
GAZİ ÜNİVERSİTESİ
Gazi
1926

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK KİMLİK AVI SALDIRILARI DENETİMİ

Öznur Şifa AKÇAM

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAYIS 2025

ETİK BEYAN

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

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

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

Öznur Şifa AKÇAM

05/05/2025

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK KİMLİK AVI SALDIRILARI DENETİMİ

(Yüksek Lisans Tezi)

Öznur Şifa AKÇAM

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2025

ÖZET

Kimlik avı saldırıları, bilgi sistemlerindeki güvenlik açıklarını istismar ederek önemli bilgilerin ele geçirilmesini sağlar. Bu tez çalışması, URL tabanlı kimlik avı saldırılarını tespit etmeyi amaçlamıştır. Bu amaç doğrultusunda, karakter ve kelime tabanlı öznitelik çıkarımı kullanan bir derin öğrenme modeli geliştirilmiştir. Geliştirilen model, URL adreslerini kimlik avı/oltalama (phishing) ve yasal (legitimate) şeklinde sınıflandırmıştır. Önerilen modelde Çift Yönlü Uzun Kısa Süreli Bellek (BiLSTM) algoritması kullanılmıştır. Modelin geliştirilme sürecinde GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setleri üzerinde deneyler gerçekleştirilmiş; çapraz doğrulama performansı ise Mendeley Data Web Page Phishing Detection veri seti ile değerlendirilmiştir. Elde edilen deneysel bulgular, modelin yüksek doğruluk oranlarına sahip olduğunu göstermiştir. GramBeddings veri setinde %98,24 doğruluk ve 0,9977 AUC, Malicious and Benign URLs veri setinde %99,32 doğruluk ve 0,9986 AUC, Ebbu2017 Phishing veri setinde %98,34 doğruluk ve 0,9981 AUC, Mendeley Data Web Page Phishing Detection veri setinde ise %90,33 doğruluk ve 0,9694 AUC elde edilmiştir. Modelin benzersizliği, URL'lerin karakter tabanlı çıkarımlar ile yapısal desenlerini analiz etmesi ve kelime tabanlı çıkarımlar ile bağlamsal anlamlarını değerlendirmesidir. Bu sayede, yapısal ve anlamsal öznitelikler bir araya getirilerek güçlü ve ayrıntılı bir temsil sağlanmıştır. Birden fazla veri seti ile gerçekleştirilen deneyler, modelin yüksek doğruluk ve AUC skorları ile üstün kimlik avı tespit yeteneklerini ortaya koyduğunu göstermiştir. Ayrıca, manuel özellik çıkarımı ve uzman müdahalesine ihtiyaç duymadan, tamamen otomatik ve dil bağımsız bir sınıflandırma yaklaşımı sunulmuştur. Modelin, kısaltılmış URL'ler de dahil olmak üzere farklı URL türlerini tespit edebilmesi, sağlamlık ve esneklik açısından önemli bir avantaj sağlamıştır. Bu çalışma, yapısal ve anlamsal analizleri birleştirerek güçlü, bağımsız ve güvenilir bir kimlik avı tespit modeli ortaya koymuştur. Elde edilen bulgular, önerilen yaklaşımın siber güvenlik alanında etkili bir çözüm sunduğunu ve kimlik avı saldırılarının tespitinde yüksek performans gösterdiğini kanıtlar niteliktedir.

Bilim Kodu : 92438

Anahtar Kelimeler : BiLSTM, karakter tabanlı öznitelik çıkarımı, derin öğrenme, kimlik avı tespiti, skip-gram, kelime tabanlı öznitelik çıkarımı

Sayfa Adedi : 77

Danışman : Doç. Dr. Adem TEKEREK

PHISHING ATTACKS DETECTION USING DEEP LEARNING METHODS

(M. Sc. Thesis)

Öznur Şifa AKÇAM

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

May 2025

ABSTRACT

Phishing attacks exploit security vulnerabilities in information systems to compromise important information. This thesis aims to detect URL-based phishing attacks. For this purpose, a deep learning model using character and word-based feature extraction was developed. The developed model classifies URL addresses as phishing and legitimate. A Bidirectional Long Short-Term Memory (BiLSTM) algorithm is used in the proposed model. During the development of the model, experiments were conducted on GramBeddings, Malicious and Benign URLs, and Ebbu2017 Phishing datasets. Cross-validation performance was evaluated with the Mendeley Data Web Page Phishing Detection dataset. The experimental results show that the model has high accuracy rates. The model achieved 98.24% accuracy and 0.9977 AUC on the GramBeddings dataset, 99.32% accuracy and 0.9986 AUC on the Malicious and Benign URLs dataset, 98.34% accuracy and 0.9981 AUC on the Ebbu2017 Phishing dataset, and 90.33% accuracy and 0.9694 AUC on Mendeley Data Web Page Phishing Detection dataset. The model's uniqueness is that it analyses the structural patterns of URLs with character-based inference and evaluates their contextual meaning with word-based inference. This way, structural and semantic features are combined to provide a robust and detailed representation. Experiments with multiple datasets have shown that the model demonstrates superior phishing detection capabilities with high accuracy and AUC scores. Furthermore, a fully automatic and language-independent classification approach is presented without requiring manual feature extraction and expert intervention. The ability of the model to detect different URL types, including shortened URLs, provided a significant advantage in terms of robustness and flexibility. This study has demonstrated a robust, independent, reliable phishing detection model by combining structural and semantic analyses. The results prove that the proposed approach provides an effective solution in cyber security and shows high performance in detecting phishing attacks.

Science Code : 92438
Key Words : BiLSTM, character-based extraction, deep learning, phishing detection, skip-gram, word-based extraction
Page Number : 77
Supervisor : Assoc. Prof. Dr. Adem TEKEREK

TEŞEKKÜR

Bu tezin hazırlanma sürecinde, bilgi ve deneyimleriyle bana rehberlik eden, yolumu aydınlatan ve her aşamada desteğini esirgemeyen saygıdeğer danışman hocam Doç. Dr. Adem TEKEREK'e teşekkürlerimi sunarım. Kendisinin rehberliği ve yönlendirmeleri, çalışmamın her aşamasında büyük katkı sağlamıştır. Ayrıca, akademik gelişimime ve kişisel yolculuğuma değerli fikirleri ve desteğiyle katkıda bulunan kıymetli hocam Prof. Dr. Mehmet TEKEREK'e de teşekkür ederim. Onun yol göstericiliği, bu sürecin benim için daha öğretici ve verimli olmasını sağlamıştır. Bu süreçte daima yanımda olan, sevgisi ve desteğiyle bana güç veren aileme – anneme, babama ve ablama – sonsuz minnettarlığımı sunarım. Onların desteği, yorulduğumda yeniden güç bulmamı ve yoluma devam etmemi sağlamıştır. Son olarak, bu süreçte her zaman yanımda olan ve beni destekleyen değerli arkadaşım Fulya KAVAK'a teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. LİTERATÜR ARAŞTIRMASI 7

2.1. Phishing Saldırı Döngüsü 7

2.2. İlgili Çalışmalar 9

2.3. Araştırma Boşlukları 12

3. MATERYAL VE METOT 15

3.1. Problemin Matematiksel Tanımı 15

3.2. Veri Setleri 16

3.3. Algoritmalar 17

3.3.1. Word2Vec 17

3.3.2. LSTM 19

3.3.3. BiLSTM 21

3.3.4. CNN 21

3.3.5. DNN 23

4. ÖNERİLEN YÖNTEM 25

4.1. Veri Ön İşleme 25

Sayfa

4.2. Öznitelik Çıkarımı 27
4.2.1. Karakterizasyon 28
4.2.2. Karakter gömme katmanı 30
4.2.3. Tokenizasyon 30
4.2.4. Kelime gömme katmanı 33
4.3. Önerilen Model 34
4.3.1. BiLSTM katmanı 36
4.3.2. Dikkat (Attention) katmanı 36
4.3.3. Maksimum havuzlama (Max pooling) 38
4.3.4. Düğüm bırakma (Dropout) 39
4.3.5. Birleştirme ve sınıflandırma 39
4.4. Eğitim-Doğrulama-Test 40
5. DENEYSEL BULGULAR 43
5.1. Sınıflandırma Metrikleri 43
5.2. Deneyler 45
5.2.1. En iyi optimizasyon algoritmasının seçimi 45
5.2.2. Karakter tabanlı öznitelikler için algoritma seçimi 47
5.2.3. Kelime tabanlı öznitelikler için algoritma seçimi 48
5.2.4. En uygun URL uzunluğunun belirlenmesi 50
5.2.5. BiLSTM nöron sayısının seçimi 51
5.2.6. Sınıflandırma metriklerinin sonuçları 53
5.3. URL Kısaltma 58
5.4. Birleşik Tokenizasyon 60
5.5. Maliyet Analizi 61

Sayfa

5.6. Çapraz Doğrulama 62

6. TARTIŞMA 63

6.1. Modelin Güçlü Yönleri, Sınırlamaları ve Gelecek Çalışmalar 67

7. SONUÇ VE ÖNERİLER 71

KAYNAKLAR 73

ÖZGEÇMİŞ 77

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. Veri setlerinin istatistiksel analizi 17

Çizelge 4.1. Karakter sözlüğü gösterimi..... 28

Çizelge 4.2. Veri setlerindeki belirteçlerin istatistiksel analizi..... 31

Çizelge 4.3. Veri setleri için kelime sözlüğü gösterimi..... 33

Çizelge 4.4. Algoritma 1..... 34

Çizelge 5.1. Modelde kullanılacak optimizasyon algoritmasının seçimi 46

Çizelge 5.2. Karakter tabanlı özniteliklerin eğitimi için algoritma seçimi 47

Çizelge 5.3. Word2Vec modelinde skip-gram algoritması kullanılarak kelime tabanlı özniteliklerin eğitimi için algoritma seçimi..... 48

Çizelge 5.4. Word2Vec modelinde CBoW algoritması kullanılarak kelime tabanlı özniteliklerin eğitimi için algoritma seçimi..... 49

Çizelge 5.5. Veri setlerine göre uygun URL uzunluğunun belirlenmesi..... 51

Çizelge 5.6. Önerilen model için BiLSTM nöron sayısının belirlenmesi 52

Çizelge 5.7. Önerilen modelin farklı URL uzunluklarındaki sınıflandırma metrik değerleri 57

Çizelge 5.8. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerindeki kısaltılmış URL sayılarına dair istatistiksel veriler 59

Çizelge 5.9. Önerilen modelin kısaltılmış URL veri seti üzerindeki sınıflandırma metrik değerleri 59

Çizelge 5.10. Birleşik tokenizasyon yöntemi kullanılarak elde edilen sınıflandırma metrik değerleri 60

Çizelge 5.11. Veri setlerine göre model parametreleri ve bellek kullanımı için maliyet analizi..... 61

Çizelge 5.12. Önerilen modelin çapraz doğrulama ve tokenizasyon yöntemleri ile karşılaştırılması..... 62

Çizelge 6.1. Önerilen modelin literatürdeki mevcut çalışmalarla karşılaştırılması..... 63

Çizelge 6.2. Önerilen modelin literatürdeki mevcut çalışmalarla karşılaştırılması..... 64

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Eylül 2022 - Eylül 2024 döneminde tespit edilen kimlik avı saldırılarının aylık dağılımı 2

Şekil 1.2. Örnek URL adresi 4

Şekil 1.3. Tezin yapısını gösteren organizasyon şeması 6

Şekil 2.1. Phishing saldırı döngüsü 8

Şekil 3.1. Word2Vec algoritması: CBoW ve skip-gram 18

Şekil 3.2. LSTM mimarisi 20

Şekil 3.3. BiLSTM mimarisi 21

Şekil 3.4. CNN mimarisi 22

Şekil 3.5. Maksimum havuzlama işlemi 22

Şekil 3.6. DNN mimarisi 24

Şekil 4.1. GramBeddings veri setindeki URL uzunluklarının dağılımı 26

Şekil 4.2. Malicious and Benign URLs veri setindeki URL uzunluklarının dağılımı 26

Şekil 4.3. Ebbu2017 Phishing veri setindeki URL uzunluklarının dağılımı 27

Şekil 4.4. Mendeley Data Web Page Phishing Detection veri setindeki URL uzunluklarının dağılımı 27

Şekil 4.5. Karakter sözlüğünün karakter bulutu olarak görselleştirilmesi 28

Şekil 4.6. Karakterizasyon yaklaşımı 29

Şekil 4.7. GramBeddings veri setindeki belirteçlerin kelime bulutu olarak görselleştirilmesi 31

Şekil 4.8. Tokenizasyon yaklaşımı 32

Şekil 4.9. Önerilen modelin mimarisi 37

Şekil 4.10. GramBeddings veri setinde önerilen modelin katmanları 39

Şekil 4.11. Veri setlerinin eğitim, doğrulama ve test olarak ayrılması 41

Şekil **Sayfa**

Şekil 5.1. Önerilen modelin GramBeddings veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri 53

Şekil 5.2. Önerilen modelin Malicious and Benign URLs veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri 54

Şekil 5.3. Önerilen modelin Ebbu2017 Phishing veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri 55

Şekil 5.4. GramBeddings veri seti karışıklık matrisi 55

Şekil 5.5. Malicious and Benign URLs veri seti karışıklık matrisi 56

Şekil 5.6. Ebbu2017 Phishing veri seti karışıklık matrisi 56

Şekil 5.7. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerine ait ROC eğrileri ve AUC değerleri 58

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
s	saniye
MB	Megabyte

Kısaltmalar	**Açıklamalar**
Adam	Adaptive Moment Estimation
AOL	American Online
APWG	Anti-Phishing Working Group
AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
BiGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-Term Memory
BoW	Bag of Words
BPE	Byte Pair Encoding
CBoW	Continuous Bag of Words
CNN	Convolutional Neural Networks
DNN	Deep Neural Network
FBI	Federal Bureau of Investigation
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GAN	Generative Adversarial Network
GloVe	Global Vectors for Word Representation

Kısaltmalar	Açıklamalar
GRU	Gate Recurrent Unit
HTML	HyperText Markup Language
IC3	Internet Crime Complaint Centre
kNN	k-Nearest Neighbour
LSTM	Long Short-Term Memory
N/A	Not Available
NLP	Natural Language Processing
RMSprop	Root Mean Square Propagation
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RQs	Research Questions
SGD	Stochastic Gradient Descent
SMS	Short Message Service
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
URL	Uniform Resource Locator
Word2Vec	Word to Vector

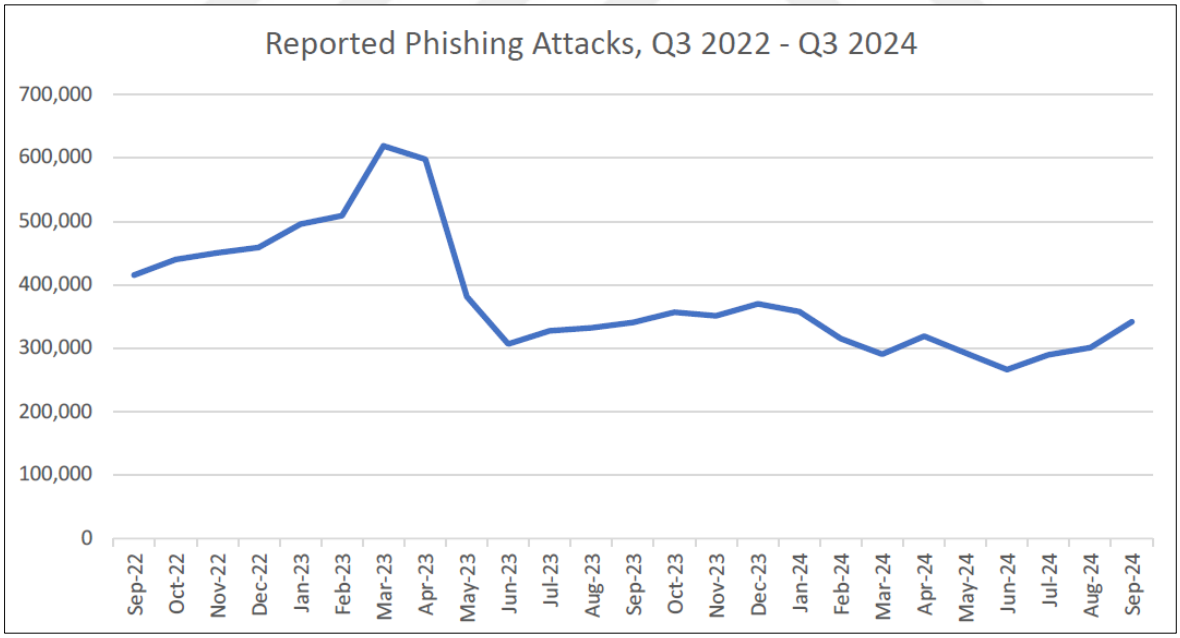
1. GİRİŞ

İnternetin yaygın kullanımı, çeşitli cihazları birbirine bağlayan ve aralarında iletişim kurulmasını sağlayan ağ teknolojisinin gelişmesinden kaynaklanır. Ağ, bireyler, kurumlar ve devletler arasındaki iletişimi kolaylaştırarak hızla artan miktarda veri aktarımına neden olur. Veriler aktarılırken aynı zamanda ağ üzerinde depolanır [1]. Ocak 2024 We Are Social Raporu'na göre, dünya genelinde 5,35 milyar kişi aktif internet kullanıcısıdır. Bu sayı, dünya nüfusunun %66,2'sinin verilerinin ağ üzerinde depolandığını gösterir. Depolanan verilerin çoğu bireylere ve kuruluşlara ait gizli bilgileri içerir [2]. Bu bilgiler kötü niyetli, saldırgan kişiler tarafından kullanılmak istenir. Saldırganlar, internet kullanıcılarına yönelik çeşitli siber saldırılar gerçekleştirerek, şifre ve banka hesap bilgileri de dahil olmak üzere kullanıcıların gizli bilgilerine erişim sağlar ve bu bilgileri kendi çıkarları için kullanır. Bu saldırılar bireylere ve kuruluşlara ciddi ekonomik zararlar verir [3, 4].

