T.C.
KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WEB SALDIRILARININ DERİN ÖĞRENME İLE TESPİT EDİLMESİ

YUNUS EMRE SEYYAR

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

DANIŞMAN

Doç. Dr. Halil Murat ÜNVER

KIRIKKALE – 2022

T.C.
KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

WEB SALDIRILARININ DERİN ÖĞRENME İLE TESPİT EDİLMESİ

YUNUS EMRE SEYYAR

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

DANIŞMAN

Doç. Dr. Halil Murat ÜNVER

KIRIKKALE - 2022

Yunus Emre SEYYAR tarafından hazırlanan "WEB SALDIRILARININ DERİN ÖĞRENME İLE TESPİT EDİLMESİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Halil Murat ÜNVER

Bilgisayar Donanımı Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

İkinci Danışman: Prof. Dr. Ali Gökhan Yavuz

Bilgisayar Mühendisliği, Türk-Alman Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. Necaattin BARIŞÇI

Bilgisayar Mühendisliği, Gazi Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Prof. Dr. Bülent TAVLI

Elektrik-Elektronik Mühendisliği, TOBB ETÜ İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Murat LÜY

Elektrik-Elektronik Mühendisliği, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Bülent Gürsel EMİROĞLU

Bilgisayar Yazılımı Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Dr. Öğr. Üyesi Ziya Cihan TAYŞİ

Bilgisayar Mühendisliği, Yıldız Teknik Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: ...../……/……

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

İmza …….……

Prof. Dr. Recep ÇALIN

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYANI

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

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

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

…….....

Yunus Emre SEYYAR

19/08/2022

ÖZET

WEB SALDIRILARININ DERİN ÖĞRENME İLE TESPİTİ

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı, Doktora Tezi

Danışman: Doç. Dr. Halil Murat ÜNVER

Ortak Danışman: Prof. Dr. Ali Gökhan YAVUZ

Ağustos, 2022, 107 sayfa

Derin Öğrenme (DÖ) tekniklerinin hızla gelişmesiyle beraber Doğal Dil İşleme (DDİ) teknikleri de hızla gelişmekte ve zenginleşmektedir. Ayrıca ilgili teknolojilerin gelişmesine bağlı olarak web uygulamalarının kullanımının da hemen her yönde arttığına tanık oluyoruz. Web uygulamaları, kişisel, finansal, savunma ve siyasi bilgileri (ör. wikileaks olayı) kullanan çok çeşitli kullanım durumlarını kapsamaktadır. Nitekim bu tür bilgilere erişmek ve bunları manipüle etmek saldırganların öncelikli amaçları arasında yer almaktadır. Bu nedenle, saldırganlar tarafından hedeflenen bilgilerin savunmasızlığı hayati bir sorundur ve bu tür bilgilerin ele geçirilmesi halinde sonuçların yıkıcı olabileceği ve bazı durumlarda potansiyel olarak ulusal güvenlik riskleri haline gelebileceği görülmektedir. Bu çalışmada, bu sorunun çözümüne yönelik, normal HTTP isteklerini ve anomali HTTP isteklerini ayırt edebilen yeni bir model önerilmiştir. Önerilen çalışmada DDİ tekniklerini, Transformatörlerden Çift Yönlü Kodlayıcı Temsilleri (BERT) modeli, Çok Katmanlı Algılayıcı (ÇKA) ve DÖ tekniklerinden Evrişimsel Sinir Ağı (ESA) teknikleri kullanılmıştır. Deneysel sonuçlarımız, önerilen yaklaşımın normal ve anomali isteklerin sınıflandırılmasında %99.98'in üzerinde bir başarı oranı ve %98.70'in üzerinde bir F1-puanı elde ettiğini ortaya koymaktadır. Ayrıca, önerilen model, web saldırısı tespit süresi 0,4 ms olarak literatürde sunulan diğer yaklaşımlardan önemli ölçüde daha düşüktür.

Anahtar kelimeler: Anomali istekler, BERT, derin öğrenme, web saldırıları, çok katmanlı algılayıcı, evrişimsel sinir ağları, doğal dil işleme, siber güvenlik

ABSTRACT

DETECTION OF WEB ATTACKS WITH DEEP LEARNING

Kırıkkale University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering, Ph. D. Thesis
Supervisor: Assoc. Prof. Halil Murat ÜNVER
Co-Supervisor: Prof. Ali Gökhan YAVUZ
August, 2022, 107 pages

Deep Learning (DL) and Natural Language Processing (NLP) techniques are improving and enriching with a rapid pace. Furthermore, we witness that the use of web applications is increasing in almost every direction in parallel with the related technologies. Web applications encompass a wide array of use cases utilizing personal, financial, defense, and political information (e.g., wikileaks incident). Indeed, to access and to manipulate such information are among the primary goals of attackers. Thus, vulnerability of the information targeted by adversaries is a vital problem and if such information is captured then the consequences can be devastating, which can, potentially, become national security risks in the extreme cases. In this study, as a remedy to this problem, we propose a novel model that is capable of distinguishing normal HTTP requests and anomalous HTTP requests. Our model employs NLP techniques, Bidirectional Encoder Representations from Transformers (BERT) model, and DL techniques. Our experimental results reveal that the proposed approach achieves a success rate over 99.98% and an F1 score over 98.70% in the classification of anomalous and normal requests. Furthermore, web attack detection time of our model is significantly lower (i.e., 0.4 ms) than the other approaches presented in the literature.

Key words: Anomalous request, BERT, deep learning, web attacks, multilayer perceptron, CNN, natural language processing, cyber security

TEŞEKKÜRLER

Tezimin hazırlanmasında hiçbir yardımını esirgemeyen sadece danışmanlık yapmayan aile olan Sayın Doç. Dr. Halil Murat Ünver'e, vermiş olduğu fikirler, destekler ve bu aşamaya gelmemde büyük paya sahip olan Sayın Prof. Dr. Ali Gökhan Yavuz'a teşekkürü bir borç bilirim. Ayrıca vermiş olduğu telkinlerle, desteklerle bu sürecin bitiminde büyük katkısı olan Sayın Prof. Dr. Bülent Tavlı'ya teşekkürlerimi sunarım. Ek olarak tez sürecinde vermiş oldukları geri bildirimlerinden ve desteklerinden ötürü Sayın. Prof. Dr. Necaattin Barışçı'ya, Sayın Doç. Dr. Murat Lüy'e ve Mehmet Boz'a teşekkür ederim.

Son olarak bu zamana kadar vermiş oldukları tüm destekler için canım annem Hatice Seyyar'a, babam Naci Seyyar'a ve kardeşlerim Eren Seyyar'a, Vildan Seyyar'a, Faik Seyyar'a ve Enise Seyyar'a teşekkürü bir borç bilirim.

İÇİNDEKİLER DİZİNİ

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

TABLOLAR DİZİNİ

Tablo **Sayfa**

KISALTMALAR DİZİNİ

Kısaltmalar		Açıklama
ESA	:	Evrişimsel Sinir Ağları / Convolutional Neural Networks (CNN)
ÇKA	:	Çok Katmanlı Algılayıcılar / Multilayer Perceptron (MLP)
ROC	:	ROC Eğrisi / Receiver Operating Characteristics
DÖ	:	Derin Öğrenme / Deep Learning (DL)
TSA	:	Tekrarlayan Sinir Ağları / Recurrent Neural Networks (RNN)
UKSH	:	Uzun Kısa Süreli Hafıza / Long Short Term Memory (LSTM)
HTTP	:	Hiper-Metin Transfer Protokolü / Hyper-Text Transfer Protocol
URL	:	Tekdüzen Kaynak Bulucu / Uniform Resource Loader
ÇYKG	:	Çift Yönlü Kodlayıcı Gösterimleri / Bidirectional Encoder Representations from Transformers (BERT)
DDİ	:	Doğal Dil İşleme / Natural Language Processing (NLP)
SQL	:	Yapılandırılmış Sorgu Dili / Structured Query Language
XSS	:	Siteler Arası Betik Çalıştırma / Cross Site Scripting
DoS	:	Hizmet Engelleme / Denial of Service Attack
DDoS	:	Dağıtık Hizmet Engelleme / Distributed Denial of Service
MITM	:	Ortadaki Adam Saldırısı / Man in the middle
YSA	:	Yapay Sinir Ağları / Neural Networks (NN)

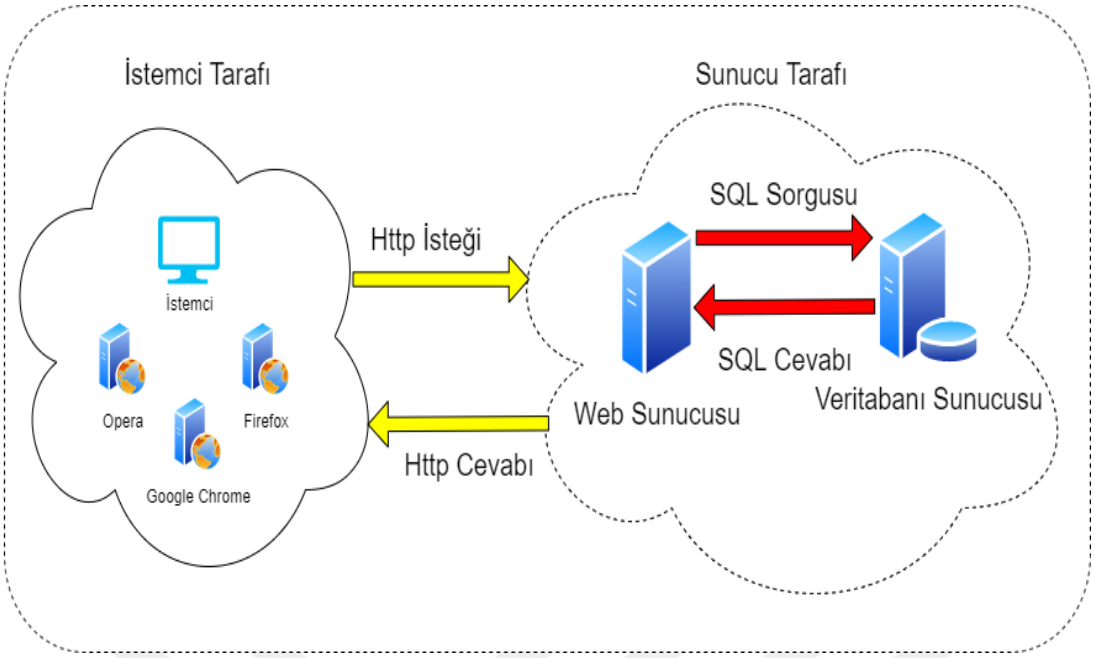
GloVe		Kelime Temsili için Global Vektörler / Global Vectors for Word Representation
OWASP	:	Açık Web Uygulaması Güvenlik Projesi/ Open Web Application Security Project
TCP	:	Gönderim Kontrol Protokolü / Transmission Control Protocol
IoT	:	Nesnelerin İnterneti / Internet of Things
OK	:	Otomatik Kodlayıcılar / Autoencoders (AE)
DVM	:	Destek Vektör Makineleri / Support Vector Machine (SVM)

1. GİRİŞ

Tezin bu bölümü web dünyası, web saldırılarının tespitine yönelik yapılan makine öğrenmesi ve doğal dil işleme yöntemleri hakkında bilgi içermektedir. Bölüm web dünyasının çalışma şeklinin anlatımı ile başlamaktadır. Devamında siber saldırılar alanında otorite olan kuruluşların web saldırılarını derecelendirmelerinin verilmesi ile devam etmektedir. Doğal Dil İşleme ve Makine Öğrenmesi yöntemleri kullanılarak çeşitli web saldırılarında başarıya ulaşan ve literatüre katkı sağlayan bazı çalışmalara yer verilmesi ile tamamlanacaktır.

1.1. Amaçlar

İnternet'in, su, gaz, elektrik ve diğer kamu hizmetleri kadar önemli hale gelmesiyle, İnternet ile ilgili teknolojiler yaşamımızda, sanki yeni bir uzuv gibi yerini almıştır. Web uygulamalarının kullanımı, İnternet'in popülaritesinin artmasıyla eş zamanlı olarak yaygınlaşmıştır. Web uygulamaları, Word Wide Web (WWW) aracılığıyla kullanılmaktadır. Aslında WWW, istemci-sunucu modeline dayanan dağıtık ve kitlesel bir bilgi sistemi olarak düşünülebilmektedir. Tarayıcılar ise istemciler ve sunucular arasındaki ilişkileri düzenleyen programlardır. Tekdüzen Kaynak Bulucu (Uniform Resource Loader-URL), istemcinin sunucuyla iletişim kurarken kullandığı İnternet Protokolü (Internet Protocol-IP) adreslerini metin haline getirmektedir. Ağdaki her cihazın benzersiz bir adresi (IP adresi) vardır. IP adresleri kolayca ezberlenemediği için URL, istemcilerin sunucuyla iletişim kurarken kullandığı IP adreslerini metin haline getirmektedir. Web mimarisinin çalışma mantığı Şekil 1'de gösterilmektedir.



Şekil 1. Web mimarisine genel bir bakış.

Bir Köprü Metni Aktarım Protokolü (Hypertext Transfer Protocol-HTTP) sunucusu çalışırken, tüm HTTP isteklerine açıktır. Sunucuya erişim sağlanması için ağ güvenlik duvarlarında yer alan HTTP kapısı (port) açık bırakılmaktadır. HTTP istekleri, geçerli HTTP istekleri gibi göründükleri, geleneksel güvenlik duvarları tarafından kabul edildiği ve kapsamlı bir şekilde araştırılmadığı için kötü amaçlı kod parçaları içerebilir. Saldırganlar, genellikle HTTP protokolü aracılığıyla web sistemlerini hedeflemektedir. Bir web sunucusu, bir istek aldığında yanıtını genelde web sayfalarıyla vermektedir. Web sunucularının, web sayfalarını, istemcilere depolamak, istemcilere hizmet sunmak ve web sayfalarını işlemek gibi çeşitli görevleri vardır. Web sunucusu ve web sayfaları arasındaki iletişim, HTTP ile kolaylaştırılmaktadır (Iqsyahiro Kresna & Rosmansyah, 2018). WWW'de en çok kullanılan protokollerden biri HTTP protokolü ya da onun güvenli uzantısı olan HTTP Güvenli (HTTP Secure) protokolüdür. Birçok protokolde olduğu gibi, HTTP ve HTTPS protokollerinde de güvenlik açıkları bulunmaktadır. Saldırganlar, bu güvenlik açıklarından yararlanarak Ortadaki Adam (Man in the Middle-MITM), Kaba Kuvvet (Brute Force), Dağıtılmış Hizmet Reddi (Distributed Denial of Service-DDoS), Yapılandırılmış Sorgu Dili Enjeksiyon (Structured Query Language-SQL) ve Siteler Arası Komut Dosyası Çalıştırma (Cross-Site Scripting-XSS) gibi saldırıları gerçekleştirmektedir (Luxemburk vd., 2021).

Ortadaki Adam saldırısında, kullanıcı ile geçit yolu arasındaki trafik yönlendirilir ve ara yönlendiricilerden biri değilken gerçek hedef gibi davranır. Adres Çözümleme Protokolü (Address Resolution Protocol-ARP) sahtekârlığı ve Güvenli Yuva Katmanı (Secure Sockets Layer-SSL) soyma tekniklerini birleştirerek sinyalleri Wi-Fi ağına gönderir. HTTPS, aktarılan verilere, kullanıcı bilgisayarının SSL başlığını ve HTTP paketi ekleyerek değiştirir (Chordiya vd., 2018).

Saldırganlar, hedefin kaynaklarını tüketmek için HTTP protokolünün GET veya POST yöntemlerini kullanarak botlardan Zaman Tetiklemeli Protokol (Time-Triggered Protocol-TTP) sel saldırısı gerçekleştirir. Belirli araçlar kullanılarak uygulama kaynak koduna erişilir ve hizmeti kesmek için bir DDoS saldırısı gerçekleştirilir (Bishnoi vd., 2021a).

HTTPS'de uygulanan doğal güvenlik önlemlerine rağmen, web sayfası parmak izleri, paket boyutları ve zamanlama bilgileri gibi bilgilerin HTTPS tarafından sızdırıldığı bildirilmiştir. Saldırganlar ise özellikle zayıf korunan web siteleri için önemli bir saldırı türü olan bu tür bilgiler kullanılarak hazırlanan özel listeler ile kullanıcıların şifrelerini tahmin etmek için Kaba Kuvvet saldırıları gerçekleştirirler (Luxemburk vd., 2021).

Web uygulamaları üzerinden kişisel bilgilerin internette paylaşılması, saldırganların iştahını kabartmaktadır. Web uygulamaları, SQL dilini kullanarak kullanıcılara ait kişisel bilgileri saklayan veri tabanları ile iletişim kurar. Açık Web Uygulama Güvenliği Projesi (Open Web Application Security Project-OWASP) 2017 raporunda yer alan önemli 10 güvenlik açığı, popülerlik sırasına göre SQL enjeksiyon ilk sırayı korurken (Wichers, 2017), 2013 OWASP raporunda ikinci sırayı tutan XSS saldırısının sıralamasını kaptırdığı görülmektedir (Wichers, 2013). SQL enjeksiyon ve XSS saldırıları, diğer saldırıların yanı sıra, veri tabanlarından kullanıcı bilgilerinin alınmasına veya web sayfalarındaki bilgilerin değiştirilmesine neden olur. Çeşitli başarılı klasik web saldırı tespit yöntemleri vardır. Ancak, geleneksel ve kural tabanlı yaklaşımlarla tatmin edici bir şekilde ele alınamayan sorunlu kullanım durumları bulunmaktadır. Örneğin, birçok farklı SQL enjeksiyon saldırısı oluşturmak mümkündür ve akla gelebilecek her saldırıya karşı koymak için çok fazla kural oluşturulması gerekmektedir. Birçok SQL enjeksiyon saldırısı nispeten basit önlemler kullanılarak engellenebilse de, yalnızca kural tabanlı karşı önlemler kullanılarak engellenmesi kolay olmayan SQL enjeksiyon saldırıları vardır. Son zamanlarda,

kalıpları tanımaya dayanan kullanım durumlarındaki Makine Öğrenimi (MÖ) ve Derin Öğrenme (DÖ) teknikleri, özellikle zorlu saldırı algılama senaryoları için daha iyi alternatifler olarak kendilerini kanıtlamıştır. Kötü niyetli saldırılar doğası gereği tekrarlayıcı olduğundan ve benzer kalıplar etrafında dönen kodları içerdiğinden, DÖ yaklaşımları bu kalıpları tanımada oldukça başarılıdır (Mac vd., 2018), (Iqsyahiro Kresna & Rosmansyah, 2018), (Luxemburk vd., 2021), (Chordiya vd., 2018), (Bishnoi vd., 2021b).

Mevcut MÖ ve DÖ tabanlı saldırı tespit yaklaşımlarında, otomatik kodlayıcı (autoencoder), word2vec, glove gibi kelime gömme (Word Embedding) yöntemleri kullanılmaktadır. Kelime gömme yöntemlerinin kullanılması, kural tabanlı modeller gibi geleneksel makine öğrenmesi yaklaşımlarına kıyasla kötü niyetli saldırı tespit görevlerinde başarı oranını arttırdığını göstermiştir (Komiya vd., 2011), (Hoang, 2019). Bununla birlikte, dönüştürücülerle ilgili son gelişmeler (örneğin, dönüştürücülerden, Çift Yönlü Kodlayıcı Temsilleri (BERT), Sağlam Bir Şekilde Optimize Edilmiş Bert Yaklaşımı (RoBERTa)) ve metin sınıflandırma görevlerindeki dikkate değer başarıları, zorlu saldırı algılama durumlarında kullanılmak üzere umut verici çözüm yöntemleri haline gelmiştir.

Son zamanlarda yaşanan Wikileaks, Yemeksepeti kullanıcı bilgilerinin ele geçirilmesi vb. ve son çalışmalar (Mac vd., 2018),(Lu Yu vd., 2019),(Y. Tian vd., 2017),(Z. Tian vd., 2020), kullanıcı kişisel bilgilerinin güvenliği ve gizliliği için web uygulama güvenliğinin ne kadar önemli olduğunu göstermektedir. Bu alanda çok sayıda çalışma olmasına rağmen, hala doldurulması gereken boşluklar olduğu tespit edilmiştir. OWASP raporları da göz önünde bulundurulduğunda, listenin ilk sıralarına yerleşmiş ve kullanıcılar için en tehlikeli saldırı çeşitleri arasında yer alan SQL enjeksiyon ve XSS saldırıları üzerine çalışma yapılmasına karar verilmiştir.

SQL enjeksiyon saldırıları, web uygulaması veri tabanlarını hedefleyen sunucu taraflı güvenlik açıklarını içermekte iken, XSS saldırıları, genellikle web uygulaması kullanıcılarını hedef alan istemci taraflı güvenlik açıklarıdır. Bahsedilen saldırılardan biri olan XSS saldırısı, istemcileri, diğer saldırı olan SQL enjeksiyon saldırısı ise sunucuları hedef aldığından SQL enjeksiyon ve XSS saldırılarını içeren HTTP isteklerinin, normal ve anomali istek olarak sınıflandırılmasının her iki taraf için de faydalı olacağını belirtmek mümkündür.

Bu tez kapsamında yukarda yer alan bilgiler göz önüne alınarak, DDİ ve DÖ modellerini temel alan HTTP isteklerini, normal ya da anomali istek olarak sınıflandıran bir mimarinin geliştirilmesi hedeflenmiştir.

1.2. Araştırma Soruları

Bu tez kapsamında yapılan çalışmanın amacı DÖ ve DDİ yöntemlerinden yararlanarak web sunucularına gelen HTTP isteklerinin, normal ya da anomali istek olarak sınıflandırılmasıdır. Önerilen mimari üç bölümden oluşmaktadır. Bunlar HTTP isteklerinin bir cümle gibi değerlendirilmesi ve kelimelere ayrılması, kelimelerin, kelime uzayında kelime vektörlerinin çıkarılması ve sınıflandırmanın yapılmasıdır. Yapılan çalışmanın amacını sorgulamak için bazı araştırma soruları aşağıda verilmiştir:

1- DDİ alanında kullanılan yöntemlerin elde etmiş olduğu başarının sentetik bir dil olan SQL dili üzerinde başarılı olup olmayacağı?

   DDİ yöntemlerinden olan BERT modelinin İngilizce üzerinde başarılı olmasının ardından, Fransızca, Çince, Arapça, Farsça ve Türkçe üzerinde de başarılı sonuçlar verdiği yapılan çalışmalar neticesinde görülmüştür. Bu bağlamda sentetik bir dil olan SQL dili üzerinde bu doğal işleme yöntemlerinin kullanılması ve başarılı sonuçlar elde edilmesi hedeflenmektedir.

2- Web saldırı tespitine yönelik yapılan çalışmalarda kullanılan veri setlerinden alınan örneklerin gerçek dünyaya uygun olup olmadığı?

   Yapılan çalışmalarda önerilen modellerin eğitilmesi ve test edilmesi aşamasında kullanılan veri setlerinden, genellikle bir normal istek bir anomali istek alınıp eğitim yapıldığı dikkatimizi çeken hususlardan olmuştur. Gerçek dünyada bu isteklerin hiçbir zaman bire bir gelmediği göz önüne alınarak 1:1, 1:10 ve 1:20 gibi örneklemler yapılarak literatürde yer alan çalışmalar kadar iyi ya da daha iyi başarı oranları elde edilmesi hedeflenmektedir.

3- HTTP istekleri için normalizasyon işleminin uygulanmasının faydası var mıdır?

   HTTP istekleri için normalizasyon işleminin yapılmadığı durumda sistemin başarısı değerlendirilecektir. Tarafımızca önerilen ve veri normalizasyonuna

ihtiyaç duymayan mimarinin, literatürde yer alan normalizayon yapılarak önerilen modellerin sınıflandırma başarıları ile karşılaştırılacaktır.

4- Önerilen modelin web saldırı tespit süresi, literatürde yer alan çalışmalar dikkate alındığında, mevcut çalışmaların tespit süresine göre daha hızlı mıdır?
Önerilen modelin, istekleri, normal ve anomali istek olarak, ek olarak istek başına ne kadar sürede sınıflandırma işlemini gerçekleştireceği ve başarı oranının ne olacağı belirlenecektir. Ayrıca elde edilen sonuçlar, literatürde yer alan ve tespit süresi verilen çalışma sonuçlarıyla karşılaştırılacaktır.

2. DERİN ÖĞRENME

Tezin bu bölümüne DÖ tarihçesi ile giriş yapılmaktadır. Bölüm, DÖ algoritmaları hakkında bilgi verilmesi ile devam etmektedir. Son olarak bölüm, tez kapsamında kullanılan DÖ algoritmaları anlatılarak tamamlanmaktadır.

2.1. Derin Öğrenme Tarihçesi

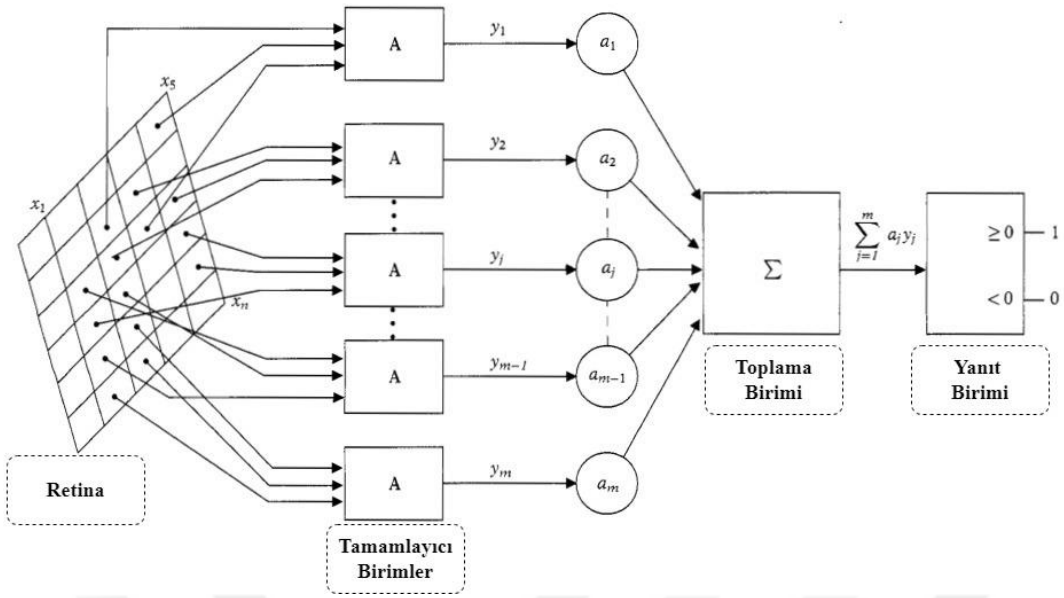
Derin öğrenme kavramı, gelişen ihtiyaçlar doğrultusunda makine öğrenmesinin bir dalı olarak ortaya çıkmıştır. Derin öğrenmenin gelişmesiyle beraber, bilgisayarlı görü, konuşma tanıma, doğal dil işleme ve ses tanıma gibi alanlarda da büyük ilerlemeler olduğu görülmüştür. Google, Facebook vb. şirketler derin öğrenmenin popülerliğinin artmasında büyük rol almışlardır. Diğer taraftan derin öğrenme tarihine bakıldığında sürecin 1940'lara dayandığı görülmektedir.

1943 yılında McCulloch ve Pitts, sinir ağları için matematik ve eşik mantığı adı verilen algoritmalara dayalı bir hesaplama modeli oluşturmasıyla başlamıştır (Mcculloch & Pitts, 1943).

Rosenblatt, basit toplama ve çıkarma kullanarak iki katmanlı bir bilgisayar sinir ağına dayanan bir örüntü tanıma algoritması olan algılayıcıyı (Perceptron) kavramını 1957 yılında ortaya atmıştır.

Algılayıcı, 1957'de psikolog Frank Rosenblatt, beynin özelliklerini ve biyolojik sistemlerin reseptörlerini içeren bir yapay sinir ağları sınıfı olarak "Algılayıcı: algılayan ve tanıyan bir otomat" önermiştir.

Şekil 2'de, Mark 1 Perceptron ağı gösterilmektedir. Daha sonra Rosenblatt, başlangıçta çeşitli teorik sinir ağları için genel bir isim olarak tasarlanan algılayıcı teriminin aslında çok özel bir donanım parçasıyla ilişkili olduğunu belirtmiştir (Rosenblatt, 1962).



Şekil 2. Mark 1 Perceptron mimarisi

Bir algılayıcının temel yapıtaşı, bir dizi X_i, i = 1, . . . , N girişini kabul eden bir öğedir ve bu girdilerin ağırlıklı toplamını hesaplamaktadır. Burada her girdi için sabit ağırlığı w sadece + 1 veya - 1 olabilmektedir. Daha sonra toplam bir θ eşiği ile karşılaştırılmaktadır ve toplamın eşiği aşıp aşmamasına bağlı olarak 0 ya da 1 olan bir y çıktısı üretilmektedir.

$$y = \begin{cases} 1 \; eğer \left(\sum_{i=1}^{N} w_i x_i\right) \geq \theta \\ 0 \; eğer \left(\sum_{i=1}^{N} w_i x_i\right) < \theta \end{cases} \tag{2.1}$$

Algılayıcı, duyu birimleri (S birimleri), ilişki birimleri (A birimleri) ve çıkış veya yanıt birimlerinden (R birimleri) oluşan bir sinyal iletim ağı olarak tanımlanmaktadır. Algılayıcının reseptörü, gözün retinasına benzemektedir ve bir dizi duyu elemanından (fotoseller) oluşmaktadır. Bir S biriminin uyarılıp uyarılmadığına bağlı olarak, ikili bir çıktı üretilmektedir. Rastgele seçilen bir dizi retina hücresi, ağın bir sonraki düzeyi olan A birimlerine bağlanmaktadır. Her A birimi, her A birimine girişler için + 1, - 1 ağırlıklarının rastgele atandığı, yukarıda bahsedilen temel yapı taşı gibi davranmaktadır. Tüm A birimleri için aynı eşik değeri belirlenmektedir.

k_{th}, A biriminin (k = 1, ... , m) ikili çıktısı Y_k, bir ak ağırlığı ile çarpılmaktadır ve tüm ağırlıkları + 1'e eşit olan temel yapıtaşı ile aynı şekilde bir toplama biriminde tüm m ağırlıklı çıktıların toplamı oluşturulmaktadır.

Her bir a_k ağırlığının pozitif, sıfır veya negatif olmasına izin verilir ve diğer ağırlıklardan bağımsız olarak değişebilir. Algılayıcının çıktısı, normalde 0'a ayarlanan bir eşiğe bağlı olarak yine ikilidir. Çıktının ikili değerleri, bir algılayıcının retinasına sunulabilen iki model sınıfını ayırt etmek için kullanılır. Verilen iki örüntü kümesini ayırt etmek için bir algılayıcının tasarımı, ağırlıklar a_k, k = 1,…,m, ve eşik değer olarak ayarlanmasını içermektedir.

Rosenblatt, algılayıcıları eğitmek için aşağıdaki yöntemin bir dizi varyasyonunu önermiştir. Sınıflandırılmış olarak bilinen desenler kümesinin bir kısmı retinaya sırayla sunulmaktadır ve tüm set gerektiği kadar sık tekrarlanmaktadır.

Algılayıcının çıktısı, bir kalıbın doğru sınıflandırılıp sınıflandırılmadığını belirlemek için izlenmektedir. Doğru sınıflandırma gerçekleşmediyse, ağırlıklar aşağıdaki "hata düzeltme" yöntemine göre ayarlanmaktadır: nth, kalıbı yanlış sınıflandırıldıysa kth ağırlığı için ak(n+1) yeni değeri hesaplanmaktadır.

$$a_k(n+1) = a_k(n) + y_k(n)\ x\ \delta(n) \qquad (2.2)$$

Burada, nth kalıbı 1 sınıfından ise δ(n) 1 olacaktır, nth kalıbı 2 sınıfından ise δ(n) -1 olacaktır. Bir desen doğru sınıflandırılmışsa, ağırlıkta herhangi bir ayarlama yapılmaz.

Tüm örüntülerin doğru bir şekilde sınıflandırılabileceği bir ağırlık kümesi varsa, örüntü sınıflarının lineer olarak ayrılabilir olduğu söylenir. Rosenblatt tarafından, model sınıfları doğrusal olarak ayrılabilir olduğunda, hata düzeltme "öğrenme" prosedürünün, tüm kalıpları doğru şekilde sınıflandıran bir ağırlık kümesine yakınsayacağı tahmin edilmiştir. Bu algılayıcı yakınsama teoreminin birçok ispatı daha sonra, en kısası A. J. Novikoff tarafından elde edildi. Daha sonraki katkılar, istatistiksel lineer diskriminant fonksiyonlarına basit algılayıcıyı ve bilinmeyen regresyon fonksiyonlarının sıfırlarını ve uçlarını bulmak için orijinal olarak geliştirilmiş olan gradyan iniş ve stokastik yaklaşım yöntemlerine hata düzeltme öğrenme algoritmasını ilişkilendirilmiştir.

Açıklanan basit algılayıcı, yalnızca S birimlerinden A birimlerine ve A birimlerinden tek R birimine ileri beslemeli bağlantıları olan bir dizi-çift algılayıcıdır. Bu ağdaki tek

uyarlanabilir öğe olan a_k ağırlıkları, doğrudan çıkış hatası açısından değerlendirilir. Bu bazen tek katmanlı algılayıcı olarak adlandırılır. Çıktı hatasını dolaylı olarak etkileyecek ayarlanabilen gizli elemanların olduğu katman yoktur. Bir veya daha fazla gizli eleman katmanına sahip bir algılayıcı, çok katmanlı algılayıcı olarak tanımlanmaktadır.

Minsky ve Papert, basit algılayıcılar hakkında, bazıları sınırlı örüntü sınıflandırması ve fonksiyona yaklaşma yeteneklerini gösteren çeşitli teoremleri kanıtlamıştır. Tek katmanlı algılayıcının "OR" mantıksal fonksiyonunu ve buna benzer birkaç fonksiyonunun uygulanamadığını göstermişlerdir. Minsky ve Papert'in sonuçları çok katmanlı algılayıcıları kapsamamaktadır (Minsky & Papert, 1969).

1974 Harvard Üniversitesi tezinde Paul Werbos, sistemin girdileri ve çıktıları arasındaki işlevsel bir ilişkiyi öğrenmek için türevlenebilir doğrusal olmayan bir sistemin ağırlıklarını uyarlamalı olarak ayarlamak için genel bir yakınsama prosedürü sundu. Prosedür, çıktılardan girdilere doğru geriye doğru çalışarak, sistemin tüm girdilerine ve ağırlıklarına veya parametrelerine göre çıktıların bazı fonksiyonlarının türevlerini hesaplar (Werbos, 1974). Bununla birlikte, Werbos'un bu çalışması, Rumelhart, Hinton ve Williams'ın bağımsız olarak, öğrenme örnekleri mevcut olduğunda örüntü sınıflandırma uygulamaları için çok katmanlı, ileri beslemeli algılayıcının ağırlıklarını uyarlamalı olarak ayarlamak için genel yöntemin özel bir durumunu bağımsız olarak popüler hale getirmesinden birkaç yıl sonrasına kadar, esasen fark edilmedi. Ağırlıkları gradyan inişini kullanarak uyarlayan bu algoritma, hata geri yayılımı veya sadece geri yayılım olarak bilinir. Çok katmanlı algılayıcı (ÇKA – Multilayer Perceptron (MLP)) ağının her bir ara katmanı boyunca çıktı katmanından türevleri yayar. 1980'lerde çok katmanlı algılayıcılar ve uygulamaları üzerindeki çalışmaların yeniden canlanması, doğrudan bu yakınsak geri yayılım algoritmasına atfedilebilir.

Giriş ve çıkış birimleri arasında yeterli sayıda ara veya "gizli" birime sahip çok katmanlı ileri beslemeli ağların "evrensel bir yaklaşım" özelliğine sahip olduğu gösterilmiştir: hemen hemen her işlevi istenen herhangi bir doğruluk derecesine yaklaştırabilirler. Ayrıca White tarafından geri yayılımın temelde stokastik yaklaşımın özel bir durumu olduğu ve bir kez daha sinir ağı öğrenme prosedürlerinin bilinen istatistiksel tekniklerle yakından ilişkili olduğu gösterilmiştir (Bishop, 1995).

