https://doi.org/10.48550/arXiv.1801.01078

Sazli M H, 2006, A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 50(01), Article 01. https://doi.org/10.1501/commua1-2_0000000026

Shahriar H, Zulkernine M, 2011, S2XS2: A Server Side Approach to Automatically Detect XSS Attacks. 7-14. https://doi.org/10.1109/DASC.2011.26

Sharma S, Zavarsky P, Butakov S, 2020, Machine Learning based Intrusion Detection System for Web-Based Attacks, 227-230. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048

Shrivastava A, Choudhary S, Kumar A, 2016, XSS vulnerability assessment and prevention in web application. 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), 850-853. https://doi.org/10.1109/NGCT.2016.7877529

Singh Verma R, Chandavarkar B R, 2019, Hard-coded Credentials and Web Service in IoT: Issues and Challenges (SSRN Scholarly Paper 3358283). https://papers.ssrn.com/abstract=3358283

Song H, Kim M, Park D, Shin Y, Lee J-G, 2023, Learning From Noisy Labels With Deep Neural Networks: A Survey. IEEE Transactions on Neural Networks and Learning Systems, 34(11), 8135-8153. https://doi.org/10.1109/TNNLS.2022.3152527

Stock B, Lekies S, Mueller T, Spiegel P, Johns M, 2014, Precise Client-side Protection against {DOM-based} {Cross-Site} Scripting. 655-670. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/stock

Suzuki K, 2011, Artificial Neural Networks—Methodological Advances and Biomedical Applications. https://doi.org/10.5772/644

Swaswati Goswami, Nazrul Hoque, Dhruba K Bhattacharyya, Jugal Kalita, 2017, An Unsupervised Method for Detection of XSS Attack. International Journal of

Network Security, 19(5). https://doi.org/10.6633/IJNS.201709.19(5).14

Wang R, Jia X, Li Q, Zhang D, 2015a, Improved N-gram approach for cross-site scripting detection in Online Social Network, 1206-1212. https://doi.org/10.1109/SAI.2015.7237298

Wang R, Jia X, Li Q, Zhang S, 2015b, Machine Learning Based Cross-Site Scripting Detection in Online Social Network, 823-826. https://doi.org/10.1109/HPCC.2014.137

Wu Y, Feng J, 2018, Development and Application of Artificial Neural Network. Wireless Personal Communications, 102(2), 1645-1656. https://doi.org/10.1007/s11277-017-5224-x

Wu Y, Wang H, Zhang B, Du K-L, 2012, Using Radial Basis Function Networks for Function Approximation and Classification, International Scholarly Research Notices, 2012, e324194. https://doi.org/10.5402/2012/324194

Wurzinger P, Platzer C, Ludl C, Kirda E, Kruegel C, 2009, SWAP: Mitigating XSS attacks using a reverse proxy. 2009 ICSE Workshop on Software Engineering for Secure Systems, 33-39. https://doi.org/10.1109/IWSESS.2009.5068456

Xiaowei Xue Y, 2011, A Survey on Web Application Security. https://www.semanticscholar.org/paper/A-Survey-on-Web-Application-Security-Xiaowei-Xue/4090c860cf6eb3c574bc41612585fb9452251b37

YEGNANARAYANA, B, 2009, ARTIFICIAL NEURAL NETWORKS. PHI Learning Pvt. Ltd.

Zhang C, 2019, Research on the Fluctuation and Factors of China TFP of IT Industry. Journal of Industrial Integration and Management, https://doi.org/10.1142/S2424862219500131

Zhang C, Lu, Y, 2021, Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Zhu X, Goldberg A B, 2009, Introduction to Semi-Supervised Learning. Springer International Publishing. https://doi.org/10.1007/978-3-031-01548-9

İnternet Kaynakları

1- https://owasp.org/Top10/A02_2021-Cryptographic_Failures/, 06.02.2024

2- https://owasp.org/www-community/attacks/xss/, 02.10.2023

3- https://owasp.org/www-project-top-ten/, 26.01.2024