Federal Soruşturma Bürosu (Federal Bureau of Investigation - FBI) İnternet Suçları 2023 raporunda, İnternet Suçları Şikayet Merkezi'nin (Internet Crime Complaint Centre - IC3) 2019-2023 yılları arasında ortalama 758 000 şikayet aldığı belirtilmiştir. Bu şikâyetlerin büyük çoğunluğu dünya genelinde bireyleri etkileyen internet dolandırıcılığı ile ilgilidir. İnternet Suçları 2023 raporuna göre, 2023 yılında IC3'e 880 418 şikâyet yapılmış ve bu şikâyetler 12,5 milyar dolar zarara sebep olmuştur. Bu şikâyetlerin 298 878'i kimlik avı/oltalama (phishing) saldırılarılarına yöneliktir [5]. Kimlik avı/oltalama saldırıları, banka, okul ve işyerleri gibi yasal kurumlardan geliyormuş gibi görünen sahte e-posta, kısa mesaj hizmeti (Short Message Service - SMS) ve web sitesi bağlantılarının kullanılmasını içerir. Saldırganlar kurban olarak seçtikleri kişilere sahte e-mail, SMS ve web sitesi bağlantısı gönderirler. Gönderilen iletiyi alan kullanıcılar bağlantıya tıkladıkları zaman saldırganlar kurbanların şifreleri, banka hesap bilgileri gibi kişisel bilgilerine veya kurumlara ait gizli bilgilere erişebilirler. Burada e-mail ve SMS gibi iletiler yem olarak kullanılır ve kullanıcılar tuzağa düşürülür. Bu durum kimlik avı/oltalama saldırıları olarak ifade edilir [6, 7]. Kimlik avı saldırıları, kullanıcıları hassas bilgiler vermeleri için kandırarak kimlik hırsızlığı ve finansal dolandırıcılığa karşı savunmasız hale getirir [8].

Kimlik avı saldırıları, ilk saldırının 2 Ocak 1996 tarihinde bir internet hizmetleri sağlayıcısı olan American Online'a (AOL) karşı gerçekleştirilmesinden bu yana artış göstermektedir

[9]. Anti-Phishing Çalışma Grubu'nun (Anti Phishing Working Group - APWG) 2024 yılı 3. çeyrek raporuna göre, Temmuz, Ağustos ve Eylül aylarında toplam 932 923 kimlik avı saldırısı tespit edilmiştir. Bu rakam, 2024'ün 2. çeyreğinde kaydedilen 877 536 saldırıya kıyasla belirgin bir artış göstermiştir. Özellikle Eylül 2024'te tespit edilen saldırı sayısı, önceki aylara göre en yüksek seviyeye ulaşmıştır. APWG'nin raporuna göre, oltalama saldırılarının eğilimi Haziran 2023'ten bu yana aylık 290 000 ile 370 000 arasında değişmektedir [10]. Şekil 1.1'de Eylül 2022'den Eylül 2024'e kadar gerçekleşen kimlik avı saldırılarının sayısını gösteren bir grafiğe yer verilmiştir. Şekil 1.1'de de görüldüğü gibi, son yıllarda kimlik avı saldırılarında genel bir artış eğilimi gözlemlenmiş, özellikle 2023'ün ortalarında yaşanan keskin düşüşün ardından saldırılar tekrar yükselişe geçmiştir. APWG tarafından yayınlanan rapor, kimlik avı/oltalama saldırılarının görülme sıklığının endişe verici bir oranda arttığını ve internet kullanıcıları için tehlike oluşturduğunu kanıtlar niteliktedir. Kişisel ve gizli bilgilerin yetkisiz erişimden korunması için, bireyleri kandırmaya yönelik gönderilen iletilerin tespit edilmesi ve oltalama saldırılarının önlenmesi önemlidir [11].



Şekil 1.1. Eylül 2022 - Eylül 2024 döneminde tespit edilen kimlik avı saldırılarının aylık dağılımı (Bkz. [10])

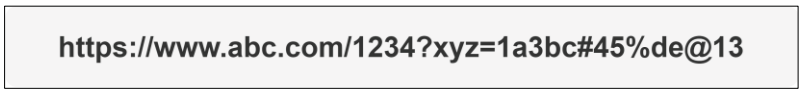
Verilerin güvenli bir şekilde korunmasını sağlamak ve oltalama saldırısı gerçekleşmeden önce kimlik avı amacıyla gönderilen iletinin (e-posta, SMS veya Tekdüzen Kaynak Konum Belirleyicisi (Uniform Resource Locator - URL) adresi) tespit edilmesi kritik bir öneme

sahiptir. Gönderilen e-posta, SMS veya URL adresinin bir oltalama saldırısına mı yoksa yasal/meşru bir adrese mi ait olduğunu belirlemek için çeşitli yaklaşımlar mevcuttur. Bunlar arasında liste tabanlı, sezgisel, makine öğrenmesine dayalı ve hibrit yöntemler yer almaktadır [12]. Liste tabanlı yaklaşımlarda, beyaz liste ve kara liste olarak bilinen listelerden yararlanılır. Beyaz liste, yasal/meşru iletiler hakkında bilgi içerirken, siyah liste oltalama saldırıları hakkında bilgi içerir [13]. Örneğin, web sitelerinin adres bilgileri URL formatında tutulur. URL, bir web tarayıcısında bir web sitesinin adresini göstermek için kullanılan standart bir yapıdır [14]. Liste tabanlı yaklaşımlarda, beyaz liste güvenilir web sitelerinin adreslerini, kara liste ise kimlik avı amaçlı web sitelerinin adreslerini içerir [15]. Bu yöntemlerin temel amacı, kötü niyetli URL'leri mevcut beyaz ve kara listelerdeki adreslerle karşılaştırarak tespit etmektir [16]. Ancak, liste tabanlı yaklaşımlar yeni oltalama saldırılarını tespit etmede yetersiz kaldığından, bu tür tehditleri belirlemek için alternatif tespit yöntemlerinin geliştirilmesine ihtiyaç duyulmuştur [17]. Geliştirilen yöntemler hem sezgisel hem de makine öğrenmesi tabanlı yaklaşımları kapsar. Doğal dil işleme (Natural Language Processing - NLP) alanındaki ilerlemeler ve makine öğrenmesinin alt dalı olan derin öğrenme tekniklerinin gelişimi, kimlik avı tespiti için yenilikçi çözümler sunulmasına olanak sağlamıştır. Bunun yanı sıra, birden fazla yöntemin bir araya getirilmesiyle oluşturulan hibrit yaklaşımlar da geliştirilmeye başlanmıştır [18, 19]. Sezgisel yaklaşımlar, oltalama amacı taşıyan web sitesi adreslerinin en yaygın özelliklerini belirlemeye çalışırken, makine öğrenmesi, derin öğrenme ve hibrit yöntemler, belirli bir veri kümesi üzerinde eğitim yaparak bir URL adresinin oltalama amaçlı mı yoksa yasal bir siteye mi ait olduğunu tespit etmeye yönelir [16, 20].

Oltalama saldırıları, hem bireysel hem de kurumsal kullanıcıları hedef alarak finansal kayıplara, veri hırsızlığına ve kurumsal itibar zedelenmesine yol açar. Bu tür saldırıların giderek artan sayısı ve çeşitlenen yöntemleri, kullanıcı güvenliği açısından ciddi bir tehdit oluşturur. Bu gerekçeye dayanarak, bu tez çalışmasının temel bilimsel problemi, web sitesi URL'lerinin kimlik avı (phishing) mı yoksa yasal (legitimate) mı olduğunu doğru bir şekilde tespit etmektir. Özellikle, kimlik avı saldırılarının neden olduğu yüksek maliyet ve kullanıcı zafiyetlerinin önüne geçebilmek için güvenilir ve etkili bir tespit mekanizmasına ihtiyaç duyulur. Günümüzde, kimlik avı saldırılarının sayısındaki ve tekniklerindeki artış, veri güvenliği ihlallerine yol açtığı için mevcut oltalama tespit yöntemlerinin etkinliğini sorgulamayı gerektirir. Geleneksel yöntemler, saldırganların geliştirdiği yeni tekniklere karşı yetersizdir. Bu durum, daha yenilikçi ve etkili çözümlerin geliştirilmesini zorunlu hale

getirmiştir. Bu tez çalışması kapsamında, kimlik avı tespit sürecini iyileştirmek için karakter ve kelime tabanlı öznitelik çıkarımına dayalı bir derin öğrenme modeli geliştirilmiş ve web sitesi URL'lerinin kimlik avı (phishing) veya yasal (legitimate) adresler olarak sınıflandırılması amaçlanmıştır. Önerilen yöntemin temelinde, karakter ve kelime tabanlı özellik çıkarım tekniklerinin birleştirilmesi yer alır. Bunun temel nedeni, tek bir yöntemin sınırlamalarını gidermek ve daha yüksek doğruluk oranına sahip bir sınıflandırma modeli oluşturmaktır. Bu yaklaşım, hem karakteristik örüntüleri hem de metinsel içerik özelliklerini değerlendirerek daha kapsamlı bir saldırı tespit analizi sağlar. URL adresi, Şekil 1.2'de gösterildiği gibi bir dizi karakterden oluşan metinsel bir ifadedir. Bu ifadenin yasal bir web sitesine mi yoksa kimlik avı amacıyla oluşturulmuş sahte bir web sitesine mi ait olduğunu belirlemek, NLP'de bir metin sınıflandırma problemidir [6]. Bu sorunu daha derinlemesine incelemek amacıyla, çalışma da belirli araştırma soruları ele alınmış ve detaylı bir şekilde analiz edilmiştir. Temel hedef, iletilen URL adresinin kimlik avı saldırıları için kullanılan sahte bir adres mi yoksa gerçek bir adres mi olduğunu belirlemektir. Bu doğrultuda, URL'leri sınıflandırmak için karakter ve kelime tabanlı özellik çıkarımına dayalı bir derin öğrenme modeli önerilmiştir. Çalışma kapsamında, kimlik avı tespitine yönelik geliştirilen bu modelin etkinliğini değerlendirmek için aşağıdaki araştırma sorularının (Research Questions - RQs) yanıtları araştırılmıştır.

RQ1. Farklı URL adreslerinin sınıflandırılmasında karakter tabanlı özellik çıkarımı ile kelime tabanlı özellik çıkarımı arasındaki farklar nelerdir? Bu özellik çıkarma yöntemlerinin birleştirilmesinin sınıflandırma performansı üzerindeki etkisi nedir?
RQ2. Farklı örnek sayılarına sahip veri setlerinde örnek sayısının derin öğrenme algoritmalarında özellik çıkarımı ve sınıflandırma başarısı üzerindeki etkileri nelerdir?
RQ3. Veri setindeki dengesiz etiket dağılımı, özellik çıkarımı ve derin öğrenme algoritmalarının sınıflandırma performansını nasıl etkiler?



Şekil 1.2. Örnek URL adresi

Bu tez çalışmasının mevcut literatüre başlıca katkıları aşağıdaki gibidir:

- Yenilikçi Yaklaşım: Bu çalışma, karakter ve kelime tabanlı özellik çıkarım yöntemlerini birleştiren yenilikçi bir yaklaşım sunar. Bu birleşim, URL'lerin yapısal ve içerik

yönlerini kapsamlı bir şekilde temsil ederek, sınıflandırma modelinin zengin bir öznitelik kümesiyle çalışmasını sağlar.

- Etkili Entegrasyon: Karakter tabanlı öznitelik çıkarımı, URL'lerin yapısal özelliklerini temsil etmek için etkili bir yöntemdir. Kelime tabanlı öznitelik çıkarımı ise URL'lerin anlamsal içeriklerini yansıtmak için etkili bir yöntemdir. Bu iki yöntemin birleştirilmesi, modelin farklı türdeki URL'leri daha etkili bir şekilde tanımasını ve genellemesini sağlamıştır.

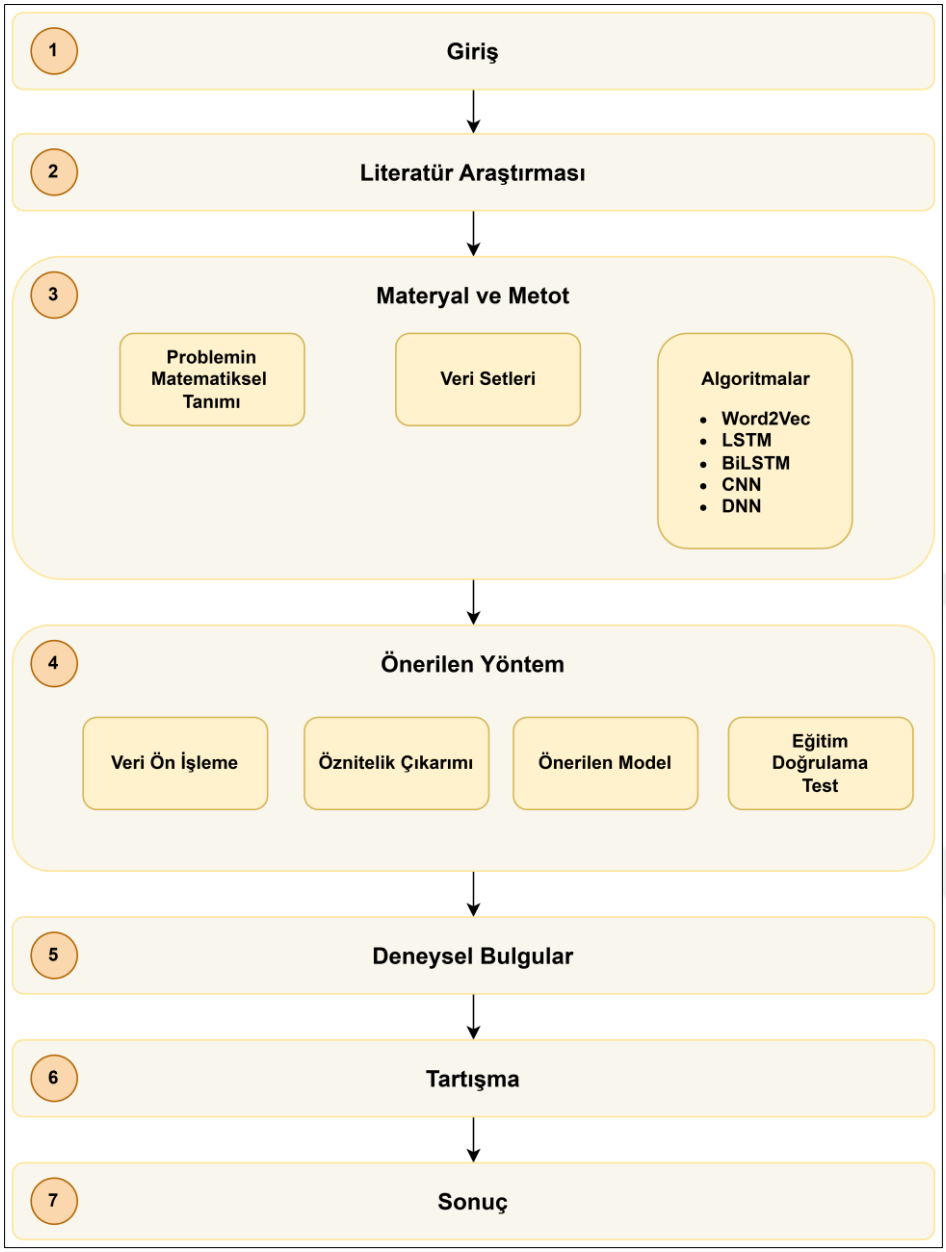
- Gelişmiş Tespit Yeteneği: Karakter ve kelime tabanlı yöntemlerin birleştirilmesiyle URL'lerin hem karakter hem de kelime düzeyinde analiz edilmesi mümkün olmuştur. Bu yaklaşım, tek bir yöntemle tespit edilemeyen oltalama kalıplarının/desenlerinin belirlenmesini sağlamış ve modelin kapsamlı tespit yeteneğini artırmıştır. Böylece, daha karmaşık oltalama URL'leri daha doğru bir şekilde tespit edilebilir, hata toleransı artırılır ve sınıflandırma doğruluğu iyileştirilir.

- Deneysel Başarı: Kısaltılmış URL'lerden oluşan benzersiz bir veri kümesiyle gerçekleştirilen testler, önerilen modelin bu tür URL'lerde de başarılı performans sergilediğini ortaya koymuştur.

- Manuel Özellik Çıkarımına Bağımlılığın Olmaması: Önerilen model, manuel özellik çıkarımı ve uzman bilgisine ihtiyaç duymadan çalışır. Bu durum tespit sürecini daha verimli hale getirir.