Kelley, 1960 yılında yayınladığı “Optimal Uçuş Yollarının Gradyan Teorisi” başlıklı makalesinde, sürekli geri yayılım modelinin ilk halini göstermiştir. Bu model ilerleyen zamanlarda YSA’larda da karşımıza çıkmıştır. 1962 yılında Dreyfus, “Varyasyonel problemlerin sayısal çözümü” adlı makalesinde, geri yayılım modellerinde yapmış olduğu değişikliklerle, gelişime katkı sağlamıştır. Ivakhnenko ve Lapa, 1965 yılında polinom aktivasyon fonksiyonunu kullanan ilk ÇKA oluşturmuşlardır. Minsky ve Papert, Rosenblatt'ın algılayıcısının XOR gibi karmaşık fonksiyonları çözemeyeceğini gösterdikleri “Perceptrons” kitabını 1969 yılında yayınlamışlardır.

Ivakhnenko, 1971 yılında 8 katmanlı derin sinir ağı oluşturmuştur (Ivakhnenko, 1971).

El yazısı karakterler vb. görsel kalıpları tanıyabilen ilk evrişimsel sinir ağı mimarisi olan Neocognitron Fukushima tarafından 1980 yılında oluşturulmuştur.

1982 yılında ise Hopfield, tekrarlayan bir sinir ağı olan Hopfield Ağı'nı sunmuştur. İçeriğe göre adreslenebilir bir bellek sistemi olarak hizmet eden ve modern derin öğrenme çağının TSA modelleri için temel olmuştur (Hopfield, 1982).

1985 yılında ise Ackley, Hinton ve Sejnowski, rastgele tekrarlayan bir sinir ağı olan Boltzmann Makinesini önermişlerdir. Boltzmann Makinesini sadece giriş ve gizli katmanı vardır. Sejnowski, 1986 NeTalk’u sunmuştur. NeTalk metin olarak gösterilen İngilizce kelimeleri telaffuzunu öğrenen bir sinir ağı olarak hizmet vermiştir.

Hinton, Rumelhart ve Williams 1986 yılında “Öğrenme Temsillerini Geri Yayılım Hatalarıyla Öğrenme” başlıklı makalelerinde, sinir ağlarında geri yayılımın başarılı bir şekilde uygulandığını göstermişlerdir.

Smolensky, 1986 yılında giriş ve gizli katmanda katman içi bağlantının olmadığı bir Boltzmann Makinesi varyasyonu olan ve Kısıtlı Boltzmann Makinesi (RBM) olarak tanıtmıştır (Smolensky Paul, 1986).

LeCun, 1989 yılında el yazısı rakamlarını tanımak için ve evrişimli sinir ağını eğitmek için geri yayılımı kullanmıştır (LeCun vd., 1989).

Hochreiter, 1991 yılında derin sinir ağının öğrenilmesini son derece yavaşlatan kaybolan gradyan sorununu tanımlamıştır (Hochreiter, 1998).

Hochreiter ve Schmidhuber, 1997 yılında yayınladıkları makalede Uzun Kısa Süreli Bellek (UKSB) üzerine çalışmışlardır (Hochreiter & Schmidhuber, 1997).

Hinton ve arkadaşları, 2006 yılında birden fazla Kısıtlı Boltzmann Makinesi'ni katmanlar halinde bir araya getirip Derin İnanç Ağları adını verdikleri çalışmayı yayınlamışlardır (Hinton vd., 2006).

NG ve ekibi, 2008 yılında eğitim süresini hızlandırmak ve derin sinir ağlarını eğitmek için GPU'ları kullanmaya başlamıştır (Ng vd., 2009).

Stanford'da üniversitesinde profesör olarak çalışan Li, 2009 yılında derin öğrenme topluluğunun, etiketli veri bulma konusunda çekmiş olduğu etiketli veri üzerine bir çalışma gerçekleştirerek, 14 milyon etiketli veriden oluşan ImageNet'i başlatmıştır (K. Li vd., 2009).

Bengio, Bordes, Glorot, 2011 yılında "Deep Sparse Rectifier Neural Networks" başlıklı makalelerinde ReLU aktivasyon fonksiyonunun kaybolan gradyan problemini önleyebileceğini göstermişlerdir (Bengio vd., 2011).

Bir GPU uygulamalı ESA modeli olan AlexNet 2012 yılında Krizhevsky tarafından tasarlanmıştır. Imagenet'in görüntü sınıflandırma yarışmasını yüksek doğrulukla kazanmıştır (Krizhevsky vd., 2012).

2014 yılında GAN olarak da bilinen Çekişmeli Üretici Sinir Ağı, Goodfellow tarafından 2014 yılında tanıtılmıştır. GAN'lar, gerçek benzeri verileri sentezleme yeteneği nedeniyle derin öğrenmenin uygulanma alanlarından olan moda, sanat ve bilimde yerini almıştır (Goodfellow vd., 2014).

2.2. Öğrenme Çeşitleri

Denetimli öğrenme, etiketli veri kümelerinin kullanımıyla tanımlanan bir makine öğrenimi yaklaşımıdır. Bu veri kümeleri, verileri sınıflandırmak veya sonuçları doğru bir şekilde tahmin etmek için algoritmaları eğitmek veya denetlemek için tasarlanmıştır. Model, etiketlenmiş girdi ve çıktıları kullanarak doğruluğunu ölçebilmekte ve zamanla öğrenebilmektedir.

Denetimli öğrenme, veri madenciliği sırasında iki tür probleme ayrılabilmektedir: sınıflandırma ve regresyon:

Sınıflandırma problemleri, test verilerini ayırmak ve belirli kategorilere doğru bir şekilde atamak için denetimli öğrenme algoritmaları kullanmaktadır. Doğrusal

sınıflandırıcılar, Destek Vektör Makineleri, Karar Ağaçları ve Rastgele Orman, yaygın sınıflandırma algoritması türleri arasında yer almaktadır.

Regresyon, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi anlamak için bir algoritma kullanan başka bir denetimli öğrenme yöntemi türüdür. Bazı popüler regresyon algoritmaları Lineer Regresyon, Lojistik Regresyon ve Polinom Regresyondur.

Denetimsiz öğrenme, etiketlenmemiş veri kümelerini analiz etmek ve kümelemek için makine öğrenimi algoritmalarını kullanmaktadır. Bu algoritmalar, insan müdahalesine ihtiyaç duymadan verilerdeki gizli kalıpları keşfetmektedir. Denetimsiz öğrenme modelleri üç ana görev için kullanılmaktadır: kümeleme, ilişkilendirme ve boyutluluk azaltma.

Kümeleme, etiketlenmemiş verileri benzerliklerine veya farklılıklarına göre gruplandırmaya yönelik bir veri madenciliği tekniğidir.

İlişkilendirme, belirli bir veri kümesindeki değişkenler arasındaki ilişkileri bulmak için farklı kurallar kullanan başka bir denetimsiz öğrenme yöntemi türüdür.

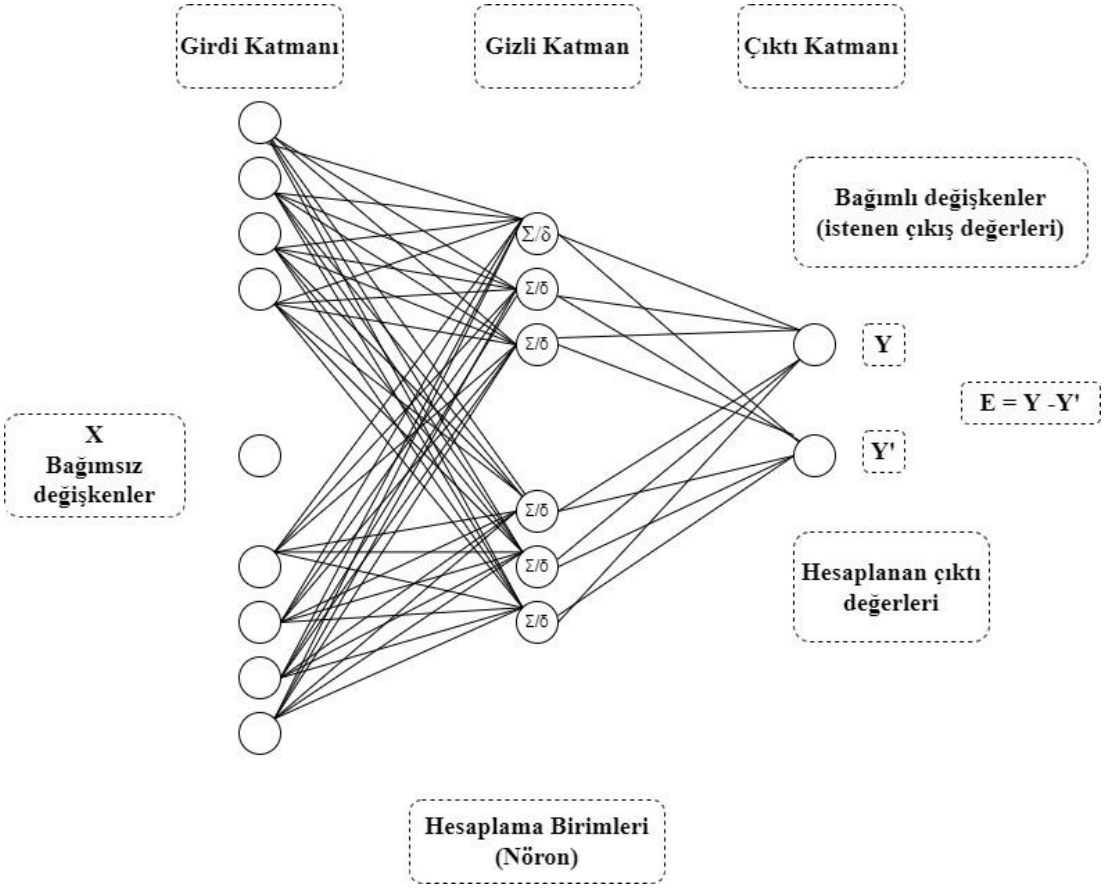
Boyut azaltma, belirli bir veri kümesindeki özelliklerin veya boyutların sayısı çok yüksek olduğunda kullanılan bir öğrenme tekniğidir. Veri bütünlüğünü korurken, veri girişi sayısını yönetilebilir bir boyuta düşürmektedir. Popüler bir boyut azaltma yöntemine Temel Bileşen Analizi (Principal Component Analysis-PCA) denir.

Pekiştirmeli öğrenmede algoritma, belirli bir durum kümesi için bir hedef duruma yol açan eylemleri öğrenmektedir. Çevre ile etkileşime girerek her durum veya eylemden sonra geri bildirim sinyalleri alan geri bildirim tabanlı bir öğrenme modelidir. Bu geribildirim bir ödül olarak çalışmaktadır ve aracının amacı, performanslarını iyileştirmek için olumlu ödülleri en üst düzeye çıkarmaktır. Modelin pekiştirmeli öğrenmedeki davranışı, insanların bir şeyleri deneyimleri yoluyla öğrenmesi ve çevre ile etkileşime girmesi nedeniyle insan öğrenmesine benzetilmektedir.

2.3. Çok Katmanlı Algılayıcı

Çok katmanlı ileri beslemeli sinir ağları olarak da adlandırılan geri beslemeli (backpropagation) öğrenme algoritmalarına sahip Çok Katmanlı Algılayıcılar (ÇKA) çok popülerdir ve çok çeşitli problemler için diğer sinir ağı türlerinden daha fazla

kullanılmaktadır. ÇKA algoritmasının temeli denetimli öğrenme yöntemine dayanmaktadır. Ağ, bilinen çıktılara sahip verilerdeki örneklere dayalı bir model oluşturmaktadır. Bir ÇKA algoritması, bir ilişki için dolaylı bilgileri içeren örneklerden, yalnızca bu ilişkiyi çıkartmaktadır.



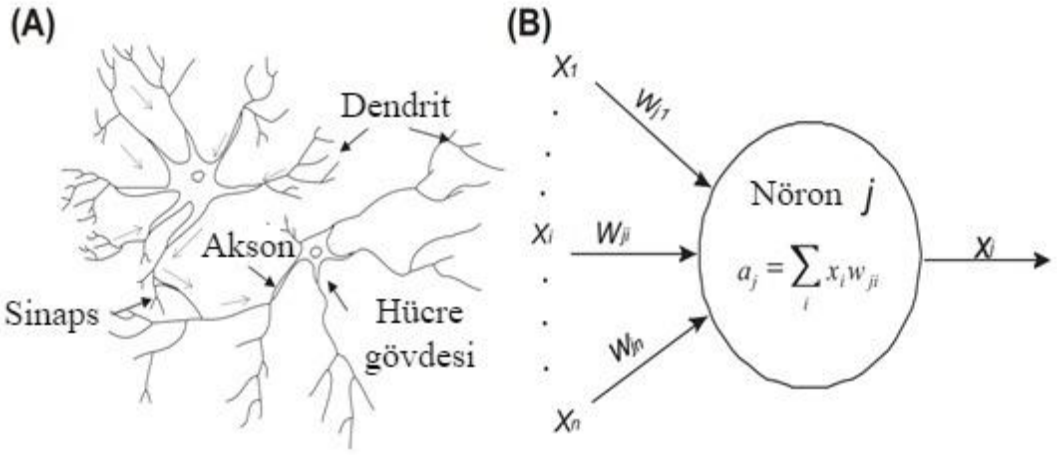
Şekil 3. Bir giriş katmanı, bir veya daha fazla gizli katman(lar) ve bir çıkış katmanı içeren üç katmanlı ileri beslemeli sinir ağı.

Bir ÇKA mimarisi Şekil 3'te olduğu gibi, çevrim içi hesaplama elemanlarına (nöronlar ve işlem birimleri) sahip üç katmandan (girdi, gizli ve çıktı) oluşur. Bilgi, girdi katmanından çıktı katmanına gizli katman aracılığıyla akmaktadır. Bir katmandaki tüm nöronlar, bitişik katmanlardaki nöronlara tamamen bağlıdır. Bu bağlantılar, hesaplama sürecinde ağırlıklar (bağlantı yoğunluğu) olarak temsil edilmektedir. Ağırlıklar, sinyalin ağda yayılmasında önemli bir rol oynamaktadır. Sinir ağının problem çözme ilişkisi hakkında bilgisi içerirler. Girdi katmanındaki nöronların sayısı modeldeki bağımsız değişkenlerin sayısına bağlıyken, çıktı katmanındaki nöronların sayısı bağımlı değişkenlerin sayısına eşittir. Çıkış nöronlarının sayısı, tek veya çoklu olabilmektedir. Ayrıca, hem gizli katmanların sayısı hem de nöronların sayısı, modelin

karmaşıklığına bağlıdır ve ÇKA modelinin geliştirilmesinde önemli parametreler olarak tanımlanmaktadır.

2.3.1. Öğrenme Algoritması

Bir ÇKA, istenen hedef değerler ile modelden hesaplanan değerler arasındaki hataları en aza indirmek için eğitilmekte ya da öğretilmektedir. Ağ, yanlış yanıt verirse veya hatalar belirli bir eşikten büyükse, bunları en aza indirmek için ağırlıklar güncellenmektedir. Böylece hatalar azaltılmakta ve sonuç olarak ağın gelecekteki yanıtlarının doğru olması beklenmektedir. Öğrenme yönteminde, girdi veri kümeleri ve istenen hedef model çiftleri sırayla ağa sunulmaktadır. Bir ÇKA'nın öğrenme algoritması, bir ileri yayılma adımı ve ardından bir geriye yayılma adımı içermektedir. Bir ÇKA, ileri yayılma ve geri yayılma aşamalarını tamamlayarak öğrenmektedir.



Şekil 4. (a) Biyolojik bir nöronun ve **(b)** Bir ağdaki temel işleme elemanı olarak bir yapay nöronun şematik diyagramları. Oklar bilgi akışının yönünü göstermektedir.

İleri yayılım aşaması, girdi katmanına bir girdi modelinin sunulmasıyla başlamaktadır. Bir ÇKA, dendritlerin, nöronlardan sinyal alıp hücre gövdesine gönderdiği biyolojik sinir sistemlerinde olduğu gibi, giriş ve çıkış nöronları aracılığıyla bilgiyi alıp ve özetlemektedir. Genel yapı ise Şekil 4-a ve 4-b'te örneklenmektedir. Bir ÇKA eğitimi, istenen hedef ve hesaplanan model çıktısı arasındaki hatayı en aza indirmek için yinelemeli bir gradyan algoritmasına dayanmaktadır.

p deseni için gizli katmanın j nöronuna net girdisi ($NET_{p,j}$), girdi katmanının her çıktısının ($x_{p,i}$; girdi değeri) toplamının, ağırlıkla ($v_{p,ji}$) çarpımı olarak

hesaplanmaktadır. Gizli katmanın ($z_{p,j}$) nöronunun j çıktısını ve çıktı katmanının ($o_{p,k}$) nöron k çıktısını aşağıdaki gibi hesaplamak için bir aktivasyon fonksiyonu uygulanmaktadır:

$$f(NET) = \frac{1}{1+\exp(-\lambda NET)} \tag{2.3}$$

Burada, "1", bir aktivasyon fonksiyonu katsayısıdır ve NET, $z_{p,j}$ veya $o_{p,k}$ olarak aşağıdaki gibi ifade edilmektedir:

$$z_{p,j=} = f(\sum_{i} x_{p,i} v_{p,ji}) \tag{2.4}$$

$$o_{p,k=} = f(\sum_{j} z_{p,j} w_{p,kj}) \tag{2.5}$$

burada $v_{p,ji}$ ve $w_{p,kj}$ girdi katmanının i nöronu ile gizli katmanın j nöronu arasındaki ve p deseni için sırasıyla gizli katmanın j nöronu ve çıkış katmanının k nöronu arasındaki bağlantıların ağırlıklarından oluşmaktadır. Ağırlıklar küçük rastgele sayılar olarak başlatılmaktadır. Doğrusal fonksiyonlar, Eşik fonksiyonları ve Sigmoid fonksiyonlar gibi çeşitli transfer fonksiyonları kullanılabilmektedir. Doğrusal olmaması nedeniyle genellikle bir Sigmoid fonksiyonu kullanılmaktadır. Bu fonksiyonlar Şekil 5'te görüldüğü gibi çalışmaktadır. Öğrenme algoritması, hatayı en aza indirmek için ağırlıkları ($v_{p,ji}$ ve $w_{p,kj}$) değiştirmektedir. p desenindeki her bir nörondaki hataların (Ep) toplamı aşağıdaki gibi hesaplanmaktadır:

$$E_p = \frac{1}{2}\sum_{k}(d_{p,k} o_{p,k})^2 \tag{2.6}$$

$$TE = \sum_{p} E_p \tag{2.7}$$

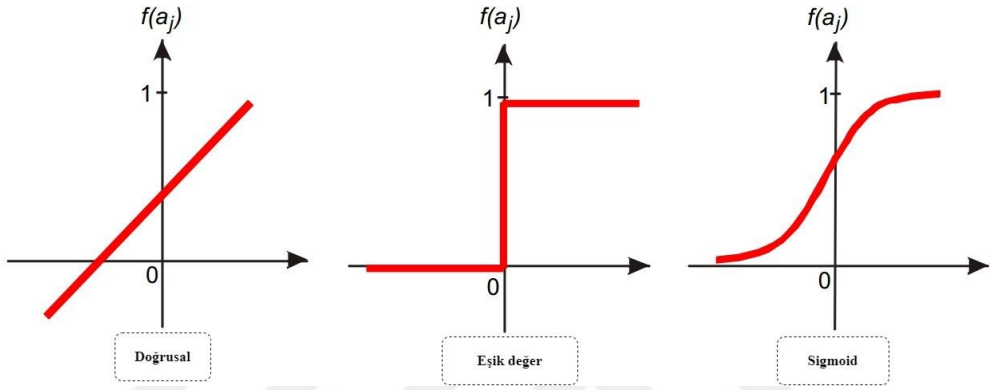
Burada, $d_{p,k}$, nöron k'deki p desenine karşılık gelen hedef değerini temsil etmekte ve TE, bir yineleme sırasındaki toplam hatayı temsil etmektedir.

İleri yayılma aşaması, aktivasyon seviyesi hesaplamaları gizli katman(lar) aracılığıyla çıkış katmanına yayılırken devam etmektedir. Ardışık her katmanda, her nöron

girdilerini toplanmakta ve ardından çıktısını hesaplamak için bir transfer fonksiyonu uygulanmaktadır. Ağın çıktı katmanı daha sonra nihai yanıtı, yani tahmini hedef değeri üretmektedir.

Geri yayılım aşaması, her nöronla ilişkili hata değeri, o nöronla ilişkili hata miktarını yansıtmaktadır. Sonuç olarak, nöron uygun ağırlık ayarı için geri yayılmaktadır. Çıkış nöronlarındaki ağırlıklar aşağıdaki gibi güncellenmektedir:

$$\delta_{p,k(0)} = o_{p,k}(1 - o_{p,k})(d_{p,k} - o_{p,k}) \tag{2.8}$$



Şekil 5. YSA modellerinde yaygın olarak kullanılan üç tip transfer fonksiyonu.

$$\Delta w_{p,kj}\,(t+1) = \eta\delta_{p,k(o)}o_{p,k} + \alpha\Delta w_{p,kj} \tag{2.9}$$

$$w_{p,kj}\,(t+1) = w_{p,kj}(t) + \Delta w_{p,kj}(t+1) \tag{2.10}$$

Burada, $\delta_{p,k(o)}$, p deseni için çıktı katmanının k nöronundaki hata sinyalini, η öğrenme oranı katsayısını ve α momentum katsayısını temsil etmektedir.

Uygun öğrenme hızı ve momentum katsayısı deneyimler sonucunda seçilmektedir. Her iki değerin de sıfır ile bir arasında bir aralığı bulunmaktadır. η için büyük bir değer ağda kararsızlığa ve yetersiz öğrenmeye neden olabilirken, aşırı derecede küçük bir değer aşırı yavaş öğrenmeye neden olabilmektedir. Tipik olarak, öğrenme oranı, ağda verimli öğrenmeyi sağlamak için değiştirilmektedir.

Örneğin, daha iyi bir öğrenme performansı elde etmek için, h'nin değeri başlangıçta yüksek olabilmektedir ve daha sonra öğrenme oturumu sırasında düşebilmektedir.

Momentum katsayısı, salınımlı ağırlık değişikliklerinden kaçınmak için tanımlanmaktadır. Mevcut ağırlığı hesaplamak için önceki ağırlıkların etkisine katkıda bulunmaktadır.

Gizli katmandaki ağırlıkların ayarlanması, tüm bu ürünlerin toplanması ve ardından ezme fonksiyonunun türevinin aşağıdaki gibi çarpılmasıyla üretilen hata sinyali (δ) dışında çıkış katmanındakine benzer bir şekilde gerçekleştirilmektedir:

$$\delta_{p,j(h)} = z_{p,j}(1 - z_{p,j}) \sum_{k} \delta_{p,k(o)} w_{p,kj} \tag{2.11}$$

Burada, $\delta_{p,j(h)}$, p deseni için gizli katmanın j nöronundaki hata sinyali olarak tanımlanmaktadır (Park & Lek, 2016).

Öğrenme algoritması aşağıda gösterildiği gibi özetlenmektedir:

Adım 0 (Giriş):

• Ağa bir dizi eğitim çifti girilmektedir.

Adım 1 (Başlatma):

• Ağırlıklara küçük rastgele değerler atılmaktadır.

• Öğrenme hızı, momentum katsayısı ve TEmax (maksimum tolere edilebilir hata) gibi parametreler ayarlanmaktadır.

Adım 2 (Eğitim döngüsü):

• Ağ giriş modeli giriş katmanına uygulanmaktadır.

Adım 3 (İleriye yayılma):

• Sinyal ağ üzerinden ileri doğru iletilmektedir.

• Ağ çıkış vektörü hesaplanmaktadır.

Adım 4 (Çıkış hatasının ölçümü):

• Çıkışların her biri için hataları, istenen hedef ve ağ çıkışı arasındaki fark hesaplanmaktadır.

Adım 5 (Hata geri yayılımı):

• Ağırlıkları hatayı en aza indirecek şekilde ayarlamak için hataları geriye doğru yayılmaktadır.

Adım 6 (Bir yineleme döngüsü):

• Tüm eğitim verisi setinin bir kez döngüye girip girmediğini kontrol edilmektedir.

• Tüm eğitim veri seti için Adım 2 ile 5 arası tekrarlanmaktadır.

7. Adım (Toplam hata kontrolü):

• Mevcut toplam hatanın (TE) kabul edilebilir olup olmadığını kontrol edilmektedir.

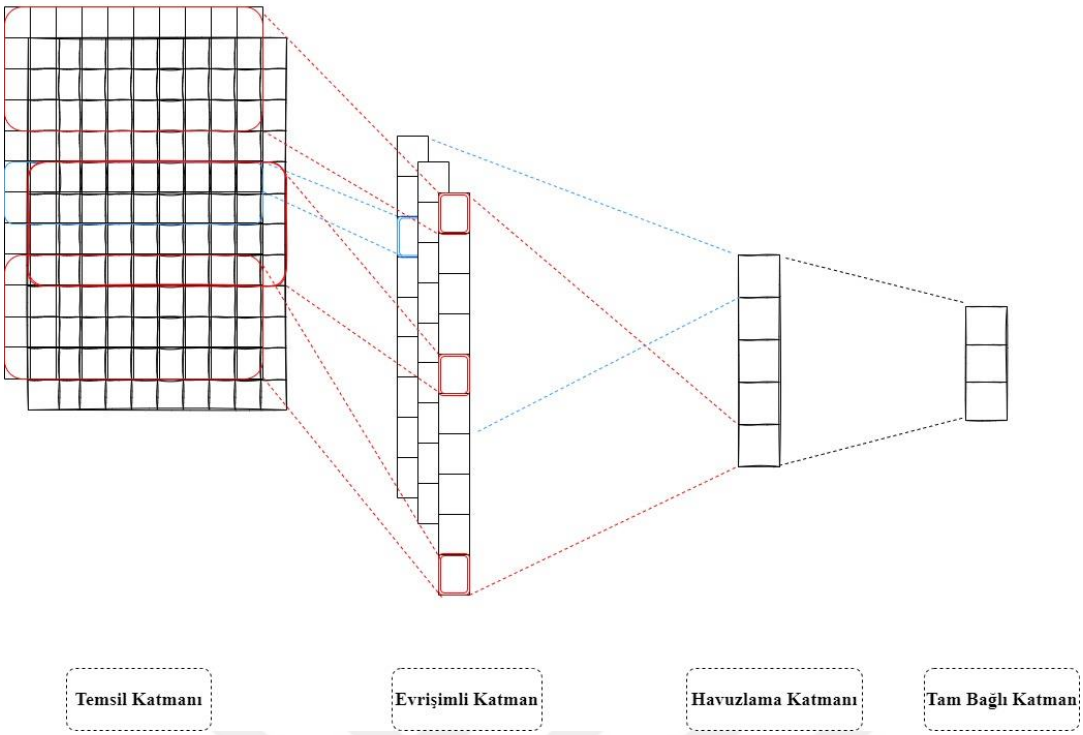
• TE < TEmax ise, eğitim sürecini sonlandırılmaktadır ve son ağırlıkların çıktısını alınmaktadır; aksi takdirde, tüm sistem için toplam hata kabul edilebilir şekilde düşük olana veya önceden tanımlanmış yineleme sayısına ulaşılana kadar, 2. adımdan 6. adıma kadar tekrarlanmaktadır.

2.4. Evrişimsel Sinir Ağı

Evrişimsel Sinir Ağı (ESA), bir tür tipik yapay sinir ağıdır. Bu tür ağlarda, her katmanın çıktısı, bir sonraki nöron katmanının girişi olarak kullanılmaktadır. Çok katmanlı evrişim işlemi, çıktı katmanına kadar her katmanın sonuçlarını doğrusal olmayan şekilde dönüştürmek için uygulanmaktadır.

2.4.1. ESA Mimarisi

ESA üç tip katmandan oluşmaktadır. Bu katmanlar evrişim katmanları (convolutional layer), havuzlama katmanları (pooling layer) ve tam bağlantılı katmanlar (fully-connected layer) olarak tanımlanmaktadır. Bu katmanların birleştirilmesiyle, bir ESA mimarisi meydana getirilmektedir.



Şekil 6. ESA model yapısı

ESA modelinin temel fonksiyonu dört temele dayandırılabilmekte ve Şekil 6'da sunulmaktadır.

1. Diğer YSA formlarında olduğu gibi, giriş katmanı kelimelerin temsil değerlerini tutmaktadır.

2. Evrişim katmanı, ağırlıkları ile giriş hacmine bağlı bölge arasındaki skaler çarpımı hesaplayarak, girişin yerel bölgelerine bağlı nöronların çıkışını belirlemektedir. Doğrultulmuş lineer birim genellikle ReLU (Rectified Linear Unit) (Agarap, 2018) olarak kısaltılmaktadır. Önceki katman tarafından üretilen aktivasyonun çıktısına Sigmoid vb. gibi bir aktivasyon fonksiyonunu uygulamayı amaçlamaktadır.

3. Havuzlama katmanı, verilen girdinin uzamsal boyutluluğu boyunca basitçe alt-örnekleme gerçekleştirerek ve bu aktivasyon içindeki parametre sayısını daha da azaltmaktadır.

4. Tam bağlantılı katmanlar daha sonra standart YSA'larda bulunan aynı görevleri yerine getirerek ve aktivasyonlardan sınıflandırma için kullanılmak üzere sınıf puanları üretmeye çalışmaktadır. Performansı artırmak için bu katmanlar arasında ReLU kullanılması da önerilmektedir. Bu basit dönüşüm yöntemiyle, ESA'lar,

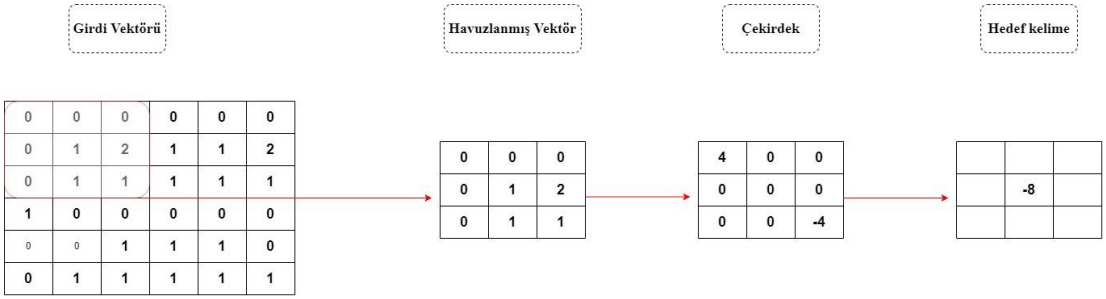
sınıflandırma, regresyon amaçları, sınıf puanları üretmek için evrişimsel ve alt-örnekleme tekniklerini kullanarak orijinal girdi katmanını katman-katman dönüştürebilmektedir.

Bir ESA genel mimarisini basitçe anlamaktan ziyade bu ESA modellerinin oluşturulması ve optimizasyonu üzerinde zaman harcayarak, daha iyi kavranabileceği ifade edilmektedir.

2.4.2 Evrişimsel Katman

Adından da anlaşılacağı gibi, evrişim katmanı, ESA'larin nasıl çalıştığı konusunda hayati bir rol oynamaktadır. Katman parametreleri, öğrenilebilir çekirdeklerin kullanımına odaklanmaktadır. Bu çekirdekler genellikle uzamsal boyutluluk açısından küçüktür, ancak girdinin derinliğinin tamamı boyunca yayılmaktadırlar. Veri bir evrişimli katmana çarptığında, katman, bir iki boyutlu (two-dimensional 2-D) aktivasyon haritası üretmek için her filtreyi girdinin uzamsal boyutluluğu boyunca sarmaktadır.

Girdide gezinirken, çekirdekteki her bir değer için skaler olarak hesaplama yapılmaktadır. Bu hesaplamayla ağ, girdinin belirli bir uzamsal konumunda belirli bir özellik görmesiyle filtreyi öğrenmektir. Bunlar genellikle aktivasyonlar olarak bilinmektedir. Şekil 7'de olduğu gibi özetlenebilmektedir.



Şekil 7. Evrişimsel bir katmanın görsel bir temsili. Çekirdeğin merkez elemanı, daha sonra hesaplanan ve kendisinin ve yakındaki kelimelerin ağırlıklı toplamı ile değiştirilen giriş vektörünün üzerine yerleştirilir.

Her çekirdek, evrişim katmanından tam çıktı hacmini oluşturmak için derinlik boyutu boyunca istiflenecek olan ve karşılık gelen bir aktivasyon haritasına sahip olmaktadır.

YSA'ları görüntü vb. gibi girdiler üzerinde eğitmek, modellerin etkili bir şekilde eğitilmesi için çok büyük olmasına neden olmaktadır. Bu, standart YSA nöronlarının

tam bağlantılı tarzına inmektedir. Bu nedenle bağlantı fazlalığından oluşan yükü hafifletmek için bir evrişim katmanındaki her nöron, giriş hacminin yalnızca küçük bir bölgesine bağlanmaktadır. Bu bölgenin boyutuna genel olarak nöronun alıcı alan (receptive field) boyutu denmektedir. Derinlik boyunca bağlantının büyüklüğü hemen hemen her zaman girdinin derinliğine eşittir.

Ağa, girdi olarak (64 × 64 × 3) boyutunda bir görüntü verilmesi ve alıcı alan boyutunu (6 × 6) olarak ayarlanması halinde, evrişim katmanındaki her nöron toplam 108 ağırlığa sahip olacaktır. (6 × 6 × 3) Burada 3, hacmin derinliği boyunca bağlantının büyüklüğü olarak değerlendirilmektedir. Açıklığa kavuşturmak için, YSA formlarında görülen standart bir nöron, her biri 12.288 ağırlığa sahip olmaktadır.

Evrişimsel katmanlar ayrıca çıktısının optimizasyonu yoluyla modelin karmaşıklığını önemli ölçüde azaltabilmektedir. Bunlar, derinlik, adım ve sıfır dolgu ayarı olmak üzere üç hiperparametre aracılığıyla optimize edilir.

Evrişimli katmanlar tarafından üretilen çıktı hacminin derinliği, katman içindeki nöronların sayısı aracılığıyla girdinin aynı bölgesine manuel olarak ayarlanabilmektedir. Bu, gizli katmandaki tüm nöronların önceden her bir nörona doğrudan bağlı olduğu diğer YSA formlarında görülebilmektedir. Bu hiperparametreyi azaltmak, ağın toplam nöron sayısını önemli ölçüde en aza indirebilir, ancak modelin örüntü tanıma yeteneklerini de önemli ölçüde azaltabilmektedir.

Alıcı alanı yerleştirmek için girdinin uzamsal boyutluluğu etrafındaki derinliğin ayarlandığı adım da tanımlanabilmektedir. Adım sayısı olarak 1 belirlenmesi halinde, son derece büyük aktivasyon haritası üreten yoğun şekilde örtüşen bir alıcı alan elde edilmektedir. Alternatif olarak, adımın daha büyük bir sayıya ayarlanması halinde, örtüşme miktarı azalmaktadır ve daha düşük uzamsal boyutlarda bir çıktı elde edilmektedir.

Sıfır doldurma, girdinin sınırlarını doldurmanın basit bir işlemidir ve çıktı hacimlerinin boyutsallığı konusunda daha fazla kontrol sağlamak için etkili bir yöntemdir.

Bu teknikleri kullanarak, evrişim katmanlarının çıktısının uzamsal boyutluluğunu değiştirilebilmektedir. Bunu hesaplamak için aşağıdaki formülü kullanılmaktadır.

$$\frac{(V - R) + 2Z}{S + 1} \tag{2.12}$$

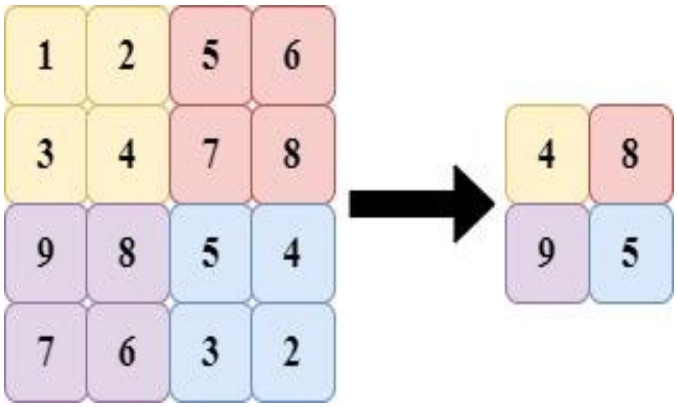
V'nin girdi hacim boyutunu (yükseklik×genişlik×derinlik) temsil ettiği yerde, R alıcı alan boyutunu temsil eder, Z sıfır dolgu seti miktarıdır ve S adıma atıfta bulunmaktadır. Bu denklemden hesaplanan sonuç bir tam sayıya eşit değilse, nöronlar verilen girdiye düzgün bir şekilde sığamayacakları için adım yanlış ayarlanmış olarak kabul edilmektedir.