- Dilden Bağımsız Sınıflandırma: Önerilen yöntem, üçüncü taraf hizmetlerine bağımlı olmadan çalışan, dilden bağımsız bir sınıflandırma modeli sunar.

Çalışmanın geri kalanı aşağıdaki şekilde düzenlenmiştir: Bölüm 2'de kimlik avı/oltalama saldırılarının tespitine yönelik mevcut literatür incelendikten sonra elde edilen bilgilere yer verilmiştir. Bölüm 3'te kullanılan veri kümeleri ile algoritmalar, Bölüm 4'te ise önerilen model detaylı bir şekilde açıklanmıştır. Bölüm 5'te deneysel çalışmaların sonuçları analiz edilmiştir. Bölüm 6'da elde edilen sonuçlar önceki çalışmalarla karşılaştırılmış ve Bölüm 7'de çalışma sonlandırılmıştır. Çalışmanın düzeni ve akışı, Şekil 1.3'te sunulmuştur.



Şekil 1.3. Tezin yapısını gösteren organizasyon şeması

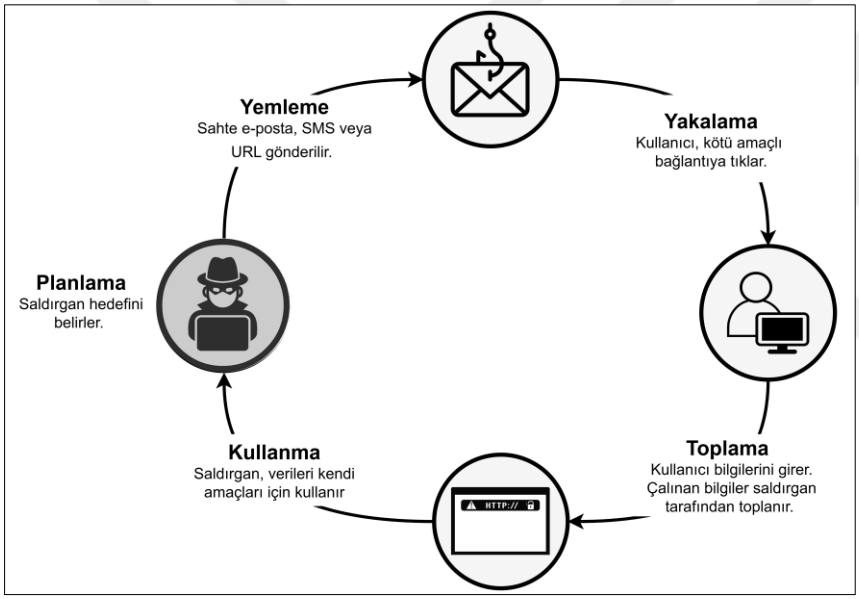
2. LİTERATÜR ARAŞTIRMASI

Oltalama (phishing) saldırısı, saldırganların sahte e-posta, SMS veya web sitesi kullanarak kullanıcıları kandırdığı ve bilgilerini ele geçirdiği bir siber saldırı türüdür. Bu saldırılar, genellikle kullanıcı adı, şifre, kredi kartı bilgileri gibi gizli verilere erişmek amacıyla gerçekleştirilir. Günümüzde phishing saldırıları farklı yöntemlerle uygulanabilir ve bu yöntemler, saldırının hedef kitlesine ve kullanılan tekniklere göre değişiklik gösterir. Phishing saldırılarının en yaygın türlerinden biri olan e-posta phishing, saldırganların güvenilir görünen sahte e-postalar göndererek kullanıcıları kötü amaçlı bağlantılara yönlendirmesiyle gerçekleştirilir. Bunun yanı sıra, belirli kişi veya kurumları hedef alan spear phishing, üst düzey yöneticileri hedefleyen whaling, SMS yoluyla yapılan smishing, telefon aramalarıyla gerçekleştirilen vishing ve sahte web siteleri üzerinden yapılan website phishing gibi farklı oltalama yöntemleri bulunur. Bu çeşitlilik, phishing saldırılarının ne kadar geniş bir alana yayıldığını ve bireyler ile kurumlar için büyük bir tehdit oluşturduğunu gösterir [21].

2.1. Phishing Saldırı Döngüsü

Phishing saldırıları, rastgele gerçekleşen olaylar değildir; aksine, saldırganların planlı bir şekilde ilerlediği aşamalı bir süreçtir. Şekil 2.1'de phishing saldırı döngüsü gösterilmektedir. Bu döngü, saldırganların kullanıcıları nasıl tuzağa düşürdüğünü anlamak açısından oldukça önemlidir. Saldırı süreci beş temel aşama üzerinden açıklanabilir. İlk olarak, saldırgan hedefini belirler (Planlama). Bu aşamada, saldırgan kimleri tuzağa düşüreceğine ve hangi bilgilere erişeceğine karar verir. Daha sonra, hedeflenen kişileri kandırmak için sahte e-postalar, SMS'ler veya web siteleri hazırlanır ve bunlar kullanıcının ilgisini çekecek şekilde tasarlanır. E-posta, SMS gibi iletiler genellikle sahte web sitelerine yönlendirilen bağlantılar içerir (Yemleme). Kullanıcı, bu sahte bağlantıya tıkladığında ve bilgilerini girdiğinde saldırgan, kullanıcıyı başarıyla tuzağa düşürmüş olur (Yakalama). Ardından, elde edilen bilgiler saldırgan tarafından toplanır ve işlenir (Toplama). Son olarak, saldırgan bu verileri kendi amaçları doğrultusunda kullanır; örneğin, başka saldırılarda değerlendirebilir, finansal kazanç sağlamak için satabilir veya kimlik hırsızlığı gibi kötü niyetli faaliyetlerde bulunabilir (Kullanma). Bu döngünün anlaşılması, phishing saldırılarına karşı bilinçlenmek ve korunma yöntemleri geliştirmek açısından önemlidir [3].

Phishing saldırıları her geçen gün daha karmaşık hale gelmekte ve saldırganlar, kullanıcıları kandırmak için daha ikna edici yöntemler geliştirmektedir. Bu saldırılarda, kullanıcıları sahte web sitelerine yönlendirmek amacıyla genellikle yem olarak URL içeren iletiler kullanılmaktadır. Bu nedenle, phishing saldırılarının tespit edilmesi ve önlenmesi için etkili yöntemler geliştirmek amacıyla son yıllarda birçok araştırmacı tarafından çeşitli çalışmalar yürütülmüştür. Bu tez çalışması kapsamında, web sitelerine ait URL adreslerinin oltalama (phishing) veya yasal (legitimate) olup olmadığını sınıflandırmak için derin öğrenme tabanlı bir model geliştirilmesi hedeflenmiştir. Bu doğrultuda, literatürdeki mevcut çalışmalar incelenerek phishing URL'lerin tespitine yönelik derin öğrenme yaklaşımları değerlendirilmiştir.



Şekil 2.1. Phishing saldırı döngüsü

Kimlik avı saldırılarını tespit etmek, farklı karakterlerin birleşiminin URL'leri oluşturduğu göz önüne alındığında, özellik mühendisliğine bağlıdır. İlk olarak, metinsel ifadelerde bulunan ayırt edici özelliklerin belirlenmesi gerekir. URL metninden özelliklerin çıkarılması sarmalayıcı, filtre ve gömülü yöntemler kullanılarak gerçekleştirilebilir. Sarmalayıcı ve filtre yöntemleri, manuel olarak özellik çıkarımına dayanır ve çoğunlukla geleneksel makine öğrenimi algoritmaları ile birlikte uygulanır. Ayrıca, ki-kare testi ve korelasyon gibi istatistiksel yöntemler de bu kategoriye dahildir [18]. Manuel özellik çıkarma ve uzman girdisi gereksinimleri göz önüne alındığında, metinsel içeriği matematiksel olarak formüle edilmiş dizilere çevirmek için derin öğrenme ve doğal dil işleme alanlarında farklı yöntemler geliştirilmiştir [19]. URL adresleri, her karakterin analiz edildiği karakter tabanlı yöntemler

ve URL'lerin belirteçlere (token) bölündüğü kelime tabanlı yöntemler kullanılarak matematiksel dizilere çevrilir. Çevrilen URL'ler, gömülü yöntemlerin kullanıldığı bir derin öğrenme algoritmasının giriş katmanını oluşturur [22].

2.2. İlgili Çalışmalar

Şahingöz ve ark. [6] tarafından yapılan çalışmada, web sitelerinin URL'leri Naive Bayes, rastgele orman, k-En Yakın Komşu (kNN), Adaboost, K-star, sıralı minimum optimizasyon ve karar ağacı gibi çeşitli algoritmalar kullanılarak kimlik avı veya yasal URL'ler olarak sınıflandırılmıştır. Bu algoritmalar, doğal dil işleme tabanlı ve kelime vektörü özelliklerinin yardımıyla kullanılmıştır. Araştırmacılar çalışmalarında, Ebbu2017 Phishing Veri Seti olarak adlandırılan kendi oluşturdukları bir veri kümesini kullanmışlardır. Doğal dil işleme tabanlı 40 ve kelime vektörü tabanlı 1701 özniteliğin kullanıldığı sınıflandırma sürecinde, rastgele orman algoritması %97,98'lik doğruluk oranıyla en iyi sonucu vermiştir.

Word to Vector (Word2Vec), kelimeleri sayısal gösterimler olan vektörlere dönüştüren bir derin öğrenme modelidir. Word2Vec, kelime gömme katmanında kullanılacak ağırlık matrisinin değerini döndürür. Bu değer Word2Vec derin öğrenme modelini eğittikten sonra elde edilen ilk ağırlık değeridir ve gömme katmanında kullanılır [22]. Wang ve ark. [23] ile Afzal ve ark. [24] tarafından yapılan çalışmalar, URL adreslerini kelime gömme yöntemleri ve Word2Vec ile derin öğrenme algoritmalarının kullanabileceği bir matris yapısına dönüştüren çalışmalardır. Wang ve ark. [23] çalışmalarında, Tekrarlayan Sinir Ağı (Recurrent Neural Network - RNN) ve Evrişimsel Sinir Ağı (Convolutional Neural Network - CNN) algoritmalarının birleşimi olan PDRCNN adlı bir model kullanarak öznitelikleri çıkarmışlar ve sınıflandırma yapmışlardır. Çalışmada, Alexa ve PhishTank'tan temin edilen veri seti kullanılmış ve %97 doğruluk oranına ulaşılmıştır.

Word2Vec'e ek olarak, Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri (Bidirectional Encoder Representations from Transformers - BERT) ve Kelime Temsili için Küresel Vektörler (Global Vectors for Word Representation - GloVe) dahil olmak üzere metinden vektör temsilleri oluşturmak için başka teknikler de kullanılmaktadır. Bu yöntemler, doğal dil işleme alanındaki bir dizi zorluğu ele almak için tasarlanmıştır. Özellikle, metin sınıflandırma ve dil modelleme amacıyla kullanılırlar. Glove ve BERT, Word2Vec'e benzer kelime gömme teknikleridir [25, 26]. Benavides-Astudillo ve ark. [26] çalışmalarında, Glove

kelime gömme tekniğini kullanmıştır. Elsadig ve ark. [25] ise, BERT kelime gömme tekniğini kullanmıştır. Benavides-Astudillo ve ark. [26], web sayfalarının Hiper Metin İşaretleme Dili (Hyper Text Markup Language - HTML) kodundan kelimeleri çıkarmış ve bunları vektör temsiline dönüştürmüştür. Bu vektör temsilleri, daha sonra Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM), Çift Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-Term Memory - BiLSTM), Kapılı Tekrarlayan Birim (Gated Recurrent Unit - GRU) ve Çift Yönlü Kapılı Tekrarlayan Birim (Bidirectional Gated Recurrent Unit - BiGRU) algoritmalarına girdi olarak sağlanmıştır. BiGRU algoritması, %97,39 doğruluk oranıyla sınıflandırma işleminde en iyi sonucu vermiştir. Elsadig ve ark. [25] çalışmalarında, Phishing Site Predict veri setini kullanmışlardır. Veri setindeki URL adresleri için BERT ile kelime tabanlı öznitelik çıkarımı yapılmıştır. CNN algoritması kullanılarak sınıflandırma gerçekleştirilmiş ve %99,66'lık doğruluk değeri elde edilmiştir.

Yang ve ark. [27], URL adreslerini karakter dizilerine dönüştürmek için karakter gömme yöntemini kullanmışlardır. Bu dönüşümün ardından, CNN ve LSTM algoritmalarıyla ayırt edici öznitelikler çıkarılmış ve ardından softmax fonksiyonu kullanılarak sınıflandırma gerçekleştirilmiştir. Softmax çıktısı, dinamik kategori belirleme algoritmasının giriş verisi olarak değerlendirilmiştir. Eğer CNN-LSTM sınıflandırmasının sonucu belirlenen eşik değerini aşarsa, oltalama tespiti doğrudan çıktı olarak verilir; ancak aşmazsa, ikinci bir tespit adımı başlatılır. Bu adımda, önceki adımdan elde edilen derin öğrenme özellikleri, istatistiksel veriler, web sayfası kodu ve metin özellikleriyle birleştirilerek XGBoost algoritması kullanılarak sınıflandırma yapılır. Bu çok boyutlu özellik füzyon yöntemi, %98,99 doğruluk oranı elde ederek oltalama tespit süresini azaltmıştır. Benzer şekilde, Alshehri ve ark. [28], oltalama URL'lerini sınıflandırmak için Derin Sinir Ağı (Deep Neural Network - DNN) kullanmışlar ve %98,13 doğruluk oranı elde etmişlerdir.

Özcan ve ark. [29], Şahingöz ve ark. [6]'nın çalışmasındaki 40 NLP tabanlı özniteliği kullanarak karakter gömme ve NLP özniteliklerini birleştiren hibrit bir yöntem önermişlerdir. URL'ler karakter dizilerine dönüştürülerek BiLSTM modeline giriş olarak verilmiş, NLP öznitelikleri ise DNN modeline giriş olarak kullanılmıştır. Her iki modelden elde edilen belirleyici öznitelikler, sınıflandırma sürecinde bir araya getirilmiştir. LSTM, CNN, DNN ve RNN modelleri karşılaştırıldığında, en yüksek performansı DNN-BiLSTM hibrit modeli göstermiş ve Ebbu2017 Phishing ile PhishTank veri setlerinde sırasıyla %98,79 ve %99,21 doğruluk oranlarına ulaşmıştır. Benzer şekilde, karakter tabanlı yaklaşımları

inceleyen çalışmalardan biri de n-gram gömme yöntemidir. N-gram gömme, kelime veya karakter n-gramları oluşturarak metni vektörlere dönüştüren standart bir NLP sınıflandırma yöntemi olarak bilinir. Bozkır ve ark. [20] çalışmalarında, karakter tabanlı n-gram gömme yöntemini kullanarak URL'leri matrislere dönüştürüp CNN modeline giriş olarak vermişlerdir. Daha sonra çift yönlü LSTM (BiLSTM) kullanılarak ardışık ilişkiler yakalanmıştır. Ardından, dikkat mekanizması (attention) ve tam bağlantılı (fully connected) katmanlar kullanılarak oltalama ve yasal URL'ler sınıflandırılmıştır. Karakter tabanlı n-gram gömme yöntemi, n-gramlar arasındaki anlamsal ilişkileri yakalayamadığı için bu tür özniteliklere odaklanılmamıştır. Araştırmacılar kendi oluşturdukları ve GramBeddings ismini verdikleri bir veri seti kullanmışlar ve %98,27 doğruluk oranı elde etmişlerdir.

Bazı çalışmalarda da, karakter ve kelime gömme yöntemleri birlikte kullanılarak oltalama tespiti için URL adresleri sınıflandırılmıştır. Rasymas ve Dovydaitis [30], çalışmalarında karakter, kelime gömme ve sözcüksel (lexical) öznitelikleri bir araya getirmiştir. Karakter ve kelime gömme katmanlarının çıktıları önce CNN algoritmasına, ardından GRU algoritmasına giriş olarak verilmiş ve 60 sözcüksel özniteliğin işlendiği çoklu yoğun (dense) katmanlar kullanılmıştır. Tüm çıktıların tek bir sınıflandırma sürecine entegre edilmesiyle %94,4 doğruluk oranı elde edilmiştir. Benzer şekilde, Zheng ve ark. [22] da, URL adreslerini giriş olarak kullanmış; kelime ve karakter gömme matrislerini ayrı ayrı oluşturmuş; ardından bunları bir otoyol ağı (highway network) modülünde birleştirerek CNN algoritmasına aktarmışlardır. Bu yöntem ile %98,30 doğruluk oranı elde etmişlerdir.

Mobil teknolojilerin yaygınlaşmasıyla birlikte, akıllı cihazlar kimlik avı saldırılarına karşı daha savunmasız hale gelmiştir. Rao ve ark. [31], bu soruna çözüm sunmak amacıyla mobil cihazlar için PhishDump adlı bir uygulama tasarlamışlardır. Modelleri, çoklu topluluk yaklaşımını kullanarak LSTM'den türetilen öznitelikleri Destek Vektör Makinesi (Support Vector Machine - SVM) ile sınıflandırmış ve kendi veri setlerinde %97,30 doğruluk oranı, karşılaştırmalı bir veri setinde ise %98,50 doğruluk oranı elde etmiştir.