Evrişim katmanı, içindeki toplam parametre sayısını büyük ölçüde azaltmak için yöntemler geliştirilmiştir. Parametre paylaşımı, bir bölge özelliğinin belirli bir uzamsal bölgede hesaplanması faydalıysa, başka bir bölgede de faydalı olacağı varsayımıyla çalışmaktadır. Çıktı hacmi içindeki her bir bireysel aktivasyon haritasını aynı ağırlıklar ve sapma ile kısıtlanması halinde, evrişim katmanı tarafından üretilen parametre sayısında büyük bir azalma görülecektir. Bunun bir sonucu olarak, geri yayılım aşaması meydana geldiğinde, çıktıdaki her bir nöron, toplam gradyanı derinlik boyunca toplanabilecek, böylece her birinin aksine yalnızca tek bir ağırlık kümesini güncelleyebilecektir.

2.4.3 Havuzlama Katmanı

Havuzlama katmanları, temsilin boyutsallığını kademeli olarak azaltmayı, böylece parametre sayısını ve modelin hesaplama karmaşıklığını daha da azaltmayı amaçlamaktadır. Havuzlama katmanı, girdideki her aktivasyon haritası üzerinde çalışır. Maksimum, minimum ve ortalama gibi havuzlama fonksiyonları kullanılarak boyutsallığını ölçeklendirmektedir.

Çoğu ESA'da bunlar, girdinin uzamsal boyutları boyunca 2'lik bir adımla uygulanan (2 × 2) boyutlu filtreye sahip maksimum havuzlama katmanları biçiminde gelmektedir. Bu, derinlik hacmini standart boyutunda tutarken aktivasyon haritasını orijinal boyutunun %25'ine kadar ölçeklendirebilmektedir.



Şekil 8. Maksimum havuzlama yöntemi örneği

Genellikle, havuzlama katmanlarının adımları ve filtreleri, katmanın girdinin uzamsal boyutluluğunun tamamı boyunca genişlemesine izin verecek şekilde (2 × 2) boyutlu filtreye ayarlanmaktadır. Bu ayarlama işlemine Şekil 8 örnek olarak verilebilmektedir. Ayrıca, adım sayısının 2 olarak ve filtre boyutunun (3 × 3) olarak ayarlandığında örtüşen havuzlama kullanılabilmektedir. Filtre boyutunun 3'ün üzerinde olması genellikle modelin performansını büyük ölçüde düşürmektedir. Evrişim katmanında genele olarak maksimum havuzlama kullanılmaktadır. Maksimum havuzlamanın dışında, ESA mimarilerinin genel havuzlama ve ortalama havuzlama içerebileceğini hatırlatmakta fayda görülmektedir.

Genel olarak havuzlama katmanları, L1/L2 fonksiyonları ve ortalama havuzlama dahil olmak üzere çok sayıda ortak işlemi gerçekleştirebilen havuzlama nöronlarından oluşmaktadır.

2.4.4 Tam Bağlantılı Katman

Tam bağlantılı katman, içlerindeki herhangi bir katmana bağlanmadan, iki bitişik katmandaki nöronlara doğrudan bağlı nöronları içermektedir. Bu, nöronların geleneksel YSA formlarındaki bağlanma şekline benzemektedir.

2.5. Tekrarlayan Sinir Ağları

TSA, eylem tanıma, sahne etiketleme ve dil işleme gibi sıralı öğrenme problemlerinde yaygın olarak kullanılmış ve etkileyici sonuçlar elde edilmiştir. ESA gibi ileri beslemeli ağlarla karşılaştırıldığında, bir TSA, son gizli durumun, bir sonraki duruma

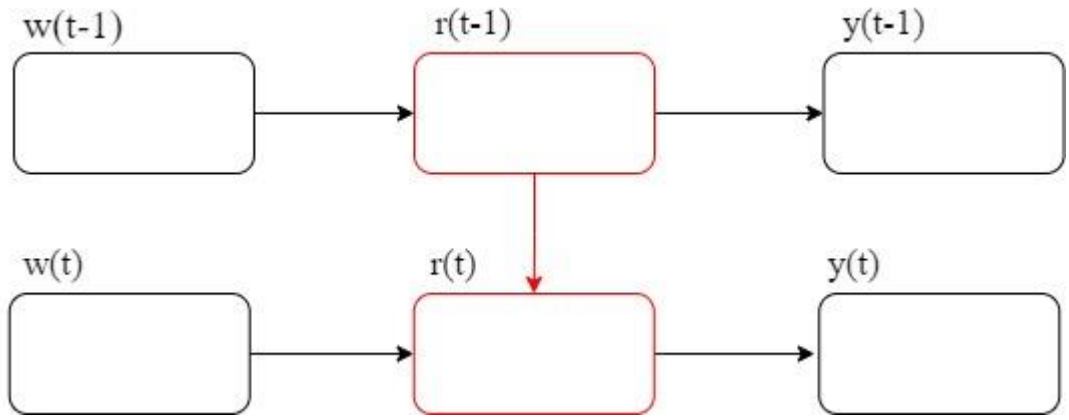
girdi olduğu tekrarlayan bir bağlantıya sahiptir. Durumların güncellenmesi şu şekilde açıklanabilir:

$$h_t=\sigma(\mathrm{W}x_t + Uh_{t-1} + b) \quad (2.13)$$

Burada,

Burada, $x_t \in R^M$ ve $\mathbf{h}_t \in R^N$, sırasıyla t adımında giriş ve gizli durumdur. $W \in R^{N\times M}$, $U \in R^{N\times N}$ ve $b \in R^N$, mevcut girdi ve tekrarlayan girdi için ağırlıklar ve nöronların bias değerleridir. σ, nöronların eleman bazında aktivasyon fonksiyonudur ve N, bu TSA katmanındaki nöronların sayısıdır.

TSA'ların eğitimi, tekrarlayan ağırlık matrisinin tekrar tekrar çarpımı nedeniyle gradyan kaybolması ve patlama probleminden muzdariptir. Gradyan problemlerini ele almak için Uzun Kısa Süreli Bellek (Long Short-Term Memory-LSTM-UKSB) ve Kapılı Tekrarlayan Birim (Gated Recurrent Unit-GRU) gibi çeşitli TSA varyantları önerilmiştir. Şekil 9'da basit bir TSA gösterilmiştir.



Şekil 9. Basit bir TSA modeli

2.6. Softmax

Softmax aktivasyon fonksiyonu, çoklu sınıflandırma problemlerine çözüm olarak kullanılmaktadır. Doğrusal ya da sigmoid aktivasyon fonksiyonlarının çoklu sınıflandırma problemlerinde yetersiz kalması nedeniyle geliştirilmiştir.

Softmax fonksiyonu, Sigmoid aktivasyon fonksiyonuna benzer şekilde her sınıfın olasılığını döndürmektedir. Softmax aktivasyon fonksiyonunun denklemi aşağıda verilmektedir.

$$Softmax(S_i) = \frac{\exp(S_i)}{\sum_j \exp(S_j)} \tag{2.14}$$

Burada S, çıktı katmanının nöronlarından gelen değerleri temsil etmektedir. Üstel, doğrusal olmayan fonksiyon olarak görev almaktadır. Bu değerler, normalleştirmek ve daha sonra olasılıklara dönüştürmek için üstel değerlerin toplamına bölünmektedir. İlgili sınıflara ait olasılık değerlerinin toplamı 1'e eşittir.

2.7. Model Değerlendirmesi

Önerilen modelin performansını sınıflandırma doğruluğu açısından değerlendirmek için kullanılan birkaç metrik vardır. Performans değerlendirmelerinde kullanılan metrikler şunlardır: TP (True Positive-Gerçek Pozitif), TN (True Negative-Gerçek Negatif), FP (False Positive-Yanlış Pozitif) ve FN (False Negative-Yanlış Negatif) değerleri kullanılarak hesaplanan Doğruluk, Kesinlik, Hassasiyet, F1-ölçümü. F1-ölçüm değerini elde etmek için diğer metriklerin hesaplanması gerekmektedir.

Doğruluk, denklem 2.15'de verilen doğru tahminlerin toplam tahmin sayısına oranı olarak tanımlanmaktadır.

$$Doğruluk = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.15}$$

Kesinlik, denklem 2.16'de gösterildiği gibi pozitif olarak tahmin edilen değerlerden kaçının gerçekte pozitif olduğunu ifade eden bir ölçümdür.

$$Keskinlik = \frac{TP}{TP + FP} \tag{2.16}$$

Hassasiyet, geri çağırma, hatırlatma, pozitif olarak tahmin edilmesi gereken değerlerin ne kadarının pozitif olarak tahmin edildiğini ölçmek için kullanılmaktadır. Hassasiyet için kullanılan başka bir terim ise doğru pozitif oranıdır (True Positive Rate). Her ikisi de, denklem 2.17'de gösterildiği gibi modelin gerçek pozitif değerleri ne kadar doğru tahmin ettiğini ölçmek için kullanılmaktadır.

$$Hassasiyet = \frac{TP}{TP + FN} \tag{2.17}$$

Gerçek Negatif Oran (True Negative Rate-TNR), denklem 2.18'te sunulduğu gibi modelin gerçek negatif değerleri ne kadar doğru tahmin ettiğinin bir ölçüsüdür.

$$TNR = \frac{TN}{TN + FP} \tag{2.18}$$

Yanlış Pozitif Oranı (False Positive Rate-FPR), denklem 2.19'te verilen değer 0 olmasına rağmen 1 olduğu tahmin edilenlerin oranıdır.

$$FPR = \frac{FP}{FP + TN} \tag{2.19}$$

Yanlış Negatif Oran (False Negative Rate-FNR), Denklem 2.20'da verilen gerçek değer 1 olmasına rağmen 0 olduğu tahmin edilenlerin oranı olarak tanımlanır.

$$FNR = \frac{FN}{FN + TP} \tag{2.20}$$

ROC (Receiver Operating Characteristics - Alıcı Çalışma Karakteristiği), Denklem 2.21'de verilen bir Hassasiyet / TNR raporunu oluşturmak için kullanılmaktadır.

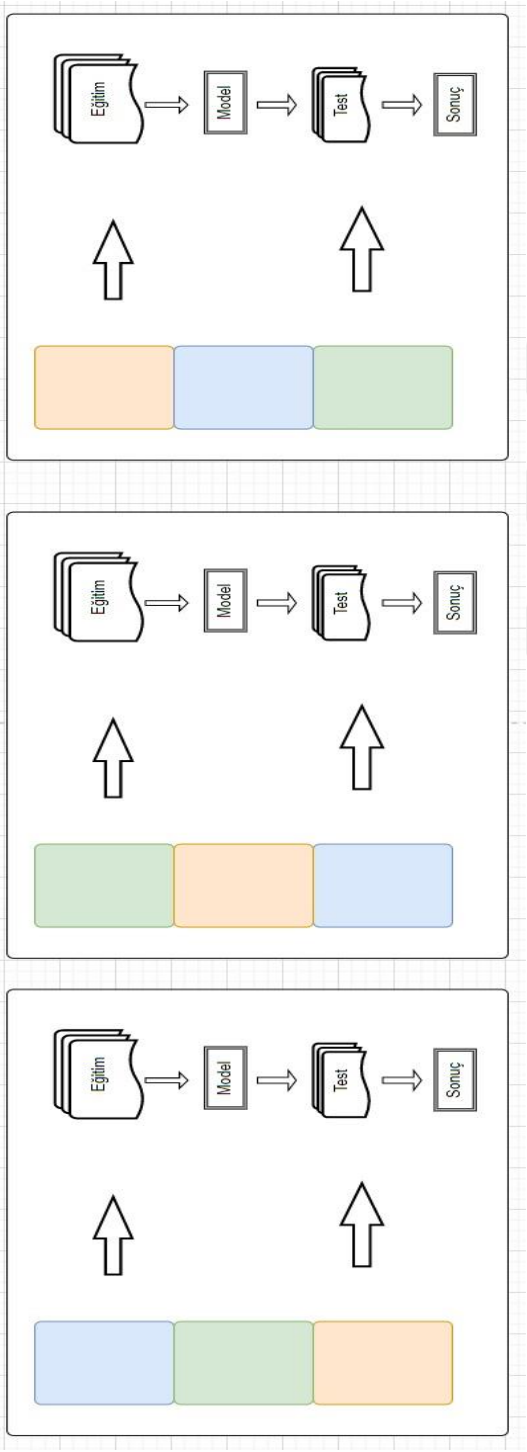
$$ROC = \frac{Keskinlik}{TNR} \tag{2.21}$$

F1-puanı aslında denklem 2.22'de verilen Hassasiyet ve Geri Çağırma değerlerinin harmonik ortalamasıdır.

$$F1 - puanı = 2 * \frac{Keskinlik * Hassasiyet}{Keskinlik + Hassasiyet} \tag{2.22}$$

Çapraz doğrulama (K-cross validation), fikri 1930'larda ortaya çıkmıştır. Çapraz doğrulama, verileri iki bölüme ayırarak öğrenme algoritmalarını değerlendirmenin ve karşılaştırmanın istatistiksel bir yöntemidir: biri bir modelin öğrenmesi veya eğitilmesi, diğeri ise doğrulanması için kullanılmaktadır. Tipik çapraz doğrulamada, eğitim ve doğrulama kümeleri, her bir veri noktasının doğrulanma şansı olacak şekilde birbirini izleyen turlarda çaprazlanmaktadır. Çapraz doğrulamanın temel biçimi k-katlı (k-fold) çapraz doğrulamadır. Diğer çapraz doğrulama biçimleri, k-katlı çapraz doğrulamanın özel durumlarıdır veya tekrarlanan k-kat çapraz doğrulama turlarını içermektedir. K-katlı çapraz doğrulamada, veriler ilk önce eşit (ya da neredeyse eşit) boyutlu bölümlere ya da katlara bölünmektedir. Ardından, her yinelemede verilerin

farklı bir katının doğrulama için tutulmaktadır. Kalan k1 katının öğrenme için kullanılmaktadır. Şekil 8'de k eğitim ve doğrulama yinelemeleri gösterilmektedir. Her sınıfın, verilerin %50'sini oluşturduğu bir ikili sınıflandırma probleminde, verileri her kattaki her sınıf örneklerin yaklaşık yarısını oluşturacak şekilde düzenlenmesinin en iyi seçim olduğu belirtilmiştir (Refaeilzadeh vd., 2020).



Şekil 10. Üçlü çapraz doğrulama prosedürü

3. DOĞAL DİL İŞLEME

Tezin bu bölümü, doğal dil işlemenin tarihçesi, gelişimi ve bu gelişmeye bağlı olarak gelişen teknikler hakkında bilgilerle başlamaktadır. Tezin takip eden bölümü ise DDİ tekniklerine yer verilmektedir. Ek olarak tez kapsamında yapılan çalışmanın temelini oluşturan BERT modelinden bahsedilmektedir.

3.1. Doğal Dil İşleme Gelişimi

İnsanlar birbirleriyle iletişim kurmak için dil aracını kullanırlar. Bu dil kelimelerden ya da işaretlerden oluşmaktadır. DDİ çalışmaları ise insanlar tarafından kullanılması tercih edilen dil aracının makine tarafından öğrenilmesini, anlaşılmasını hedeflemektedir. DDİ, öğrenme süreci doğal alan bir dili üretme, çıkarım yapma, anlama, değerlendirme gibi farklı aşamalardan oluşmaktadır.

DDİ, 1950 yıllarında yapay zeka teknolojisinin bir alt alanı olarak konumlandırılmıştır. Gelişen teknoloji ile birlikte temel bir disiplin alanı haline gelmiştir.

Aslında, DDİ tarihinin 1950'ler öncesine dayandığı rivayet edilmektedir. Ancak başlangıç olarak Alan Turing tarafından 1950 yılında yayınlanmış olan, bir makinenin insana eşdeğer davranışlar sergileme yetenek testi içeriğini oluşturan "Bilgisayar Makineler ve Zeka" başlıklı makalesine dayandırılmaktadır (Turing, 2012).

Bu gelişmeyi, 1954 yılında gerçekleştirilmiş olan Georgetown deneyi takip etmektedir. Bu deney ise 60 kelimeden fazla kelimenin bir dilden başka bir dile tercüme edilmesinden oluşmaktadır (Hutchins, 1997).

1960'lı yıllarda bazı dikkate değer çalışmalar olduğu belirtilmiştir. Bunlar; kısıtlı kelime dağarcığına sahip bir doğal dil sistemi olan SHRDLU, diğeri ise psikoterapi simülasyonu olan ELIZA olarak kayıtlarda yerlerini almışlardır (Kuipers & Prasad, 2022).

1966 yılında hazırlanan ALPAC raporu ile 1980 yılının sonunda ortaya konulan İstatistiksel Makine Çeviri sistemlerine kadar ilerlemenin oldukça yavaş olduğu kaydedilmiştir (Cieri vd., 2022).

1970'li yıllarda ise ontoloji üzerine çalışmalar gerçekleştirilmiştir. Bu çalışmalara MARGIE, SAM, PAM, TaleSpin, QALM, Politics, Plot Units ve PARRY, Racter, Jabberwacky gibi sohbet robotları örnek verilmektedir.

1980'li yıllara gelindiğinde, elle yazılmış karmaşık kurallar kümesi ile sürdürülen çalışmalar yerine MÖ algoritmalarının, dil işlemeye dahil edilmesine karar verilmiştir. Bu kararın verilmesinde ise, tümleşik devre ve fiziki boyutu arasındaki ilişkiyi temel alan Moore yasası ve Derlem (corpus) yaklaşımı etkili olmuştur. Karar ağaçları, Markov modeli gibi algoritmalarla yapılan sonuçlar, başarı elde edileceğini göstermiştir. IBM Research'teki Otomatik Çevirim (Machine Translation)'in başarılı olması dikkate değer çalışmalar arasında yerini almıştır.

Takip eden yıllarda ise araştırmalarda, giderek denetimsiz ve yarı denetimli öğrenme algoritmaları kullanılmaya başlanmıştır.

2010'lu yıllarda ise temsili öğrenme, kelime temsil yöntemleri ve derin sinir ağlarının DDİ işlemlerinde yerlerini almaya başlamışlardır. Bu tekniklerin popülerleşmesi ve dil modelleme, dil ayırma gibi dil işleme çalışmalarında başarılı sonuçlar vermeleriyle yaygınlaşmaları hızlanmıştır.

Metin madenciliği, doğal dil metinlerinden anlamlı bilgiler toplamaya çalışan bir alandır. Belirli amaçlar için faydalı olabilecek bilgilerin çıkarımı amacıyla metin analizi olarak tanımlanabilmektedir. Modern kültürde metin, resmi bilgi alışverişi için en yaygın araç yerine geçmektedir. Metin madenciliği alanı genellikle gerçek fikir, bilgi veya fikirlerin bağlantılı olduğu metinlerle ilgilenmektedir ve başarı kısmi olsa bile bu tür metinlerden otomatik bilgi çıkarmaya çalışma motivasyonu oldukça zorlamaktadır.

Geleneksel kütüphaneler, kullanıcıların, belgenin yazarını, başlığı, konu başlığını, anahtar kelimelerini vb. tanımalarına olanak sağlayan üst verilerle (metadata) temsil edilen fihristler sunmaktadır.

Üst verilerin otomatik olarak çıkarılması (örneğin konular, dil, yazar, anahtar kelimeler) metin madenciliği tekniklerinin başlıca uygulamaları arasında yer almaktadır. Modern otomatik belge erişim teknikleri, üst veri oluşturma aşamasını

atlar ve doğrudan belgelerin tam metni üzerinde çalışmaktadır. Temel fikir, belge koleksiyonundaki her bir kelimeyi indekslemektir. Belgeler, kelime torbası olarak adlandırılan, belgelerin içerdikleri kelime kümesi ve her bir kelimenin belgede ne sıklıkta göründüğünü içermesiyle temsil edilmektedir.

Bu gösterimin sözcük sırası tarafından verilen sıralı bilgileri atmasına rağmen, oldukça etkili ve popüler birçok çıkarma alma tekniğinin temelini oluşturmaktadır. Bir kelimenin nasıl tanımlanacağı, sayılarla ne yapılacağı gibi basit problemler her zaman basit geçici buluşsal yöntemlerle çözülmektedir. Birçok pratik sistem, uygun sıkıştırma teknikleri buna olan ihtiyacı ortadan kaldırsa da öncelikle verimlilik nedeniyle ortak kelimeleri veya etkisiz kelimeleri atmaktadır.

Bilgi alımı, çıkarılan belgelerin kullanıcı tarafından aranan belirli bilgileri yoğunlaştırmak veya çıkarmak için işlendiği belge alımının bir uzantısı olarak kabul edilebilmektedir. Bu nedenle, belge alımını, kullanıcı tarafından sorulan sorguya odaklanan bir metin özetleme aşamasını veya bazı teknikleri kullanan bir bilgi çıkarma aşamasını izleyebilmektedir. E-posta adresleri ve URL'ler gibi bazı yapay varlıklar, makine işleme görevine özel olarak tasarlandıkları için tanınması kolaydır. Uygun model için genellikle düzenli bir ifadeyle kodlanmış basit bir dilbilgisi tarafından açık bir şekilde tespit edilebilmektedirler. Tarihler, sayılar ve para miktarları, basit sözlüksel gramerlerle yakalanabilecek iyi örneklerdir. Ancak pratikte işler çoğu zaman göründüğü kadar kolay olmamaktadır. Farklı kalıpların çoğalması ve yeni kalıpların ortaya çıkma potansiyeli bulunmaktadır. İşlemedeki ilk adım, genellikle girdiyi sözcük belirteçlerine veya kelimelere bölmektir.

Web arama motorları, şüphesiz en yaygın olarak kullanılan belge erişim sistemleridir. Ancak, arama sorguları genellikle yalnızca birkaç kelime veya kelime öbeği ile sınırlıdır. Buna karşılık, uzmanlar tarafından gelişmiş belge alma sistemlerine yapılan sorgular genellikle çok daha karmaşık ve spesifik olmaktadır.

Birçok pratik görev, dünyadaki nesneleri veya varlıkları temsil eden dilsel yapıları tanımlamayı içermektedir. Genellikle birden fazla sözcükten oluşan bu terimler tek sözcük öğeleri olarak işlev görmekte ve bu şekilde tanımlanmaları halinde birçok belge işleme görevi için önemli ölçüde ilerleme geliştirilebilmektedir. Belgeler arasında arama, bağlantı kurma ve çapraz başvuru yapma, tarama dizinlerinin oluşturulmasına yardımcı olabilmekte ve belirli işlemler için belge içeriği için bir vekil

görevi gören makinede işlenebilir meta veriler içerebilmektedirler. Bu öğelerin bazıları, kişi adları ve kuruluşların listeleri, gazetelerden alınan konumlar hakkında bilgiler, kısaltmalar ve kısaltma sözlükleri vb. kullanılarak sözlük tabanlı bir yaklaşımla tespit edilebilmektedir.

Bilgi çıkarma, metin madenciliğinin başlıca alt alanlarından biri olan doğal dil girdi şablonları doldurma görevine atıfta bulunmak için kullanılmaktadır. Bilgi çıkarmanın sonucu, ilgili varlığı belirlemek ve her biri için gösterilene benzer bir şablon doldurmaktır.

Metin özetleme ve belge alımından farklı olarak, bu anlamda bilgi çıkarma, insanlar tarafından yaygın olarak üstlenilen bir görev değildir, çünkü çıkarılan bilgiler, ayrı ayrı alınan her bir varlıktan gelmektedir.

Bir bilgi çıkarma sisteminin ilk işi, varlık çıkarma olarak tanımlanmaktadır. Bu işlemi takip eden işlem ise metnin söz dizimsel çözümlemesini içeren, çıkarılan varlıklar arasındaki ilişkiyi belirlemektir.

Bilgi çıkarmayı bir adım daha ileri götürerek, çıkarılan bilgiler sonraki bir adımda bilginin nasıl çıkarılacağına ilişkin kurallardan ziyade metnin içeriğini karakterize eden kuralları öğrenmek için kullanılabilmektedir.

Aşağıda tez kapsamında kullanılan BERT modeli ve DDİ'de sıklıkla kullanılan tekniklere sırasıyla değinilmektedir.

3.2. Tek Sıcak Kodlama

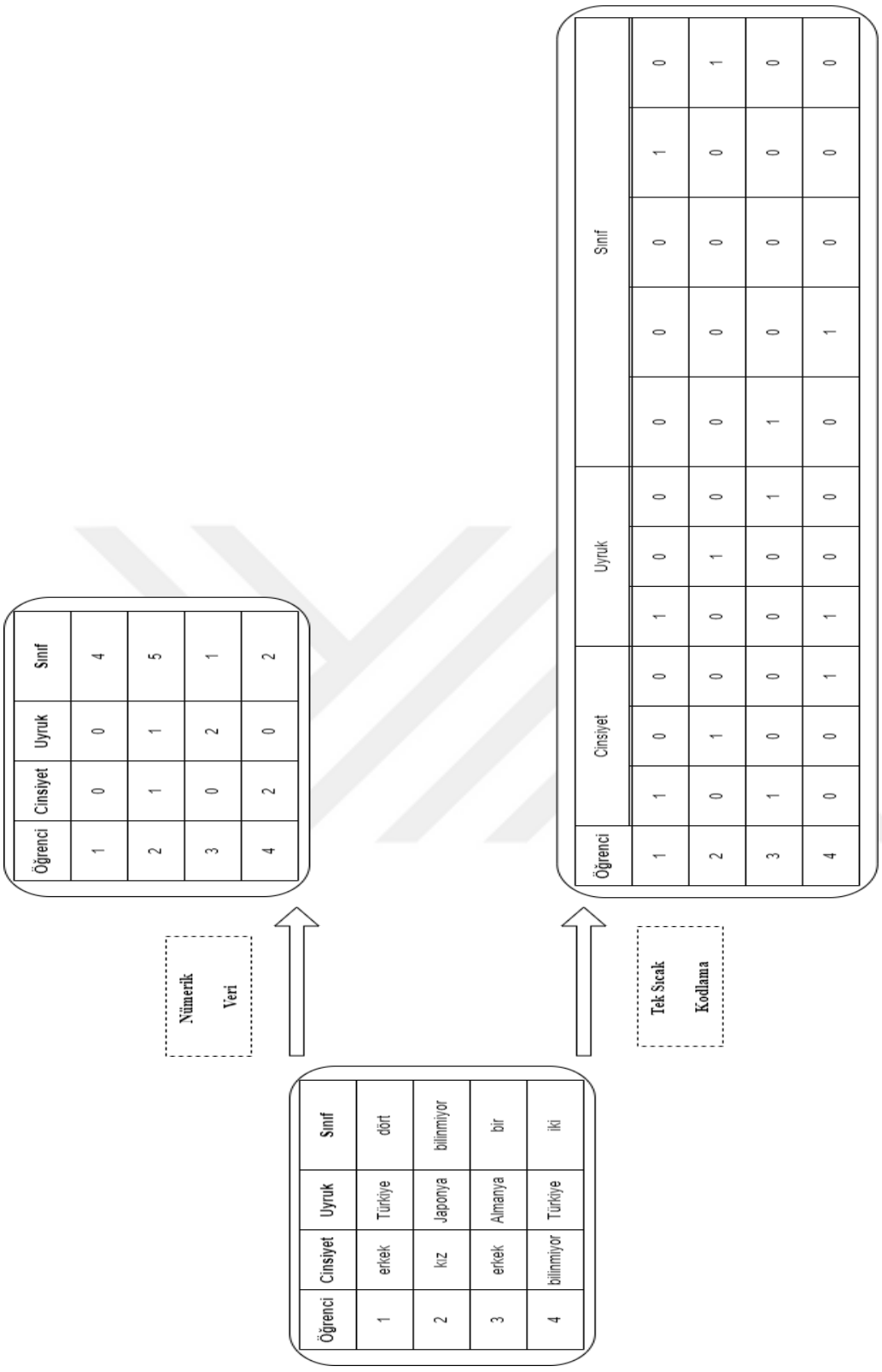
Tek sıcak kodlama (One Hot Encoding), bir tür kodlama yöntemidir. İlk olarak 1954 yılında Huffman tarafından önerilmiştir. Bir algoritma tasarlanırken, tek sıcak kodlama basitliği nedeniyle en popüler hedef kodlama teknikleri arasında yerini almıştır.

Tek sıcak kodlamanın avantajı, kategorik girdileri ikili hale getirebilmesi ve böylece sınıflandırma için birçok algoritmada özellikler arasındaki mesafeleri veya benzerlikleri hesaplamak için yaygın olarak kullanılan Öklid uzayından bir vektör olarak kabul edilebilmeleridir (Cassel & Kastensmidt, 2006).

Tek sıcak kodlama ile aynı kategorik özniteliklerin tüm değerlerinin, birbirine eşit uzaklıkta olduğu örtük olarak belirtilmektedir.

Tek sıcak kodlamada, orijinal özellik vektörü çok boyutlu bir matrise dönüştürülmektedir, matrisin boyutu bu özellikteki durumların sayısıdır ve her boyut belirli bir durumu temsil etmektedir. Bu işlem, belirli bir durum için, özellik matrisinin yalnızca bir boyutunun "1" olarak diğer tüm durum boyutlarının "0" olarak işaretlenmesiyle yapılmaktadır. Tek sıcak kodlamanın nasıl çalıştığına dair bir örnek şekil ile gösterilmektedir.

Şekil 11'deki örnekte, dört öğrenci üç özellikle tanımlanır: cinsiyet, uyruk ve sınıf. "Cinsiyet" özelliği için üç durum mevcuttur: erkek, kadın ve bilinmiyor. "Ülke" için örnekte üç ülke bulunmaktadır: Türkiye, Japonya ve Almanya. "Sınıf" için, toplamda dört dereceye sahip olduğu varsayılmaktadır ve dört durum olarak temsil edilebilir: bir, iki, dört ve bilinmeyen. Ancak uygulamada bazı değerlerin mevcut değerlerin dışında kalma olasılığı vardır. Örneğin "Ülke" ele alındığında, mevcut üç değerin yanı sıra, yeni örnekler dahil edildiğinde daha fazla seçenek mevcuttur. Sürekli özelliklerle karşılaşıldığında bu sorun daha ciddi olmaktadır. Bir çözüm olarak, eğitim setinin ve test setinin tüm özellikleri, etiketlerden bağımsız olarak ilk önce tek sıcak kodlama ile bir araya getirilip ve işlenmektedir. Tek sıcak kodlamanın bir avantajı, sınıflandırılmış tipte bir özelliği sayısallaştırdığımızda, sayısallaştırılmış değerdeki farkın modelin eğitim etkisi üzerindeki etkisini ortadan kaldırabilmesidir. Kesin olmak gerekirse, "1000" olarak kodlanan aynı özellik değeri, model eğitimi sürecinde "1" olarak kodlanandan daha büyük bir ağırlığa sahip olabilir. Kodlama sürecinde bu tür yan etkileri ortadan kaldırmak için tek sıcak kodlama devreye alınmaktadır. Ayrıca, veri kümesinde eksik veri sorunları olduğunda, tek sıcak kodlama, eksik değerleri yeni bir boyut olarak listeleyerek, eksik veri kümesinin tamamlanmasını sağlayarak bunu halledebilmesidir (Lean Yu vd., 2020).



Şekil 11. Tek sıcak kodlamanın işlenmesi.

Tek sıcak kodlama tekniğinin, diğer herhangi bir atama yöntemine göre büyük bir avantajı, eksik bir değeri benzer veya hesaplanmış bir değer olarak simüle etmeye

çalışmamasıdır. Bunun yerine eksik değerleri, veri yapısına müdahale eden simülasyonlardan kaçınmak amacıyla başka bir bakış açısı bulan yeni bir sınıf olarak ele alması şeklinde tanımlanmaktadır.

3.3. Kelime Torbası

Kelime torbası (Bag of Words), metin modellemesinde kullanılan bir DDİ tekniği olarak tanımlanmaktadır. Büyük miktardaki veriyi anlamak ve verilerden içgörüler elde etmek için verilerin kullanılabilir hale getirilmesi gerekmektedir. DDİ bu işlemi yapmamıza yardımcı olmaktadır.

Teknik anlamda metin verileri ile özellik çıkarma yöntemi olarak tanımlanabilmektedir. Bu yaklaşım, belgelerden özellik çıkarmanın basit ve esnek bir yoludur.

Bir kelime torbası, bir belgedeki kelimelerin oluşumunu açıklayan metnin bir temsilidir. Sadece kelime sayıları takip edilmektedir, gramer ayrıntıları ve kelime sırası dikkate alınmamaktadır. Belgedeki kelimelerin sırası veya yapısı ile ilgili herhangi bir bilgiye bakılmadan her kelime atıldığı için kelime torbası olarak adlandırılmaktadır. Model, belgenin neresinde olduğu ile değil, sadece bilinen kelimelerin belgede bulunup bulunmadığı ile ilgilenmektedir.

Metinle ilgili en büyük sorunlardan biri, dağınık ve yapılandırılmamış olmasıdır. MÖ algoritmaları yapılandırılmış ve iyi tanımlanmış sabit uzunluklu girdileri tercih etmektedir. Kelime torbası tekniği kullanılarak değişken uzunluktaki metinler, sabit uzunluktaki vektörlere dönüştürülebilmektedir.

Çok ayrıntılı bir düzeyde, MÖ algoritmaları, metinsel veriler yerine sayısal verilerle çalışmaktadır. Kelime torbası tekniği kullanılarak, bir metin eşdeğer sayı vektörüne dönüştürülebilmektedir.

Metin sınıflandırmasını, kelime torbası tekniğine dayandıran yaklaşımlar bulunmaktadır. Bazı kelime torbası tabanlı modeller aşağıdaki gibi listelenmektedir. Bunların arasında Derin Ortalamalı Ağlar (Deep Averaging Network - DAN), BoW'un ortalamasını almaya dayanan n katmanlı derin bir ÇKA modeli, Basit Kelime Gömme Modelleri (Simple Word Embedding Models - SWEM) önceden eğitilmiş sözcük yerleştirmeleri için farklı havuzlama stratejilerini araştırır ve önceden eğitilmiş sözcük yerleştirmelerinin üzerinde doğrusal bir katman kullanan fastText. Bu modeller,

sözcük konumu ve sırasını göz ardı ederek giriş dizisindeki tüm belirteçlerin oluşumunu sayar ve ardından sözcük yerleştirmelerine ve tam olarak bağlı ileri beslemeli katman(lar)a güvenmektedir (Galke, 2022).

3.4. Terim Frekansı – Ters Belge Frekansı

TF-IDF (Term Frequency - Inverse Document Frequency), Terim Frekansı ve Ters Belge Frekansı olmak üzere iki farklı kelimenin birleşiminden oluşmaktadır. Sırasıyla Terim Frekansı ve Ters Belge Frekansı açıklanacaktır.

Terim frekansı (TF)**,** bir belgede bir terimin kaç kez bulunduğunu ölçmek için kullanılmaktadır. Örneği 5000 kelimelik bir "Anadolu" belgemiz var ve belgede "Türk" kelimesi tam olarak 10 kez geçmektedir. Belgelerin toplam uzunluğunun çok küçükten büyüğe değişebileceği çok iyi bilinmektedir, herhangi bir terimin büyük belgelerde küçük belgelere kıyasla daha sık ortaya çıkma olasılığı bulunmaktadır. Bu nedenle, bu sorunu gidermek için, bir belgedeki herhangi bir terimin varlığı, terim sıklığını bulmak için o belgede bulunan toplam terimlere bölünmektedir. Bu durumda, "Anadolu" belgesindeki "Türk" kelimesinin terim sıklığı TF = 10/5000 = 0.002 olarak bulunmaktadır.

$$TF_{ij} = \frac{f_{ij}}{n_j} \tag{3.1}$$

Burada F_{ij}, j belgesindeki i teriminin sıklığıdır. n_j ise, j belgesindeki toplam sözcük sayısıdır.

Ters Belge Frekansı (IDF), Bir belgenin terim sıklığı hesaplandığında, algoritmanın tüm anahtar kelimelere eşit davranılıp davranılmadığına, yanlış olan "ve" gibi etkisiz kelimeler (stop words) olup olmadığına önem verilmemektedir. Tüm anahtar kelimelerin önemi farklı olarak değerlendirilmektedir. "ve" etkisiz kelimesi bir belgede 2000 kez geçmektedir ama hiçbir faydası bulunmamaktadır veya çok daha faydası bulunmaktadır. IDF tam olarak bunun için devreye girmektedir. Ters belge sıklığı, sık kullanılan sözcüklere daha az ağırlık vermektedir ve seyrek olan sözcüklere daha fazla ağırlık vermektedir.

$$IDF = 1 + \log(\frac{N}{c_i}) \tag{3.2}$$

Burada N, derlemdeki toplam belge sayısıdır. C, i kelimesini içeren belge sayısıdır.