Oltalama saldırılarını tespit etmeye yönelik bazı çalışmalarda, birden fazla yöntem bir arada kullanılmıştır. Örneğin, Xiao ve ark. [32] tarafından yapılan çalışmada, dengeli bir eğitim veri seti oluşturmak için Üretici Çekişmeli Ağlar (Generative Adversarial Network - GAN) kullanılmış ve sentetik bir URL veri seti oluşturulmuştur. Araştırmacılar URL adreslerini sınıflandırmak için, CNN algoritmasını çok başlıklı (multi-head) öz-dikkat (self-attention)

mekanizması ile birleştirerek kullanmışlardır. GAN tarafından oluşturulan dengeli veri kümesinin, dengesiz veri kümesine kıyasla daha üstün sonuçlar verdiğini tespit etmişler ve %95,6 doğruluk oranı elde etmişlerdir.

Afzal ve ark. [24]'nın gerçekleştirdiği çalışmada, Kaggle'dan elde edilen açık erişimli bir veri seti kullanılarak, denetimli ve denetimsiz öğrenme algoritmalarını bir araya getiren URLdeepDetect adlı hibrit bir model geliştirilmiştir. Bu model, sınıflandırma işlemini LSTM ve K-Means kümeleme algoritmalarıyla gerçekleştirmiş ve sırasıyla %98,30 ve %99,70 doğruluk oranlarına ulaşmıştır. Benzer şekilde, Roy ve ark. [33] da, Kaggle'dan temin edilen bir veri seti kullanarak URL'leri LSTM, BiLSTM ve GRU algoritmaları ile sınıflandırmış ve BiLSTM algoritması ile en yüksek doğruluk oranı olan %99,0'a ulaşmışlardır.

Aslam ve ark. [34] ile Zaimi ve ark. [35] tarafından yapılan çalışmalarda, araştırmacılar derin öğrenme modelleri kullanarak oltalama URL tespitinde yüksek doğruluk oranlarına ulaşmayı amaçlamıştır. Aslam ve ark. [34], karakter düzeyinde öznitelikler kullanan iki aşamalı AntiPhishStack modelini geliştirmişlerdir. Alexa, Common Crawl, Yandex ve PhishTank'dan elde ettikleri veri seti ile %96,04 doğruluk oranı elde etmişlerdir. Bu modelde, XGBoost'un son sınıflandırıcı olarak kullanıldığı yığılmış (stacked) yaklaşımı benimsenmiştir. Bunun yanı sıra, Zaimi ve ark. [35] veri dengesizliği sorununu permütasyon öznitelik önemi (permutation feature importance) ve SMOTE-Tomek yöntemi ile ele almışlar ve Mendeley veri seti üzerinde XGBoost, CNN, LSTM ve hibrit modelleri kullanarak %94,12 ile %97,82 arasında değişen doğruluk oranlarına ulaşmışlardır. Ayrıca, Zaimi ve ark. [35], daha önce çıkarılmış URL özniteliklerini barındıran bir veri setinden yararlanmışlardır. Her iki çalışmada da, oltalama tespitinde yüksek doğruluk oranları elde edilirken, veri dengesizliği ve öznitelik sınırlamaları ile ilgili zorluklarla karşılaşılmıştır.

2.3. Araştırma Boşlukları

Literatürdeki mevcut çalışmalar çoğunlukla karakter veya kelime tabanlı öznitelik çıkarımı üzerine yoğunlaşmıştır. Karakter tabanlı öznitelikler, bağımsız olarak ya da önceden belirlenmiş özniteliklerle birlikte kullanılmıştır. Ancak, oltalama saldırılarının sürekli evrim geçirmesi, önceden çıkarılan bu özniteliklerin güvenilirliği konusunda belirsizlik yaratmaktadır. Bu nedenle, URL adreslerine dayalı olarak benzersiz öznitelikler

belirleyebilen yeni modellerin geliştirilmesi kritik bir gereklilik hâline gelmiştir. Bununla birlikte, karakter ve kelime tabanlı öznitelik çıkarımını bir arada kullanan çalışmaların sayısı sınırlıdır. Örneğin, Rasymas ve Dovydaitis [30], karakter ve kelime tabanlı özniteliklerle birlikte sözcüksel (lexical) öznitelikleri de içeren bir model geliştirmişlerdir. Ancak, bu özniteliklerin oltalama tespitinde belirgin bir ayırt edici güce sahip olmadığını tespit etmişlerdir. Zheng ve ark. [22] tarafından yapılan çalışmada ise veri setindeki dengesizlik önemli bir kısıtlama olarak öne çıkmıştır. Özellikle, yasal URL'lerin sayıca oltalama URL'lerinden fazla olması, önerilen modelin performansını olumsuz etkilemiştir.

Yapılan literatür taraması, karakter ve kelime tabanlı öznitelik çıkarımının genellikle ayrı ayrı kullanıldığını, ancak bu iki yöntemi bir arada ele alan çalışmaların yetersiz olduğunu ortaya koymuştur. Bu doğrultuda, bu tez çalışmasında karakter ve kelime tabanlı öznitelik çıkarım yöntemleri birleştirilerek yeni bir model geliştirilmiş ve bu eksikliğin giderilmesi hedeflenmiştir. Geliştirilen modelin etkinliğini değerlendirmek amacıyla dört farklı veri seti üzerinde deneysel analizler gerçekleştirilmiştir. Bu yenilikçi yaklaşım, çalışmanın literatürdeki diğer araştırmalardan farklılığını vurgulamakta ve oltalama tespiti için daha etkili çözümler sunma potansiyelini ortaya koymaktadır.

Kimlik avı/oltalama saldırıları, bilgi ve bilgisayar sistemlerinde bulunan güvenlik açıklarını ve zafiyetleri istismar ederek ciddi bir risk oluştur. Literatürdeki çalışmalar kimlik avı saldırılarının tespitine katkı sunmuş olsa da, yeni yöntemlerin ortaya çıkmasıyla bu saldırıların sayısı ve karmaşıklığı giderek artmaktadır. Bu nedenle, kimlik avı saldırılarını tespit etmeye yönelik savunma tekniklerinin güçlendirilmesi gerekir. Bu çalışmada, kimlik avı saldırılarını belirlemek için dört farklı veri seti kullanılmıştır. Veri setleri, farklı sayılarda URL adreslerinden oluşmaktadır. URL adresleri çeşitli karakterlerden meydana geldiği için, derin öğrenme algoritmaları ile doğrudan ham verilerin kullanımı mümkün değildir. Bu verilerin matematiksel bir formata dönüştürülmesi gerekir. Bu doğrultuda, veri ön işleme ve öznitelik çıkarımı süreçlerinde hem karakter hem de kelime odaklı yöntemler aynı anda uygulanmıştır. Karakterler ve kelimeler için ayrı ayrı iki farklı öznitelik grubu oluşturulmuştur. Elde edilen öznitelikler, yüksek boyutlu tam sayılardan oluşmaktadır. Bu öznitelikleri daha düşük boyutlu ve yoğun vektörlere dönüştürmek amacıyla bir gömme (embedding) katmanı kullanılmıştır. Gömme katmanının çıktısı, derin öğrenme algoritmaları için daha anlamlı ve verimli girdiler oluşturarak modelin etkin bir şekilde öğrenmesine katkıda bulunmuştur. Karakter ve kelime tabanlı çıkarılan öznitelikler,

BiLSTM algoritması kullanılarak ayrı ayrı eğitilmiş ve dikkat (attention) katmanı ile desteklenmiştir. Eğitimden elde edilen karakter ve kelime tabanlı öznitelikler birleştirilmiş ve ardından sınıflandırma işlemi gerçekleştirilmiştir. Elde edilen bulgular, literatürdeki önceki çalışmalarla karşılaştırıldığında, sürekli değişen kimlik avı saldırılarını kontrol altına almak için bu savunma yönteminin etkili bir şekilde kullanılabileceğini ortaya koymuştur. Bu noktada, karakter tabanlı ve kelime tabanlı öznitelik çıkarma yöntemlerinin bir arada kullanılması, kimlik avı URL'lerinin tespitinde kapsamlı ve çok yönlü bir yaklaşım sunmaktadır. Karakter tabanlı yöntem, URL'lerdeki her bir karakteri analiz ederek ince desenleri ve anormallikleri tespit edebilirken, kelime tabanlı yöntem kelimeler arasındaki anlamsal ilişkileri incelemeye odaklanmaktadır. Bu kombinasyon, kimlik avı URL'lerinin hem karakter hem de kelime düzeyinde analiz edilmesine olanak tanıyarak daha kapsamlı bir tespit süreci sağlar. Örneğin, bir yöntemin tespit edemediği kimlik avı URL'si diğer yöntem tarafından belirlenebilir, böylece modelin hatalara karşı dayanıklılığı artırılabilir. Ayrıca, iki farklı seviyedeki yöntemin entegrasyonu, daha gelişmiş ve karmaşık kimlik avı saldırılarının tespit edilmesini kolaylaştırmaktadır. Bu yaklaşım, kimlik avı URL'lerinin tespitinde daha yüksek doğruluk ve güvenilirlik sağlayarak kullanıcıların güvenliğini artırmakta ve internet ortamının daha güvenilir hale gelmesine katkıda bulunmaktadır.

3. MATERYAL VE METOT

Bu bölümde, çalışmada ele alınan problem matematiksel olarak ifade edilmekte, kullanılan veri kümeleri ve algoritmalar açıklanmaktadır.

3.1. Problemin Matematiksel Tanımı

Kimlik avı (phishing) ve yasal (legitimate) URL'lerin sınıflandırılması, metin sınıflandırma problemi kapsamında ele alınmaktadır. URL adresleri ve bunlara ait etiketlerden oluşan bir veri seti, Eş. 3.1'de gösterildiği şekilde tanımlanmıştır.

$$D = \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\} \quad (3.1)$$

Eş. 3.1'de D, veri setini temsil eder. $\{a_1, a_2, \dots, a_n\}$ değişkenleri URL adreslerini, $\{b_1, b_2, \dots, b_n\}$ değişkenleri ise URL adreslerinin etiketlerini ifade eder. b değişkeni $\{0,1\}$ olmak üzere iki farklı değer alabilir. 0 değeri ($b = 0$), ilgili URL adresinin yasal/meşru (legitimate) sınıfa ait olduğunu gösterirken, 1 değeri ($b = 1$), URL adresinin kimlik avı/oltalama (phishing) sınıfına ait olduğunu belirtir.

Farklı karakterlerden oluşan URL adreslerinin derin öğrenme algoritmalarında kullanılabilmesi için vektörlere dönüştürülmesi gerekir. Eş. 3.2'de, a_t değişkeni URL'yi, x_t ise URL adresinden elde edilen öznitelik vektörünü temsil eder. URL adresi a_t, x_t'ye dönüştürülür.

$$a_t \rightarrow x_t \quad (3.2)$$

Eş. 3.3, f fonksiyonunun R^n uzayından (n boyutlu reel sayılar kümesi) R uzayına (reel sayılar kümesi) geçişini gösterir. R^n, f fonksiyonunun tanım kümesini, R ise değer kümesini temsil eder.

$$f: R^n \rightarrow R \quad (3.3)$$

Bir URL adresinin oltalama (phishing) (1) veya yasal (legitimate) (0) sınıfına ait olup olmadığını belirlemek için, f fonksiyonu bir tahmin yapar. Bu tahmin, Eş. 3.4'te gösterilmiştir. b değişkeni 0 ve 1 değerlerini aldığından, tahmin işleminde sigmoid fonksiyonu kullanılır. Sigmoid fonksiyonunun çıktısı, 0 ile 1 aralığında bir değer üretir. Eş. 3.4 ve Eş. 3.5'te σ, sigmoid fonksiyonunu temsil eder. Eş. 3.4'te $f(x_t))$, modelin sınıflandırma sonucuna ait tahmin değerini gösterir. b_t, t zamanındaki URL adresinin tahmini değeridir (Eş. 3.4). Eşik değeri 0,5 seçilmiştir. Eğer sigmoid fonksiyonunun çıktısı 0,5 veya daha büyükse, URL adresi oltalama (phishing) olarak sınıflandırılır. Eğer sigmoid fonksiyonunun çıktısı 0,5'ten küçükse, URL adresi yasal (legitimate) olarak sınıflandırılır [24, 27].

$$\hat{b}_t = \sigma\big(f(x_t)\big) \tag{3.4}$$

$$\sigma\,(z) = \frac{1}{1 + e^{-z}} \tag{3.5}$$

3.2. Veri Setleri

URL kimlik avı sınıflandırma sorununu ele almak için tez kapsamında, farklı sayılarda URL adresleri içeren dört halka açık veri seti kullanılmıştır. İlk veri seti, toplam 799 991 URL içeren ve %50'si kimlik avı, %50'si yasal olacak şekilde dengelenmiş olan GramBeddings Veri Seti [20]'dir. İkinci veri seti, Kaggle'dan temin edilen Malicious and Benign URLs Veri Seti [36]'dir ve toplam 450 176 URL içerir. %23,2 kimlik avı ve %76,8 yasal URL ile dengesiz bir dağılıma sahiptir. Üçüncü veri seti, Ebbu2017 Phishing Veri Seti [6]'dir ve toplam 73 575 URL içerir. %50,5 kimlik avı ve %49,5 yasal URL ile büyük oranda dengeli bir dağılıma sahiptir. Son veri seti ise Mendeley Data Web Page Phishing Detection Veri Seti [37]'dir ve 11 430 URL bulunur. %50 kimlik avı, %50 yasal URL olacak şekilde dengelidir.

GramBeddings, Ebbu2017 Phishing ve Mendeley Data Web Page Phishing Detection veri setleri kimlik avı ve yasal sınıflar açısından dengeli bir yapıya sahipken, Malicious and Benign URLs veri seti dengesizdir. Çizelge 3.1'de, bu veri setlerine ilişkin ayrıntılı istatistiksel bilgilere yer verilmiştir. Bu çalışmada, önerilen modeli geliştirmek amacıyla ilk üç veri seti kullanılmıştır. Daha küçük boyutta olan dördüncü veri seti ise alternatif

yöntemlerin test edilmesini sağlamak ve modelin daha küçük bir veri kümesinde nasıl performans gösterdiğini değerlendirmek amacıyla çalışmaya dahil edilmiştir.

GramBeddings, Malicious and Benign URLs, Ebbu2017 Phishing ve Mendeley Data Web Page Phishing Detection veri setleri çeşitli nedenlerle seçilmiştir. Öncelikle, bu veri setlerinin tamamı halka açık ve erişilebilirdir. Ayrıca, farklı sayılarda kimlik avı ve yasal URL adresleri içerirler. Üçü dengeli bir dağılıma sahipken, biri dengesizdir. Bu özellikler, URL kimlik avı sınıflandırma sorununu ele almak için bu veri setlerini uygun bir seçenek haline getirmiştir.

Çizelge 3.1. Veri setlerinin istatistiksel analizi

	GramBeddings Veri Seti [20]	Malicious and Benign URLs Veri Seti [36]	Ebbu2017 Phishing Veri Seti [6]	Mendeley Data Web Page Phishing Detection Veri Seti [37]
Toplam URL sayısı	799 991	450 176	73 575	11 430
Kimlik avı URL sayısı	399 998 (%50)	104 438 (%23,2)	37 175 (%50,5)	5715 (%50)
Yasal URL sayısı	399 993 (%50)	345 738 (%76,8)	36 400 (%49,5)	5715 (%50)
Maksimum URL uzunluğu	7833	2314	3992	1633
Minimum URL uzunluğu	9	8	9	5
Ortalama URL uzunluğu	58,7934	52,4561	61,8635	53,7309

3.3. Algoritmalar

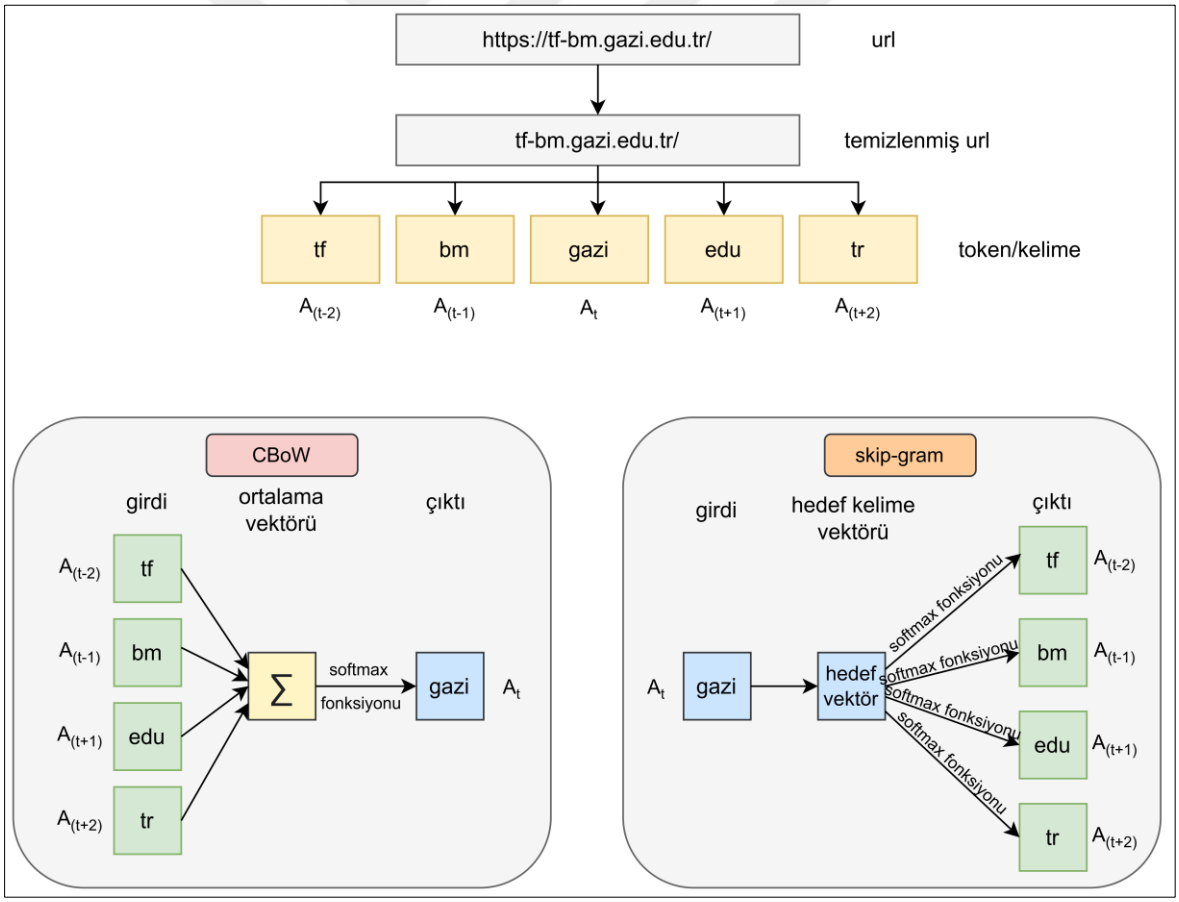
Bu bölümde, tez kapsamında geliştirilen modelin oluşturulmasında kullanılan algoritmalar detaylı olarak açıklanmaktadır.

3.3.1. Word2Vec

Word2Vec, yapay sinir ağlarına benzer bir yaklaşım kullanır. Bir doğal dil işleme modelidir. Bu model, kelimeleri sayısal vektörler halinde temsil ederek anlamlarını matematiksel olarak kodlar. Böylece, kelimeler arasındaki anlamsal ilişkiler vektörler üzerinden analiz edilebilir. Ayrıca, benzer anlamlara sahip kelimeleri temsil eden vektörler, matematiksel

olarak birbirine yakın konumlanır [38]. Word2Vec modelinde, kelimeler arasındaki anlamsal yakınlık sayısal olarak ifade edilir. Kelimelerin vektörlere dönüştürülmesi, derin öğrenme algoritmalarının uygulanmasını mümkün kılar.

Word2Vec algoritması, kelime vektörlerini oluşturmak için Continuous Bag of Words (CBoW) ve Skip-gram olmak üzere iki farklı mimari yaklaşım kullanır. Bu yaklaşımlar, kelimeler arasındaki ilişkileri belirlemek amacıyla geliştirilmiştir [24]. Şekil 3.1'de, CBoW ve Skip-gram algoritmalarının nasıl çalıştığına dair bir görsele yer verilmiştir ve bu görsel üzerinden algoritmaların çalışma mantığı anlatılmıştır. Burada, giriş verisi olarak bir metin kullanılmış olup, bu çalışmada URL adresleri üzerinde uygulanmıştır. Öncelikle URL adresinin 'https://' kısmı kaldırılır. ('https://' kısmının kaldırılması zorunlu değildir; bu durum herhangi bir soruna neden olmaz. 'https' de bir kelime olarak ele alınabilir.)



Şekil 3.1. Word2Vec algoritması: CBoW ve skip-gram

Metin, '/', '.' ve '-' gibi belirli karakterlere göre kelimelere veya tokenlara ayrılır. Bu süreçte algoritmaların çalışma mantığını açıklamak için hedef kelimenin 'gazi' olduğu varsayılır.

‘tf’, ‘bm’, ‘edu’ ve ‘tr’ kelimeleri, bağlam (context) kelimeleri olarak belirlenir. Word2Vec modeli, pencere boyutu (window size) adı verilen bir parametre kullanır. Bu parametre, hedef kelimenin sol ve sağ tarafında kaç kelimenin dikkate alınacağını belirler. CBoW algoritması kullanılarak bağlam kelimeleri giriş verisi olarak alınır ve hedef kelime tahmin edilmeye çalışılır. Örneğin, pencere boyutunun 2 olduğu varsayılırsa, ‘tf’, ‘bm’, ‘edu’ ve ‘tr’ kelimeleri giriş olarak seçilir ve vektörlere çevrilir. Bu vektörlerden ortalama bir vektör oluşturulur ve hedef kelime olan ‘gazi’ kelimesini tahmin etmek için softmax fonksiyonu (Eş. 3.6) kullanılır. Skip-gram algoritması, CBoW algoritmasının tersi bir mantıkla çalışır. Bu yöntemde, hedef kelimenin vektörü oluşturularak bağlam kelimeleri tahmin edilmeye çalışılır. Tahmin sürecinde Softmax fonksiyonu (Eş. 3.6) kullanılır ve tahmin edilen bağlam kelimeleri ile gerçek bağlam kelimeleri arasındaki farkın en aza indirilmesi hedeflenir. Sonuç olarak, kelimeler arasındaki anlamsal ilişkiler matematiksel olarak belirlenir.

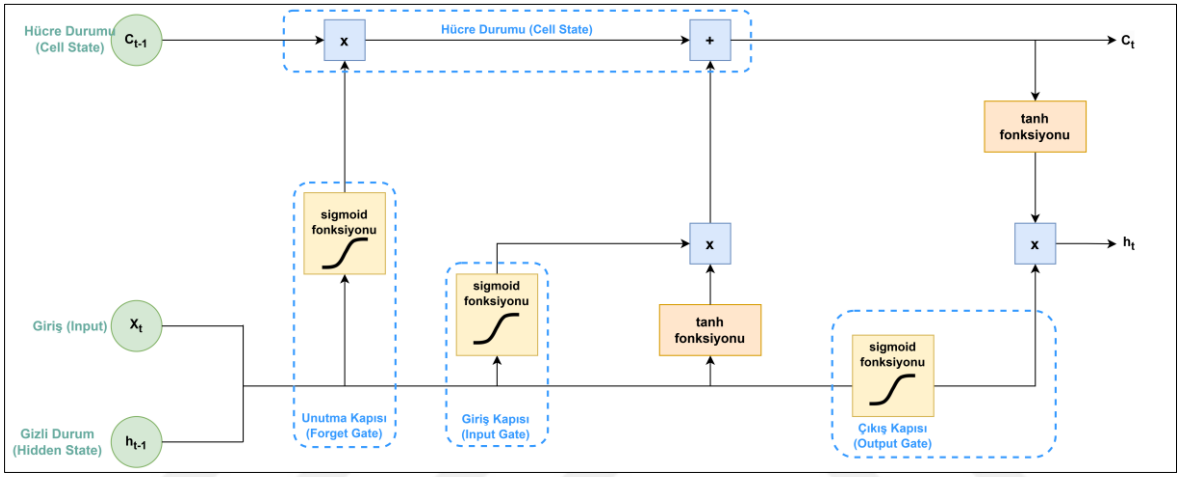
$$\text{softmax}(\text{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{3.6}$$

3.3.2. LSTM

LSTM, özellikle RNN’lerde yaygın olarak görülen kaybolan gradyan (vanishing gradient) sorununu önlemeyi hedefleyen bir yinelenen sinir ağı (Recurrent Neural Network - RNN) modelidir. Bu sorun özellikle derin ileri beslemeli ağlarda ortaya çıkar ve katman sayısının artması, pratik eğitim sürecini zorlaştırır. LSTM’ler, RNN’lerin kısa vadeli bağımlılıklar konusundaki kısıtlarını giderirken, uzun vadeli bağımlılıkların öğrenilmesini de destekler [18, 39]. LSTM mimarisi, hücre durumu (cell state), unutma kapısı (forget gate), giriş kapısı (input gate) ve çıkış kapısı (output gate) gibi ana bileşenlerden meydana gelir. Şekil 3.2’deki görsel LSTM’nin mimari yapısını gösterir. Hücre durumu, uzun vadeli belleği koruyarak bilgilerin saklanmasına veya unutulmasına olanak tanır. Unutma kapısı, hücre durumunda hangi bilgilerin saklanacağı veya yok sayılacağına karar verirken, giriş kapısı ise yeni bilgilerin hücre durumuna nasıl ekleneceğini düzenler. Çıkış kapısı, bir sonraki gizli durumu (hidden state) belirler [31].

$(w_1, w_2, \dots, w_n)$ bir kelime dizisini temsil eder. (h_t, c_t), t zamanındaki LSTM durumunu ifade eder. Burada h_t, gizli durumu, c_t ise hücre durumunu temsil eder. LSTM modeli, giriş vektörü (x_t) ile önceki durum bilgilerini (h_{t-1}, c_{t-1}) girdi olarak alır. Unutma kapısı, hücre

durumunda hangi bilgilerin korunacağı veya yok edileceğini belirler. Unutma kapısı, önceki gizli durum (h_{t-1}) ve giriş vektörü (x_t) kullanılarak belirlenir ve bu işlem Eş. 3.7 ile ifade edilir. Hücre durumundaki (c_{t-1}) her bir bilgi için 0 ile 1 arasında bir değer üretilir. Üretilen bu değere bağlı olarak, 0 bilgilerin silineceğini, 1 ise bilgilerin korunacağını ifade eder. Eş. 3.7'de, f_t unutma kapısının çıktısını temsil eder. σ, sigmoid aktivasyon fonksiyonunun sonucudur. w_f, ağırlık matrisini, b_f ise bias (önyargı) değerini ifade eder. $[h_{t-1}, x_t]$ ise gizli durum ve giriş vektörünün birleşimini gösterir.



Şekil 3.2. LSTM mimarisi

$$f_t = \sigma\left(w_f * [h_{t-1}, x_t] + b_f\right) \tag{3.7}$$

Giriş kapısı, Eş. 3.8 ve Eş. 3.9'da verilen formüllere göre hesaplanır. Bu hesaplama sürecinde, giriş kapısı vektörü ve yeni aday hücre durumu bileşenleri de göz önünde bulundurulur. Burada, i_t değişkeni giriş kapısı çıktısını temsil eder. $\tilde{c}_t$ değişkeni yeni aday hücre durumunu ifade eder. w_i ve w_c değişkenleri ağırlık matrislerini temsil eder. b_i ve b_c değişkenleri ise bias değerlerini ifade eder.

$$i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i) \tag{3.8}$$

$$\tilde{c}_t = \tanh\left(w_c * [h_{t-1}, x_t] + b_c\right. \tag{3.9}$$

Yeni hücre durumu, Eş. 3.10'da belirtilen formüle göre hesaplanır. Bu hesaplamada unutma kapısı ve giriş kapısının çıktıları kullanılır. c_t değişkeni güncellenmiş hücre durumunu, c_{t-1} değişkeni ise önceki hücre durumunu temsil eder.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{3.10}$$

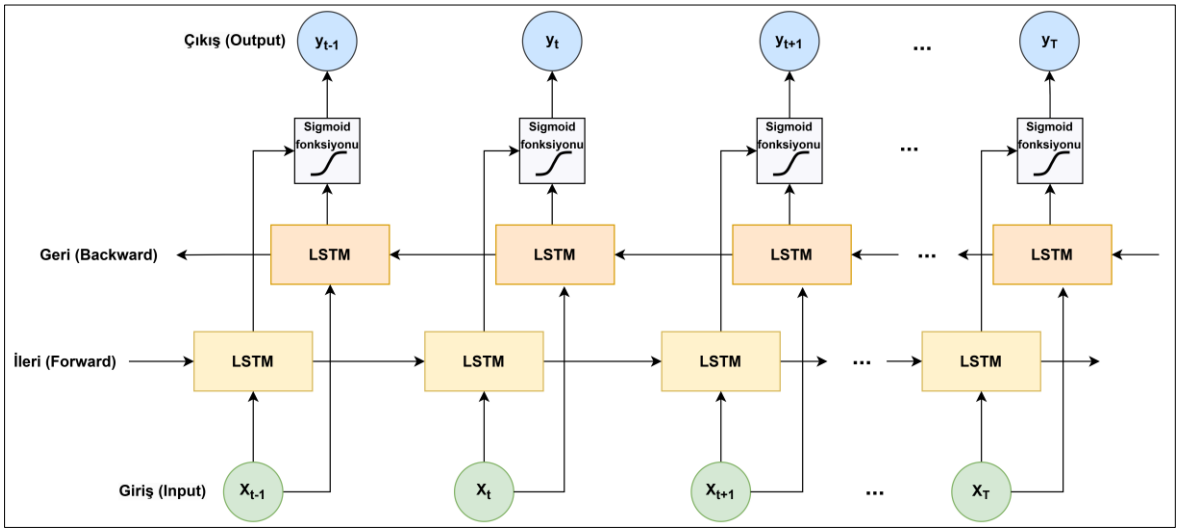
Çıkış kapısı, hangi bilginin gizli durum olacağını belirler ve bu işlem Eş. 3.11 ve Eş. 3.12 kullanılarak gerçekleştirilir. o_t değişkeni çıkış kapısının çıktısını, h_t değişkeni güncellenmiş gizli durumu, w değişkeni ağırlık matrisini ve b değişkeni ise bias değerini ifade eder.

$$o_t = \sigma(w_o * [h_{t-1}, x_t] + b_o) \tag{3.11}$$

$$h_t = o_t * \tanh(c_t) \tag{3.12}$$

3.3.3. BiLSTM

BiLSTM algoritması, LSTM'nin her iki yönde çalışmasını temel alan bir prensibe dayanır. Şekil 3.3'te gösterildiği üzere, BiLSTM modeli ileri ve geri yönlü LSTM katmanlarından oluşur. Bu çift yönlü çalışma yapısı, sıralı verilerde hem geçmiş hem de gelecek bilgilerin değerlendirilmesini sağlar. Böylece, algoritma daha fazla bağlamsal bilgi yakalayarak verileri daha etkili bir şekilde işler. Sonuç olarak, LSTM'ye kıyasla daha kapsamlı ve doğru tahminler üretebilir. BiLSTM algoritmasının matematiksel ifadeleri, LSTM'nin denklem yapısıyla aynıdır [33, 40].

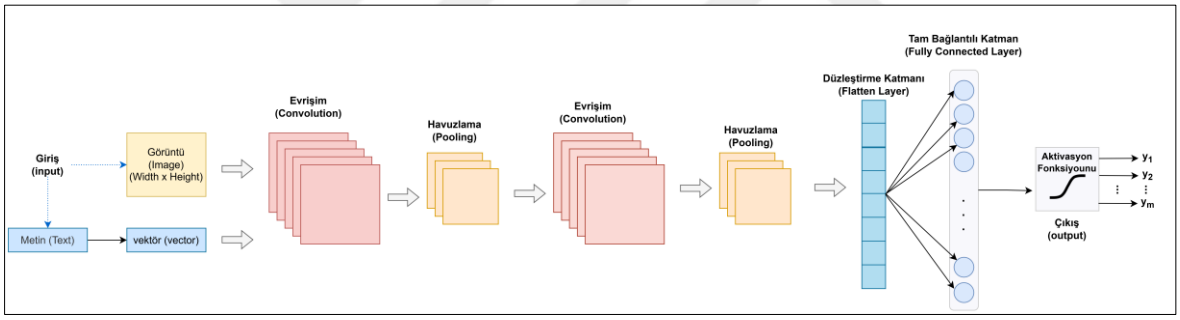


Şekil 3.3. BiLSTM mimarisi

3.3.4. CNN

CNN, görüntü işleme, bilgisayarla görme ve tıbbi görüntülerden hastalık teşhisi gibi alanlarda sıkça tercih edilen bir derin öğrenme algoritmasıdır. Bir CNN'nin temel yapısı; evrişim, havuzlama ve tam bağlantılı katmanlardan oluşur (Şekil 3.4). Görüntülerden oluşan veri kümelerinde, genellikle CNN uygun algoritma olarak tercih edilir. Çünkü evrişimsel sinir ağları, görüntüleri doğrudan girdi verisi olarak alabilir ve işleyebilir. Evrişim katmanı, girdiye uygulanan filtreler (kernels) içerir ve bu sayede öznitelik haritaları çıkarılır [32]. Evrişim katmanında genellikle ReLU aktivasyon fonksiyonu tercih edilir (Eş. 3.13). Aktivasyon fonksiyonları, çok katmanlı sinir ağlarında doğrusal olmayan dönüşümler için kullanılır [39].

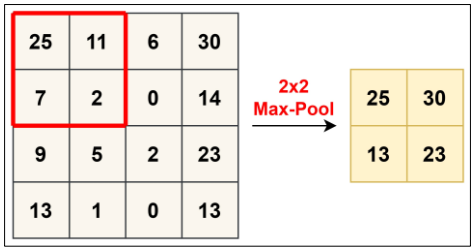
$$ReLU(x) = \max(0, x) \tag{3.13}$$



Şekil 3.4. CNN mimarisi

Filtreler evrişim katmanında kullanıldıktan sonra veri boyutu azalır. Bu azalmayı ve bilgi kaybını önlemek için padding uygulanır. Havuzlama katmanında maksimum, ortalama vb. havuzlama işlemleri gerçekleştirilir. Şekil 3.5'te gösterildiği gibi maksimum havuzlamada, filtre kapsadığı alan içindeki en büyük değeri belirlerken, ortalama havuzlamada ise filtrenin kapsadığı alandaki değerlerin ortalaması hesaplanır. Bu yaklaşım, sinir ağının doğru karar verebilmesi için yeterli bilgi içeren daha küçük çıktılar üretilmesini sağlar. Evrişim ve havuzlama katmanlarının amacı, girdiden önemli öznitelikleri çıkarmaktır [39].