Terim Frekansı - Ters Belge Sıklığı, kısımlarında, bir kelimenin belgelerde daha fazla veya daha yüksek oranda geçmesinin, daha yüksek terim sıklığı vereceği ve belgelerde daha az kelimenin geçmesinin, belirli bir belgede aranan anahtar kelime için daha yüksek önem (IDF) sağlayacağı anlaşılmaktadır. TF-IDF hesaplaması ise sadece terim frekansı (TF) ve ters belge frekansının (IDF) çarpımından elde edilmektedir. TF ve IDF'yi sırasıyla hesaplamıştır. TF-IDF'yi hesaplamak için aşağıdaki şekilde gerçekleştirilmektedir (Qaiser & Ali, 2018). (TF-IDF = 0,002*0,3010 = 0,000602)

$$w_{ij} = TF_{ij} + IDF_i) \tag{3.3}$$

Burada w_{ij}, j belgesindeki TF-IDF puan hesaplamasını göstermektedir.

Esasen, TF-IDF, belirli bir belgedeki kelimelerin göreceli sıklığını, o kelimenin tüm belge bütünü üzerindeki ters oranına kıyasla belirlemektedir. Sezgisel olarak, bu hesaplama belirli bir kelimenin belirli bir belgede ne kadar alakalı olduğunu belirlemektedir. Tek veya küçük bir belge grubunda ortak olan kelimeler, makaleler ve edatlar gibi yaygın kelimelerden daha yüksek TF-IDF numaralarına sahip olma eğilimindedir (Aizawa, 2003).

3.5. N-Gram

N-gram, uzun bir dizenin N karakterli bir dilimini temsil etmektedir. Literatürde bu terim, bir dizgede (örneğin, bir kelimenin birinci ve üçüncü karakterinden oluşan bir N-gram) herhangi bir birlikte meydana gelen karakter kümesi kavramını içerebilse de, burada sadece bitişik dilimler için anlatılmaktadır. Tipik olarak, dize bir dizi örtüşen N-gram'a bölünmektedir. Aynı anda birkaç farklı uzunlukta N-gram kullanılmaktadır. Ayrıca, kelime başı ve kelime sonu durumlarını eşleştirmeye yardımcı olmak için dizenin başına ve sonuna boşluklar eklenmektedir. (Boşlukları temsil etmek için alt çizgi (“_”) karakterini kullanılmaktadır.

Böylece, “METİN” kelimesi aşağıdaki N-gramlardan oluşmaktadır:

bi-gram: _M, ME, ET, Tİ, İN

üç-gram: _ME, MET, ETİ, TİN, N_ _

dört-gram: _MET, METİ, ETİN, TİN _, İN _ _

Genel olarak, boşluklarla doldurulmuş k uzunluğundaki bir dizi, k+1 bi-gram, k+1 tri-gram, k+1 quad-gram vb. içermektedir. N-gram tabanlı eşleştirme, posta adreslerini yorumlama, metin alma, ve çok çeşitli diğer doğal dil işleme uygulamalarında başarılar elde etmiştir. N-gram temelli eşleştirmenin sağladığı temel fayda, doğasından kaynaklanmaktadır: her dize küçük parçalara ayrıldığından, mevcut olan hatalar, yalnızca sınırlı sayıda parçayı etkilemekte ve geri kalanını bozulmadan bırakmaktadır. İki dizide ortak olan N-gramları sayılırsa, çok çeşitli metinsel hatalara dirençli olan benzerliklerinin bir ölçüsü elde edilmektedir (Cavnar vd., 1994).

3.6. Özellik Vektörü

Harris dağılım hipotezinden esinlenerek (Harris, 1968), benzerlik ölçümleri, iki kelimeyi karakterize eden bir çift ağırlıklı özellik vektörünü karşılaştırmaktadır. Özellikler tipik olarak, aynı bağlamda karakterize edilen kelimeyle birlikte ortaya çıkan diğer kelimelere karşılık gelmektedir. Daha sonra, benzer bağlamlarda bulunan farklı kelimelerin anlamsal olarak benzer olduğu varsayılmaktadır (Geffet & Dagan, 2004).

Tablo 1. Benzerlik listesinde örnek en benzer kelimeler

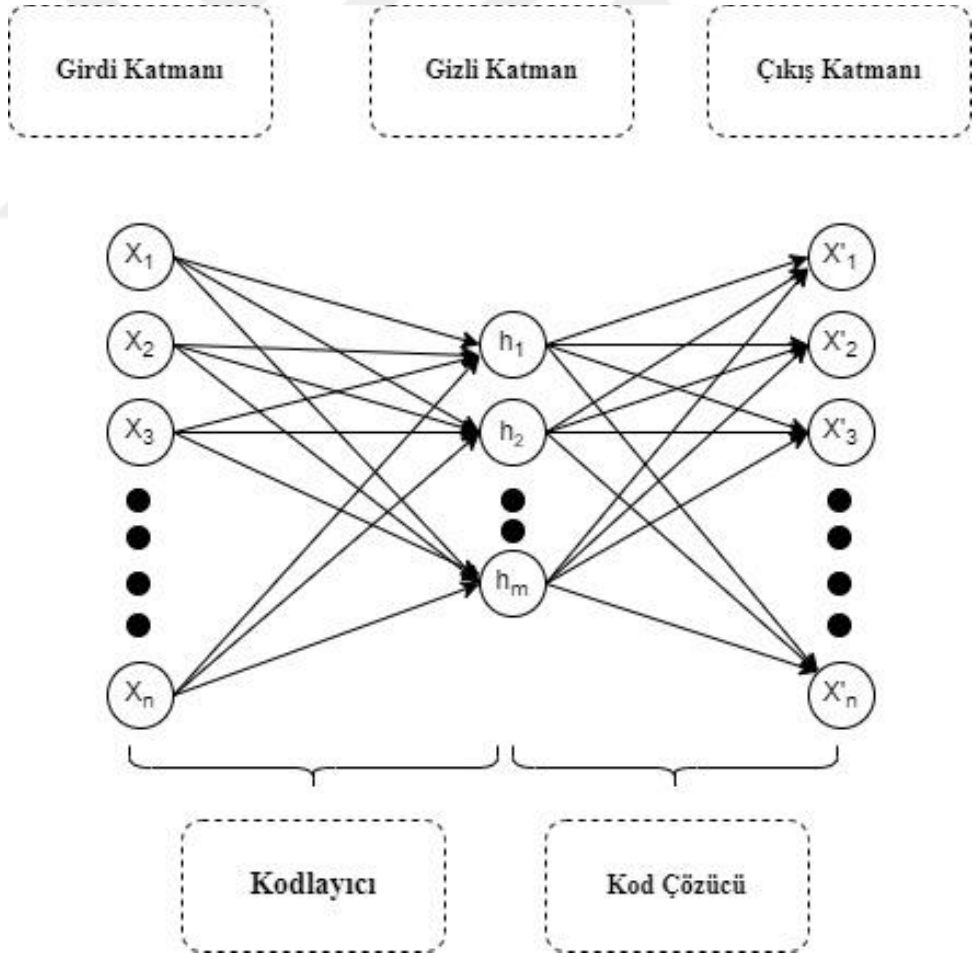
Ulus	1	*şehir	8
Bölge	2	Bölge	9
Durum	3	Alan	10
*dünya	4	*şehir	11
Ada	5	Cumhuriyet	12
Bölge	6	*şirket	13
*ekonomi	7	*endüstri	14

Tablo 1'e göre, dağılımsal benzerlik, anlamsal benzerliğe göre biraz dağınık bir kavram yakalamaktadır. Yapısal olarak, eğer iki kelime dağılım olarak benzerse, o

zaman bazı bağlamlarda bir kelimenin ortaya çıkması, diğer kelimenin de bu tür bağlamlarda ortaya çıkma olasılığının olduğunu göstermektedir. Ancak, verilen bağlamda diğeriyle değiştirilirken ilk kelimenin anlamının korunmasını sağlamamaktadır. Örneğin şirket-hükümet gibi benzer semantik türdeki kelimeler, anlam koruyucu anlamda ikame edilemeseler bile, dağılımsal olarak benzer olarak ortaya çıkma eğilimindedir.

3.7. Otomatik Kodlayıcılar

Otomatik Kodlayıcılar (AutoEncoders), 80'lerin sonlarında ortaya çıkan bir tür denetimsiz sinir ağıdır. Otomatik Kodlayıcılar, doğrusal olmayan öznitelikleri otomatik olarak çıkarmak için güçlü bir araç olarak kabul edilmiştir. Şekil 12'de gösterildiği gibi, temel bir Otomatik Kodlayıcı, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşmaktadır.



Şekil 12. Otomatik kodlayıcı mimarisi

Her katmandaki nöron sayısı n, m, n'dir. Giriş katmanı ve gizli katman bir kodlayıcıyı oluşturmaktadır. Gizli katman ve çıktı katmanının birleşmesiyle bir kod çözücü oluşturulmaktadır. Kodlayıcı, $x = \{x_1, x_2,..., x_n\}$ yüksek boyutlu girdi verilerini, bir f fonksiyonu ile $h = \{h_1, h_2,..., h_m\}$ düşük boyutlu gizli bir temsile kodlamaktadır:

$$h = f(x) = s_f(W_x + b) \tag{3.4}$$

burada s_f bir aktivasyon fonksiyonudur. Kodlayıcı, bir (m×n) ağırlık matrisi W ve bir bias vektörü $b \in R_m$ ile parametrelendirilmektedir.

Şekildeki kod çözücü, gizli temsili h'yi bir fonksiyonla $x = \{x_1, x_2,..., x_n\}$ yeniden yapılandırmasını eşlemektedir.

$$x' = g(h) = s_g(W'h + b') \tag{3.5}$$

burada s_g, kod çözücünün aktivasyon fonksiyonunu temsil etmektedir. Kod çözücünün parametreleri, bir sapma vektörü $b \in R_n$ ve bir n×m ağırlık matrisi W'den oluşmaktadır.

s_f ve s_g fonksiyonu, genellikle doğrusal olmayan aktivasyon fonksiyonudur, örneğin hiperbolik tanjant ve sigmoid fonksiyonu (Zhang vd., 2017). Doğrusal olmayan aktivasyon fonksiyonu, OK'nin temel bileşen analizinden (PCA) daha yararlı özellikler öğrenmesine yardımcı olmaktadır (Japkowicz vd., 2000), (Hinton & Salakhutdinov, 2006).

Otomatik kodlayıcı, x ve x arasındaki yeniden yapılandırma hatasını en aza indirecek şekilde eğitilmiştir. Yeniden yapılandırma hatasını formüle etmenin kare hatası ve çapraz entropi olmak üzere iki yolu vardır. Formülleri aşağıda gösterilmiştir:

• Kare hatası:

$$E_{AE}(x,x') = ||x-x'||^2 \tag{3.6}$$

• Çapraz entropi:

$$E_{AE}(x,x') = -\sum_{i=1}^{n} (x_i \log x'_i + (1 - x_i) \log(1 - x'_i)) \tag{3.7}$$

Otomatik Kodlayıcının kayıp fonksiyonunu oluşturmak için yeniden yapılandırma hatasının hesaplanmasına düzenli bir terim eklenebilmektedir:

$$L_{AE}(\mathrm{x},\mathrm{x}') = \left(\sum_{x \epsilon R^n} E_{AE}(x, x')\right) + \lambda * Regularization \tag{3.8}$$

Kayıp fonksiyonu, stokastik gradyan inişi (Stochastic Gradient Descent - SGD) veya alternatif en küçük kareler (Alternative Method to Least Squares - ALS) ile optimize edilebilmektedir.

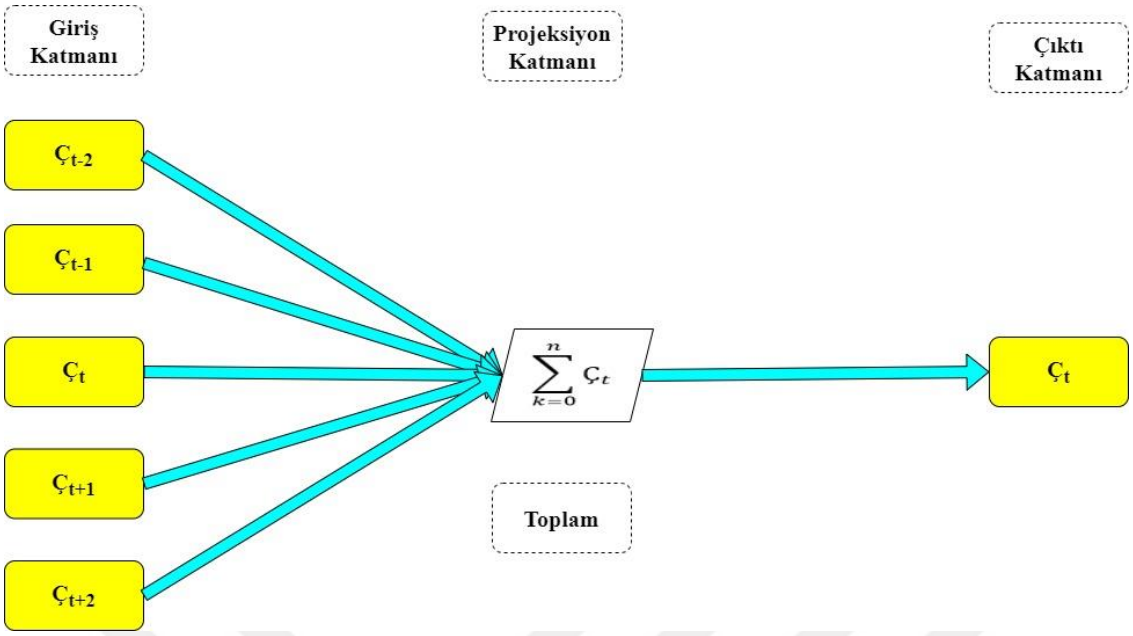
3.8. Word2Vec

Mikolov ve arkadaşları, Tekrarlayan Sinir Ağı Dil Modelini (Recurrent Neural Network Language Models - RNNLM), Bengio ve arkadaşlarının önermiş olduğu kelime vektörlerinin ilk kez tek bir gizli katmana sahip sinir ağı kullanılarak öğrenildiği Sinir Ağ Dil Modeli (Neural Network Language Models - NNLM)mimarisinden esinlenerek önermişlerdir. SADM mimarisinin karmaşıklığını azaltmak ve daha fazla veri üzerinde verimli ve hızlı bir şekilde çalışılması hedeflenmiştir.

Mimari, içinde Sürekli Kelime Çantası (Continuous Bag of Words) CBOW ve Skip-gram mimarilerini barındırmaktadır (Mikolov vd., 2013a).

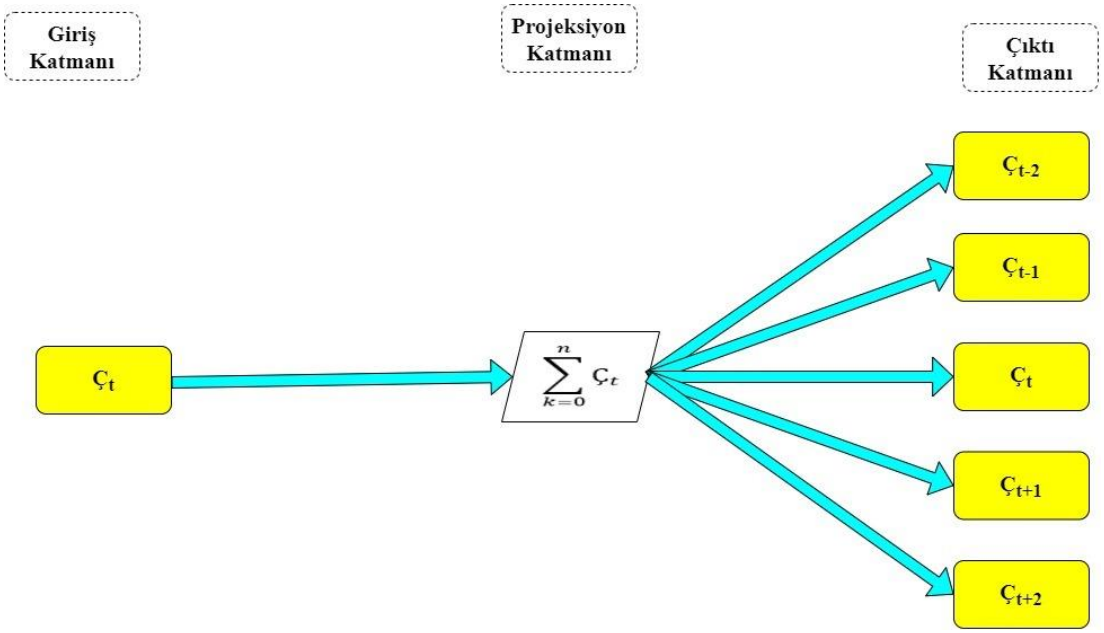
$$\sum_{k=0}^{n} \mathrm{Ç}_t \tag{3.9}$$

Önerilen ilk olarak, SADM mimarisine benzeyen CBOW mimarisidir ve Şekil 13'te verilmiştir. Bağlamın sürekli dağıtılmış temsilini kullanımaktadır. Doğrusal olmayan gizli katmanın kaldırıldığı ve projeksiyon katmanının tüm kelimeler için paylaşılmış ve tüm kelimeler aynı konuma yansıtılmıştır. Mevcut kelime bağlama göre tahmin edilmektedir. Giriş ve projeksiyon katmanı arasındaki ağırlık matrisinin, SADM mimarisindeki ile aynı şekilde tüm kelime konumları için paylaşılmıştır.



Şekil 13. Sürekli kelime çantası mimarisi.

İkinci olarak Skip-gram mimarisidir ve Şekil 14'te verilmiştir. Bu mimaride, mevcut kelimeyi bağlama göre tahmin etmek yerine, aynı cümledeki başka bir kelimeye göre bir kelimenin sınıflandırmasını en üst düzeye çıkarmayı hedeflenmiştir. Sürekli iz düşüm katmanına sahip bir log-lineer modele girdi olarak mevcut her kelime kullanılmıştır ve mevcut kelimeden önce ve sonra belirli bir aralıktaki kelimeler tahmin edilmiştir.



Şekil 14. Skip-Gram mimarisi

Aralığın artırması sonucunda elde edilen kelime vektörlerinin kalitesinin iyileştiği, aynı zamanda hesaplama karmaşıklığını da arttırmıştır. Daha uzak kelimeler genellikle yakın kelimelere göre mevcut kelimeyle daha az ilgili olduğu görülmüştür.

3.9. GloVE

Pennington ve arkadaşları, daha önce yapılan çalışmalardan farklı olarak kelime vektörü öğrenimi için uygun başlangıç noktasının, olasılıkların kendisinden ziyade birlikte meydana gelme olasılıklarının oranları ile olması gerektiğini öne sürmüştür (Pennington vd., 2014). Kelime Temsili için Global Vektörler (Global Vectors for Word Representation- GloVe) olarak adlandırılan çalışma kelime-kelime birlikte kullanılmasının hesaplama değerine dayanmaktadır ve anlamsallık ön plana çıkmaktadır. Bu hesaplamada bir kelimenin başka bir kelime ile bir arada görülmesini, X matrisi olarak tanımlanmıştır. X_{ij} girişi, j sözcüğünün i sözcüğü bağlamında kaç kez oluştuğunun tablolaştırılması olarak formüle edilmiştir. Sonuç olarak oluşum bilgisi matrisi, bir kelimeyi yani satırları, bu kelimeyi büyük bir bütünde yani bağlamda, sütunlarda ne sıklıkta görüldüğünün sayılması ile hesaplanmıştır. Herhangi bir kelimenin ise i sözcüğü bağlamında hesaplamasını ise $X_i = \sum k\ X_{ik}$ olarak tanımlamışlardır. J sözcüğünün, i sözcüğü bağlamında görülme olasılığı aşağıdaki şekilde hesaplanmıştır:

$$P_{ij} = \mathrm{P(j|i)} = \frac{X_{ij}}{X_i}, \tag{3.10}$$

Bu kelimelerin ilişkisi, çeşitli deneme k sözcükleri ile birlikte bulunma olasılıklarının oranına bakarak incelenebilir, P_{ik} / P_{jk} oranının i, j ve k olmak üzere üç kelimeye bağlı olduğuna dikkat edilmiştir, ve aşağıdaki gibi hesaplanmıştır:

$$\mathrm{F}\left(\left(w_i\text{-}\mathrm{w}_j, \widetilde{w}_k\right)\right) = \frac{P_{ik}}{P_{jk}}, \tag{3.11}$$

Vektör uzayları, doğası gereği doğrusal yapılar olduğundan, vektör farklılıkları göz önüne alındığında hesaplama aşağıdaki hali almıştır:

$$\mathrm{F}\left((w_i\text{-}\mathrm{w}_j)^T, \widetilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}, \tag{3.12}$$

Maliyet fonksiyonu hesaplamalar sonrasında aşağıdaki hali almıştır:

$$J = \sum_{i,k} (w_i^T . \tilde{w}_k + b_i + \hat{b}_k - \log(X_{ik}))^2 \quad (3.13)$$

3.10. FastText

Facebook araştırmacılarının, kelime temsillerini öğrenmek ve metin sınıflandırması yapmak için hızlı ve etkili bir yöntem olarak açık kaynak şeklinde hazırlanan bir kelime gömme projesidir.

FastText kelime gömme projesinin temel amacı, kelime temsillerini öğrenmek yerine kelimelerin içyapısını dikkate almaktır. Bu, morfolojik olarak zengin diller için oldukça faydalıdır, böylece farklı morfolojik kelimelerin temsillerini bağımsız olarak öğrenilebilmektedir.

FastText, giriş metninin üzerine bir pencere kaydırarak veya merkezdeki sözcüğü, aynı zamanda sürekli sözcük çantası olarak da bilinen kalan bağlamdan öğrenerek) ya da tüm bağlam sözcüklerini ortadaki sözcükten jump-gram olarak öğrenerek çalışmaktadır. Bu yaklaşım Word2Vec'e çok benzemektedir, çünkü her ikisi Mikolov tarafından tasarlanmıştır (Mikolov vd., 2013b), (Mikolov vd., 2019).

Bununla birlikte, Word2Vec'in aksine, FastText kelimelerin alt bölümleri için vektörleri de öğrenebilmektedir: karakter n-gramları. Bu, örneğin aşk, sevilen ve seven sözcüklerinin hepsinin, farklı bağlamlarda ortaya çıkma eğiliminde olsalar bile benzer vektör temsillerine sahip olmasını sağlamaktadır. Bu özellik, yoğun çekimli dillerde öğrenmeyi geliştirmektedir.

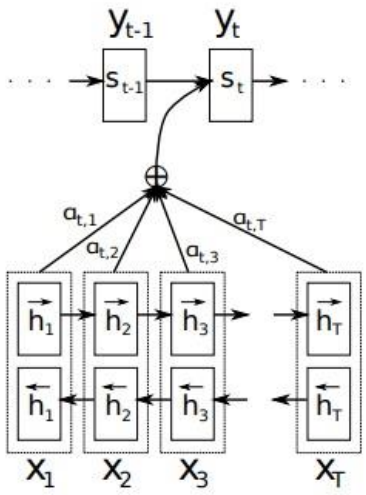
3.11. Dikkat Mekanizması

Dikkat mekanizması (Attention Mechanism), odağın yönlendirmesine dayanmaktadır ve verileri işlerken belirli kıstaslara daha fazla dikkat edilmektedir. Dikkat, bir ağın mimarisinin bir bileşenidir ve karşılıklı bağımlılığı yönetmektedir ve ölçmektedir. Ağın Dikkat bileşeninin çıkış cümlesindeki her kelime için yapacağı şey, giriş cümlesindeki önemli ve ilgili kelimeleri haritalamak ve bu kelimelere daha yüksek ağırlıklar atayarak çıkış tahmininin doğruluğunu arttırmaktır.

Dikkat mekanizması, birçok DÖ alanında uygulama alanı bulunmasına sebep olan durum, DDİ görevlerinde uygulamasından ve otomatik çevirideki uzun diziler sorununu yönelik çözüm olarak hazırlanmasıdır.

Dikkat mekanizması, kod çözme işlemi sırasında giriş dizisinin tüm gizli durumlarını korumaktadır ve kullandığı için doğrudan uzun giriş dizileri konusunu ele almaktadır. Bunu, kod çözücü çıktısının her bir zaman adımı arasında tüm kodlayıcı gizli durumlarına benzersiz bir eşleme oluşturarak yapmaktadır. Bu, kod çözücünün yaptığı her çıktı için, tüm girdi dizisine erişimi olduğu ve çıktıyı üretmek için bu diziden belirli öğeleri seçici olarak seçebileceği anlamına gelmektedir.

Dikkat mekanizmasına yönelik öne çıkan iki algoritmaya değinilmektedir. Bunlar:

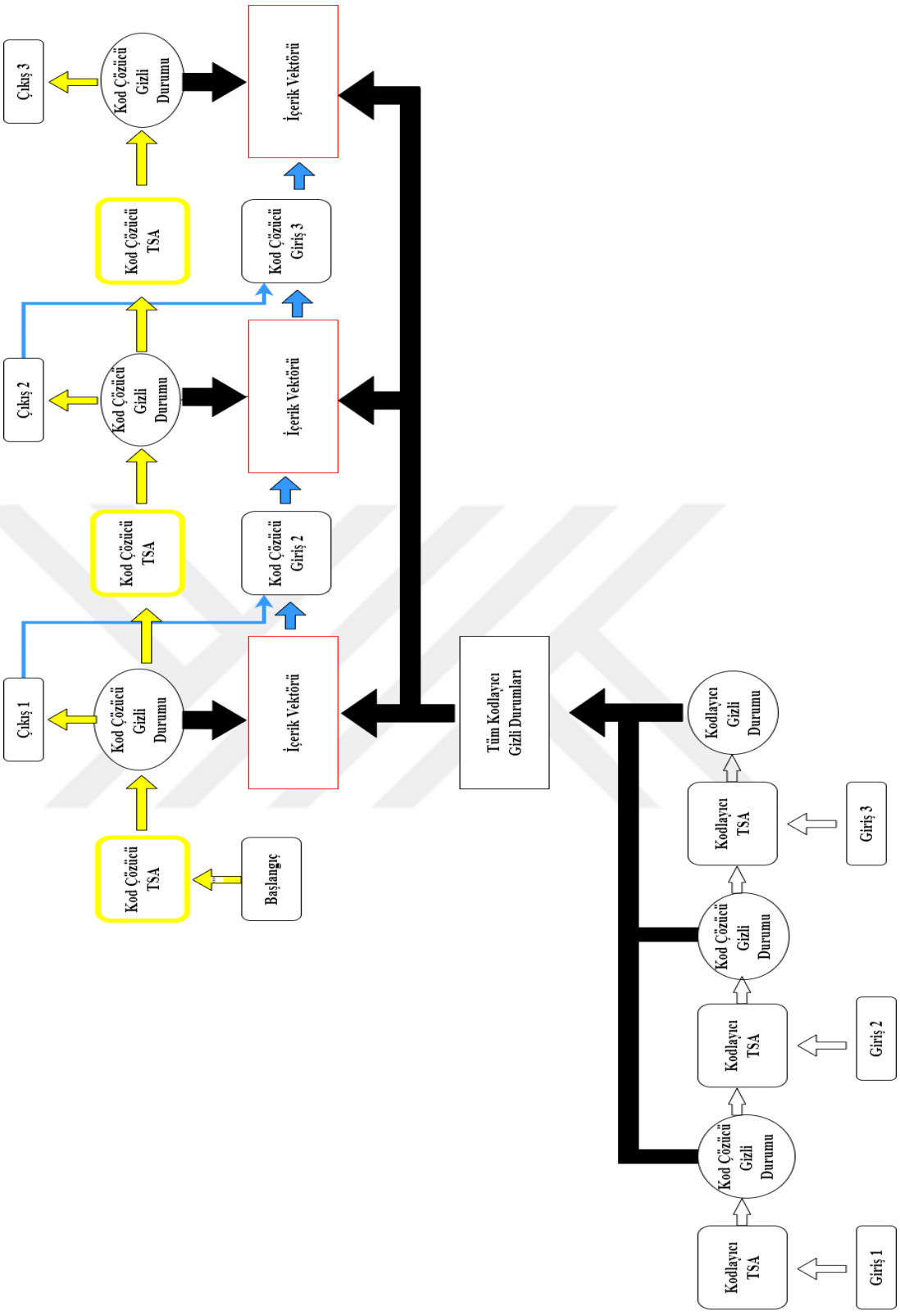


Şekil 15. Bahdanau dikkat mekanizması

Bahdanau Dikkat Algoritması (Bahdanau vd., 2015)

- Kodlayıcı Gizli Durumlarının Üretilmesi - Kodlayıcı, giriş sırasındaki her öğenin gizli durumlarını üretmektedir.
- Önceki kod çözücü gizli durumu ile kodlayıcının gizli durumlarının her biri arasındaki Hizalama Puanlarının Hesaplanması, Son kodlayıcı gizli durumu, kod çözücüdeki ilk gizli durum olarak kullanılabilmektedir.
- Hizalama Puanlarını Softmaxing - her kodlayıcı gizli durumu için hizalama puanları birleştirilmektedir ve tek bir vektörde temsil edilmektedir ve Softmax'e dönüştürülmektedir.
- Bağlam Vektörünü Hesaplama - kodlayıcı gizli durumları ve ilgili hizalama puanları, bağlam vektörünü oluşturmak için çarpılmaktadır.

- Çıktının Kodunun Çözülmesi - bağlam vektörü önceki kod çözücü çıkışıyla birleştirilmektedir ve yeni bir çıktı üretmek için önceki kod çözücü gizli durumuyla birlikte o zaman adımı için Kod Çözücü TSA'yı beslenmektedir.
- İşlem (adım 2-5), bir belirteç üretilene veya çıktı belirtilen maksimum uzunluğu geçene kadar kod çözücünün her zaman adımı için kendini tekrar etmektedir.



Şekil 16. Bahdanau dikkat mekanizması (Loye, 2019)

Luong Dikkat Algoritması

- Kodlayıcı Gizli Durumlarının Üretilmesi - Kodlayıcı, giriş sırasındaki her öğenin gizli durumlarını üretmektedir.
- Kod Çözücü TSA - önceki kod çözücü gizli durumu ve kod çözücü çıkışı, o zaman adımı için yeni bir gizli durum oluşturmak üzere Kod Çözücü TSA'dan geçirilmektedir.
- Hizalama Puanlarının Hesaplanması - yeni kod çözücü gizli durumu ve kodlayıcı gizli durumları kullanılarak hizalama puanları hesaplanmaktadır.
- Hizalama Puanlarını Softmaxing - her kodlayıcı gizli durumu için hizalama puanları birleştirilmekte, tek bir vektörde temsil edilmekte ve ardından Softmax'e dönüştürülmektedir.
- Bağlam Vektörünü Hesaplama - kodlayıcı gizli durumları ve ilgili hizalama puanları, bağlam vektörünü oluşturmak için çarpılmaktadır.
- Nihai Çıktıyı Üretmek - bağlam vektörü, yeni bir çıktı üretmek için tamamen bağlı bir katmandan geçirildiği gibi 2. adımda oluşturulan kod çözücü gizli durumu ile birleştirilmektedir.
- İşlem (adım 2 - 6), bir belirteç üretilene veya çıktı belirtilen maksimum uzunluğu geçene kadar kod çözücünün her zaman adımı için kendini tekrar etmektedir.

3.12. Dönüştürücü

Dönüştürücüler yani transformerlar, doğal dil metni, genom dizileri, ses sinyalleri veya zaman serisi verileri gibi sıralı verileri işlemek için birçok sinir ağı tasarımında kullanılmaktadır.

Kısaca bir dönüştürücü sinir ağının işlevi, bir vektör dizisi biçiminde bir giriş cümlesi alıp, kodlayıcı ile bir vektöre dönüştürmek ve ardından kod çözücü ile başka bir diziye çevirme olarak tanımlamaktadır.

Dönüştürücülerin, önemli bir parçası dikkat mekanizmasıdır. Dikkat mekanizması bir girdideki diğer belirteçlerin belirli bir belirtecin kodlanması için ne kadar önemli olduğunu temsil emektedir.

Örneğin bir dilden başka bir dile çeviri işleminde, dikkat mekanizması, dönüştürücünün, bir kelimeyi nasıl tercüme edeceğine karar vermek için mevcut

kelimenin hem solundaki hem de sağındaki belirli kelimelere odaklanmasını sağlamaktadır.

Dönüştürücü sinir ağı, giriş olarak bir cümle almaktadır. Kelime gömme vektör dizisi ve konumsal kodlama dizisi olmak üzere, iki diziye dönüştürmektedir.

Kelime gömme vektörü, metnin sayısal bir temsilidir. Bir sinir ağının bir metni işleyebilmesi için kelimeleri gömme temsiline dönüştürmesi gerekmektedir. Gömme gösteriminde, sözlükteki her kelime bir vektör olarak temsil edilmektedir. Konumsal kodlama, kelimenin orijinal cümledeki konumunun bir vektör olarak gösterilmesini temsil etmektedir.

Dönüştürücüler, kelime gömme vektörlerini ve konumsal kodlamaları bir araya getirerek sonucu bir dizi kodlayıcıdan geçirmektedir. Ardından bir dizi kod çözücüden geçirmektedir. TSA ve UKSB'lerin aksine, tüm girdi sırayla alınmayıp aynı anda ağı beslenmektedir.

Kodlayıcıların her biri, girdilerini kodlama adı verilen başka bir vektör dizisine dönüştürmektedir. Kod çözücüler, kodlamaları, farklı çıktı sözcüklerinin bir olasılık dizisine geri dönüştürerek kodlayıcının tam tersini yapmaktadır. Çıktı olasılıkları, Softmax fonksiyonu kullanılarak başka bir doğal dil cümlesine dönüştürülebilmektedir.

Her kodlayıcı ve kod çözücü, ilgili bilgileri içermeyen sözcükleri maskeleyen, belli diğer kelimelerden ilgili bilgileri içeren bir giriş kelimesinin işlenmesine izin veren dikkat mekanizması adı verilen bir bileşen içermektedir.

Bu işlem için birçok kez hesaplama gerektiğinden, GPU'ların sunduğu paralel hesaplamadan yararlanılarak paralel olarak birden fazla dikkat mekanizması uygulanmaktadır. Bu uygulama işlemi ise çok başlı dikkat mekanizması olarak tanımlanmaktadır. Bir sinir ağından aynı anda birden fazla kelime geçirme, dönüştürücülerin UKSB ve TSA'lara göre bir avantajı olarak gösterilmektedir.

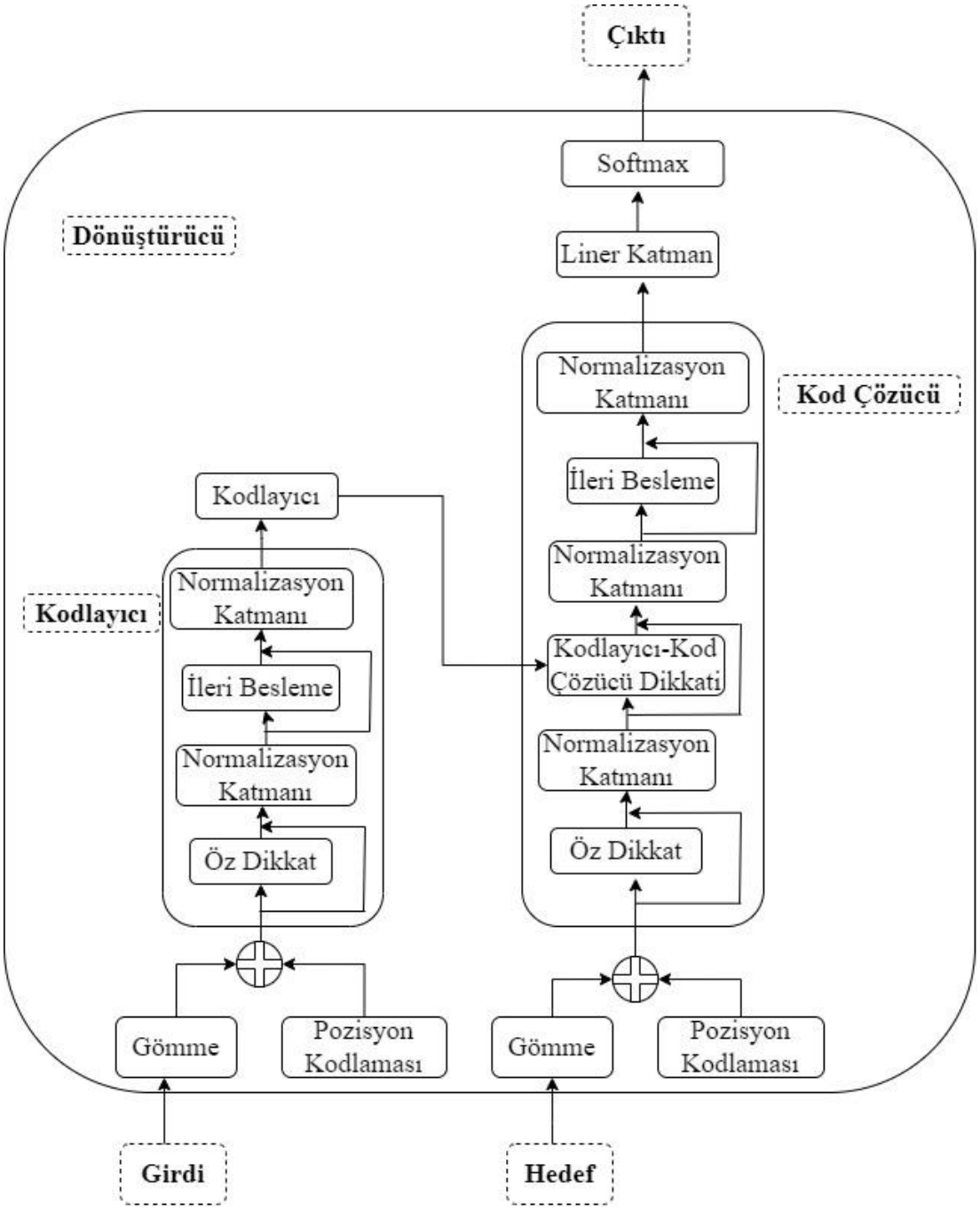
Pratikte dikkat, bir dönüştürücü sinir ağında üç farklı şekilde kullanılmaktadır:

(1) Kodlayıcı-kod çözücü dikkati: Hedef dizisi, giriş dizisine dikkat etmektedir.