Tam bağlantılı katmandan önce, iki veya üç boyutlu veriler düzleştirilerek tek boyutlu bir vektöre dönüştürülür. Tek boyutlu forma dönüştürülen bu vektör, tam bağlantılı katmanlardaki yapay sinir ağına giriş verisi olarak sunulur.



Şekil 3.5. Maksimum havuzlama işlemi

Tam bağlantılı katmanlarda dropout işlemi uygulanabilir. Dropout sürecinde, rastgele seçilen bazı nöronlar devre dışı bırakılır ve tam bağlantılı katmanda bu devre dışı bırakılan nöronlarla bağlantılar kurulmaz. Dropout işlemi, modelin ezberlemeye aşırı bağımlı hale gelmesini önler [41]. Tam bağlantılı katmandan sonra, ikili sınıflandırma yapılacaksa sigmoid aktivasyon fonksiyonu (Eş. 3.5) tercih edilir. Eğer, çoklu sınıflandırma gerçekleştirilecekse softmax aktivasyon fonksiyonu (Eş. 3.6) kullanılır. Son olarak, sınıflandırma işlemi gerçekleştirilir ve bir tahminde bulunulur.

3.3.5. DNN

Derin sinir ağları (DNN), giriş katmanı, birden fazla gizli katman ve çıkış katmanından oluşur. Bu yapı, yapay sinir ağlarının daha ileri bir versiyonunu temsil eder. DNN'ler, ham veriden öznitelikler çıkarabilme yeteneğine sahiptir. Şekil 3.6'da, DNN algoritmasının katman yapısı görselleştirilmiştir. Girdi katmanında, ham veri alınır ve gizli katmana iletilir. Gizli katmanlar birden fazla katmandan oluşabilir. Ağın derinliği, sahip olduğu gizli katmanların miktarına göre belirlenir. Bu katmanlarda, yüksek seviyeli veri öznitelikleri çıkarılır [42]. DNN'ler, ileri besleme ve geri yayılım prensipleriyle çalışır. İleri besleme mekanizması, giriş verisinin işlenmesini ve anlamlı özniteliklerin çıkarılmasını sağlar. Girdi verisi, gizli katmanlardan geçirilerek burada karmaşık ve yüksek seviyeli öznitelikler elde edilir [13]. Gizli katmanlardaki nöronların bağlantı ağırlıkları, önceki katmandaki nöronun çıktısıyla çarpılır ve bias değeri ile toplanır (Eş. 3.14). Elde edilen sonuç, ReLU, sigmoid veya tanh gibi aktivasyon fonksiyonlarından birine uygulanarak mevcut nöronun çıktısı elde edilir. Bu işlem, gizli katmanlardaki nöronlar boyunca devam eder. İleri beslemeli derin sinir ağı modeli, girdi verisinden ayırt edici öznitelikleri öğrenir. Eş. 3.14 ve Eş. 3.15'te, k_j bir nöronun aktivasyon fonksiyonuna girmeden önceki toplam girdisini temsil eder. w_{ij}, ağırlık matrisi; x_i giriş değerleri yani önceki katmandan gelen çıktılar; b_j ise bias değeridir. σ ise aktivasyon fonksiyonudur (ReLu, sigmoid, tanh, vb.) [29].

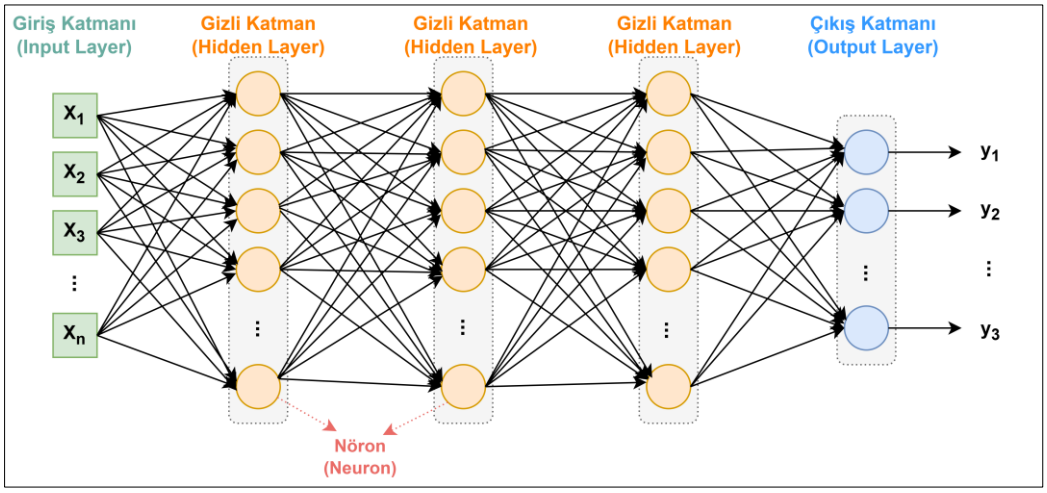
$$k_j = \sum_i (w_{ij} * x_i) + b_j \tag{3.14}$$

$$y_j = \sigma(k_j) \tag{3.15}$$

Kayıp fonksiyonu, ağın tahminlerini değerlendirmek için kullanılır. Sınıflandırma problemleri kapsamında, çapraz entropi (cross-entropy) tercih edilen kayıp fonksiyonudur (Eş. 3.16). Kayıp fonksiyonunun türevi geri doğru hesaplanarak, ağırlıklar ve bias değerleri bu doğrultuda güncellenir. Bu işlem, geri yayılım (backpropagation) olarak ifade edilir [39]. Eş. 3.16'da, y_i gerçek etiket değerini, $\hat{y}_i$ ise model tarafından tahmin edilen değeri temsil eder.

$$L = -\sum_i y_i \log(\hat{y}_i) \tag{3.16}$$

Derin sinir ağlarında ağırlık ve bias değerlerinin güncellenmesi, modelin tahmin doğruluğunu iyileştirir. Ağın son katmanında, yaygın olarak kullanılan iki aktivasyon fonksiyonundan biri olan sigmoid veya softmax seçilir ve çıktı değişkenleri için tahmin edilen değerler üretilir.



Şekil 3.6. DNN mimarisi

4. ÖNERİLEN YÖNTEM

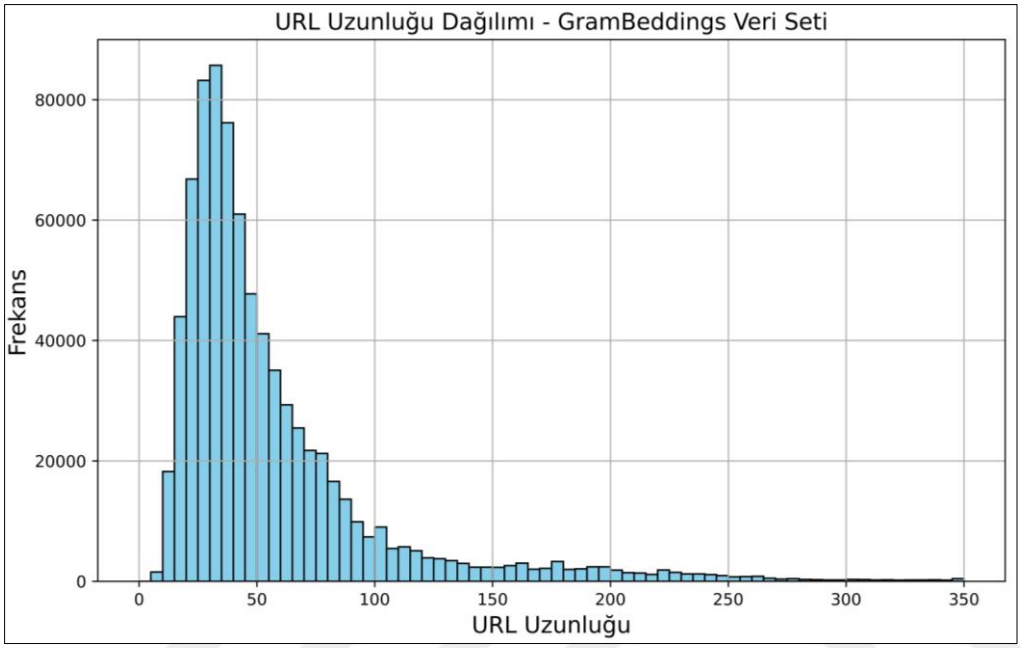
Bu bölümde, tez kapsamında kullanılan veri setlerindeki URL'lere uygulanan ön işleme adımları, URL'lerden karakter ve kelime tabanlı özniteliklerin çıkarılması ile geliştirilen sınıflandırma modelinin gelişim süreci ayrıntılı bir şekilde açıklanmıştır.

4.1. Veri Ön İşleme

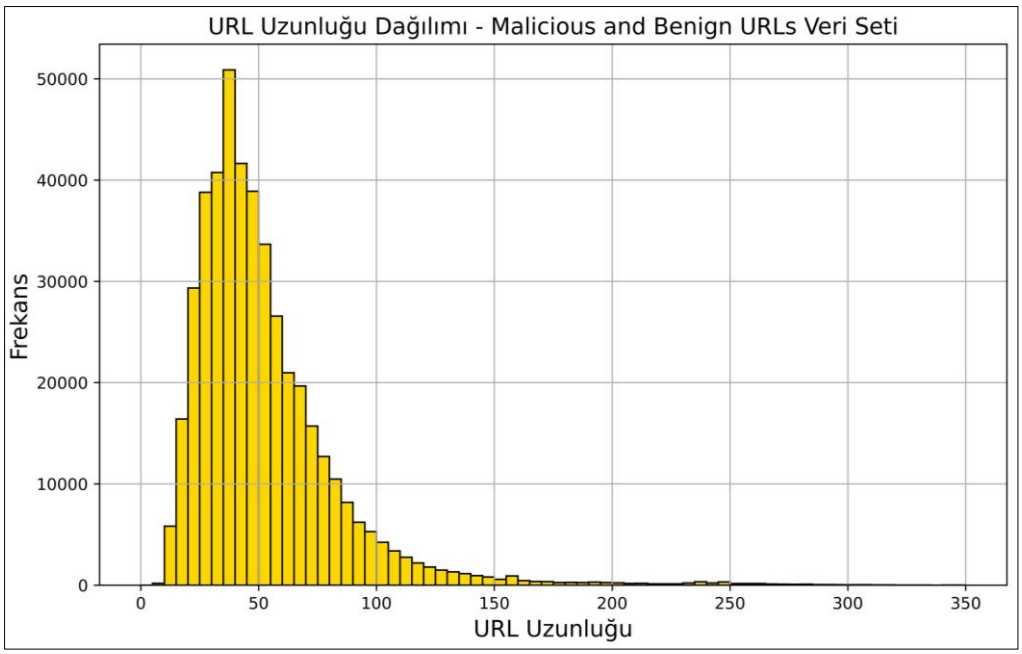
Ham veriyi anlamlı ve analiz edilebilir hale getirmek için ön işleme sürecinden geçirmek gereklidir. Bu aşama, verimlilik açısından önemlidir. URL adresleri ilk olarak ham veri şeklinde alınmış ve sonrasında ön işleme tabi tutulmuştur. Ön işleme sürecinde aşağıdaki adımlar izlenmiştir.

- Veri setindeki sütun adları 'url' ve 'class' olarak güncellenmiştir. Böylece, elde edilen veri seti n satır ve iki sütundan oluşur. Burada n, URL sütunundaki URL adreslerinin sayısını temsil eder. 'class' terimi ise URL adreslerinin karşılık geldiği oltalama (phishing) veya yasal (legitimate) sınıfı ifade eder. Bu çalışmada dört farklı veri setinden yararlanılmıştır. Her bir veri setindeki URL adreslerinin sayısı farklıdır.
- Gereksiz veriler temizlenmiş, eksik veriler ise tamamlanmıştır.
- Veri setindeki tüm URL'ler 'https://' veya 'http://' ön eklerini içermektedir. Ancak, bu bileşenler ayırt edici bir özellik taşımadığından tüm URL'lerden çıkarılmıştır.
- Veri setindeki 'class' sütunu, URL adreslerinin gerçek etiket değerlerini temsil etmektedir. Eğer bir URL adresi 'phishing' (oltalama), 'legitimate' (yasal/meşru), 'malicious' (zararlı) veya 'benign' (zararsız) olarak kategorik biçimde etiketlenmişse, etiket kodlama (label encoding) işlemi uygulanmıştır. Bu süreçte kategorilere atanan sayısal değerler şu şekilde belirlenmiştir: 'phishing' için 1, 'legitimate' için 0, 'malicious' için 1 ve 'benign' için 0. Etiket kodlamanın amacı, kategorik verileri algoritmalarının işleyebileceği sayısal bir formata dönüştürmektir.
- Dört veri setinin her biri için toplam URL sayısı, oltalama (phishing) URL sayısı, yasal (legitimate) URL sayısı, maksimum URL uzunluğu, minimum URL uzunluğu ve ortalama URL uzunluğu hesaplanmıştır. İlgili istatistiksel sonuçlara Tablo 1'de yer verilmiştir.

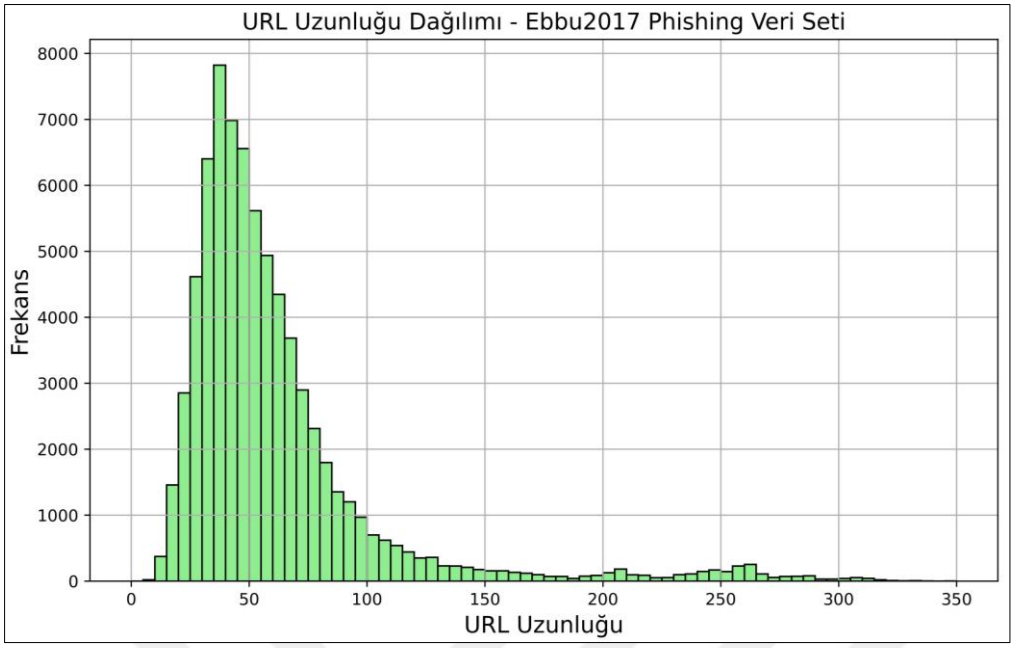
Şekil 4.1, GramBeddings; Şekil 4.2, Malicious and Benign URLs; Şekil 4.3, Ebbu2017 Phishing; ve Şekil 4.4, Mendeley Data Web Page Phishing Detection veri setlerindeki URL uzunluklarının dağılımını gösterir. Grafikler, farklı URL uzunluklarının frekans dağılımını sütunlar şeklinde gösterir.