(2) Kodlayıcı öz dikkati: Giriş dizisi kendine dikkat etmektedir.

(3) Kod çözücüde öz dikkati: Hedef dizisi kendine dikkat etmektedir.

Dikkat mekanizmaları, bir modelin cümlenin başka herhangi bir noktasındaki giriş sözcüklerinden ve gizli durumlarından bilgi çekilmesine izin vermektedir.



Şekil 17. Dönüştürücü mimarisi

Dikkat mekanizması fonksiyonu, bir sorgu ve bir dizi anahtar/değer çifti almaktadır. Sorguya en çok benzeyen anahtarlara karşılık gelen değerlerin ağırlıklı bir toplamını

vermektedir. Dikkat mekanizması fonksiyonu, dönüştürücü sinir ağının giriş vektörlerinin bir alt kümesine odaklanmasını sağlamaktadır.

Bir dönüştürücü sinir ağı Şekil 17'de ve en yaygın formül, ölçeklenmiş skaler çarpım dikkati olarak gösterilmektedir:

$$\text{Dikkat}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.14}$$

Q	d_k boyutunda bir sorgu vektörü
K	d_k boyutundaki anahtarların bir vektörü
V	d_k boyut değerlerinin bir vektörü
d_k	Dikkat anahtarlarının boyutu. Tasarım zamanında seçilen bir hiperparametredir.

Dikkat mekanizmasının dönüştürücüde kullanıldığı yere bağlı olarak Q, K ve V farklı kaynaklardan gelebilmektedir.

Dikkat hesaplaması, çok başlı dikkat mekanizmasında paralelleştirilmektedir, böylece cümledeki birden çok pozisyon için dikkati aynı anda hesaplanabilmektedir. Çoklu konumlar hesaplama, yukarıda yer alan formüldeki vektörler birleştirilerek yapılmaktadır.

3.12.1. Dönüştürücü Sinir Ağı ve TSA

TSA'lar, dönüştürücülerden temelde farklı bir tasarıma sahiptir. Bir TSA, girdi sözcüklerini birer birer işlemektedir ve zaman içinde gizli bir durum vektörünü korumaktadır. Her girdi kelimesi, sinir ağının birkaç katmanından geçirilmektedir ve durum vektörünü değiştirmektedir. Teoride, belirli bir zamanda durum vektörü, geçmişten gelen girdiler hakkında bilgi tutabilmektedir. Bununla birlikte, genellikle modelin gizli durumu, erken girdiler hakkında çok az kullanılabilir bilgiyi korumaktadır. Yeni girdiler, bir durumun üzerine kolayca yazabilmektedir ve bilgi kaybına neden olabilmektedir. Bu, bir TSA'nın performansının uzun cümlelerde

düşme eğiliminde olduğu anlamına gelmektedir. Buna uzun süreli bağımlılık sorunu denilmektedir.

Bu, tüm giriş dizisinin aynı anda işlendiği ve dikkat mekanizmasının her bir çıkış kelimesinin her bir giriş ve gizli durumdan çekilmesine izin veren dönüştürücü tasarımıyla çelişmektedir.

TSA'lar girişi sırayla işlediğinden, GPU'lar gibi yüksek performanslı bilgi işlemden yararlanmak zordur. Paralel işleme ve çok kafalı dikkat mekanizmalarına sahip dönüştürücü tasarımı, farklı girdi sözcükleri bir GPU üzerinde aynı anda işlenebildiğinden çok daha hızlı eğitim ve yürütmeye olanak tanımaktadır.

3.13. BERT Model

Google BERT (Çift Yönlü Kodlayıcı Gösterimleri-Bidirectional Encoder Representations from Transformers-BERT), Google yapay zeka grup araştırmacıları tarafından geliştirilen açık kaynaklı, önceden eğitilmiş bir DDİ modelidir.

BERT, genellikle önceden eğitilmiş ağırlıkları ile kullanılmaktadır, ancak BERT'ü bir transfer öğrenme yaklaşımında kullanmak için önceden eğitilmiş bu ağırlıkları eğitmek mümkündür. Transfer öğrenme, daha önce eğitilmiş bir ağın bilgisini, benzer sorunları çözmek için başka bir ağa aktararak kullanmaktır. Önceden eğitilmiş BERT modeline uygun bir çıktı katmanı eklenerek, dil işlemede klasik DDİ yöntemlerine göre çok daha performanslı sonuçlar elde edilebilmektedir (Cui vd., 2021).

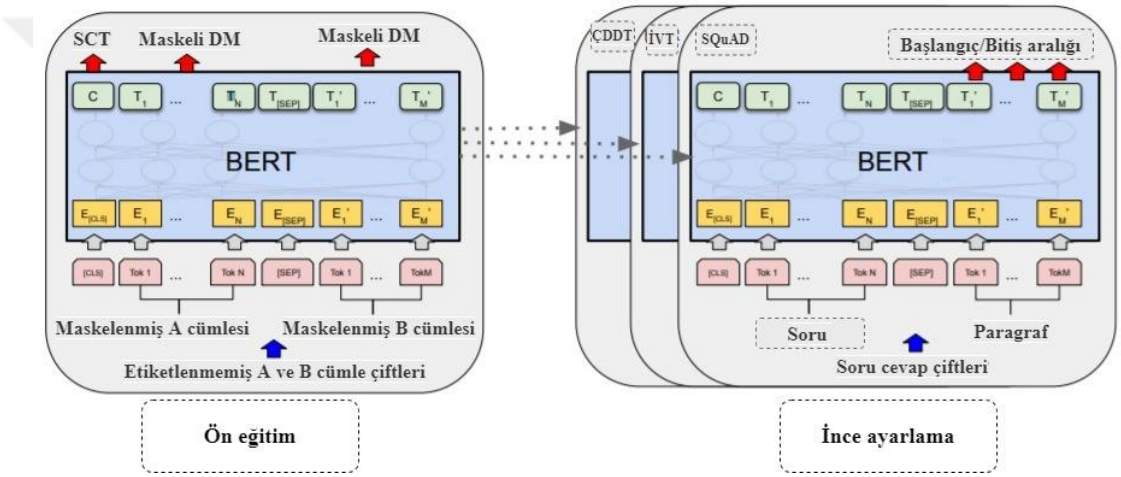
BERT, bağlamsal kısıtlamaların üstesinden gelmek ve iki yönlü ilişkiyi göstermek için iki temel öğrenme stratejisi kullanmaktadır.

Maskeli Dil Modelleme (Masked Language Model-MLM), kelime dizileri BERT modeline aktarılmadan önce, bunların %15'i [MASK] belirteci ile değiştirilmektedir. Sıradaki diğer maskesiz kelimelerin oluşturduğu bağlama dayalı olarak maskelenmiş kelimelerin orijinal değerini tahmin etmeye çalışmaktadır.

Sonraki Cümle Tahmini (Next Sentence Prediction-NSP), BERT eğitim sürecinde modele girdi olarak cümle çiftlerini almaktadır. Çiftteki ikinci cümlenin belgedeki bir sonraki cümle olup olmadığını tahmin etmeyi öğrenmektedir. Modelin eğitiminde, girdilerin %50'si için orijinal belgedeki ikinci cümle, diğer %50'sinde ise rastgele

ikinci cümle seçilmektedir. Modelin rastgele seçilen cümlenin ilk cümle ile ilgili olmadığını belirlemesi beklenmektedir.

Word2Vec, kelimeler arasındaki ilişkileri ortaya çıkarılmasını sağlayan bir çeşit algoritma aracıdır. Analiz edilen metinlerde geçen kelimelerin birbirleri ile olan uzaklık ve yakınlık ilişkilerini vektörel olarak hesaplanabilmesini sağlamaktadır. Hesaplanan bu ilişkiler kolay bir şekilde görselleştirilebilmektedir. Kullanılan bir kelimeye en yakın kelimeleri bularak öneri sistemleri oluşturulabilmektedir. Word2Vec yapısı Google'da çalışan Tomas Mikolov tarafından yönetilen bir ekibin araştırma çalışmaları ile oluşturulmuştur. Geliştirilen yapı daha sonra diğer araştırmacılar tarafından analiz edilip açıklanmıştır.



Şekil 18. BERT için genel ön eğitim ve ince ayar yöntemleri (Devlin vd., 2019)

SCT: Son Cümle Tahmini

DM: Dil Modeli

ÇDDT: Çoklu Doğal Dil Tahmini

İVT: İsim Varlık Tanıma

SQuAD: Stanford Soru Cevap Veri seti

Şekil 18'de gösterimi verilen BERT modeli, kabaca bir kodlayıcı ve kod çözücü yığını olarak ele alınmaktadır. Ancak geleneksel kodlayıcı ve kod çözücü mimarisinde girdi uzadıkça bazı kısımlar bir süre sonra unutulmaktadır. TSA gelen kelimeleri sırayla değerlendirmektedir ve kelimelerin bütünlüğünü korumaktadır. Girdi ne kadar uzun olursa, kelimeler arasındaki ilişki o kadar az olabilmektedir. TSA'da baştaki

kelimelerin değerini düşürme sorunu, kod çözücü her kelimeyi işledikten sonra üretilen Gizli Katman bilgisini ilettiği için hafifletilebilir.

Dönüştürücü, gelen verileri soldan sağa/sağdan sola paralel işleme ve çok kafalı dikkat mekanizması ile değerlendirmektedir. Performansı diğer uygulamalara göre daha yüksektir. Dönüştürücü, TSA ve Dikkat mekanizmasından, gelen metni çift taraflı incelemesi, kelimenin sağındaki ve solundaki kelimelerle olan ilişkisini iyi işlemesi, içeriği MLM ve NSP ile öğrenmesi ile fark yaratmaktadır.

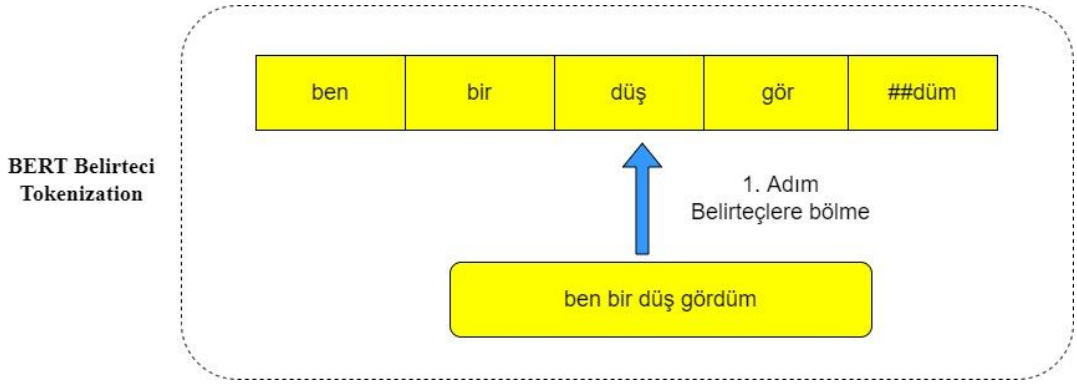
BERT, çift yönlü bir dönüştürücüdür, kodlayıcı ve kod çözücü yığını olarak kullanılmaktadır. BERT tarafından kullanılan dönüştürücü ağı, öz dikkat mekanizmalarını ve ileri besleme ağlarını içeren kodlayıcılar ve kod çözücülerden oluşmaktadır.

3.13.1. BERT Gömme

Bir kelime model tarafından bilinmese bile, tek tek alt kelime belirteçleri, modelin bir dereceye kadar anlamı çıkarması için yeterli bilgiyi tutabilmektedir. Yaygın olarak kullanılan ve diğer birçok DDİ modeline uygulanabilen böyle bir alt kelime belirleme tekniğine WordPiece adı verilmektedir (Song, 2021).

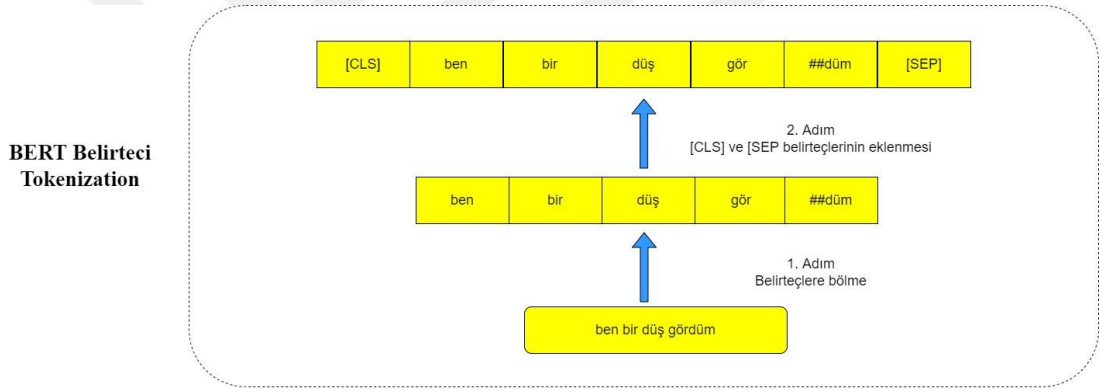
Bir BERT belirteci, cümleyi belirteçlere bölmektedir. Sınıflandırma problemini çözmek için, belirteçler, cümle başına bir [CLS] belirteci ve cümle sonuna bir [SEP] belirteci eklenmektedir. Metin işlemede kullanılan maksimum uzunluk ayarı burada da geçerlidir. Cümle maksimum uzunluktan daha kısaysa, boş alanlar sıfırlarla doldurulmaktadır. Ancak maksimum uzunluktan daha uzun ise fazlalık kısım çıkarılmaktadır. Bu işlemin tamamlanmasının ardından, cümle belirteçlere bölünmektedir ve sonunda belirteçler indekslenmektedir. Basitlik için yalnızca ilk belirtece karşılık gelen gizli dikkat gösterilmektedir. Daha sonra her bir kelime için kelime uzayındaki vektör mesafeleri belirlenmektedir (Alammar, 2020).

BERT modeli çalışmasına, ilk olarak kelimeleri, Şekil 19'da olduğu belirteçlere bölerek başlamaktadır. BERT, kelimeleri belirteçlere bölme işlemi için, BERT Belirtecini (Tokenizer) kullanmaktadır.



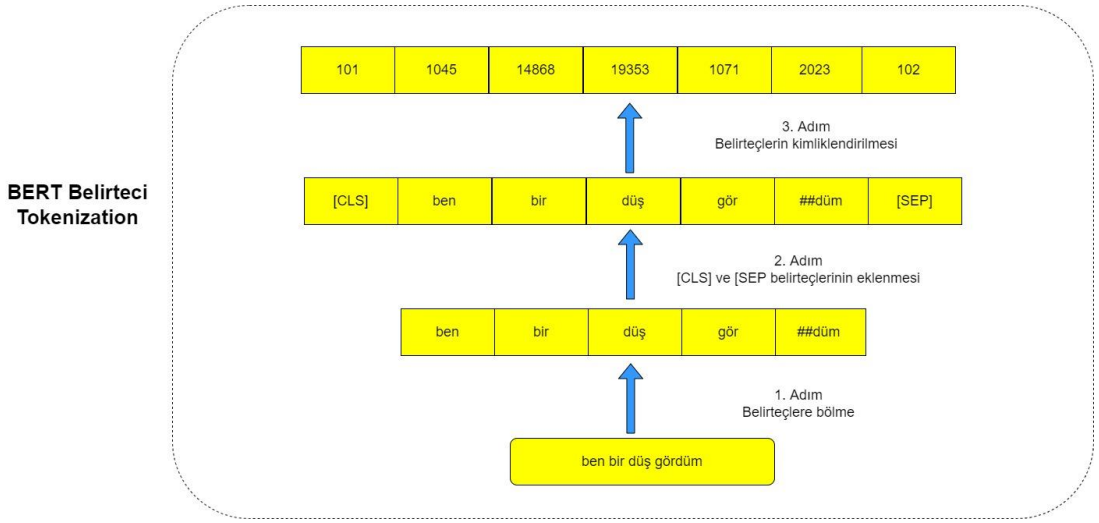
Şekil 19. BERT belirteci birinci adım

BERT belirteci ikinci adımında Şekilde 20'de sunulduğu gibi, cümle sınıflandırmaları için gereken özel belirteçleri eklemektedir. Bu özel belirteçler, [CLS] ve [SEP] belirteçleridir. Cümle başının belirlenmesi için [CLS] belirteci, cümle sonunun belirlenmesi için [SEP] belirteci kullanılmaktadır.



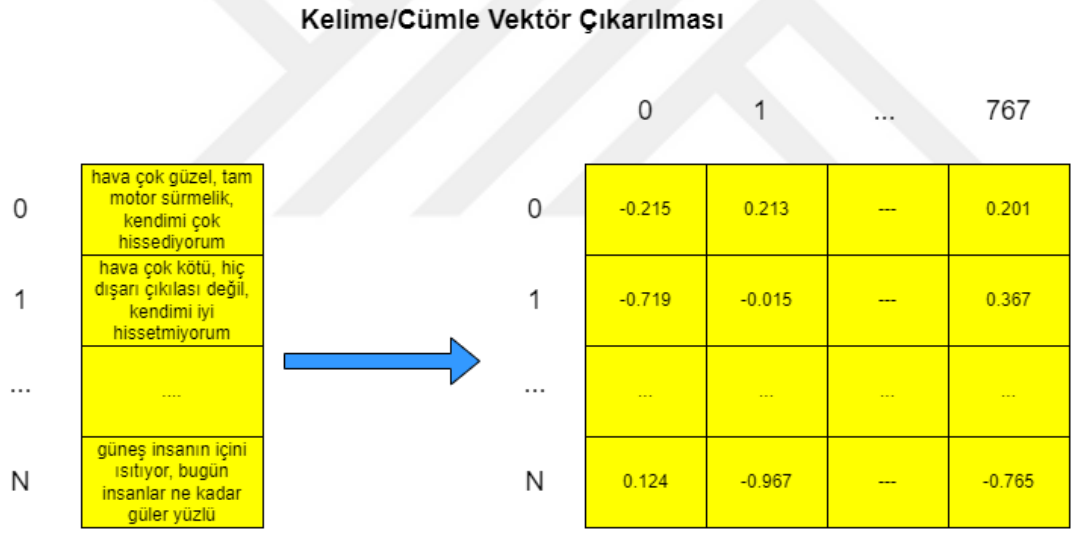
Şekil 20. BERT belirteci ikinci adım

BERT belirteci, Şekil 21'de görülen üçüncü adımında, her bir belirteci önceden eğitilmiş modelle elde edilen bir bilen olan gömme tablosundaki kimliğiyle değiştirmektedir.



Şekil 21. BERT belirteci üçüncü adım

Yukarıda yer alan işlemlerin tamamlanmasının ardından BERT modeli 768 boyutunda bir kelime/cümle vektörü oluşturmaktadır. Gösterimi Şekil 22'deki gibidir.



Şekil 22. BERT modeli cümle/kelime vektör çıkarımı

4. WEB SALDIRILARI

Tezin bu bölümü web saldırıları, web saldırılarının tespitine yönelik yapılan araştırmalardan MÖ ve DDİ yöntemlerini kullanan çalışmalardan oluşmaktadır. Bölüm, bu alan otorite olarak kabul edilen Açık Web Uygulamaları Güvenliği Projesinin yayınlamış olduğu ilk on saldırı raporu ile başlamaktadır. Arından web uygulama saldırı türlerinin anlatımı ile devam etmektedir. Son olarak DDİ ve MÖ yöntemleri kullanılarak web saldırı tespitinde başarı sağlayarak literatüre katkı sağlamış çalışmalara yer verilmesi ile tamamlanmaktadır.

4.1. OWASP Raporuna Göre İlk 10 Saldırı Türü

OWASP (Open Web Application Security Project), Açık Web Uygulama Güvenliği Projesi olarak tanımlanmaktadır. Güvensiz yazılımların yol açabileceği sorunlarla mücadele etme amacını prensip edinmiştir. İnceleme altına aldığı sektörler arasında mali işlemler, sağlık hizmetleri, e-ticaret, internet, sosyal medya, havayolu, enerji, eğlence, hükümet vb. çalışma alanları yer almaktadır.

OWASP'a göre bu saldırılar;

- Enjeksiyon,
- İhlal edilmiş kimlik yönetimi ve oturum çalınması,
- Siteler arası betik çalıştırma,
- Güvensiz doğrudan nesne referansları,
- Yanlış güvenlik yapılandırması,
- Hassas veri pozlama,
- İşlev seviyesi erişim kontrolü eksikliği,
- Siteler arası istek sahteciliği,
- Bilinen zafiyetli bileşenleri kullanma,
- Geçersiz ileri yönlendirmeler

olarak belirlenmiştir.

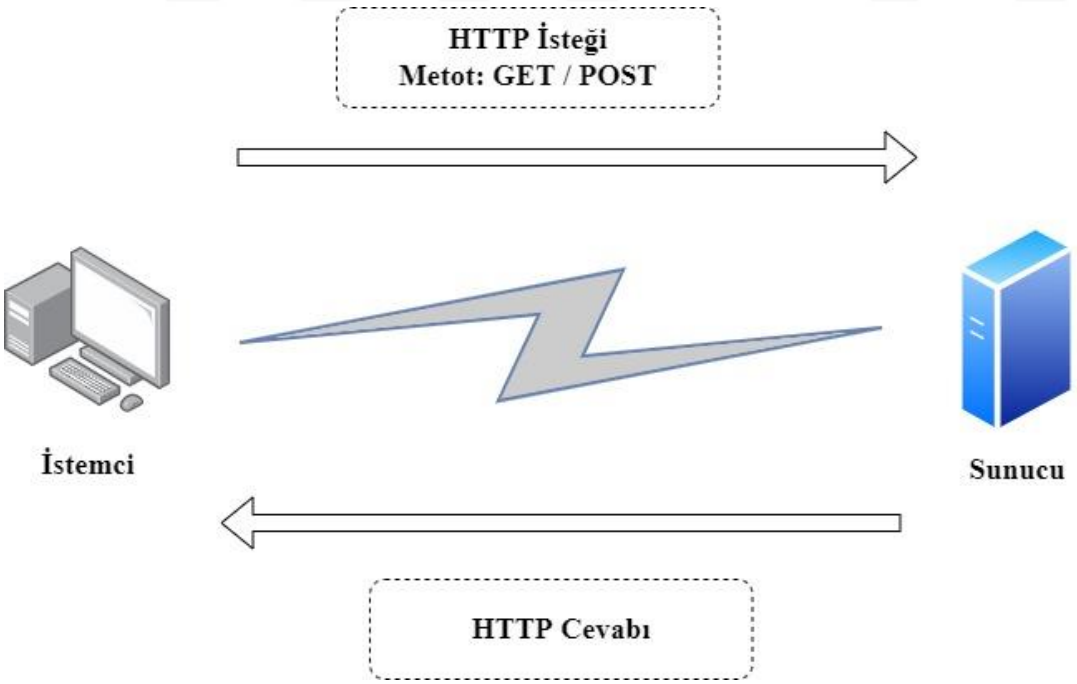
4.2. Web Saldırı Türleri

Web bilgi sisteminde, en az bir kullanıcı makinesi (istemci) ve bilgi sağlayan en az bir web sunucusu olmak zorundadır. Servis sunacak bir web sitesi için en az bir sunucu makinesi, ağ bağlantısı ve IP adresi, alan adı, web sunucu yazılımı, veri tabanı sistemleri, web uygulamaları olması gerekmektedir (Karaarslan, 2008).

Kullanıcın, web sitesi ile etkileşimini web uygulamaları sağlamaktadır. Web uygulamaları ise HTTP iletişim kurallarını kullanmaktadır. HTTP, Köprü Metin Transfer Protokolü olarak da bilinmektedir. En temel görevi web sayfalarının görüntülenmesini sağlamaktır.

İstemci, sunucunun HTTP kapısına (port) varsayılan olarak 80 numaralı porttur. Gönderim Kontrol Protokolü (Transmission Control Protocol- TCP), olarak bilenen bağlantıyla istekte bulunmaktadır. İstemci, "GET" metoduyla, onu izleyen ve isteği tanımlayan bilgileri içeren bir TCP paketi ile web sayfasını istemektedir. Sunucu ise bu isteğe bir yanıt dönmektedir. Bu yanıt olumlu ya da olumsuz olmaktadır (Karaarslan, 2008).

Örnek bir HTTP iletişimi Şekil 23'te gösterilmiştir.



Şekil 23. Tipik HTTP iletişimi

Bir HTTP sunucusu çalışması sırasında, tüm HTTP isteklerine açıktır. Sunucuya erişim sağlamak için ağ güvenlik duvarlarında HTTP kapısı açık bırakılmaktadır. HTTP istekleri, geçerli HTTP istekleri gibi göründüğü için kötü amaçlı kod içerebilmektedir, geleneksel güvenlik duvarları tarafından kabul edilmektedir ve kapsamlı bir şekilde incelenmemektedir.

Saldırganlar, genellikle HTPP/HTTPS protokolü aracılığıyla web uygulamalarına saldırmayı hedeflemektedir (Mac vd., 2018).

Web sunucusu, istemci tarafından istek yapıldığında web sayfalarını çalıştırmaktadır. Web sunucusunun, bu sayfaları depolamak, muhafaza etmek ve istemcilere sunmak gibi çeşitli görevleri bulunmaktadır. Web sunucusu ve web sayfaları arasındaki iletişim, http protokolü ile sağlanmaktadır (Iqsyahiro Kresna & Rosmansyah, 2018).

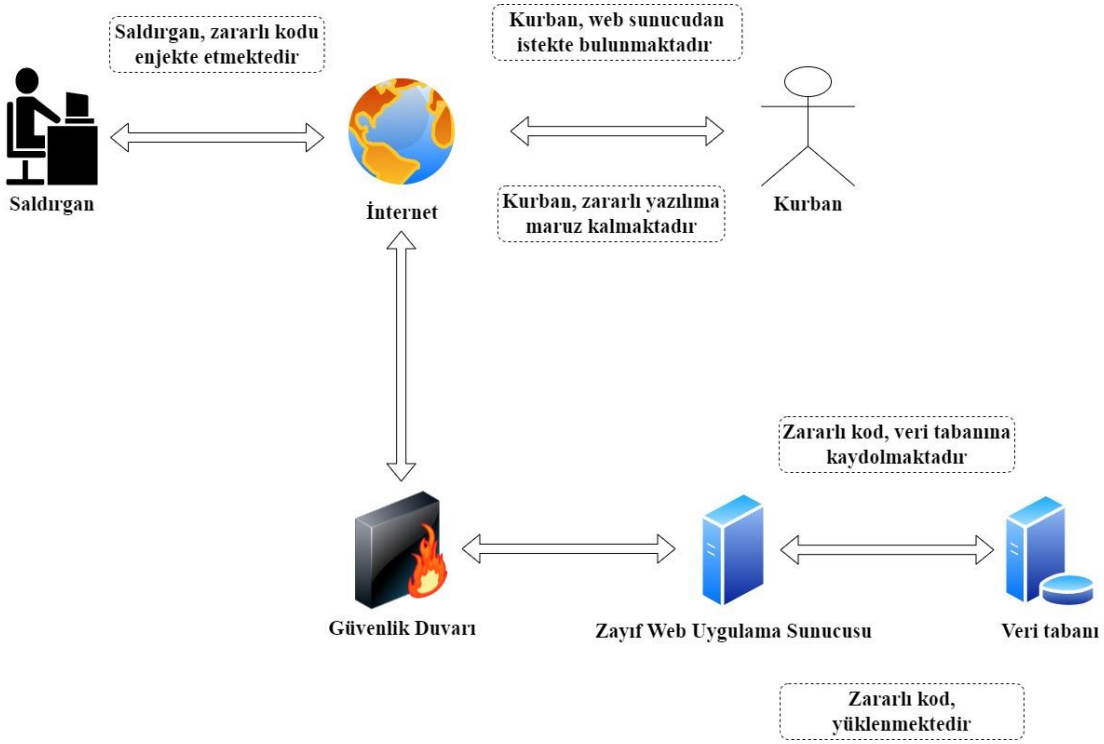
WWW'da en çok kullanılan protokollerden biri HTTP protokolü ve onun güvenli uzantısı HTTPS (HTTP Secure) protokolüdür. Birçok protokolde olduğu gibi, HTTP ve HTTPS protokollerinde de güvenlik açıkları bulunmaktadır. Saldırganlar bu açıklardan yararlanarak Ortadaki Adam, Kaba Kuvvet, Dağıtık Hizmet Reddi, SQL enjeksiyon ve XSS saldırısı gibi saldırılar gerçekleştirmektedirler. HTTPS'de web sayfası parmak izi, paket boyutları ve zamanlama bilgileri gibi bilgilerin hala sızdırıldığı görülmektedir. Saldırganlar ise bu bilgiler kullanılarak hazırlanan özel bir liste ile kullanıcıların şifrelerini tahmin etmek için kaba kuvvet saldırısı gerçekleştirmektedir (Luxemburk vd., 2021).

Ortadaki Adam saldırısında, kullanıcı ve ağ geçidi arasındaki trafik yönlendirilmektedir. ARP sızdırma (zehirleme) ve SSL soyma tekniklerini birleştirerek sinyalleri Wi-Fi ağına göndermektedir. HTTPS, aktarılan verilere kullanıcı bilgisayarının SSL başlığını ve HTTP paketini ekleyerek görüntülemektedir (Chordiya vd., 2018).

Hedefin kaynaklarını tüketmek için, HTTP protokolünün GET veya POST yöntemleri kullanılarak botlardan HTTP sel (flooding) saldırısı gerçekleştirilmektedir. Bazı araçlar kullanılarak uygulama kaynak koduna ulaşılmakta ve hizmet kesintiye uğratılarak Dağıtık Hizmet Reddi saldırısı yapılmaktadır (Bishnoi vd., 2021b).

4.2.1. SQL Enjeksiyon

SQL Enjeksiyon (SQL Injection-SQLI) saldırısı, sorguları yürütürken bir uygulamanın veri tabanı katmanında meydana gelen güvenlik açığından yararlanan bir saldırı türüdür. Saldırgan SQL enjeksiyon saldırısı kullanarak web uygulamalarının verilerini çıkarabilmekte veya değiştirebilmektedir. Saldırı, SQL deyimlerine, dize ve değişmez kaçış karakterlerinin eklenmesi sonucu, kullanıcı girişinin de yanlış bir şekilde filtrelenmesi halinde geçerli olmaktadır. Bu güvenlik açığı, kullanıcı girişinin güçlü bir şekilde yazılmadığından yaygın olarak görülmekte ve beklenmedik bir şekilde ortaya çıkmaktadır.



Şekil 24. SQL enjeksiyon mimarisi

Şekil 24'te gösterimi verilen SQL enjeksiyon saldırılarının ilk örneği, WHERE ifadesinin DOĞRU (TRUE) olarak dönüştürülmesiyle yetkisiz oturum açma saldırısı olarak tanımlanabilmektedir. WHERE ifadesi, aşağıdaki gibi veri çıkarma koşullarını açıklamak için kullanılan SQL ifadelerinin bir parçasıdır:

```
SELECT * FROM user_table WHERE id='user_id' AND password='pass';        (1)
```

1 numaralı SQL deyimi, kimliği ve parolası kullanıcı kimliğiyle uyumlu olan ve oturum açma yetkilendirmesi için geçiş yapan kullanıcı için "user_table"' tablosunda

veri çıkarmayı talep etmektedir. WHERE ifadesinin bir koşulu her zaman DOĞRU (TRUE) ise tablodaki tüm veriler çıkarılmaktadır. Bu nedenle, WHERE ifadelerinin sonucunu aşağıdaki gibi yeniden yazabilen kötü amaçlı SQL ifadelerini kullanarak bir tablodaki tüm verilere erişmek mümkün hale gelmektedir:

' OR 1=1; -- (2)

Örneğin, 2 numaralı kötü niyetli SQL deyimi, 1 numaralı SQL deyimine aşağıdaki şekilde enjekte edilebilmektedir:

SELECT * FROM user_table WHERE id='' OR 1=1' -- ' AND pass;ord='pass'; (3)

Bu enjekte edilen ve 3 numaralı hali alan SQL ifadesindeki, WHERE ifadesi her zaman DOĞRU (TRUE)'dur, çünkü 1=1 her zaman DOĞRU (TRUE)'dur. Bu nedenle, enjekte edilen 3 numaralı SQL ifadesi, id ve şifre doğru olmamasına rağmen tüm verileri kullanıcı tablosundan çıkarmaktadır, bu nedenle yetkisiz oturum açmaya müsait bir durum ortaya çıkmaktadır. Birkaç gereksiz sembol ('AND password='pass';) satır yorumu tarafından yok sayılmaktadır. Satır yorumu (--) sembolünden sonraki kelimeler yorum olarak kabul edilmektedir.

SQL enjeksiyon saldırılarının bir başka örneği, UNION deyimi tarafından yasa dışı olarak veri çalma saldırısı olarak tanımlanmaktadır. UNION ifadesi, iki SELECT ifadesinin sonuçlarını aşağıdaki gibi birleştirmek için kullanılan SQL ifadelerinin bir parçasıdır:

SELECT * FROM products_table WHERE name='product_name'; (4)

4 numaralı SQL deyimi, ürünleri aramak için adı ürün adıyla uyumlu olan ürünler tablosundan veri çıkarmayı talep etmektedir. Bu durumda SQL enjeksiyon saldırıları aşağıdaki kodu enjekte ederek yürürlüğe girmektedir:

‘UNION SELECT * FROM user_table; -- (5)

5 numaralı kötü niyetli SQL deyimi, 4 numaralı SQL deyimine aşağıdaki gibi enjekte edilmektedir.

SELECT * FROM products_table WHERE name='
' UNION

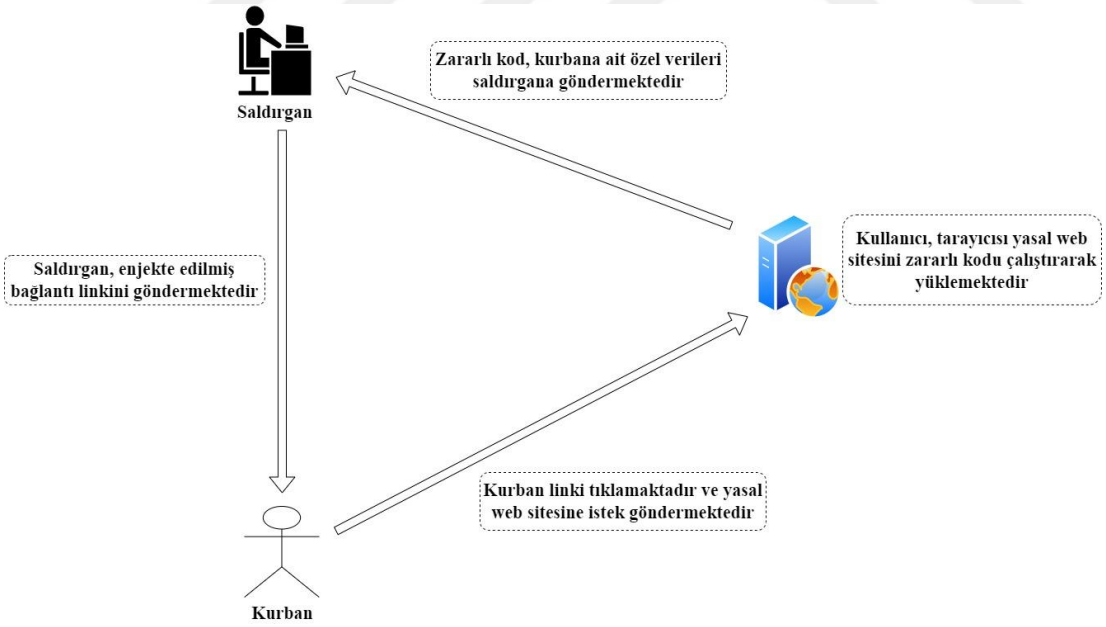
SELECT * FROM user_table; -- '; (6)

6 numaralı SQL deyimi, iki SELECT ifadesinin, birleştirilmiş sonuçlarıdır; "SELECT * FROM products_table WHERE name=' '" ve "SELECT * FROM user=tablo;" UNION. Web uygulamaları arama sonuçlarına, 6 numaralı SQL deyimi gibi yanıt vermesi durumunda, 5 numaralı kötü amaçlı SQL deyiminin enjekte edildiği sorguya cevap vermiş olmaktadır. Yani ürün tablosundaki arama sonuçlarıyla birlikte kullanıcı tablosunu da görüntülemek mümkün hale gelmektedir.