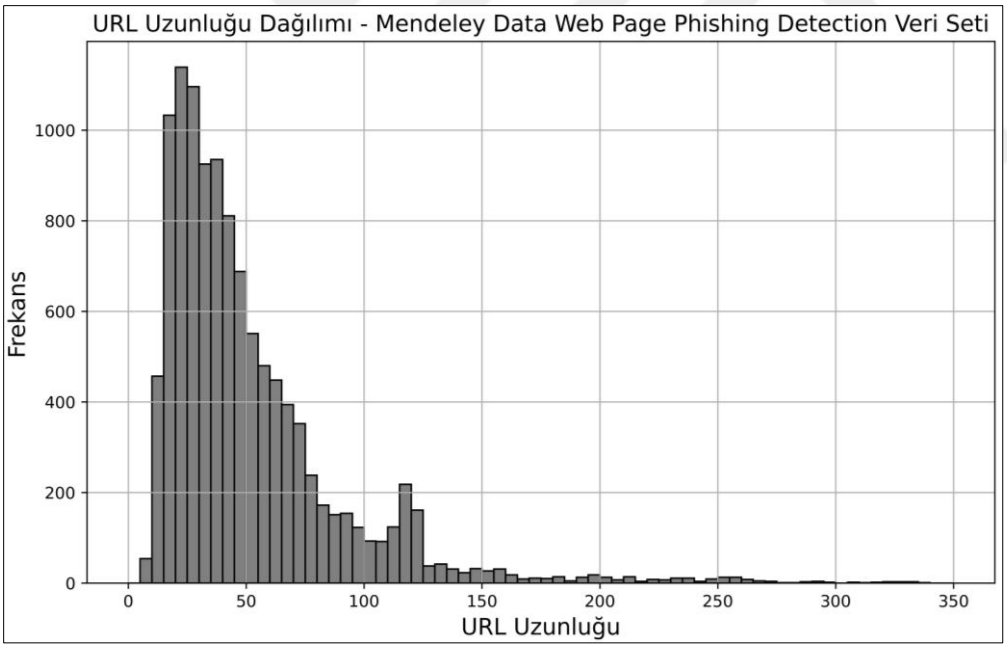
Şekil 4.1. GramBeddings veri setindeki URL uzunluklarının dağılımı



Şekil 4.2. Malicious and Benign URLs veri setindeki URL uzunluklarının dağılımı



Şekil 4.3. Ebbu2017 Phishing veri setindeki URL uzunluklarının dağılımı



Şekil 4.4. Mendeley Data Web Page Phishing Detection veri setindeki URL uzunluklarının dağılımı

4.2. Öznitelik Çıkarımı

Öznitelik çıkarma aşamasında, URL'ler karakter ve kelime düzeyinde analiz edilmiştir. Bunu sağlamak için karakterizasyon ve tokenizasyon işlemleri uygulanmıştır.

4.2.1. Karakterizasyon

URL adresleri, karakter düzeyinde analiz edilmiştir. Etiket kodlamasında olduğu gibi, kategorik verilerin sayısal olarak ifade edilmesi amacıyla her karaktere belirli bir sayısal değer atanmıştır. Çizelge 4.1, her karaktere atanan değeri gösterir. Bu işlem sonucunda karakter-değer (char, value) çiftlerinden oluşan bir karakter sözlüğü elde edilmiştir. Karakter sözlüğü 150 karakterden oluşur. Şekil 4.5'te, karakter sözlüğünden elde edilen karakterler, karakter bulutu olarak görselleştirilmiştir.

Çizelge 4.1. Karakter sözlüğü gösterimi

Karakter (karakter)	Kodlama (değer)
0 1 2 3 4 5 6 7 8 9	1-10
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	11-36
a b c d e f g h i j k l m n o p q r s t u v w x y z	37-62
' ' ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\\ ] ^ _ ` { | } ~ ¡ ¢ £ ¤ § ¨ © ª ® ¯ ° ³ ´ µ ¸ ¹ » ¼ ½ ¿ Â Ã Æ × â ã ä å æ ç è é ê ë ñ ö û ü š ž ƒ – — ' , " " • … ‰ € ™ \n	63-148
PAD	0
UNK	149

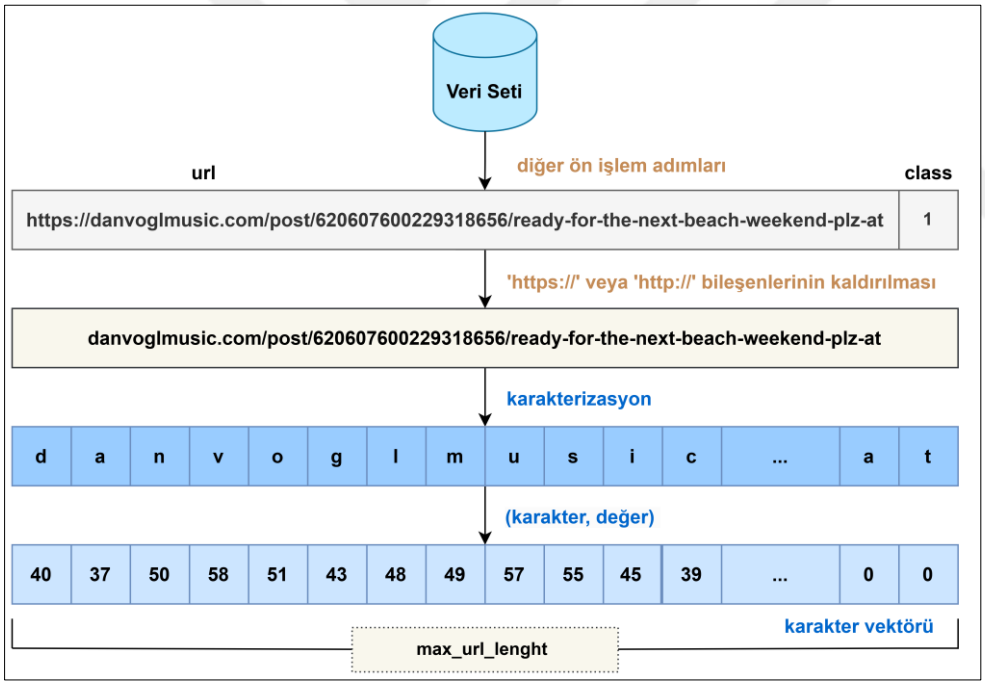
Karakter sözlüğünün oluşturulmasında Standart Amerikan Bilgi Değişim Kodu (ASCII) kodlaması kullanılmamıştır. ASCII, İngilizce dilinde kullanılan temel karakterleri kapsayan ve 128 farklı karakteri temsil eden 7 bitlik bir kodlama şemasıdır. Ancak, dil bağımlılığını engellemek için ASCII kodlaması kullanılmamıştır. Bu sayede, ASCII'de bulunmayan karakterleri içeren URL'lerin daha verimli şekilde işlenmesi sağlanmıştır.



Şekil 4.5. Karakter sözlüğünün karakter bulutu olarak görselleştirilmesi

Maksimum URL uzunluğu 250, 300 ve 350 karakter olarak belirlenmiş ve sonuçları elde etmek için deneyler gerçekleştirilmiştir. Maksimum URL uzunluğunun 300 olduğu varsayılır (Maksimum URL uzunluğunun neden 300 seçildiği, Bölüm 5'te deneysel

sonuçlarla desteklenerek açıklanmıştır). Bu durumda URL adresi, Çizelge 4.1'de belirtilen karakter sözlüğüne göre karakterler ve karşılık gelen değerlerle metinden sayısal forma çevrilir. Eğer URL uzunluğu 300 karakteri aşıyorsa, 300. karakterden sonraki bölüm kesilir ve sayısal gösterime dahil edilmez. URL uzunluğu 300 karakterden kısa olduğunda ise, eksik kısmı tamamlamak amacıyla 0 değeri eklenir. Bu durumda, sözlükte bulunan (PAD, 0) karakter çifti kullanılır. URL içerisindeki bir karakterin sözlükte bulunmaması durumunda, UNK karakteri devreye girer ve her bilinmeyen karakter için UNK karakterine karşılık gelen kodlanmış değerin indeksi bir artırılır. URL metni bu şekilde karakterize edilir ve sayısal temsillere dönüştürülür. URL adresleri, karakter vektörleri olarak adlandırılan sayısal temsillere dönüştürülmüş olur. Karakter vektörleri, tamsayı değerlerinden oluşan yüksek boyutlu vektörlerdir (Şekil 4.6). Bu vektörlerin oluşturulma yöntemi, Algoritma 1'de (Çizelge 4.4.) 22. ve 30. adımlar arasında açıklanmıştır.



Şekil 4.6. Karakterizasyon yaklaşımı

Şekil 4.6, GramBeddings veri setinde kimlik avı girişimi olarak tanımlanan bir URL adresini gösterir. Ön işleme adımlarından sonra, URL adresi kendisini oluşturan karakterlere ayrılır. max_url_length değeri 300 olarak ayarlanır. Daha sonra her karaktere bir değer atanır ve 300 uzunluğunda bir karakter vektörü oluşturulur. Elde edilen vektör yüksek boyutludur. Yüksek boyutlu karakter vektörlerine derin öğrenme algoritmalarını uygulamadan önce, düşük boyutlu, yoğun vektör temsilleri oluşturmak için gömme (embedding) katmanı kullanılır.

4.2.2. Karakter gömme katmanı

Önerilen modelin mimari yapısına Şekil 4.9'da yer verilmiştir. Her bir katmana giren ve çıkan verilerin çıkış şekli de Şekil 4.9'da gösterilmiştir. Gömme katmanı, yüksek boyutlu karakter vektörlerini düşük boyutlu ve yoğun vektör temsillerine dönüştürür. Bu katmana giriş olarak verilen vektörlerin şekli (300) iken, karakter gömme katmanından çıkan vektörün şekli (300, 128)'dir. Burada 128 değeri, karakter gömme katmanının çıkış boyutunu temsil eder. Gömme katmanı, modelin daha verimli çalışmasını sağlamak, anlamlı vektör temsilleri oluşturmak ve veriyi daha hızlı işleyerek süreci hızlandırmak amacıyla tercih edilmiştir. Yüksek boyutlu vektörlere kıyasla, düşük boyutlu yoğun vektörler daha az bellek tükettiğinden modelin eğitim maliyetini düşürür. Ayrıca, yoğun vektörlerin kullanımı, verilerin işlenmesini ve modelin eğitilmesini daha verimli hale getirir [27].

4.2.3. Tokenizasyon

URL adresi '/', '.', '-' gibi karakterlere göre belirteçlere (token) ayrılır. Her belirteç, bir kelimeyi ifade eder. Bu belirteçler, anlamlı veya anlamsız kelimeler, sayılar ve karakterlerden oluşabilir. URL'nin '/', '.', '-' gibi karakterler temel alınarak bölündüğünü belirtmek önemlidir. Çizelge 4.2'de çalışmada kullanılan veri setlerinden (GramBeddings, Malicious and Benign URLs ve Ebbu217 Phishing) elde edilen belirteçlerin istatistiksel bilgileri sunulmuştur. Çizelge 4.2 incelendiğinde, GramBeddings veri setinin toplam 5 954 984 belirteç ile en yüksek belirteç sayısına sahip olduğu görülür. GramBeddings veri setindeki belirteç sayısının daha fazla olması, veri setindeki URL adreslerinin sayısının diğer veri setlerine kıyasla daha fazla olmasından kaynaklanır. Belirteç sayısı, URL adresinin alt bölümlere ayrılabilecek belirteç sayısına bölünmesiyle hesaplanır. GramBeddings veri setinde, bir URL için maksimum belirteç sayısı 712'dir. Maksimum belirteç sayısı, belirteçlerin sayısal temsillere dönüştürülmesi sırasında vektör boyutunun belirlenmesinde önemli bir rol oynar. Minimum belirteç sayısı ise, bir URL'nin ayrılabileceği en düşük belirteç miktarını ifade eder. Çizelge 4.2, Malicious and Benign URLs veri setinin toplam 3 469 951 belirteç içerdiğini gösterir. Ebbu2017 Phishing veri seti ise 631 820 belirteç ile diğer veri setleri arasında en az belirteç içeren veri seti konumundadır. Malicious and Benign URLs veri seti için maksimum belirteç sayısı 166'dır. Ebbu2017 Phishing veri setindeki maksimum belirteç sayısı 99'dur. GramBeddings ve Malicious and Benign URLs veri setlerindeki minimum belirteç sayısı 1'dir. Ebbu 2017 Phishing veri setindeki minimum

belirteç sayısı 2'dir.

Çizelge 4.2. Veri setlerindeki belirteçlerin istatistiksel analizi

	GramBeddings Veri Seti [20]	Malicious and Benign URLs Veri Seti [36]	Ebbu2017 Phishing Veri Seti [6]
Toplam belirteç sayısı	5 954 984	3 469 951	631 820
Maksimum belirteç sayısı	712	166	99
Minimum belirteç sayısı	1	1	2
Ortalama belirteç sayısı	7,44	7,71	8,59

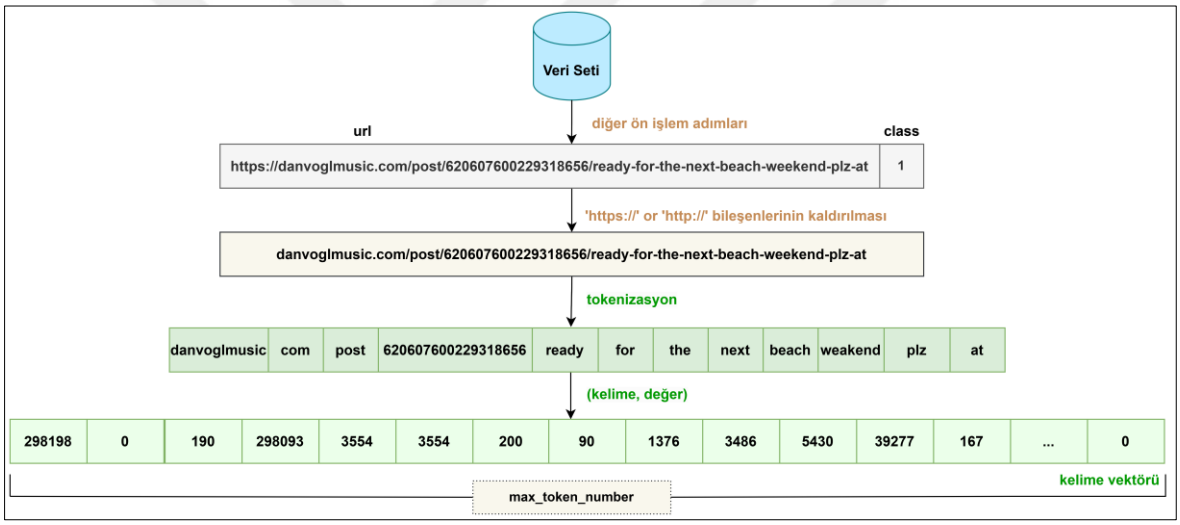
Ebbu2017 Phishing veri setinde en az sayıda belirteç içeren URL adresi 'tvfplay.com/' olarak belirlenmiştir. URL'den 'https://' ve 'http://' bileşenlerinin çıkarıldığına dikkat edilmelidir. URL'nin '.' ve '/' karakterlerine göre belirteçlere bölünmesiyle "tvfplay" ve "com" belirteçleri elde edilir. Aynı yöntem, diğer veri setlerindeki belirteç sayısını belirlemek için de uygulanmıştır. Çizelge 4.2'de gösterildiği üzere, Ebbu2017 Phishing veri setindeki ortalama belirteç sayısı, diğer veri setlerine kıyasla daha yüksektir. Şekil 4.7'de, GramBedding veri setinden elde edilen belirteçler kelime bulutu olarak görselleştirilmiştir.



Şekil 4.7. GramBeddings veri setindeki belirteçlerin kelime bulutu olarak görselleştirilmesi

Veri setlerinin her biri için ayrı bir belirteç kümesi oluşturulmuştur. Belirteçlerin sayısal temsili, Word2Vec yapay sinir ağı modeli kullanılarak elde edilmiştir. Bu model, belirteçler ile eğitilmiştir ve CBoW ile skip-gram algoritmalarını temel alarak çalışmaktadır. Bu çalışmada skip-gram algoritması kullanılmıştır. Skip-gram algoritması metinde sık geçmeyen nadir kelimeleri göz ardı etmez; nadir kelimelerin bağlamını öğrenir. Ayrıca, bir hedef kelimenin sağında ve solunda yer alan kelimeleri tahmin ettiği için CBoW'dan daha

fazla bilgi yoğunluğuna sahiptir. Bir URL adresindeki nadir bir kelime, URL adresini kimlik avına açık hale getirecek özelliklere sahip olabilir. Sonuç olarak, nadir kelimelerin de anlamsal analizini yapmak önemlidir. Word2Vec modeli, belirteçleri kullanarak bir eğitim süreci gerçekleştirir. Bu süreç, kelimeler arasındaki anlamsal ilişkileri yansıtan bir kelime sözlüğünün oluşturulmasına olanak tanır. Sözlük, kelimelerin ve bunlara karşılık gelen indeks değerlerinin bir listesinden meydana gelir. Kelime-değer (word, value) çiftleri oluşturularak, URL adreslerindeki kelimelerin vektör temsillerine dönüştürülmesi sağlanır. Vektörler, kelimelerin anlamlarını sayısal olarak ifade eden temsillerdir ve vektördeki her sayısal değer bir özniteliği gösterir. Bu vektörler, kelime vektörleri olarak adlandırılır ve genellikle yüksek boyutludur. Maksimum belirteç sayısı, kelime vektörlerinin boyutunu belirlemektedir (Şekil 4.8).



Şekil 4.8. Tokenizasyon yaklaşımı

Tablo 4.3'te, veri setlerinden elde edilen belirteçlere ve her veri seti için ayrı ayrı öğrenilen Word2Vec modeli tarafından oluşturulan kelime sözlüklerine yer verilmiştir. Sözlük, her kelimeyi ve ona karşılık gelen değeri (kelime, değer) içermektedir. Her veri seti için kelime sözlüğü analiz edilmiş ve aşağıdaki bulgular elde edilmiştir.

1. Sözlüklerde "login", "authentication" ve "update" gibi anlamlı kelimeler bulunabilir. Bunun yanı sıra, "slim", "yonks" gibi anlamsız kelimeler veya "927321" gibi sayılar da sözlükte yer alabilir. URL, "/", "." ve "-" gibi karakterlere göre belirteçlere ayrıldığından, yalnızca harflerden veya rakamlardan oluşan ifadeler de sözlüğe dahil edilebilir. URL belirteçlere ayrıldığı için tek karakterlik belirteçler de kelime olarak

değerlendirilir.