4.2.2. Siteler Arası Betik Çalıştırma

Siteler Arası Komut Dosyası Oluşturma (Cross Site Scripting-XSS) saldırısı, bir saldırganın, kurban kullanıcıların tarayıcısına kötü amaçlı JavaScript vb. kod enjekte etmesine olanak tanıyan web güvenliğindeki bir açıklıktan yararlanılmasıyla yapılmaktadır.

Kötü amaçlı komut dosyası kodu tipik olarak Jscript, HTML, VBScript, Flash vb. istemci taraflı programlama dilleriyle oluşturulmaktadır. Genellikle Jscript ve HTML, XSS saldırıları yapmak için kullanılmaktadır (Hadpawat & Vaya, 2017).



Şekil 25. XSS saldırısı süreci

XSS saldırısı, Şekil 25'teki gibi kurbanın kötü amaçlı kodu yürüten siteyi ziyaret etmesiyle gerçekleşmektedir. Bu nedenle, komut dosyasını yürüten site, kötü amaçlı komut dosyasını kullanıcının tarayıcısına iletmek için bir araç görevi görmektedir.

XSS saldırısı temel olarak yansıyan XSS (Reflected XSS), depolanan XSS (Stored XSS) ve DOM tabanlı XSS (DOM-based XSS) olarak adlandırılan üç sınıfa ayrılır.

Kalıcı ve kalıcı olmayan XSS saldırılarının sunucu taraflı XSS saldırısı olarak, ayrıca DOM tabanlı XSS saldırısını ise sunucu taraflı XSS saldırısının alt kolu olarak tanımlanmaktır (Stency & Mohanasundaram, 2021).

Yansıyan XSS, bu saldırı türü, saldırganın, kurbanın oturum çerezlerini ele geçirmesi ve kendi oturumu gibi hareket etmesi için, kötü amaçlı bir komut dosyasını çalıştırarak hedefine ulaşması olarak tanımlanmaktadır. Saldırgan, elde etmiş olduğu çerez ile herhangi bir şifre kullanmadan kurbanın izinlerini kullanarak eylemler gerçekleştirebilmektedir. Arama motorlarında formlar, URL'ler, çerezler, flash programları ve hatta videolar aracılığıyla kod enjekte edilebilmektedir ve bu şekilde saldırı gerçekleştirilmesi yaygın olarak görünmektedir. Saldırının gerçekleştirilmesinde, bir çıktı yanıtı oluşturması için kullanıcı tarafından sağlanan bilgileri kullanan web uygulamalarındaki güvenlik açıklarından yararlanılmaktadır. Bu teknikle saldırgan, kurbanı JavaScript kodunun çalıştırılması için web sitesinde herhangi bir yeri tıklaması için ikna etmesi gerekmektedir. Bu şekilde tüm trafik saldırgana yönlendirilmektedir.

Kalıcı XSS, bu saldırı türünde de, kötü amaçlı kod doğrudan web sayfasına veya savunmasız siteye doğrudan enjekte edilmektedir. Bu saldırıyı gerçekleştirmek için programlama etiketleri olarak da bilinen JavaScript gibi komut dosyaları gerekmektedir. Saldırının gücü, bu kodların ilk saldırı gerçekleştirildikten sonra tüm kullanıcılar için web üzerinde kalıcı hale getirilmesine izin vermektedir. Bu eylemlerin sonucu, bir kullanıcı web sitesinin enjekte edilmiş bir XSS kodunun bulunduğu bölümüe her girdiğinde, web tarayıcısında programlanan tüm eylemleri gerçekleştirmesi olarak ortaya çıkmaktadır. Bu ikinci tür daha tehlikelidir çünkü saldırgan tarafından gönderilen komut dosyaları sunucuda kalıcı olarak saklanmaktadır ve daha sonra bu web sitesini ziyaret eden kullanıcılara gösterilmektedir. Web sitelerinin sunucularında depolanan içeriğe kötü amaçlı komut dosyalarının enjekte edilmesi mantığına dayanmaktadır.

DOM tabanlı XSS, bu saldırı türü daha karmaşık olarak kabul edilmektedir. DOM tabanlı Tip 0 ya da DOM tabanlı XSS saldırısı olarak da bilinmektedir. Kötü amaçlı kod bir URL aracılığıyla enjekte edilmektedir, ancak web sitesinin bir parçası olarak

kaynak koda yüklenmemektedir. Kötü amaçlı kod sunucuya ulaşmadığı için tespiti daha zordur, bu nedenle yerel bir XSS olarak kabul edilmektedir. Kurban, virüslü bir web sayfası açmakta ve kötü amaçlı kod, önceden herhangi bir doğrulama olmaksızın çalışan bir web tarayıcı dosyasına kendisini yüklemek için bir güvenlik açığından yararlanmaktadır. Bu saldırıda, daha önce belirtilen saldırıların aksine sunucu dahil olmamaktadır. Her ikisi de kullanıcının bir bağlantıya tıklamasını gerektirdiğinden, yansıyan türün saldırısına benzemektedir, ancak oturum çerezlerini elde etmek daha etkili bir yol olarak değerlendirilmektedir. Üç saldırı türü arasındaki fark, özellikle yürütüldükleri yere göre belirlenmektedir. Doğrudan ve dolaylı XSS sunucu tarafında yürütülürken, DOM XSS'de sunucunun herhangi bir müdahalesi yoktur, bu nedenle istemci tarafında gerçekleşmektedir. Saldırılar arasındaki diğer bir fark, yürütme zamanı ve yeridir. İlk ikisinde, HTML ile programlanan girişlerden kaynaklanan isteklerin işlenmesi sırasında kötü amaçlı kodlar enjekte edilmektedir. DOM ile kötü amaçlı kod, istemcide yürütme süresi boyunca doğrudan uygulamaya enjekte edilmektedir.

4.2.3. Siteler Arası İstek Sahteciliği

Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery-CSRF), saldırganın, kullanıcıları yapmayı düşünmedikleri eylemleri kandırıp yaptırmasına izin veren web güvenlik açıklarından biri olarak tanımlanmaktadır. Bu teknikle saldırgan farklı web sitelerinin birbirine müdahalesini önlemek için tasarlanan Aynı Kök Politikası (Same Origin Policy-SOP)'nı kısmen atlatmaktadır.

4.2.4. Uzaktan Dosya Ekleme

Uzaktan Dosya Ekleme (Remote File Inclusion-RFI), bir saldırganın bir komut dosyası aracılığıyla web sunucusuna uzaktan bir dosya eklemesine izin veren bir saldırı türü olarak tanımlanmaktadır. Bu saldırı, veri hırsızlığına veya manipülasyonuna neden olmaktadır. Web uygulamalarının, istemci tarafında diğer saldırılara yol açabilen JavaScript gibi kötü amaçlı kod yürütülmesine sebep olmaktadır. Bu güvenlik açığı, uygun doğrulama olmadan kullanıcı tarafından sağlanan girdinin kullanılması nedeniyle oluşmaktadır.

4.2.5. Yerel Dosya Ekleme

Yerel Dosya Ekleme (Local File Inclusion-LFI), bir sunucudaki dosyaları web sunucusuna ekleyen bir saldırıdır. Bu saldırı, web sunucusunda kötü amaçlı kod yürütülmesine neden olmaktadır. Bu güvenlik açığı, bir sayfanın tam olarak temizlenmemesi ve izinlerin düzgün verilmemesi halinde ortaya çıkmaktadır, örneğin dosya yolu geçiş karakterlerinin enjekte edilmesi olarak gösterilmektedir. Temizle (sanitize) işlemini atlatmak için dosya yoluna bir boş karakter eklenmektedir.

4.2.6. Dizin ya da Dosya Yolu Geçişi

Dizin ya da Dosya Yolu Geçişi (Directory Traversal-DT), bir uygulamanın eriştiği ancak saldırganların erişemediği ve saldırganlara sergilenmeyen dosyalara, erişim sunan bir güvenlik açığıdır. Saldırı, yetersiz güvenlik doğrulamasından veya kullanıcı tarafından sağlanan dosyası adlarının yetersiz temizlenmesinden yararlanmaktadır. Böylece üst dizine geçişi temsil eden karakterler, dosya API'sine iletilmektedir.

4.2.7. E-posta Çıkarma

E-posta çıkarma (Email Extraction), e-posta ayıklama olarak da adlandırılmaktadır. Web uygulamalarını tarayarak e-posta adreslerini ve diğer kişisel iletişim bilgilerini çıkarma işlemi olarak tanımlanmaktadır. Bu e-postalar daha sonra promosyon kampanyaları ve benzer pazarlama amaçları için kullanılmaktadır.

4.2.8. Spam Yorum Gönderme

Spam yorum gönderme (Comment Spamming), popüler arama motorları tarafından döndürülen arama sonuçlarında spam gönderenin web sitesinin sıralamasını değiştirme olarak tanımlanmaktadır. Sıralamanın yükselmesiyle, bu sitenin potansiyel ziyaretçi ve ödeme yapan müşterilerinin sayısının artırmasına olanak tanımaktadır. Saldırı, ziyaretçilerin köprüler içeren içerik göndermesine izin veren web uygulamalarını hedeflemektedir. Saldırgan, tanıtılan siteye bağlantılar içeren herkese açık çevrimiçi forumlara otomatik olarak rastgele yorumlar veya ticari hizmet tanıtımları göndermektedir.

4.2.9. HTTP Protokolü İhlali

HTTP Protokolü İhlali (HTTP Protocol Violation), HTTP protokol tanımına uymayan isteklerin gönderilmesi durumudur. Bu ihlaller, geçersiz HTTP yöntemleri,

parametrelere geçersiz baytların dahil edilmesi vb. işlemleri içermektedir. Bu tür trafikler, standart web tarayıcıları yerine, özel komut dosyaları tarafından oluşturulmaktadır.

4.3. Literatür: Makine Öğrenmesi, Derin Öğrenme ve Doğal Dil İşleme Tabanlı Web Saldırı Tespit Çalışmaları

2008 yılında Karaarslan, yaptığı tez çalışmasında web uygulama güvenliği temel alarak web altyapısına yönelik farkındalık artırmaya ve saldırılara karşı açıklık tespit etmeye yönelik bir model sunmuştur. Modelin testleri Ege Üniversitesi laboratuvarlarında gerçekleştirilmiş olup, kampüs ağında kullanılarak sistemin güvenliği ortaya konulmuştur (Karaarslan, 2008).

2011 yılında Komiya ve arkadaşları, web uygulama aktivitelerinin hayatı kolay hale getirmesi, kullanıcılara ait birçok bilgiyi içermesi ve fazlaca güvenlik açığı içermesi nedeniyle zararlı yazılım tespitine yönelik çalışmışlardır. Zararlı web kodlarının sınıflandırılmasına yönelik MÖ ile bazı çalışmalar gerçekleştirmişlerdir. Naive Bayes (NB), Destek Vektör Makinaları (Support Vector Machine-SVM-DVM), K-En Yakın Komşu(K-Nearest Neighbors-KNN-KEYK) algoritmalarıyla çalışmalar yapılmıştır. Çeşitli veri setleri kullanılmış, tüm algoritmalarda %98 üzerinde doğruluk oranı ile çalışmalarının performanslarını sunmuşlardır (Komiya vd., 2011).

2012 yılında Shar ve arkadaşları, yaptıkları çalışmada SQL enjeksiyon ve XSS saldırıları üzerinde durmuşlardır. Çalışmalarında Naive Bayes (NB), C4.5 ve ÇKA sınıflandırıcılarını kullanmışlardır en yüksek sonu C4.5 ve ÇKA modellerinde %85 üzeri doğruluk oranı olarak ortaya koymuşlardır (Shar & Tan, 2012).

2013 yılında Demirol ve arkadaşları, SQL enjeksiyon saldırıları üzerine bir çalışma hazırlamışlardır. Bu çalışmada, SQL enjeksiyon saldırılarının nasıl yapılacağına yönelik ve bu saldırılara karşı alınacak yöntemlere yönelik bilgilendirmeler aktarmışlardır (Demirol vd., 2013).

2016 yılında Sevri, yapmış olduğu tez çalışmasında web saldırılarının tespit edilmesine yönelik bir web sitesi hazırlamıştır. Bu çalışmada, web sitesine gelen normal ve anormal trafiklerden yararlanılarak bir veri seti hazırlanmıştır. Sınıflandırma için Karar Ağaçları (Decision Tree-DT-KA), Naive Bayes (NB) ve K-En Yakın Komşu algoritmaları kullanılmıştır. Kullanılan bu algoritmalardan en

yüksek sonuç, Naive Bayes algoritmasında ve %94,59 doğruluk oranı olarak kayda geçilmiştir. (Sevri, 2016).

2016 yılında Tekerek ve arkadaşları, web uygulamalarına yönelik http protokolü kullanılarak yapılan saldırıları temel alan bir çalışma yürütmüşlerdir. İmza tabanlı ve anomali tabanlı bir model önerilmiştir. CSIC 2010, ECML-PKDD 2007 ve WUGD 2015 veri setlerini kullanılarak model test edilmiştir. Önerdikleri modelin başarısını ise ortalama %95 doğruluk oranı olarak sunmuşlardır (Tekerek vd., 2016).

2016 yılında Baykara ve arkadaşları, web sunucu kayıtlarını analiz etmek için web tabanlı bir platform oluşturmuşlardır. Web saldırılarını tespit etmeye yönelik yapılan bu çalışmada veri madenciliği yöntemleri kullanılmıştır ve SQL enjeksiyon, DDos ve XSS saldırılarını tespit amaçlı çalışmalar gerçekleştirilmiş ve sistem yöneticisine analiz sonuçlarının sunulduğu ifade edilmiştir (Baykara vd., 2016).

2016 yılında Kaytan, yapmış olduğu tez çalışmasında MÖ ile web tabanlı oltalama saldırıları tespiti üzerine yoğunlaşmıştır. NSL-KDD, Cup 99 ve UCI veri setleri kullanılan çalışmada sınıflandırma için Multiple Kernel Boost Destek Vektör Makinesi kullanılmıştır. Modelin performansı için 5li ve 10lu çapraz doğrulama testi denenmiş ve en yüksek doğruluk oranı % 95,93 olarak ölçülmüştür (Kaytan, 2016).

2017 yılında Liang ve arkadaşları, kişisel verilerin bulutta saklanmasının artması nedeniyle web tabanlı uygulamalara yönelmişlerdir. Kural tabanlı saldırılarla baş etmenin zorluklarını göz önüne alan, bir model hazırlamışlardır. Bu model iki aşamalı TSA içermektedir. Model, ilk aşamada yalnızca normal isteklerle denetimsiz olarak eğitilmiştir, ikinci aşamada ise denetimli olarak hem normal hem anomali isteklerle eğitilmiştir. CSIC 2010 ve WAF veri setleri kullanılmış olup en iyi sonuç UKSB kullanılan modelde % 98,42 olan tespit ile sonlandırılmıştır (Liang vd., 2017).

2018 yılında Hoang, web sitelerini bozmaya yönelik saldırılar üzerine çalışma gerçekleştirmiştir. Normal ve saldırıya uğramış web sitelerini ayırmak için MÖ tabanlı bir yöntem önermiştir. Özellik çıkarımı için n-gram yöntemi kullanılan yöntemde sınıflandırma için WEKA üzerinde Naive Bayes ve J48 Tree (C4.5) algoritmaları kullanılmıştır. Deneylerini %93 üzerinde doğruluk ve % 1 altında yanlış pozitif oranı ile tamamlamıştır (Hoang, 2019).

2018 yılında Dong ve arkadaşları, web isteği sorgu dizilerinin analizi üzerinde durmuşlardır. Bu analiz web saldırganlarını tespit amaçlı tasarlanmıştır. Saklı Markov

Modeli ve Destek Vektör Makineleri (DVM) ile bir hibrit model oluşturulmuştur. Modelin performansı %99,5 F1-ölçüm değeri ve 1000 web sitesinin analizinin 6.7 s süre ile sonuçlandırılmıştır (Dong vd., 2018).

2018 yılında Mac ve arkadaşları, Derin Otomatik Kodlayıcı kullanarak web saldırılarını tespit etmeye yönelik bir çalışma sunmuşlardır. Modelin eğitim ve testleri için CSIC 2010 veri seti kullanılmıştır. Hesaplama süresi olarak istek başına 5.1 ms ve modelin performansı %95 F1-ölçüm değeri ile elde etmişlerdir (Mac vd., 2018).

2019 yılında Şanlıöz ve arkadaşları, web sitelerini hedef alınan saldırı yöntemlerinden Oltalama saldırısı üzerine çalışma gerçekleştirmişlerdir. Sınıflandırma için Sınıflandırma ve Regresyon Ağaçları (CART), J48 (C4.5) algoritması, Adaboost algoritması, Rastgele Orman (Randım Forest-RF-RO) ve Sinir Ağları (Neural Network-NNet-SA) kullanılmıştır (Şanlıöz vd., 2019).

2019 yılında Yu ve arkadaşları, web uygulamalarına yönelik metin analizi temelli SQL enjeksiyon saldırılarını tespit etmeye yönelik bir çalışma hazırlamışlardır. Kelime yerleştirme için Word2Vec kullanılmıştır, sınıflandırıcı olarak ise Destek Vektör Makineleri(DVM) kullanılmıştır. Modelin performansı, istek başında tespit süresü 0.89 ms ve %1,9 yanlış negatif oranı ile sonuçlandırılmıştır (Lu Yu vd., 2019).

2019 yılında Tian ve arkadaşları, nesnelerinin internetinin gelişmesi ve bulut veri merkezlerine veri aktarımının oldukça fazla olması nedeniyle bu alana yönelik yapılan web saldırılarının tespiti üzerinde çalışmalar gerçekleştirmişlerdir. Yapılan çalışma eş zamanlı çalışan iki DÖ modelinden oluşturulmuştur. Özellik çıkarımı için M-ResNet ve Word2Vec, sınıflandırma içinse ESA modeli kullanılmıştır. CSIC 2010, FWAF ve HttpParams veri setleri üzerinde model eğitim ve testleri yapılmıştır. CSIC 2010 veri seti için bu modelle % 99,41 doğruluk oranı elde edilmiltir (Z. Tian vd., 2020).

2019 yılında Pan ve arkadaşları, bir Sağlam Yazılım Modelleme Aracı (RSMT) hazırlamışlardır. Bu aracın, denetimsiz ve yarı denetimli olarak web saldırı tespiti için uygunluğu değerlendirilmiştir. Daha sonra anomalileri tanımak için etiketli verilerle Otomatik Kodlayıcılar kullanarak boyut düşürülmüş ve RSMT yeniden yapılandırılmıştır. Son olarak RSMT, hem sentetik hem de kasıtlı güvenlik açıkları bırakılan web uygulamalarıyla test edilmiştir. Otomatik Kodyacılarla alınmış olan sonucu %91 F1-ölçüm değeri olarak raporlanmıştır (Pan vd., 2019).

2019 yılında Gong ve arkadaşları, model belirsizliği kavramının web güvenliği alanında kullanılmasına olanak tanımışlardır. Model belirsizliği, model tarafından yapılan tahminin güvenilirliğini tahmin etmek için kullanılmıştır. Çalışmalarında model belirsizliği bir Naive Bayes (NB) modelinin varyansı şeklinde verilmiştir. Saldırı tespit modelleri, açıklama hataları olan gerçek web günlükleri üzerinde eğitilerek, yanlış etiketlenmiş web günlüklerinin daha yüksek bir varyans kazanma eğiliminde olduğu kanıtlanmıştır (Gong vd., 2019).

2019 yılında Bakour ve arkadaşları, akıllı telefonlardaki web sayfalarından gelen tehditlere karşı bir makale yayınlamışlardır. Bu çalışmada, kötü amaçlı web kodlarının web sayfalarıyla enjekte edilmesine karşı önlem olarak bir model geliştirilmiştir. Sonuç olarak, kötü amaçlı yazılımlardan korunmaya yönelik %10 daha iyi performans sağlandığı ortaya konulmuştur (Bakour vd., 2019).

2020 yılında Liu ve arkadaşları, SQL ve XSS saldırılarına yönelik çalışmalar gerçekleştirmiştir. CSIC 2010 veri seti olarak kullanılmıştır, Saklı Markov Modeli (Hidden Markov Model-HMM-SMM) de kullanılarak sonuçlar elde edilmiştir. Modelin performansı, %99,85 doğruluk oranı ile sunulmuştur (Liu vd., 2020).

2020 yılında Fidalgo ve arkadaşları, SQL enjeksiyon saldırıları üzerine bir çalışma hazırlamışlardır. Bu çalışmada PHP dilimlerinden hazırlanmış web sitelerini hedef almışlar, eğitim ve testlerini tamamlamışlardır. Veri setini, kendileri SARD isimli bir uygulama kullanarak hazırlamışlardır. Sınıflandırma için ESA, TSA, UKSB modelleri kullanılmış ve en iyi sonuç olan %95 doğruluk oranı ile çalışmalarını yayınlamışlardır (Fidalgo vd., 2020).

2020 yılında Rahman ve arkadaşları, Bangladeş'teki e-ticaret sitelerindeki web uygulama açıklıklarını esas alan bir çalışma ortaya koymuşlardır. Yapmış oldukları çalışma 9 adet saldırı çeşidi üzerine olmuştur. Sonuç olarak, ilk sırada gelen saldırı çeşidi olarak Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery-CSRF), takip eden ikinci saldırı olarak ise Siteler Arası Komut Dosyasının (XSS) olduğu gözlemlemişlerdir (Rahman & Ahmed, 2020).

2020 yılında Calzavara ve arkadaşları, kara kutu sistemiyle hazırladıkları MÖ destekli Siteler Arası İstek Sahteciliği (Cross-Site Request Forgery-CSRF) saldırılarını bulma aracı tasarlamışlardır. Hazırlamış oldukları modelin gerçek dünya için uygun olduğunu ifade etmişlerdir (Case vd., 2020).

2020 yılında Aliero ve arkadaşları, SQL enjeksiyon yöntemlerine yönelik yapılan çalışmaların eksikliği nedeniyle yeni bir çalışma başlatmışlardır. Bu kapsamda bir kara kutu modeli ortaya koymuşlardır. Önerilen modelin performansını ise %90 F1-ölçüm değeri ile ortaya koymuşlardır (Saidu vd., 2020).

2020 yılında Wong ve arkadaşları, Nesnelerin İnterneti (Internet of Things-IoT) cihazlarının öneminde ciddi bir artış görmeleri nedeniyle bu alana yönelik yapılan saldırılardan Ortadaki Adam (Man in The Middle-MITM) saldırısı üzerine yoğunlaşmışlardır. Mesaj Kuyruğu Telemetri Aktarımı (Message Queuing Telemetry Transport-MQTT) protokolünü kullanan IoT cihazlarına yönelik yeni bir Ortadaki Adam (MITM) saldırısı şeması sunmuşlardır. Kötü niyetli mesajlar üretmek için BERT modeli ve farklı MÖ yöntemleri kullanılmıştır. ÇKA modelinin anomali tespit mekanizmalarına karşı etkili olduğunu göstermişlerdir (Luo vd., 2020).

2020 yılında Li ve arkadaşları, tarafından yayınlanmış bir makalede, zararlı kelimelerle yapılan saldırıların tespitine yönelik bir çalışma yapılmıştır. Bu çalışmada zararlı kelimelerin, semantik ve gramatik açıdan benzerleri ile yer değiştirilmesi amaçlanmıştır. Bu işlem için BERT modeli kullanılmış ve en iyi sonuç olarak %97,8 doğruluk oranı ile sunulmuştur (L. Li vd., 2019).

2020 yılında Samy ve arkadaşları, nesnelerin interneti cihazlarına yönelik yapılan saldırılarla ilgili bir çalışma hazırlamışlardır. Bu çalışmanın temeli, sis hesaplamaya dayanmaktadır. Çalışma kapsamında, 6 adet DÖ modeli incelenmiştir. 5 farklı veri seti kullanılan modelin performansı ise %99 üzerinde doğruluk oranı ve F1-ölçüm değeri olarak ortaya konulmuştur (Samy, 2020).

2020 yılında Tekerek, yapmış olduğu çalışmada web tabanlı saldırılara odaklanmıştır. Veri ön işleme adımı için Kelime Torbası (Bag of Words-BoW) ve sınıflandırma işlemi için ESA DÖ modeli kullanılan çalışmada CSIC 2010 veri setleri üzerinde eğitim ve test süreçleri tamamlanmış ve en iyi sonuç olarak %97,07 doğruluk oranı ve %97,51 F1-ölçüm değeri elde edilmiştir (Tekerek, 2021).

2021 yılında Avcı ve arkadaşları, SQL enjeksiyon saldırıları üzerine bir çalışma sunmuşlardır. Yapılan çalışmada şirketlerin, en önemli verilerinin veri tabanlarında tutulduğunu belirtilmiştir. SQL enjeksiyon yöntemi kullanılarak gerçekleştirilebilecek saldırı şekillerine ve alınacak tedbirlere yönelik bir çalışma sunulmuştur (Avcı vd., 2021).

2021 yılında Durai ve arkadaşları, bulut üzerindeki web saldırılarına yönelik ontoloji tabanlı bir çalışma gerçekleştirmişlerdir. Bu çalışmada kural tabanlı bir sistem önerilmiştir. Yapılan çalışmada sonuç olarak, var olan sistemden daha iyi tespit sağlandığı raporlanmıştır (Durai vd., 2021).

2021 yılında Rojas, tarafından yapılan çalışmada istenmeyen mesaj filtrelerinin çalışma mantığı göz önüne alınarak aldatma hileleri üzerinde durulmuştur. Gelen mesajlarda yer alan kelime dağarcığındaki kelimelerin, BERT modeli kullanılarak eş anlamlılarını çıkarılmıştır. Sınıflandırma için Destek Vektör Makineleri (DVM) kullanmış ve %95 doğruluk oranı alındığı gösterilmiştir (Rojas-Galeano, 2021).

2021 yılında Alvares, tezinde benzer kötü amaçlı yazımları sınıflama üzerine bir çalışma gerçekleştirmiştir. Çalışmasında kelime yerleştirmeleri için BERT modeli kullanmış, sınıflandırma için Destek Vektör Makineleri (SVM), Lojistik Regresyon, Rastgele Ormanlar ve ÇKA gibi sınıflandırıcılar kullanmıştır (Alvares, 2021).

2021 yılında Gong ve arkadaşları, açıklama hatalarının model eğitimini yanlış yönlendirebildiği mantığını temel alan bir çalışma hazırlamışlardır. Bu çalışmada web saldırı tabanlı DÖ tarafından yapılan tahmini değerlendirmek için ölçüm belirsizliği modeli önerilmiştir. Apache-2006, CSIC 2010, Apache-2017 veri setleri üzerinde eğitim ve testler gerçekleştirilmiştir. ESA kullanılan modelin performansını % 99 üzeri bir doğruluk oranı ile sunulmuştur (Gong vd., 2021).

2021 yılında Montes ve arkadaşları, web uygulamalarına yönelik saldırılar üzerine çalışmışlardır. CSIC 2010 ve DRUPAL, veri seti olarak tercih edilmiştir. Çalışma kapsamında RoBerta (A Robustly Optimized BERT Pretraining Approach) ve OCSVN kullanılmıştır. Sonuç olarak DRUPAL veri setinde %95 doğru pozitif oranı, CSIC 2010 veri setinde %47 doğru pozitif oranı ortaya konulmuştur (Montes vd., 2021).

2022 yılında yapmış oldukları çalışmada Shadid ve arkadaşları, web saldırılarını algılama, azaltma ve gerçek zamanlı olarak saldırgan profili oluşturma için bir çalışma yapmışlardır. Oluşturmuş oldukları veri seti üzerinde eğitim ve test işlemlerini gerçekleştirmişlerdir. Yaptıkları çalışma, bir DÖ modelinin, Çerez Analiz Motoru ile iç içe olan bir hibrit yaklaşımını içermektedir. Bu çalışmanın başarısı, hazırlamış oldukları veri seti ve gerçek dünyada test edilmiş ve sırasıyla %99,94 ve %98,74 doğruluk oranı el edeilmiştir (Shahid, Aslam, Abbas, Khalid, vd., 2022).

2022 yılında yapmış oldukları başka bir çalışmada Shadid ve arkadaşları, aldatma teknikleri kullanılarak yapılan web saldırıları üzerine yoğunlaşmışlardır. Bu çalışmanın odağına ise DÖ tabanlı bir sınıflandırıcı ve saldırgan profili oluşturmaya yardımcı olan bir tanımlama bilgisi analiz motoru içeren hibrit saldırı algılama modülü oturmuştur. Önerilen saldırı tespit modülü, %99,94 doğruluk oranı sağlamıştır (Shahid, Aslam, Abbas, Afzal, vd., 2022).

2022 yılında Gaurav ve arkadaşları, XSS saldırı tespitine yönelik bibliyometrik bir genel bakış sunmuşlardır. 2009-2022 yılları arasında bu alanda yapılan çalışmaların artmasını göz önüne almışlardır. Sürdürülebilir girişimcilik alanına önemli bir katkı sağlamakta, alanın evrimi ve mevcut durumuna kapsamlı bir genel bakışın yanı sıra, alandaki çeşitli bakış açılarının, tanımların ve eğilimlerin kapsamlı, sentezlenmiş ve organize bir özetini sunmuşlardır (Gaurav vd., 2022).

2022 yılında Alaoui ve Nfaoui, yapmış oldukları araştırmada, 2010 ile Eylül 2021 arasında DÖ tabanlı web uygulamaları güvenliği üzerine yayınlanan makaleler üzerine analiz çalışması yapmışlardır. Bu çalışma sonucunda standart gerçek dünya web saldırıları veri kümeleri oluşturulmasının, ağ saldırı tespiti gibi benzer alanlarda başarılı olan saldırı sistemlerinin web saldırıları içinde uygulanabileceği ifade edilmiştir. Web uygulamaları güvenliğindeki uzmanlık ile MÖ'deki uzmanlık arasında köprü kurmanın, web saldırılarının tespiti için metin madenciliğinden yararlanmanın, DÖ tabanlı web saldırıları algılama modellerini geliştirmek ve karşılaştırmak için ortak bir çerçeve oluşturulmasının önemli olduğuna değinmişlerdir (Alaoui & Nfaoui, 2022).

2022 yılında Roy ve arkadaşlarının, gelişen teknolojiyi kullanarak hazırlamış oldukları çalışmada SQL enjeksiyon saldırıları esas alınmıştır. Veritabanında yer alan mantıksal kusurlar nedeniyle saldırganlar yeni bir tür mantıksal yük göndermeleri çalışmalarının temelini oluşturmuştur. Yaptıkları araştırmada, Kaggle SQL enjeksiyon veri setinde en iyi sonucu veren sistemin, diğer sistemlere göre %2 daha iyi olduğu ve %98,33 doğruluk oranıyla Naive Bayes (NB) modelinin olduğunu raporlamışlardır (P. Roy & Rani, 2022).

2022 yılında Kshirsagar ve Kumar, izinsiz giriş tespit sistemlerine yönelik geliştirilen makine öğrenmesi modellerinin eksik kaldığı performans ve zaman konularını ele almışlardır. Yaptıkları çalışmada Filtreleme Tabanlı Öznitelik Seçme Yöntemlerini

kullanmışlar ve özelliklerin dörtte birini seçerek web saldırı tespiti gerçekleştirmişlerdir. CICIDS 2017 veri seti üzerinde yapılan deneylerde, J48 algoritması ile %99,99 bir doğruluk oranı elde edilmiştir (Kshirsagar & Kumar, 2022).

5. WEB SALDIRI TESPİTİ İÇİN ÖNERİLEN SİSTEM

Tezin bu bölümü, tez kapsamında web saldırı tespitine yönelik yapılan çalışmaları ve literatüre sağlanan katkıları içermektedir. Bölüm, çalışmanın gerçekleştirildiği uygulama ortamı tanıtılması ile başlamaktadır. Ardından bölüm, çalışma kapsamında kullanmış olduğumuz veri setleri açıklamaları ve örnekleri ile devam etmektedir. Son olarak bölüm, önerilen ÇKA ve ESA tabanlı modellerin, BERT modeli ile birleştirilmesiyle elde edilen mimarilerin sunulması ile tamamlanmaktadır.

5.1. Uygulama Ortamı

Gündelik işlemlerimizi yapmış olduğumuz bilgisayarlar ile derin ağların eğitilmesi zor bir süreç olarak tanımlanmaktadır. Öncelikle kullanılan bilgisayarın DÖ kütüphanelerini destekleyen ekran kartına sahip olması gerekmektedir. DÖ kütüphanelerini destekleyen ekran kartlarına sahip olunsa bile işlem günlerce belki aylarca sürebilmektedir. Bu nedenler göz önüne alınarak önerilen sistemin modellenmesi, gerçeklenmesi ve performanslarının değerlendirilmesi için, 2 adet 16 çekirdekli AMD işlemci, 64GB RAM ve NVIDIA RTX 2080 Ti ekran kartına sahip bir iş istasyonu kullanılmıştır. Ek olarak iş istasyonu, Ubuntu 20.04 işletim sistemine sahiptir.

Python programlama dili, tez kapsamında yapılan çalışmaların içerdiği tüm yöntemleri geliştirmek için kullanılmıştır. Önerilen sistemin modellenmesinde DÖ kütüphanelerinden biri olan Pytorch kütüphanesi (Stevens vd., 2020) kullanılmıştır.

5.2. Uygulamada Kullanılan Veri Setleri

Web saldırılarının sınıflandırılmasında kullanılacak veri seti/setlerinin seçimi büyük önem taşımaktadır. Aslında bu çalışmanın ana performans ölçütü, normal ve anomali URL isteklerinin doğru sınıflandırılmasıdır. Bu çalışmada, tamamen web saldırı kalıplarından oluşan ve literatürde birçok çalışmada kullanılmış olan HttpParams, CSIC 2010 ve FWAF veri setlerinin kullanılması tercih edilmiştir. Web saldırılarının

tespitine yönelik önerilen çözüm yöntemlerinin, performans değerlendirmelerini diğer birçok çözüm yöntemi ile karşılaştırabilmek için bu veri setleri tercih edilmiştir.

Önerilen mimarilerde kullanılan ilk veri seti, 36.000 normal URL isteği ve 25.000'den fazla anomali URL isteği içeren CSIC 2010 (Giménez vd., 2010) veri setidir. Bu veri seti, bir e-ticaret web uygulamasının trafiğini dinlenerek otomatik olarak oluşturulmuştur. CSIC 2010 veri seti, SQL enjeksiyon, XSS saldırısı gibi farklı saldırı türlerini içermektedir.

Tablo 2. CSIC 2010 veri setinde yer alan normal URL istek örneği

/root/public/code/cp_html2txt.php?page=http://192.168.202.118:8080/moclyxlwqyfjnp?

Tablo 3. CSIC 2010 veri setinde yer alan anomali URL istek örneği

/examples/jsp/num/index.php?id=’union/**/select/**/0,0,1369222205,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0–

Burada URL isteği içerisine “UNION” ve “SELECT” kodları enjekte edilmiştir.

İkinci veri seti, Web Saldırıları Güvenlik Duvarı (Web Application Firewall - WAF) tarafından kaydedilen HTTP trafikleriden oluşturulmuştur (Ahmad, 2017). Yinelenen öğeler olmadan daha büyük bir veri seti elde edilmesi için, farklı etki alanlarından URL'ler, etki alanı birleştirme yoluyla adlardan, kaynak yollarından ve nitelik anahtarlarından birleştirilmiştir. 1.290.000 normal URL isteği ve 48.000 anomali URL isteğinden oluşmaktadır.

Tablo 4. FWAF veri setinde yer alan normal URL istek örneği.

http://localhost:8080/tienda1/index.jsp

Tablo 5. FWAF veri setinde yer alan anomali URL istek örneği

http://localhost:8080/tienda1/publico/pagar.jsp?modo=insertar&precioA=51&B1=Confirmar

Burada URL isteği içerisine “INSERT” kodu enjekte edilmiştir.

Üçüncü veri seti GitHub'da web saldırıları çalışmalarında en fazla kullanılan ve HttpParams (Anonymous, 2020) olarak adlandırılan veri setidir. Bu veri seti farklı araçlarla üretilmiş olup, 19.304 normal URL istek ve 11.763 anomali URL istek içermektedir. Bu anomoli URL isteklerinden 10.852 tanesi SQL enjeksiyon saldırı etiketi, 532 tanesi XSS saldırı etiketi, 89 tanesi Komut enjeksiyonu (CMDI) etiketi ile etiketlenmiştir ve dengesiz bir veri seti olarak değerlendirilmektedir.

Tablo 6. HttpParams veri setinde yer alan normal URL istek örneği

"cauifield@tuviaje.com.af","24","norm","norm"

Tablo 7. HttpParams veri setinde yer alan anomali URL istek örneği

"1’));select (case when (8601=1220) then 8601 else 8601*(select 8601 from mysql.db) end)#","88","sqli","anom"

Burada URL isteği içerisine “SELECT” içinde “SELECT” kodu enjekte edilmiştir.

Tablo 8. Veri setlerine ait sayısal detaylar

Veri Seti Adı	**Toplamı**	**İstek Tipi**	
		Normal İstek Sayısı	Anomali İstek Sayısı
FWAF	1.338.000	1.290.000	48.000
CSIC 2010	71.000	36.000	25.000
HttpParams	31.067	19.304	11.763

URL istekleri, protokol, etki alanı adı, üst düzey etki alanı, klasör, dosya adı ve dosya uzantısı gibi birkaç zorunlu ve isteğe bağlı bölümden oluşmaktadır. Bu bölümlerden oluşturulan bir URL isteği, normal bir istek olarak kabul edilmektedir. Ancak, SQL veya komut dosyası dillerinden (örneğin, SELECT, UNION, ALERT) kod enjekte edilmesiyle oluşturulan bir URL isteği, anomali istek olarak kabul edilmektedir. Tez kapsamında yapılan çalışmada kullanılan veri setlerinden çıkarılan normal ve anomali URL istek örnekleri Tablo 2,3,4,5,6 ve 7'de gösterilmektedir.

5.3. Sistem Mimarisi

Saldırganlar genellikle web saldırıları gerçekleştirmek için mevcut URL yapıları ile oynamayı tercih etmektedirler. Saldırganlar, URL yapılarına, eklemiş oldukları script türlerinden oluşturdukları kod parçacıkları vasıtasıyla, kullanıcıların bilgilerinin yer aldığı kişisel bilgisayar, telefon, sunucu vb. gibi cihazlara erişmeye çalışmaktadırlar. Dolayısıyla, URL dizelerindeki her bir kelimenin ya da karakterin neyi temsil ettiği önemli bir durum almaktadır. URL dizelerindeki, her bir kelimeyi ve karakteri korumak gerçekten çok önem arz etmektedir.

Bu durum göz önüne alındığında, URL isteklerinin içerisinde geçen her bir kelimenin ya da karakterin, URL bütünlüğü açısından değerlendirilme ihtiyacı ortaya çıkmaktadır. Bu nedenle URL isteklerinin her birinin, bir cümle gibi ele alınmasının uygun olduğu değerlendirilmiş ve cümle olarak kabul edilmiştir.

Son on yılda, DÖ tekniklerindeki çığır açan gelişmeler nedeniyle DDİ alanında da önemli gelişmeler elde edilmiştir. DDİ tekniklerindeki gelişmelerle beraber yeni yöntemler ortaya çıkmıştır. Word2Vec, GloVe ve FastText gibi yeni çıkan bu metotların metin işleme gibi konulardaki başarıları yapılan çalışmalarla ispatlanmıştır. Ayrıca metin işleme yöntemlerinin web saldırı tespitine yönelik çalışmalarda kullanıldıkları ve oldukça başarılı sonuçlar verdiği rapor edilmiştir. Son zamanlarda ise Dönüştürücülerin (transformer), özellikle BERT modelinin DDİ alanında parladığı, yapılan çalışmalarla ortay konulmuştur.

BERT modeli, bilgi çıkarma, duygu analizi ve soru cevaplama gibi çeşitli DDİ görevlerinde oldukça etkili olduğunu kanıtlamıştır (A. Roy & Pan, 2021).

Literatürde metin girişlerinden elde edilen öznitelik vektörlerine dayalı birçok başarılı sınıflandırma yaklaşım örneği bulunmaktadır. Bu tür çalışmaların başarısından

esinlenerek, tez kapsamında önerilen mimari için benzer bir yaklaşım kullanılmasına karar verilmiştir.

Kelime gömme yöntemleri sabit uzunlukta veri girişi kabul etmektedir. Bu nedenle URL istekleri için sabit uzunluk belirlenmesine ihtiyaç duyulmuştur. Bu kelime uzunluğuyla ilgili detaylar, önerilen modeller içerisinde açıklanmaktadır.

Cümle olarak yani metin olarak kabul ettiğimiz URL isteklerinin küçük parçalara bölünerek (tokenlaştırılması), sayısal değerlere dönüştürülmesi gerekmektedir. Bu dönüştürme işlemi için BERT Belirteci kullanılarak, özellik vektörleri elde edilmiştir.

URL isteklerinin belirteçlere ayrılmasından (tokenlaştırılmasından) sonra elde edilen kelimeler veya cümleler sayısal değerlere dönüştürülmektedir. Daha sonra kelime uzayında, kelime vektörleri oluşturmak üzere birleştirilmektedir.

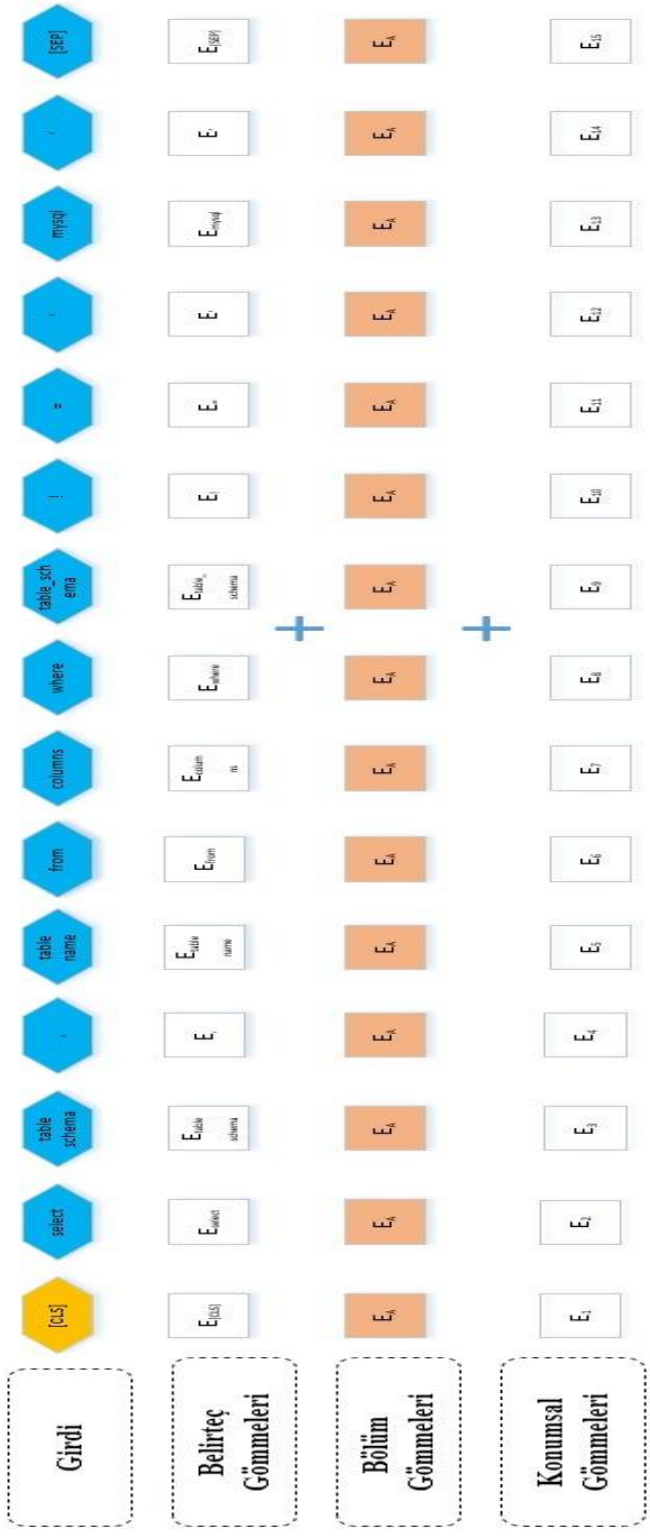
BERT belirteci, kelime temsil işlemini tamamlamasının ardından çıktı olarak 768 uzunluğa sahip vektörler üretmektedir. İşlem süresinin kısalması için tüm cümle, tek bir 768 uzunluğuna sahip bir vektör olarak elde edilmesine ihtiyaç duyulabilmektedir. Bu tek kelime vektörü elde etmek için birkaç yöntem bulunmaktadır. Bunlardan en basit olanı, ikinci ve son gizli katmanın çıktılarının ortalamasının alınmasıyla hesaplanmaktadır. Tez kapsamında yapılan çalışmalarda bu yöntem tercih edilmiştir.

5.3.1. ÇKA Tabanlı Model

Elde edilen bu özellik vektörleri sınıflandırma için ÇKA'ya giriş olarak verilerek, URL istekleri, normal ya da anomoli istek olarak sınıflara ayrılma işlemi gerçekleştirilmiştir.

Metin verilerinin otomatik analizindeki ilk adım, metnin, tipik olarak DDİ tekniklerine başvurulmasıyla gerçekleştirilen, bir bilgisayar tarafından uygun şekilde yorumlanabilen bir temsile dönüştürülmesi olduğu ifade edilmektedir.

Bu nedenle önerilen sistem mimarisi bir metin sınıflandırma altyapısı üzerine kuruludur. Birçok metin sınıflandırma şemasında olduğu gibi, metnin BERT belirteci aracılığıyla simgeleştirilmesi, aşağıda şekilde gösterilmiştir. Bu işlem, önerilen mimarinin birincil adımını oluşturmaktadır. BERT gömme işlemi Şekil 26'da gösterilmektedir.



Şekil 26. BERT sembol gömmeleri

BERT sembol yerleştirmeleri (belirteç yerleştirmeleri) ve cümlelerdeki kelimelerin konumları, görevdeki girdi alanları olarak (konumsal-position) bilgileri ve cümle eşleştirmelerini ifade etmek için kullanılmaktadır. Birinci ve ikinci cümle arasında bir

ayrım yapılmaktadır. Benzersiz bir yerleştirme öğrenme bölümü (segment), temsil ve girdi için ek yerleştirmeler içermektedir.

Kelime gömme yöntemlerinin, sabit uzunlukta verileri, giriş verisi olarak kabul ettiği yukarda belirtilmiştir. Bunun için bir giriş veri uzunluğu belirlenmesi gerekmektedir. Bu nedenle maksimum uzunluk parametresi olarak 80 uzunluğunda bir giriş verisi kullanmanın, URL sorgularının büyük çoğunluğu için yeterli olduğu ampirik olarak yapılmış olan deneylerle belirlenmiştir. Önerilen modelde, BERT belirteci, önceden belirlenen maksimum uzunluk parametresi olarak 80 uzunluğa sahip bir dizi kelime almaktadır. Bu işlemin gerekliliği ise küme boyutlarının tekdüzeliğini garanti edilmesi şeklinde açıklanmaktadır. 80 olarak belirlenen maksimum uzunluk parametresindan daha kısa uzunluğa sahip cümleler ise bu uzunluğu sağlamak için kalan kısımları sıfırlarla doldurulmaktadır. BERT belirteci, giriş URL isteklerini işleyerek, 80x768 boyutunda bir özellik vektörleri çıkarmaktadır. Vektör ağırlıklarına sahip bir 80x768 matris örneği Tablo 9'da sunulmuştur.

Tablo 9. BERT Belirteç işleminden sonra kelime vektör ağırlıkları temsili ile 80x768 matris örnekleri. Her satır bir kelimeye karşılık gelmektedir ve sorgulardaki maksimum kelime sayısı seksen ile sınırlandırılmaktadır

	0	**.**	**.**	**.**	**767**
0	-0.3988637030124664	.	.	.	- 0.420408129692
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
79	-0.5857216119766235	.	.	.	-0.376996338367462

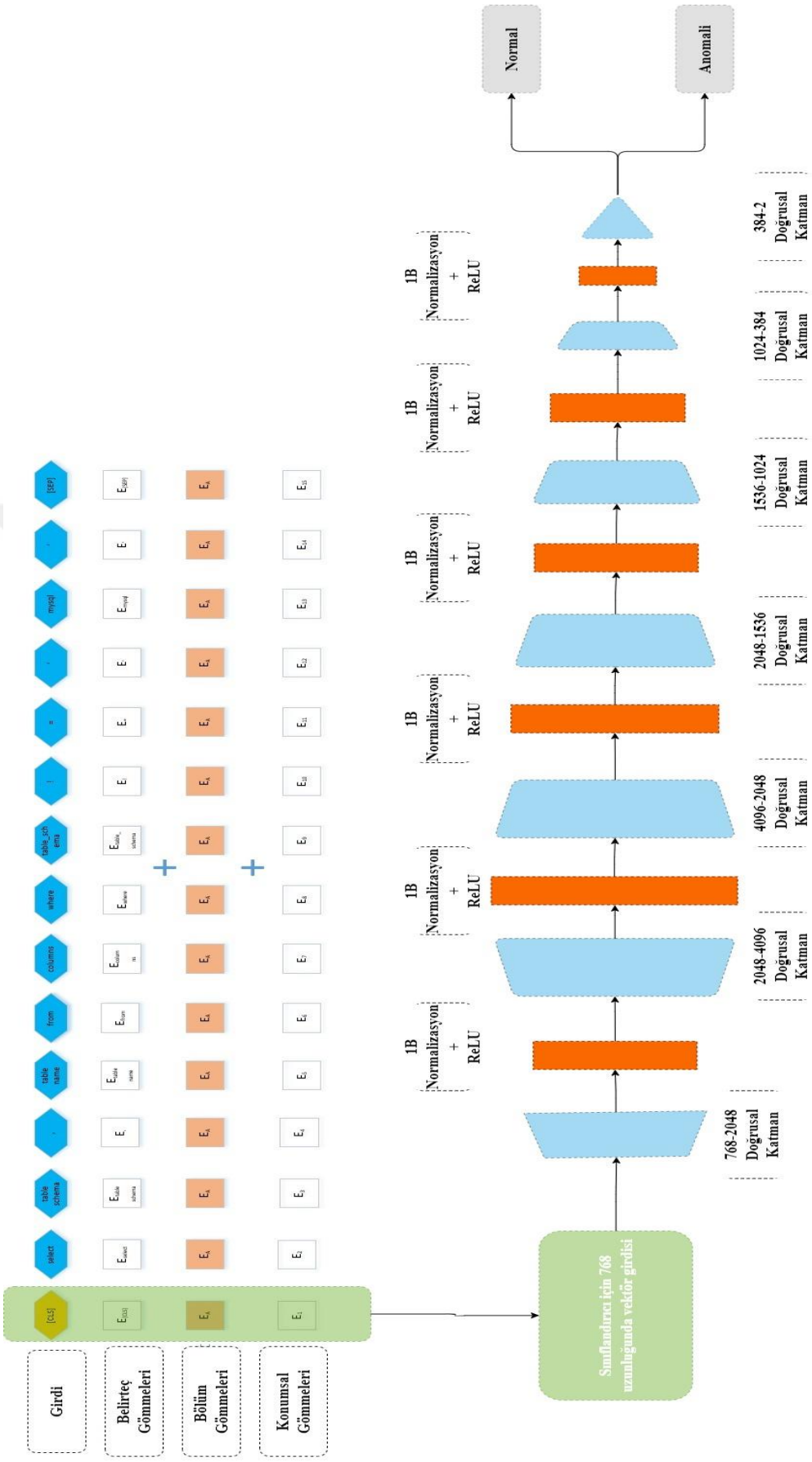
BERT belirteç sürecinin tamamlanmasının ardından 80x768 boyutunda bir özellik vektörü elde edilmektedir. Elde edilen bu özellik vektörü giriş matrisi olarak kullanılmaktadır. Giriş matrisindeki satır vektörlerinin her biri, satırlarda yer alan ilk sütun ([CLS] belirteci) ve son ([SEP] belirteci) sütun hariç olmak üzere, kalan

sütunlar, girdi sorgusundaki belirli sözcüklere karşılık gelmektedir. İlk satır, tüm cümlenin (URL isteklerinin), sınıflandırıcı vektörü olarak tanımlanmaktadır.

Tez kapsamında yapılan deneyler, BERT modeli tarafından hesaplanan olasılık değerlerinin, bir URL isteğinin normal ya da anomali istek olarak, yüksek performansla belirlemesi için yeterli olmadığını göstermiştir.

Bu sorunu çözmek için, BERT modelinin üretmiş olduğu çıktıyı, girdi olarak kabul eden bir ileri beslemeli sinir ağ, yani bir ÇKA'nın, önerilen mimariye dahil edilmesine karar verilmiştir. Bu nedenle, URL isteklerinin sınıflandırılması ve ÇKA modelinin eğitilmesi için BERT modeli tarafından elde edilen ve her biri 768 uzunluğunda olan satır vektörleri kullanılmıştır.

Yapılan çalışma kapsamında kullanılan ÇKA tabanlı mimari, her biri toplu normalleştirme (batch normalization) ve Rektifiye Edilmiş Doğrusal Birim (ReLU) katmanlarının izlediği 6 tam bağlantılı katmandan (yani doğrusal katmanlardan) oluşmaktadır. Önerilen ÇKA tabanlı mimarinin son katmanı, belirli bir URL sorgusu için tahminler veren bir Softmax (Pang vd., 2021) katmanı olarak belirlenmiştir. Doğrusal katmanlar ağırlıkları ve özellikleri belirlerken, diğer katmanlar çoğunlukla çıktıları düzenlemek için kullanılmıştır.



Şekil 27. Önerilen ÇKA tabanlı sistem mimarisi

ÇKA tabanlı sistem mimarisi Şekil 27'de gösterilmektedir. Literatürdeki yer alan diğer sonuçların aksine, önerilen mimari, BERT belirteci dışında, veri ön işlemeye ihtiyaç duymamaktadır. Veri ön işlemeye ihtiyaç duyulmaması da, daha düşük işlem süreleriyle tespitin tamamlanmasına neden olmaktadır.

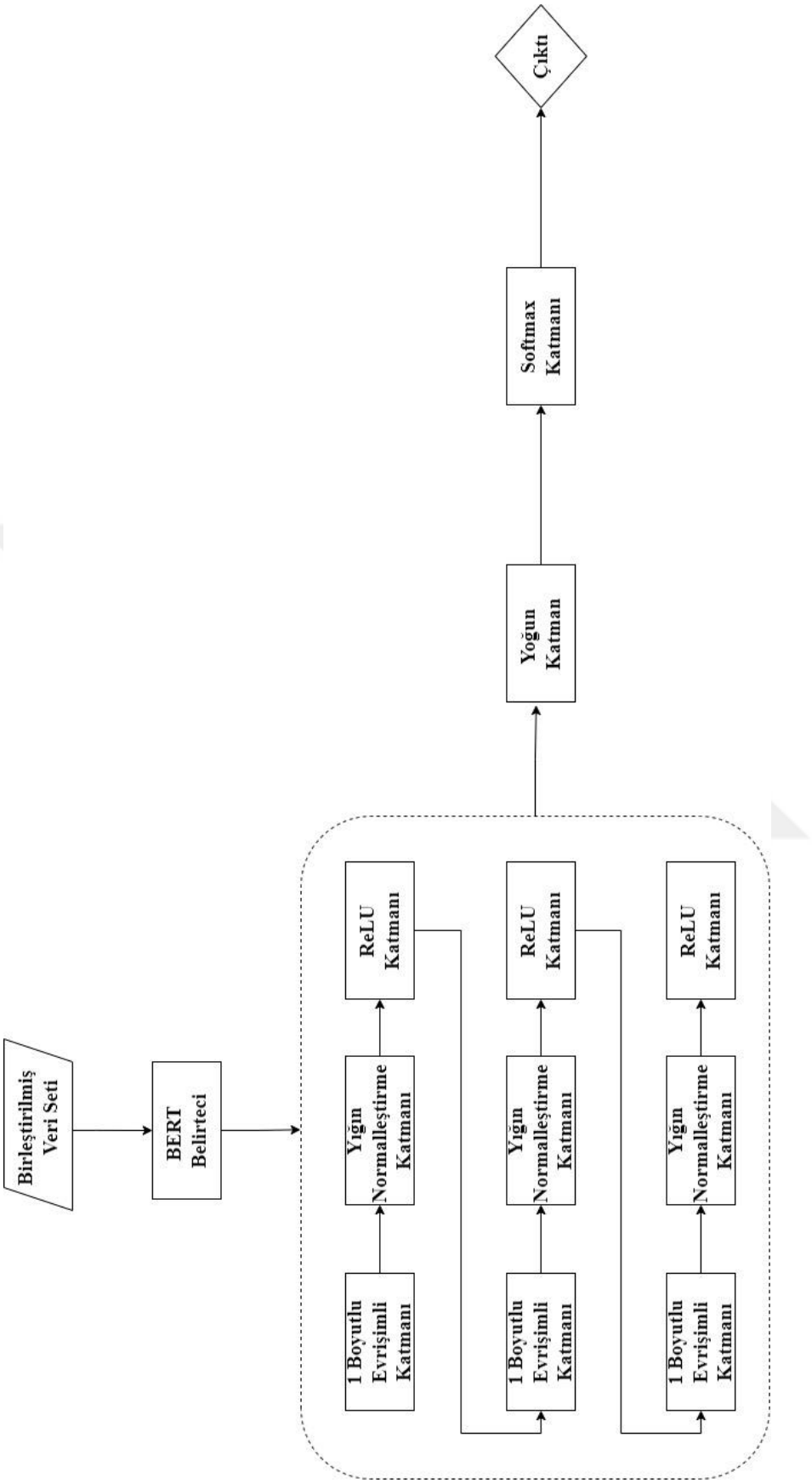
Yapılan deneysel gözlemlerle, ÇKA tabanlı sistem mimarisinin performans değerlendirmelerinin, %99,99 doğruluk oranı ve %97,4 F1-ölçüm değeri olduğu tespit edilmiştir.

5.3.2. ESA-Tabanlı Model

Kelime gömme yöntemlerinin, sabit uzunlukta verileri, giriş verisi olarak kabul ettiğini ve bunun için belirli bir uzunluk parametresinin belirlenmesine ihtiyaç olduğu daha önceki bölümlerde ifade edilmiştir. URL istekleri için maksimum uzunluk parametresinin belirlenmesi için ampirik deneyler yapıldığı ve 80 olarak belirlendiğinden de ÇKA tabanlı mimari anlatımında bahsedilmiştir. BERT belirteci, tarafından alınan URL istekleri 80 uzunluğunda girdiler almakta ve özellik vektörlerini çıkartmaktadır. BERT belirteci, giriş URL'lerini işleyerek, 80x768 boyutunda bir özellik vektörleri çıkarmaktadır. 80x768 boyutundaki özellik vektörleri giriş olarak ESA tabanlı modele verilmektedir. Küme boyutlarının tekdüzeliğini garanti etmek için belirlenen maksimum kelime uzunluğu olan 80 uzunluktan daha kısa olan cümleler ise sıfırlarla doldurmaktadır.

Web saldırı tespit sorununa ikinci çözüm olarak ESA tabanlı bir model önerilmiştir. Bu önerilen model, BERT belirteci tarafından üretilen 80x768 boyutundaki özellik vektörleri girdi olarak kabul etmektedir.

Önerilen ESA tabanlı mimari, 4 katmandan oluşan bir boyutlu ESA modelinden oluşmaktadır. Modelin ilk 3 katmanı, 1 boyutlu evrişimli katman, yığın normalleştirme katmanı ve ReLU katmanından oluşmaktadır. Modelin son katmanı ise URL sorgularının normal ya da anomali istek olarak sınıflandırılması için tahminler veren bir Softmax katmanından oluşmaktadır.



Şekil 28. Önerilen ESA tabanlı sistem mimarisi

Önerilen ESA tabanlı modelin genel mimarisi, Şekil 28'de gösterilmiştir. Her biri 768 uzunluğunda olan vektörler URL isteklerinin sınıflandırılması için ESA modelinin eğitiminde kullanılmıştır. Yapılan çalışmada yer alan BERT model ve ESA tabanlı modelin eğitimleri, birleştirilmiş yeni veri seti (CSIC 2010, FWAF, HttpParams) ile gerçekleştirilmiştir. Hem eğitim hem de test aşamalarında yeni veri seti kullanılmıştır. Yeni veri setinin, verilerinin %85'i eğitim için, %15'i ise doğrulama için iki bölüme ayrılmıştır. Farklı derinliklerde oluşturulan ESA-tabanlı modellerle %95 üzerinde doğruluk oranı ve F1-ölçüm değeri elde edilmiştir. İstek başına tespit süresi olarak 0,3 ms değerleri görülmüştür. Literatürde yer alan tespit sürelerine göre oldukça başarılı olduğu görülmüştür. Ancak önerilen modelin performans değerleri, doğruluk oranı ve F1-ölçüm değerleri açısından istenilen kadar iyi sonuç elde edilememiştir.

Yapılan deneyler kapsamında, ESA da dahil olmak üzere birçok DÖ mimarisi denenmiştir. Ancak bu mimariler arasında en iyi sonuç, ESA tabanlı modelde elde edilmiştir.

Web saldırı tespit sorunun çözümüne yönelik yapılan çalışmalar da göz önüne alındığında, önerilen modellerin genel değerlendirmesi sonucunda hızlı tespit süresi, doğruluk oranı ve F1-ölçüm değerleri göz önüne alındığında ÇKA tabanlı modelinin daha iyi bir çözüm olduğu kanısına varılmıştır. Bu huşulardan dolayı, ÇKA tabanlı mimari ile elden sonuçlarla devam edilmesine karar verilmiştir.

6. TARTIŞMA

Tez kapsamında yapılmış olan çalışmalar sonucunda tarafımızca önerilen mimarilerin, performans değerlerini elde etmek için hesaplama ve deneyler, 2 adet 16 çekirdekli AMD işlemci, 64GB RAM ve NVIDIA RTX 2080 Ti ekran kartına sahip bir iş istasyonu kullanılarak gerçekleştirilmiştir. Mimarilerin tasarlanması, oluşturulması ve performans değerlendirme işlemlerinin tamamlanmasında Python, PyTorch kütüphanesi ve BERT modeli, yazılım araçları olarak kullanılmıştır.

Önerilen modelin eğitilmesinin başlandığı ilk zamanlarda, her bir adım (epoch) için hesaplama süresinin ortalama dört dakika olduğu tespit edilmiştir. Yapılan gözlemler sonrasında her bir adımda en çok zaman alan işlemin, girdi olarak verilmiş olan URL isteklerinden özellik vektörlerinin çıkarılması için BERT belirteci (tokenizer) tarafından işlenmesi sürecinin olduğu tespit edilmiştir. Bu nedenle sorguları tek tek işlemek yerine, ilk olarak tüm sorguları birlikte işlemeye ve elde edilen özellik vektörlerinin kaydedilmesinin denenmesine karar verilmiştir.

Tez kapsamında ele almış olduğumuz web saldırı tespit problemine yönelik en iyi ESA ve ÇKA mimarisini elde etmek için; katman sayılarına, nöron sayılarına ve kayıp fonksiyonlarına yönelik ampirik olarak deneyler yapılarak, önerilen modellerin son halleri elde edilmiştir. Yapılan deneyler sırasında, modellerin hepsi 200 adım için eğitilmiş ve çok sayıda ESA ve ÇKA tabanlı mimarili modeller geliştirilerek en iyi performans gösteren mimari belirlenmiştir. Daha sonra seçilmiş olan modelin, adımlar arasındaki doğruluk oranlarındaki farkın 0.0001'in altına düşmesi için gerekli ince ayarlamalar yapılmış ve iyileştirmeleri belirlenen değere ulaşılana kadar ince ayarlamalara devam edilmiştir. Model eğitimlerinin, 350. adıma kadar devam ettirilmesine rağmen, eğitim süreçlerinin neredeyse her defasında çok daha erken bir adımda düzlüğe ulaştığı gözlemlenmiştir.

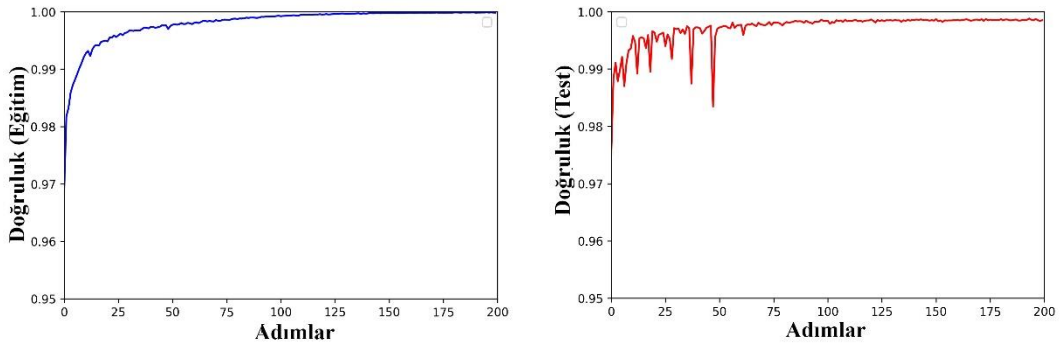
Web saldırı tespit problemi için literatürde yer alan ve çözüm önerilen çalışmalarda farklı veri setleri kullanılmıştır. Model eğitimleri için ilk olarak SQL enjeksiyon, XSS, Komut/Kabul Enjeksiyonu (Command Injection-CMDI) saldırıları gibi 7 farklı

kategoride web saldırı örneklerini barındıran ve 50.000 normal ve anomali istekten hazırlanmış olan CSIC 2010 veri setinin kullanılması kararlaştırılmıştır. Yapılan tez çalışının amacı, normal ve anomali istekleri tespit etmek için bir sistem tasarlanması olması nedeniyle, CSIC 2010 veri setindeki girişleri normal ve anomali istek olarak yeniden etiketlenme işlemine tabi tutulmuştur. Sonuç olarak yedi farklı kategoriden oluşan istekler, normal ve anomoli olarak 2 kategori olacak şekilde etiketleme işlemi yeniden gerçekleştirilmiştir.

Tez çalışması kapsamında ilk ÇKA tabanlı model, iki kategoride etiketlenerek oluşturulan CSIC 2010 veri setinden çıkarılan verilerle test edilen ve %99 doğruluk oranı veren üç gizli katmana sahip basit bir YSA olarak tasarlanmıştır. Bu ÇKA tabanlı model, HttpParams gibi başka bir veri setinde test edildiğinde, doğruluk oranı açısından modelin performansının rastgele tahminlerden biraz daha iyi sonuç verdiği kaydedilmiştir. Bunun sonucunda, CSIC 2010 ve HttpParams veri setlerinde yer alan örneklerin, karşılaştırılabilir olarak değerlendirilemeyecek kadar birbirine benzemediği göz önüne alınmıştır. Bu hipotezin doğruluğunun test edilebilmesi için, ÇKA tabanlı modelin, HttpParams veri setiyle eğitilmesine kadar verilmiştir. Eğitim aşamasından sonra, önerilen model, CSIC 2010 veri setiyle test edilmiştir. Elde edilmiş olan sonuçların bir önceki sonuçlarla benzerlik göstermiş olduğu gözlenmiştir.

Yukarda belirtilen gözlemler sonucunda, yapılan tez çalışması kapsamında, tek bir veri seti kullanılmasına karar verilmiştir. Bu yeni ve tek veri seti, CSIC 2010, FWAF ve HttpParams veri setlerinin birleştirilmesiyle oluşturulmuştur. Bu üç veri setinin birleştirilmesindeki temel amaç, tek biçimliliğin sağlanması olarak düşünülmüştür. Önerilen mimarilerin performans değerlendirmeleri, yeni birleşik veri seti kullanılarak gerçekleştirilmiştir.

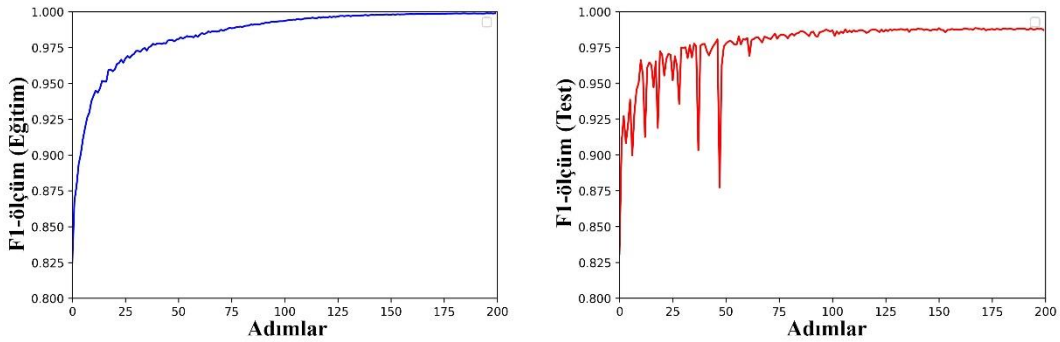
Bu çalışma kapsamında tavsiye edilen modellerin, performans değerlendirmelerinin hesaplanması için doğruluk oranı ve F1-ölçüm değerleri hesaplanmıştır. İlk olarak bu çalışma kapsamında tarafımızca tavsiye edilen modelin doğruluğunu test etmek için model doğrulama tekniklerinden Sınama Seti Yöntemi (Holdout Method) (Oxford & Daniel, 2001) kullanılmıştır. Birleştirilmiş yeni veri seti eğitim ve test veri seti olarak ikiye bölünmüş ve bu yeni veri setinin %85'i önerilen modelin eğitimi için, %15'i ise önerilen modelin testi için kullanılmıştır.



(a) (b)

Şekil 29. (a) Eğitim sırasındaki doğruluk değeri, **(b)** Test esnasındaki doğruluk değeri

Önerilen modelin, eğitim ve test aşamalarının adımlar bazında doğruluk oran değerleri Şekil 29-a ve 29-b'de verilmiştir. 200 adımdan sonra performans değerlerinin sabit bir değere ulaştığı gözlenmiştir. Eğitim aşamasında doğruluk oranı, 7. adımdan sonra %98,00 doğruluk değerini geçmiştir. 91. adımdan sonra doğruluk oranı %99,90 doğruluk değerinin üzerine seyir etmiştir. Önerilen modelin, doğruluk oranı yakınsama seviyesini belirleyen 162. adımdan sonra %99,98 ulaşılması olmuştur. Test aşamasında doğruluk oranı, 47. adımdan sonra %98,30 doğruluk değerini geçmiştir. 49. adımdan sonra doğruluk oranı %99,70 doğruluk oranın altına düşmemiştir. 103. adım sonrasında doğruluk oranı %99,80 doğruluk oranından hep yüksek olmuştur.



(a) (b)

Şekil 30. (a) Eğitim sırasındaki F1-ölçüm değerleri, **(b)** Test sırasındaki F1-ölçüm değerleri

Önerilen modelin, eğitim ve test aşamalarının adımlar bazında F1-ölçüm değerleri Şekil 30-a ve 30-b'de sunulmuştur. Eğitim aşamasında F1-ölçüm değeri, 20. adımdan başlayarak %96,00 F1-ölçüm değerini geçmiştir. 81. adımdan sonra F1-ölçüm değeri, %99,80 F1-ölçüm değerinin üzerine seyir etmiştir. Adım sayısının 150'yi geçmesinden sonra F1-ölçüm değeri, %99,80 F1-ölçüm değeri civarına yerleşmiştir. Test kapsamında önerilen modelin, F1-ölçüm değeri 47. ve 49. adımlardan sonra sırasıyla %88,00 ve %97,00 F1-ölçüm değerlerini geçmiştir. F1-ölçüm değeri, 172. adımın ardından %98,70 F1-ölçüm değerine yaklaşmıştır. F1-ölçüm değerlerindeki farklılıkların, öncelikle F1-ölçüm değer hesaplamalarında kullanılan ağırlıklı kayıplardan kaynaklandığı değerlendirilmiştir.

Web saldırı tespitinin ne kadar önemli olduğu, hem literatürde yer alan çalışmalar göz önüne alındığında hem de önceki bölümlerde verilmiş olan açıklamalarla vurgulanmıştır. Bu nedenlerden dolayı önerilen modelin, elde etmiş olduğu doğruluk oranlarını alternatif doğrulama yöntemlerinden biri olan k-katmanlı çapraz doğrulama (Anguita vd., 2012) ile tekrar test sürecinden geçirilmiştir.

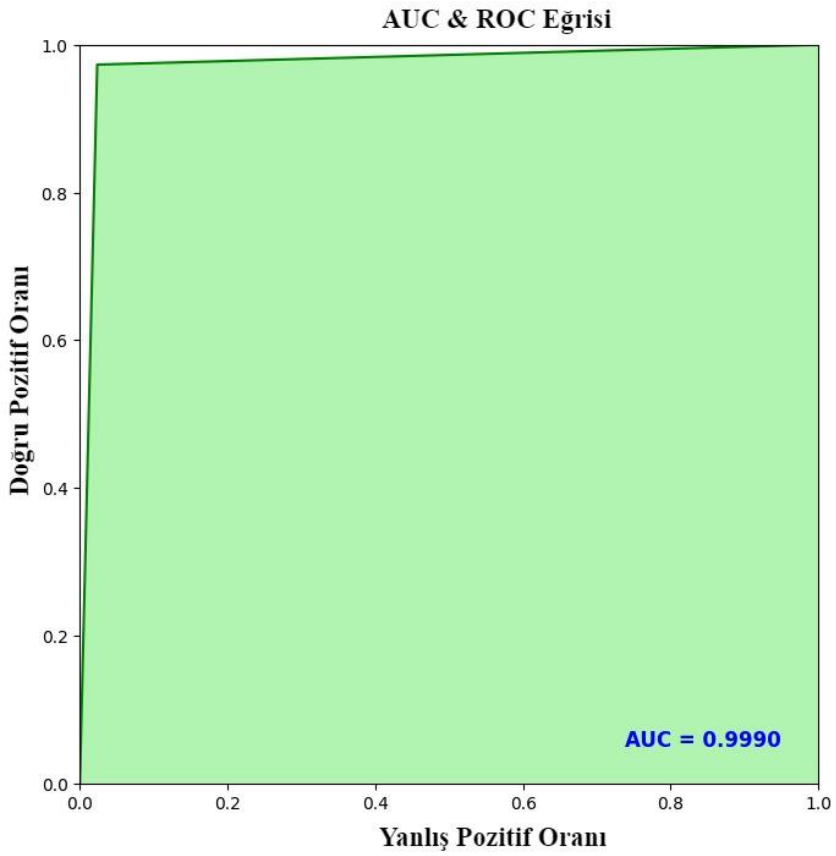
Tablo 10. Önerilen ÇKA tabanlı modelin 10-katmanlı çapraz doğrulama işlem sonucu

	Eğitim Doğruluk Oranı	**Test Doğruluk Oranı**	**Eğitim F1-Ölçümü**	**Test F1-Ölçümü**
1.Katman	%99,83	%99,79	%98,51	%98,21
2.Katman	%99,85	%99,74	%98,67	%97,59
3.Katman	%99,83	%99,73	%98,56	%97,66
4.Katman	%99,84	%99,80	%98,60	%97,15
5.Katman	%99,84	%99,78	%98,65	%98,16
6.Katman	%99,83	%99,76	%98,59	%98,98
7.Katman	%99,84	%99,72	%98,55	%98,56

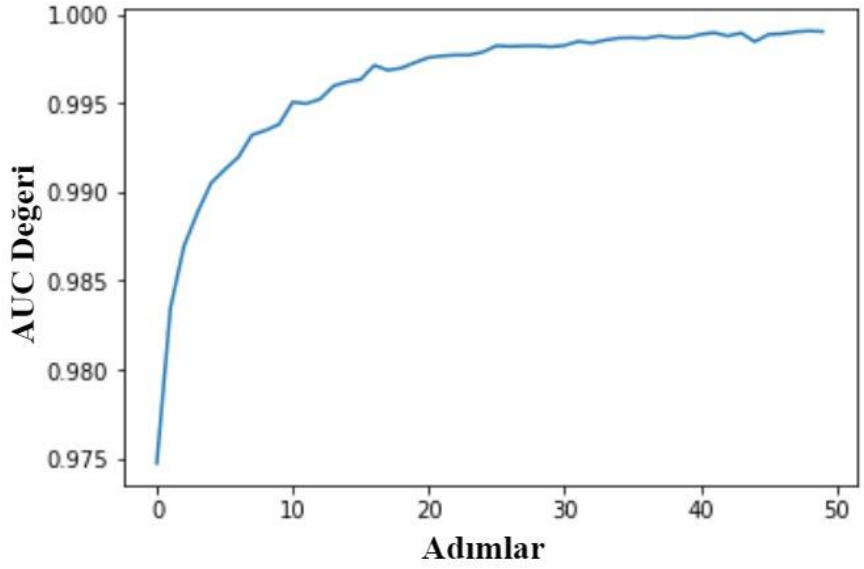
Tablo 10. (Devamı) Önerilen ÇKA tabanlı modelin 10-katmanlı çapraz doğrulama işlem sonucu

	Eğitim Doğruluk Oranı	**Test Doğruluk Oranı**	**Eğitim F1-Ölçümü**	**Test F1-Ölçümü**
8.Katman	%99,84	%99,78	%98,59	%98,15
9.Katman	%99,83	%99,77	%98,51	%98,09
10.Katman	%99,84	%99,66	%98,48	%98,15
Ortalama Değer	**%99,84**	**%99,75**	**%98,57**	**%98,07**

Tez çalışması kapsamından önerilen modelin, 10-katmanlı çapraz doğrulama işlem sonuçları Tablo 10'da verilmiştir. Her katman için en yüksek değerlerin ortalaması alınarak hesaplamalar gerçekleştirilmiştir. Eğitim için doğruluk oranı %99,84, test için doğruluk oranı %99,75, eğitim için F1-ölçümü %98,57, test için F1-ölçümü %98,07 olarak elde edilmiştir (Seyyar vd., 2022).



(a)



(b)

Şekil 31. (a) 10-Katmanlı çapraz doğrulama için AUC-ROC eğrisi **(b)** 50 adım için değişen AUC değerleri

Ayrıca, 10-katmanlı çapraz doğrulama için AUC (Area Under the Curve - Eğri Altındaki Alan) ve ROC (Alıcı İşlem Karakteristiği - Receiver Operating Characteristics) eğrisi çıkarılmıştır ve Şekil 31-a ve 31-b'de sunulmuştur. 50 adım için AUC'deki değişiklik şekil'de çizilir. Elde etmiş olduğumuz sonuçlar, tarafımızca önerilen bu modelin web saldırı tespit işleminde, URL'leri doğru sınıflandırma göz önüne alındığında, güvenilirlik açısından sorunu olmadığını ortaya koyulmuştur.

Tez çalışması kapsamında önerilen mimaride, BERT modelinin BERT belirteci, vasıtasıyla cümle olarak alınmış olan URL isteklerini işleyerek, öznitelik vektörlerini oluşturmaktadır. BERT belirteci tarafından çıkarılan öznitelik vektörleri giriş olarak sınıflandırıcıya verilmektedir ve URL istekleri, normal ya da anomali istek olarak sınıflanmaktadır. Yapılan çalışmada, farklı MÖ teknikleri kullanılmıştır, ardından DÖ tekniklerini ile deneyler yapılmıştır ve en iyi sonuç alınan iki model detaylandırılmıştır. İlk olarak, sınıflandırma için doğrusal bir çıktıya sahip, farklı derinliklerde 1 boyutlu ESA modeli denenmiştir. Bu yaklaşımla yapılan ilk deneyler sonucunda belli bir doğruluk oranı düzeyine ulaşılmış olunsa da, web saldırı tespit sorununa yönelik, tatmin edici bir çözüm olarak sunulması için yeterli olmadığı değerlendirilmiştir. Daha sonra sınıflandırıcı olarak bir ÇKA modeli ile deneyler yapılmasına karar verilmiş ve uygun model seçimi için çeşitli deneyler yapılmıştır. Web saldırı tespit süresi ve kesinlik arasındaki en iyi uzlaşmayı bulmak için ÇKA modelinde ince ayarlamalar yapılmıştır. Yapılan deneyler sonucunda, tarafımızca önerilmiş olan modelin performans değerlendirmelerinin, %99,98 doğruluk oranı ve %98,70 F1-ölçüm değerine ulaştığı gözlenmiştir. Ayrıca, BERT modelinin öznitelik çıkarım süresi göz önüne alınmadığında, önerilen modelin istek başına saldırı tespit süresi 0,4 ms ile, literatürde yer alan ((Dong vd., 2018; Mac vd., 2018), (Lu Yu vd., 2019)) çalışmalarda verilen saldırı tespit sürelerinin yarısından daha az olduğunu gösterilmiştir.

Tablo 11. Önerilen modelin, literatürle karşılaştırılması

Çalışma	Kullanılan Veri Seti	Kullanılan Yöntem	Modelin Performans Değerlendirmesi	Tespit Süresi
(Komiya vd., 2011)	GERÇEK	NB, DVM, KEK	>%98	-
(Shar & Tan, 2012)	SENTETİK	NB, C4.5, ÇKA	>%85	-
(Sevri, 2016)	SENTETİK	KA, NB, KEK	%94,59	-
(Tekerek vd., 2016)	CSIC2010, ECML, PKDD2007, WUGD 2015	-	%95	-
(Kaytan, 2016)	NSL-KDD, Cup99, UCI	Multiple Kernel Boost DVM	%95,93	-
(Liang vd., 2017)	CSIC2010, WAF	TSA, UKSB	%98,42	-
(Hoang, 2019)	-	NB, J48 (C4.5)	>%93	-
(Dong vd., 2018)	GERÇEK WEB SİTELERİ	SMM + DVM	%99.5	6.7 (s)
(Mac vd., 2018)	CSIC2010	OK	%95	5.1 (ms)
(Liu vd., 2020)	CSIC2010	SMM	%99,85	-
(Lu Yu vd., 2019)	-	Word2Vec + DVM	%1,9	0.89 (ms)
(Z. Tian vd., 2020)	CSIC2010, FWAF, HttpParams	M-ResNet + Word2Vec, ESA	%99,41	-
(Pan vd., 2019)		OK	%91	-
(Fidalgo vd., 2020)	SENTETİK	ESA, TSA, UKSB	%95	-
(Saidu vd., 2020)	-	Blackbox	%90	-

Tablo 11: (Devamı) Önerilen modelin, literatürle karşılaştırılması

Çalışma	Kullanılan Veri Seti	Kullanılan Yöntem	Modelin Performans Değerlendirmesi	Tespit Süresi
(Tekerek, 2021)	-	BoW + ESA	%97,07	-
(Rojas-Galeano, 2021)	-	BERT + DVM	%95	-
(Gong vd., 2020)	Apache2006, CSIC2010, Apache2017	ESA	%99	-
(Montes vd., 2021)	CSIC2010, DRUPAL	RoBerta + OCSVN	%95	-
(Shahid, Aslam, Abbas, Khalid, vd., 2022)	-	DÖ	%98,74	-
(Shahid, Aslam, Abbas, Afzal, vd., 2022)	-	DÖ	%99,94	-
(P. Roy & Rani, 2022)	-	NB	%98,33	-
(Kshirsagar & Kumar, 2022)	CICIDS2017	Filtreleme Tabanlı Öznitelik Seçme Yöntemleri	%99,99	-
Önerilen Model	**CSIC2010 FWAF httpParams**	**BERT + ESA, ÇKA**	**%99,9**	**0,4 ms**

Tez sürecinde yapılan çalışmalar kapsamında ulaşılan doğruluk oranları, literatürde yer alan çalışmaların doğruluk oranlarıyla benzer olduğu ya da daha iyi olduğunu Tablo 11'de gösterilmektedir. Ayrıca literatürde yer alan ((Dong vd., 2018; Mac vd., 2018), (Lu Yu vd., 2019)) ve istek başına tespit süresi verilmiş olan çalışmalarla karşılaştırıldığında, çalışma kapsamında tavsiye edilen modelin tespit süresinin önemli ölçüde düşük olduğu görülmüştür. Ek olarak, literatürde yer alan web saldırı tespitine yönelik çözüm yöntemlerinden, farklı olarak tarafımızca önerilen çözüm yönteminin

ön işleme sürecini gerektirmemesinin, büyük bir avantaj olduğu değerlendirilmiştir. Literatürdeki mevcut çalışmalar, model eğitim ve test süreçlerinde genellikle normal ve anomali istek sayılarını hemen hemen eşit sayıda seçtikleri görülmüştür. Gerçek hayatta anormali isteklerin gerçekleşme sıklığını neredeyse yansıtmayan veri kümelerini kullanmaktadırlar. Gerçek hayatta istatistikler göz önüne alındığında, anomali isteklerin oluşma sıklığı, normal isteklerin oluşma sıklığından çok daha düşük olduğu gözlenmektedir.

Özet olarak, çalışma kapsamında yapılmış olan çalışmalar, DDİ tekniklerinden bir tanesi olan BERT modelini, web saldırı tespiti için başarılı bir şekilde kullanılabileceğini göstermiştir ve literatürde BERT modeli kullanılarak web saldırı tespiti yapılan ilk çalışma olarak yerini almıştır. Ayrıca BERT modelinin, önerilen mimariye dahil edilmesiyle, veri normalizasyonuna ihtiyaç duyulmadan web saldırılarının sınıflandırmasının yüksek performansla yapıldığı, üstelik son derece düşük saldırı tespit süresi ile gerçekleştirildiği ortaya konulmuştur.

7. SONUÇ

Saldırganların, web uygulamalarının güvenlik açıklarını tespit etmede kullanmış oldukları yöntemler arasında, URL yapılarıyla oynanması da yer almaktadır. Literatürde yer alan çalışmalar göz önüne alınarak, URL isteklerinin bir cümle gibi ele alınacağı belirlenmiştir. Cümleler ve cümleyi ilgilendiren işlemler söz konusu olduğunda da DDİ teknikleri devreye girdiği bilinmektedir. DÖ tekniklerindeki gelişmeler ışığında, DDİ teknikleri de fazlaca yol kat etmiştir. DDİ tekniklerinden olan BERT modelinin, İngilizce, Fransızca, Arapça gibi çeşitli dillerde göstermiş olduğu yüksek başarı performansı nedeniyle sentetik bir dil olan SQL dilinde de başarı sağlayacağı görülmüştür.

Tez kapsamında yapılan çalışmada, web saldırılarının tespiti için BERT modeli ile ÇKA ve ESA tabanlı modellerin bir arada kullanıldığı yeni bir yaklaşım önerilmiştir. URL isteklerinin kelime temsili için BERT modeli uygulanmıştır. BERT modeli ile elde edilen kelime temsilleri ile URL isteklerinin normal ya da anomali istek olarak sınıflara ayrılması işlemi için de ESA ve ÇKA tabanlı sınıflandırıcılar kullanılmıştır. Deneysel sonuçlar, BERT modelinin, URL istekleri temsil etme konusunda kabiliyetinin oldukça yüksek olduğunu göstermiştir. Ayrıca, yüksek performanslı web saldırısı tespit çalışmalarında önemli ölçüde yarar sağlayacağı ortaya konulmuştur. Bu bağlamda yapılan çalışmaların literatüre olan yeni katkıları aşağıdaki şekilde sıralanmıştır:

Hem eğitim hem de doğrulama aşamalarında kullanılan üç farklı veri kümesini (CSIC 2010, FWAF, httpParams) birleştirerek tek bir veri seti elde edilmiştir. Bu nedenle, bu çalışma kapsamında tavsiye edilen mimari, birden çok veri setinden almış olduğu verileri başarıyla genelleştirebilmiştir. Yapılan araştırmalar neticesinde, bu veri setlerinin diğer çalışmalarda genellikle ayrı ayrı kullanılmış oldukları görülmüştür. Bununla birlikte, önerilen mimari, bilindiği kadarıyla sorgu sınıflandırması yoluyla web saldırısı tespiti bağlamında bu üç veri setini de entegre eden ilk mimaridir.

Literatürde yer alan çalışmalarda genel olarak model doğruluğunu belirlemek için normal ve anomali istek seçimleri için 1:1 oranı kullanılırken, önerilen modelin doğruluğunu belirlemek için normal ve anomali istek seçimleri için 1:1, 1:10 ve 1:20 oranları kullanılmıştır, gerçek dünya senaryolarının daha iyi temsili sağlanmıştır.

Veri normalleştirme, diğer web saldırı tespit çalışmalarında kullanılan ve işlem yükünü artıran bir süreç olarak değerlendirilmektedir. Önerilen mimaride, veri normalleştirmesi gerektirmemektedir, ayrıca performans değerlendirme sonuçları literatürde web saldırısı tespiti konusunda bildirilen sonuçlarla aynı ya da daha iyi olarak elde edilmiştir. Bununla birlikte, 0,4 ms olan işlem tespit süresi, literatürde işlem süresi verilen diğer çalışmalara göre daha düşük tespit süresi ile yerini almıştır.

Tez çalışması kapsamında önerilen mimarinin performans değerlendirmeleri, URL isteklerini başarılı bir şekilde temsil etme ve sınıflandırma yeteneğine sahip olduğunu ortaya koymaktadır. Ayrıca %97 F1-ölçüm değeri ve %99 üzerinde doğruluk oranları ile web saldırısı tespitini başarılı bir şekilde gerçekleştirdiğini göstermiştir. Bu nedenle önerilen mimari pratik olarak gerçek dünya uygulamalarında ve senaryolarında da kullanılabilecektir.

Önerilen modelin, yeni çıkacak GPU, işlemci, TPU teknolojili iş istasyonlarına bağlı olarak daha hızlı sonuçlar vereceği düşünülmektedir.

Ayrıca web saldırılarına yönelik yeni veri setleri hazırlanmaktadır. Bu veri setleri üzerinde de benzer başarılar elde edileceği düşünülmektedir.

Yeni gelişmeler göz önüne alınarak önerilen modellerin geliştirilmesi ve daha iyi başarı oranlarının elde edilmesi hedeflenmektedir.

KAYNAKLAR

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Ahmad, F. (2017). *Fwaf-Machine-Learning-driven-Web-Application-Firewall*. https://github.com/faizann24/Fwaf-Machine-Learning-driven-Web-Application-Firewall

Aizawa, A. (2003). *An information-theoretic perspective of tf – idf measures q*. *39*, 45–65.

Alammar, J. (2020). *(BERT): A Visual Guide to Using BERT for the First Time – Jay Alammar – Visualizing machine learning one concept at a time.* https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Alaoui, R. L., & Nfaoui, E. H. (2022). Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review. *Future Internet*, *14*(4). https://doi.org/10.3390/fi14040118

Alvares, J. L. (2021). *Malware Classification with BERT* [San José State University]. https://doi.org/10.31979/etd.7n35-garb

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The 'K'in K-fold cross validation. *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 441–446.

Anonymous. (2020). *GitHub - Morzeux/HttpParamsDataset: Dataset contains several benign and attacks samples which can be used as values in HTTP protocol.* https://github.com/Morzeux/HttpParamsDataset

Avcı, İ., Koca, M., & Atasoy, M. (2021). Windows Tabanlı Uygulamalarda SQL Enjeksiyon Siber Saldırı Senaryosu ve Güvenlik Önlemleri. *European Journal of Science and Technology*, *28*, 213–219. https://doi.org/10.31590/ejosat.995697

Bahdanau, D., Cho, K., & Bengio, Y. (2015). *Neural Machine Trasnlation*. 1–15.

Bakour, K., Murat Ünver, H., & Ghanem, R. (2019). A Deep Camouflage: Evaluating Android's Anti-malware Systems Robustness Against Hybridization of Obfuscation Techniques with Injection Attacks. *Arabian Journal for Science and Engineering*, *44*(3), 9333–9347. https://doi.org/10.1007/s13369-019-04081-5

Baykara, M., Daş, R., & Tuna, G. (2016). *Web Sunucu Erişim Kütüklerinden Web Ataklarının Tespitine Yönelik Web Tabanlı Log Analiz Platformu*. *28*(2), 291–302.

Bengio, Y., Bordes, A., & Glorot, X. (2011). Deep Sparse Rectifier Neural Networks. *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, *34*(7), 1114–1118. https://doi.org/10.1002/ecs2.1832

Bishnoi, S., Mohanty, S., & Sahoo, B. (2021a). *A Deep Learning-Based Methodology*

in Fog Environment for DDOS Attack Detection. *Iccmc*, 201–206. https://doi.org/10.1109/iccmc51019.2021.9418363

Bishnoi, S., Mohanty, S., & Sahoo, B. (2021b). A Deep Learning-Based Methodology in Fog Environment for DDOS Attack Detection. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, *Iccmc*, 201–206. https://doi.org/10.1109/ICCMC51019.2021.9418363

Case, T., Request, C., Calzavara, S., Ca, U., & Venezia, F. (2020). *Machine Learning for Web Vulnerability Detection*. *June*.

Cassel, M., & Kastensmidt, F. L. (2006). *Evaluating One-Hot Encoding Finite State Machines for SEU Reliability in SRAM-based FPGAs*.

Cavnar, W. B., Trenkle, J. M., & Mi, A. A. (1994). *N-Gram-Based Text Categorization*.

Chordiya, A. R., Majumder, S., & Javaid, A. Y. (2018). Man-in-the-Middle (MITM) Attack Based Hijacking of HTTP Traffic Using Open Source Tools. *IEEE International Conference on Electro Information Technology*, *2018-May*, 438–443. https://doi.org/10.1109/EIT.2018.8500144

Cieri, C., Liberman, M., Cho, S., Strassel, S., Fiumara, J., & Wright, J. (2022). *Reflections on 30 Years of Language Resource Development and Sharing*. 20–25. https://catalog.ldc.upenn.edu

Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3504–3514.

Demirol, D., Daş, R., & Baykara, M. (2013). *SQL Enjeksiyon Saldırılarına Karşı Güvenlik*. *May*.

Devlin, J., Kenton, M. C., & Kristina, L. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Mlm*.

Dong, Y., Zhang, Y., Ma, H., Wu, Q., Liu, Q., Wang, K., & Wang, W. (2018). *An adaptive system for detecting malicious queries in web attacks*. *61*(March), 1–16.

Durai, K. N., Subha, R., & Haldorai, A. (2021). A Novel Method to Detect and Prevent SQLIA Using Ontology to Cloud Web Security. *Wireless Personal Communications*, *117*(4), 2995–3014. https://doi.org/10.1007/s11277-020-07243-z

Fidalgo, A., Medeiros, I., Antunes, P., & Neves, N. (y.y.). Towards a Deep Learning Model for Vulnerability Detection on Web Application Variants. *Proceedings - 2020 IEEE 13th International Conference on Software Testing, Verification and Validation Workshops,*. https://doi.org/10.1109/ICSTW50294.2020.00083

Galke, L. (2022). *Bag-of-Words vs . Graph vs . Sequence in Text Classification : Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP*. *1*, 4038–4051.

Gaurav, A., Santaniello, D., Gupta, A. K., & Colace, F. (2022). *A Biblometric review of the Current State and Future Perspectives of XSS attack detection in Web based Applications*. *May*. https://doi.org/10.13140/RG.2.2.19829.65763

Geffet, M., & Dagan, I. (2004). Feature vector quality and distributional similarity. *COLING 2004: Proceedings of the 20th International Conference on*

Computational Linguistics, 247–253.

Giménez, C. T., Villegas, A. P., & Marañón, G. Á. (2010). *HTTP data set CSIC 2010*. Information Security Institute of CSIC (Spanish Research National Council).

Gong, X., Lu, J., Zhou, Y., Qiu, H., & He, R. (2020). Model Uncertainty Based Annotation Error Fixing for Web Attack Detection. *Journal of Signal Processing Systems*. https://doi.org/10.1007/s11265-019-01494-1

Gong, X., Lu, J., Zhou, Y., Qiu, H., & He, R. (2021). *Model Uncertainty Based Annotation Error Fixing for Web Attack Detection*. 187–199.

Gong, X., Zhou, Y., Bi, Y., He, M., Sheng, S., Qiu, H., He, R., & Lu, J. (2019). *Estimating Web Attack Detection via Model Uncertainty from Inaccurate Annotation*. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00019

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. Içinde Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Ed.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

Hadpawat, T., & Vaya, D. (2017). Analysis of Prevention of XSS Attacks at Client Side Formulation Of A Fast And Hardware-Efficient Visual Cryptography Scheme For Images View project Analysis of Prevention of XSS Attacks at Client Side. *Article in International Journal of Computer Applications*, *173*. https://doi.org/10.5120/ijca2017915344

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*. https://doi.org/10.1162/neco.2006.18.7.1527

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Hoang, X. D. (2019). A website defacement detection method based on machine learning. *Lecture Notes in Networks and Systems*, *63*, 116–124. https://doi.org/10.1007/978-3-030-04792-4_17

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, *6*. https://doi.org/10.1142/S0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, *79*. https://doi.org/10.1073/pnas.79.8.2554

Hutchins, J. (1997). A Chronology. Içinde *Machine Translation* (C. 12). Kluwer Academic Publishers.

Iqsyahiro Kresna, A., & Rosmansyah, Y. (2018). Web server farm design using Personal Computer (PC) Desktop. *Proceedings of 2018 10th International Conference on Information Technology and Electrical Engineering: Smart Technology for Better Society*. https://doi.org/10.1109/ICITEED.2018.8534920

Ivakhnenko, A. G. (1971). Polynomial Theory of Complex Systems. *IEEE*

Transactions on Systems, Man and Cybernetics, *1*(4), 364–378. https://doi.org/10.1109/TSMC.1971.4308320

Japkowicz, N., Hanson, S. J., & Gluck, M. A. (2000). Nonlinear autoassociation is not equivalent to PCA. *Neural computation*, *12*(3), 531–545.

Karaarslan, E. (2008). *Web Saldırı Saptama Sistemlerinin Etkinleştirilmesi İçim Sistem Farkındalığı ve Çok Katmanlı Güvenlik Önlemlerinin Gerçekleştirilmesi*.

Kaytan, M. (2016). *Web Tabanlı Oltalama Saldırılarının Makine Öğrenmesi Yöntemleri ile Tespiti*. İnönü Üniversitesi, Fen Bilimleri Enstitüsü.

Komiya, R., Paik, I., & Hisada, M. (2011). Classification of malicious web code by machine learning. *Proceedings of 2011 3rd ICAST 2011*. https://doi.org/10.1109/ICAwST.2011.6163109

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Içinde F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinberger (Ed.), *Advances in Neural Information Processing Systems* (C. 25). Curran Associates, Inc.

Kshirsagar, D., & Kumar, S. (2022). Towards an intrusion detection system for detecting web attacks based on an ensemble of filter feature selection techniques. *Cyber-Physical Systems*, 1–16. https://doi.org/10.1080/23335777.2021.2023651

Kuipers, M., & Prasad, R. (2022). *Journey of Artificial Intelligence*. *123*, 3275–3290. https://doi.org/10.1007/s11277-021-09288-0

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten Digit Recognition with a Back-Propagation Network. Içinde D. Touretzky (Ed.), *Advances in Neural Information Processing Systems* (C. 2). Morgan-Kaufmann.

Li, K., Fei-Fei, L., & Deng, J. (2009). ImageNet: Constructing a large-scale image database. *Journal of Vision*, *9*(8), 1037–1037. https://doi.org/10.1167/9.8.1037

Li, L., Guo, Q., Xue, X., & Qiu, X. (2019). *BERT-ATTACK: Adversarial Attack Against BERT Using BERT*.

Liang, J., Zhao, W., & Ye, W. (2017). *Anomaly-Based Web Attack Detection*. 80–85. https://doi.org/10.1145/3171592.3171594

Liu, C., Yang, J., & Wu, J. (2020). Web intrusion detection system combined with feature analysis and SVM optimization. *Eurasip Journal on Wireless Communications and Networking*, *2020*(1), 1–9. https://doi.org/10.1186/s13638-019-1591-1

Loye, G. (2019). *Attention Mechanism*. https://blog.floydhub.com/attention-mechanism/

Luo, T. T., Wong, H., & Luo, T. (2020). Man-in-the-Middle Attacks on MQTT-based IoT Using BERT Based Adversarial Message Generation. *KDD'20 workshop*, *November*, 1–6. https://github.com/HenryCWong/adversarialBERTMessages

Luxemburk, J., Hynek, K., & Cejka, T. (2021). Detection of HTTPS Brute-Force Attacks with Packet-Level Feature Set. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, 114–122. https://doi.org/10.1109/CCWC51732.2021.9375998

Mac, H., Khoa, B., & Centre, C. (2018). *Detecting Attacks on Web Applications using Autoencoder*. 416–421.

Mcculloch, W. S., & Pitts, W. (1943). A Logical Calculus Of The Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, *5*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Distributed Representations of Words and Phrases and their Compositionality*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2019). Advances in pre-training distributed word representations. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, *1*, 52–55.

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons*.

Montes, N., Betarte, G., Martínez, R., & Pardo, A. (2021). Web Application Attacks Detection Using Deep Learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12702 LNCS*, 227–236. https://doi.org/10.1007/978-3-030-93420-0_22

Ng, A. Y., Raina, R., & Madhavan, A. (2009). Large-scale deep unsupervised learning using graphics processors. *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, 873–880.

Oxford, R. M., & Daniel, L. G. (2001). Basic Cross-Validation: Using the" Holdout" Method To Assess the Generalizability of Results. *Research in the Schools*, *8*(1), 83–89.

Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D. C., Staples, J., & Krause, L. (2019). Detecting web attacks with end-to-end deep learning. *Journal of Internet Services and Applications*, *10*(1). https://doi.org/10.1186/s13174-019-0115-x

Pang, G., Shen, C., Cao, L., & Hengel, A. Vanden. (2021). *Deep Learning for Anomaly Detection : A Review*. *54*(2).

Park, Y. S., & Lek, S. (2016). Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. Içinde *Developments in Environmental Modelling* (C. 28). Elsevier. https://doi.org/10.1016/B978-0-444-63623-2.00007-4

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. 1532–1543. http://nlp.

Qaiser, S., & Ali, R. (2018). *Text Mining : Use of TF-IDF to Examine the Relevance of Words to Documents Text Mining : Use of TF-IDF to Examine the Relevance of Words to Documents*. *July*. https://doi.org/10.5120/ijca2018917395

Rahman, A., & Ahmed, B. (2020). *Analyzing Web Application Vulnerabilities : An Empirical Study on E-Commerce Sector in Bangladesh*. 5–10.

Refaeilzadeh, P., Tang, L., Liu, H., Angeles, L., & Scientist, C. D. (2020). Encyclopedia of Database Systems. *Encyclopedia of Database Systems*. https://doi.org/10.1007/978-1-4899-7993-3

Rojas-Galeano, S. (2021). *Using BERT Encoding to Tackle the Mad-lib Attack in SMS*

Spam Detection. http://arxiv.org/abs/2107.06400

Roy, A., & Pan, S. (2021). *Incorporating medical knowledge in BERT for clinical relation extraction*.

Roy, P., & Rani, P. (2022). *SQL Injection Attack Detection by Machine Learning Classifier*. *Icaaic*, 394–400.

Saidu, M., Imran, A., Kashif, G., Qureshi, N., & Fo, M. (2020). *An algorithm for detecting SQL injection vulnerability using black-box testing*. 249–266. https://doi.org/10.1007/s12652-019-01235-z

Samy, A. (2020). *Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning*. 74571–74585. https://doi.org/10.1109/ACCESS.2020.2988854

Şanlıöz, Ş. G., Kara, M., Aydın, M. A., & Balık, H. H. (2019). *Makine Öğrenmesi Yöntemleri İle Oltalama Websitesi Saldırı Tespiti Attack Detection of Web Phishing With Machine Learning Methods*.

Sevri, M. (2016). *Web Saldırılarının Tespitine Yönelikr Yapay Zeka Tabanlı Bir Güvenlik Modülü Geliştirilmesi*. Gazi Üniversitesi, Bilişim Enstitüsü.

Seyyar, Y. E., Yavuz, A. G., & Unver, H. M. (2022). An Attack Detection Framework Based on BERT and Deep Learning. *IEEE Access*, *July*, 1–1. https://doi.org/10.1109/access.2022.3185748

Shahid, W. Bin, Aslam, B., Abbas, H., Afzal, H., & Khalid, S. Bin. (2022). A deep learning assisted personalized deception system for countering web application attacks. *Journal of Information Security and Applications*, *67*(April), 103169. https://doi.org/10.1016/j.jisa.2022.103169

Shahid, W. Bin, Aslam, B., Abbas, H., Khalid, S. Bin, & Afzal, H. (2022). An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling. *Journal of Network and Computer Applications*, *198*(November 2021), 103270. https://doi.org/10.1016/j.jnca.2021.103270

Shar, L. K., & Tan, H. B. K. (2012). Predicting common web application vulnerabilities from input validation and sanitization code patterns. *2012 27th IEEE/ACM International Conference on Automated Software Engineering, ASE 2012 - Proceedings*, 310–313. https://doi.org/10.1145/2351676.2351733

Smolensky Paul. (1986). Information Processing in Dynamical Systems: Foundations of Harmony Theory. *Journal of Japan Society for Fuzzy Theory and Systems*, *4*(2), 220–228.

Song, X. (2021, Aralık). *A Fast WordPiece Tokenization System*. Google AI Blog. https://ai.googleblog.com/2021/12/a-fast-wordpiece-tokenization-system.html

Stency, V. S., & Mohanasundaram, N. (2021). A Study on XSS Attacks: Intelligent Detection Methods. *Journal of Physics: Conference Series*, *1767*, 12047. https://doi.org/10.1088/1742-6596/1767/1/012047

Stevens, E., Antiga, L., & Viehmann, T. (2020). *Deep learning with PyTorch*. Manning Publications.

Tekerek, A. (2021). A novel architecture for web-based attack detection using convolutional neural network. *Computers & Security*, *100*, 102096. https://doi.org/10.1016/j.cose.2020.102096

Tekerek, A., Gemci, C., & Bay, Ö. F. (2016). *Web Tabanlı Saldırı Önleme Sistemi Tasarımı ve Web tabanlı saldırı önleme sistemi tasarımı ve gerçekleştirilmesi : yeni bir hibrit model*. https://doi.org/10.17341/gummfd.63355

Tian, Y., Wang, J., Zhou, Z., & Zhou, S. (2017). *CNN-Webshell*. 75–79. https://doi.org/10.1145/3171592.3171593

Tian, Z., Luo, C., Qiu, J., Du, X., & Guizani, M. (2020). A Distributed Deep Learning System for Web Attack Detection on Edge Devices. *IEEE Transactions on Industrial Informatics*, *16*. https://doi.org/10.1109/TII.2019.2938778

Turing, A. M. (2012). Computing machinery and intelligence. Içinde *Machine Intelligence: Perspectives on the Computational Model* (ss. 1–28). https://doi.org/10.1525/9780520318267-013

Werbos, P. (1974). Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

Wichers, D. (2013). OWASP Top-10 2013. *OWASP Foundation*. https://wiki.owasp.org/

Wichers, D. (2017). OWASP Top-10 2017. *OWASP Foundation, February*. https://wiki.owasp.org/

Yu, Lean, Zhou, R., Chen, R., Lai, K. K., & Yu, L. (2020). Missing Data Preprocessing in Credit Classification : One-Hot Encoding or Imputation. *Emerging Markets Finance and Trade*. https://doi.org/10.1080/1540496X.2020.1825935

Yu, Lu, Luo, S., & Pan, L. (2019). *Detecting SQL Injection Attacks based on Text Analysis*. *90*(Iccia), 95–101. https://doi.org/10.2991/iccia-19.2019.14

Zhang, S., Yao, L., & Xu, X. (2017). Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders. *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 957–960.

ÖZGEÇMİŞ

Adı Soyadı : Yunus Emre SEYYAR

Yabancı Dil : İngilizce

Eğitim Durumu :

Lisans : Erciyes Üniversitesi - 2008

Yüksek Lisans : Kırıkkale Üniversitesi - 2013

Çalıştığı Kurum/Kurumlar ve Yıl/Yıllar :

: TÜBİTAK 2012 - Halen

: Vakıfbank 2009 - 2012

: Nurol Teknoloji A.Ş. 2007 - 2009

Yayınları (SCI) :

1. Y. E. Seyyar, A. G. Yavuz and H. M. Ünver, "An Attack Detection Framework Based on BERT and Deep Learning," in IEEE Access, vol. 10, pp. 68633-68644, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3185748.

Bildiriler :

Y. E. Seyyar, A. G. Yavuz ve H. M. Ünver "Web Saldırılarının BERT Modeli Kullanılarak Tespit Edilmesi", SİU, 30. Sinyal İşlemeleri ve İletişim Uygulamaları Kurultayı, 2022

Araştırma Alanları : Siber Güvenlik, Derin Öğrenme, Metin Sınıflandırma