2. Benzer kelimeler sözlüklerde yer alabilir ancak aynı değeri taşımayabilir. Örneğin, “php” kelimesi GramBeddings veri setinin kelime sözlüğünde 2 olarak kodlanırken, Ebbu2017 Phishing veri setinde 3 olarak kodlanmıştır. Bu durum, sözlüğün oluşumunun, Word2Vec modelindeki öğrenme süreci doğrultusunda kelimeler arasındaki anlamsal ilişki ve benzerliklere dayanmasından kaynaklanır.
3. Her veri setinin boyutu farklı olduğundan, ilgili sözlüklerin boyutu da farklılık gösterir. Örneğin; GramBeddings veri setinin sözlüğü 858 959 kelime, Malicious and Benign URLs veri setinin sözlüğü 491 509 kelime ve Ebbu2017 Phishing veri setinin sözlüğü 68 263 kelime içerir.
4. Çizelge 4.2’de sunulan veri setlerindeki toplam belirteç sayısı ile Tablo 4.3’te verilen kelime sözlüğündeki toplam kelime sayısının uyuşmadığına dikkat edilmelidir. Bir kelimenin birden fazla URL’de belirteç olarak yer alabileceği dikkate alınmalıdır. Aynı kelime farklı URL’lerde tekrar edebilir. Kelime sözlüğü belirteçlerden oluşturulmuştur ve sözlükteki her bir giriş bir kelimeyi temsil etmektedir.

Çizelge 4.3. Veri setleri için kelime sözlüğü gösterimi

Kelime Sözlükleri					
GramBeddings Veri Seti [20]		Malicious and Benign URLs Veri Seti [36]		Ebbu2017 Phishing Veri Seti [6]	
Belirteç (kelime)	Kodlama (değer)	Belirteç (kelime)	Kodlama (değer)	Belirteç (kelime)	Kodlama (değer)
com	0	www	0	com	0
www	1	com	1	www	1
php	2	html	2	amp	2
html	3	php	3	php	3
login	4	org	4	html	4
...	...	...	...	...	...
verification	71	n	71	update	71
...	...	...	...	...	...
927321	565	charles	565	interactive	565
...	....	...	...	...	...
authentification	754	thomas	754	fi	754
...	....	...	...	...	...
rlzm	858 959	yonks	491 509	jesuscreed	68 263

4.2.4. Kelime gömme katmanı

Gömme katmanı, yüksek boyutlu kelime vektörlerini düşük boyutlu ve yoğun vektör temsillerine dönüştürür. Kelime gömme katmanının giriş verileri kelime vektörlerinden

oluşur. Şekil 4.9'da gösterildiği gibi, kelime vektörlerinin şekli (max_token_number) şeklindedir. Kelime gömme katmanının çıkış şekli (max_token_number, 128) şeklindedir. Kelime gömme katmanından elde edilen vektörün çıkış şekli GramBeddings veri seti için (712,128), Malicious and Benign URLs veri seti için (166,128) ve Ebbu2017 Phishing veri seti için (99,128)'dir. Kelime gömme katmanı, başlangıç ağırlık değerinin varsayılan olarak rastgele atanmadığı yönüyle karakter gömme katmanından ayrılmaktadır. Bu ağırlık değeri, kelime sözlüğündeki kelimelerin vektör temsillerinden elde edilir. Word2Vec modeli eğitildiği için, bu ağırlık değeri önceden öğrenilmiş olup kelime gömme katmanında başlangıç ağırlık değeri olarak kullanılır. Öte yandan, karakter gömme katmanında herhangi bir eğitim süreci bulunmadığından, başlangıç ağırlık değeri rastgele atanır. URL'nin belirteçlere ayrıştırılması ve kelime vektörlerinin oluşturulmasına yönelik sözde kod, Algoritma 1'de (Çizelge 4.4) 33. ve 51. adımlar arasında sunulmuştur.

4.3. Önerilen Model

Bu tez çalışması kapsamında önerilen modelin sözde kodu, Çizelge 4.4.'te "Algoritma 1" adıyla sunulmuştur. Önerilen modelin mimari yapısı, Algoritma 1'de açıklanmış ve Şekil 4.9'da görselleştirilmiştir.

Çizelge 4.4. Algoritma 1

Algoritma 1 URL'lerin sınıflandırılmasına yönelik önerilen derin öğrenme modelinin algoritması
Girdi: URL adresleri Çıktı: Oltalama (phishing) veya yasal (legitimate)
1: Veri seti yüklenir.
2: Gereksiz veriler veri setinden çıkarılır.
3: Eksik veriler tamamlanır.
4: Veri seti iki sütundan oluşur: URL ve class (sınıf/etiket).
5: 'class' sütunundaki kategorik veriler belirlenir ve etiket kodlama (label encoding) yapılır.
6: Yasal URL adreslerine 0 değeri atanır. (legitimate = 0)
7: Kimlik avı URL adreslerine 1 değeri atanır. (phishing = 1)
8: URL adresleri ve karşı geldiği sınıf farklı değişkenlere atanır.
9: X = dataset['url']
10: Y = dataset['class']
11: original_urls = X
12: for url in original_urls(len(url)) do
13: original_url_lenghts = [] % URL'lerin uzunluklarının listesi oluşturulur.
14: end for
15: max_url = max(original_urls, key=len)
16: min_url = min(original_urls, key=len)
17: total_urls = len(original_urls)
18: average_length = sum(original_url_lenghts) / total_urls

Çizelge 4.4. (devam) Algoritma 1

19: cleaned_urls = []
20: https:// ve http:// bileşenleri URL adreslerinden çıkarılır.
21: max_url_length = 300
22: Karakter sözlüğündeki karakterler tanımlanır.
23: Karakter sözlüğü oluşturulur (char_dict)
24: for index, char in enumerate(allowed_characters) do
25: add the pair (character, index + 1) to char_dict
26: char: index + 1
27: end for
28: char_dict[‘<PAD>’] = 0 % Dolgu (padding) karakteri için 0 eklenir.
29: char_dict[‘<UNK>’] = len(char_dict) % Bilinmeyen karakter için yeni bir indeks eklenir.
30: URL adresleri karakter vektörlerine dönüştürülür.
31: Veri seti %80 eğitim (X_ train_char) ve %20 test (X_ test_char) olarak ayrılır.
32: Eğitim veri seti %80 eğitim ve %20 doğrulama olarak ayrılır.
33: URL’ler belirteçlere ayrılır (tokenize_url)
34: dataset[‘tokens’] = dataset[‘url’].apply(tokenize_url)
35: Belirteçler için Word2Vec modeli oluşturulur (sentences, vector_size, window, min_count, workers, sg)
36: Kelime sözlüğü oluşturulur ((key, index)) % key=kelime, index=değer (word, value)
37: Kelime gömme matrisi oluşturulur.
38: for word, i in word2vec_model.wv.key_to_index.items() do
39: embedding_matrix[i] = word2vec_model.wv[word] %embedding_matrix, kelime gömme katmanında önceden eğitilmiş bir ağırlık olarak kullanılır.
40: end for
41: token_lists = dataset[‘tokens’]
42: for tokens in token_lists do
43: token_number = len(tokens)
44: end for
45: total_tokens = sum(token_number)
46: min_token_number = min(token_ number) %Tüm URL’lerin tokenize edilmiş hallerindeki en kısa token sayısını temsil eder.
47: max_token_number = max(token_number) %Tüm URL’lerin tokenize edilmiş hallerindeki en uzun token sayısını temsil eder.
48: average_token_number = total_tokens / len(token_lists)
49: Veri seti %80 eğitim (X_train_word) ve %20 test (X_test_word) olarak ayrılır.
50: Eğitim veri seti %80 eğitim ve %20 doğrulama olarak ayrılır.
51: Belirteçlere ayrılmış URL’ler matematiksel temsillere/sayısal dizilere (kelime vektörleri) dönüştürülür.
52: Model oluşturulur.
53: Karakter dizisini almak için bir giriş katmanı (character_input) oluşturulur. % Bu yaklaşım, URL’lerin karakter düzeyinde bir temsilini sağlar.
54: Karakter girdisi, karakterlerin vektör temsilini içeren bir gömme katmanına aktarılır. %char_dict sözlüğü karakterlerin indekslerini temsil eder.
55: Gömme katmanının çıktısı Çift Yönlü LSTM (BiLSTM) katmanına aktarılır. Bu katman hem ileri hem de geri sıralı bilgileri kullanır.
56: BiLSTM’nin çıktısı bir dikkat katmanına iletilir. %Bu, BiLSTM çıktısının belirli kısımlarına odaklanılmasını sağlar.
57:Dikkat katmanının çıktısı bir küresel maksimum havuzlama katmanına (GlobalMaxPooling1D) iletilir. %Bu işlem, dikkat katmanının çıktısının en önemli değerlerini alarak özelliklerin boyutunu azaltır.
58: Aşırı uyumu önlemek için modele bir bırakma katmanı (Dropout) eklenir.

Çizelge 4.4. (devam) Algoritma 1

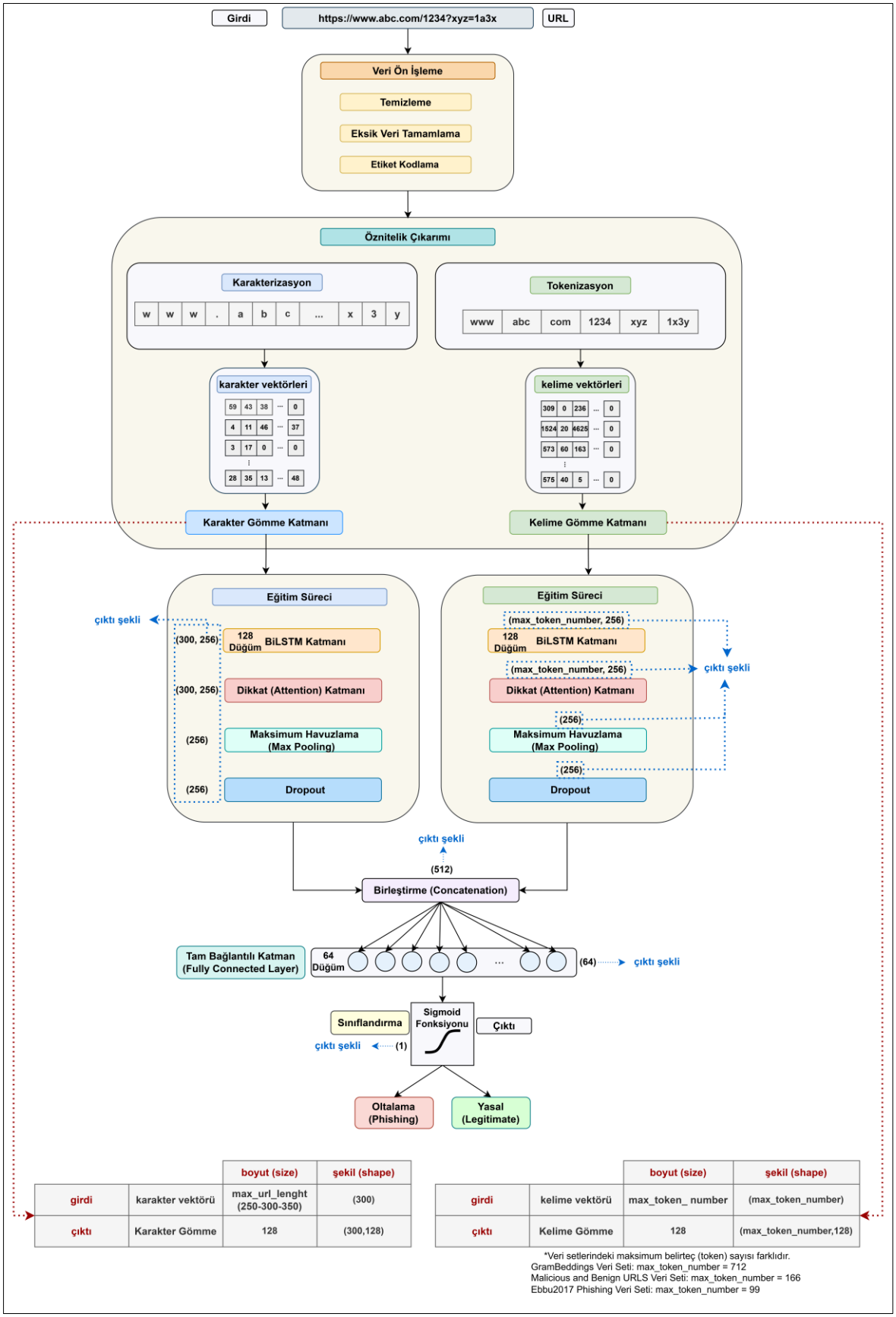
59: Kelime dizisini almak için bir giriş katmanı (word_input) oluşturulur. Bu yaklaşım, URL'lerin kelime düzeyinde temsil edilmesine izin verir.
60: Kelime girişi, önceden eğitilmiş bir kelime gömme matrisi kullanılarak bir gömme katmanına aktarılır. Bu matris Word2Vector modelinin kelime vektörlerini içerir ve eğitilemez.
61: Gömme katmanının çıktısı bir Çift Yönlü LSTM katmanına iletilir. Bu katman hem ileri hem de geri sıralı bilgileri kullanır.
62: BiLSTM'nin çıktısı bir dikkat katmanına iletilir.
63:Dikkat katmanının çıktısı bir küresel maksimum havuzlama katmanına (GlobalMaxPooling1D) iletilir.
64: Aşırı uyumu önlemek için bir bırakma katmanı (Dropout) eklenir.
65: Karakter ve kelime tabanlı özellik çıktıları tek vektörde birleştirilir.
66: ReLU aktivasyon fonksiyonu kullanılarak tam bağlı bir katman oluşturulur.
67: Sigmoid aktivasyon fonksiyonuna sahip bir çıkış katmanı eklenir.
68: Model, Adam optimizasyon algoritması ve ikili çapraz entropi kayıp fonksiyonu ile derlenir.
69: for epoch in range (30) do
70: Doğruluk (accuracy) ve doğrulama doğruluğu (validation accuracy) değerlendirilir.
71: Kayıp (loss) ve doğrulama kaybı (validation loss) değerlendirilir.
72: Karışıklık matrisi (confusion matrix) değerlendirilir.
73: Kesinlik (precision), duyarlılık (recall), f1-skor, ROC eğrisi, AUC değerlendirilir.
74: end for

4.3.1. BiLSTM katmanı

BiLSTM, çift yönlü ileri ve geri besleme mekanizması ile çalışan gelişmiş bir LSTM modelidir. Bu model, sıralı verileri işleyerek öğeler arasındaki ilişkileri öğrenir. URL verileriyle kullanılmaya uygun olan bu algoritma, çift yönlü çalışabilme özelliği sayesinde verilerden daha fazla bilgi çıkarabilir [43]. Şekil 4.9, karakter gömme katmanının BiLSTM modeline girişini gösterir. BiLSTM modeli 128 nörona/düğüme sahiptir. Çift yönlü çalıştığı için çıkış şekli 256 olur. BiLSTM katmanının çıktısı, daha sonra dikkat (attention) katmanına giriş olarak iletilir ve çıkış şekli (300, 256) olur. Buradaki 300 değeri, bir URL'nin maksimum uzunluğunu ifade eder. Aynı süreç, kelime gömme katmanından gelen veriler için de gerçekleştirilir. Kelime tabanlı öznitelikleri çıkarmak amacıyla, 128 nörona sahip BiLSTM algoritması kullanılır. Bu işlemin sonucunda, çıkış şekli (max_token_number, 256) olur. Şekil 4.10'da gösterildiği üzere, GramBeddings veri setinin maksimum belirteç sayısı 712 olduğundan, çıkış şekli (712, 256) olur.

4.3.2. Dikkat (Attention) katmanı

Dikkat (Attention) katmanı, bir ağırlıklandırma sistemi kullanarak farklı girdilere, işleyiş mantığına göre farklı ağırlıklar atar. Dikkat ağırlığı, belirli girdilerin öncelikli olarak



		boyut (size)	şekil (shape)
girdi	karakter vektörü	max_url_lenght (250-300-350)	(300)
çıktı	Karakter Gömme	128	(300,128)

		boyut (size)	şekil (shape)
girdi	kelime vektörü	max_token_ number	(max_token_number)
çıktı	Kelime Gömme	128	(max_token_number,128)

*Veri setlerindeki maksimum belirteç (token) sayısı farklıdır.
GramBeddings Veri Seti: max_token_number = 712
Malicious and Benign URLS Veri Seti: max_token_number = 166
Ebbu2017 Phishing Veri Seti: max_token_number = 99

Şekil 4.9. Önerilen modelin mimarisi

değerlendirilmesini sağlamak için hesaplanır. Böylece model, önemli bilgilerin işlenmesine öncelik tanır. Ayrıca, dikkat katmanı, uzun dizilerde bilgi kaybını engelleyerek modelin doğruluğunu ve performansını yükseltir [44]. Çalışma prensibi şu şekildedir: Süreç üç temel bileşenden oluşur: sorgu (query), anahtar (key) ve değer (value). Bu bağlamda, sorgu (query), dikkate alınması gereken kelimeyi veya karakteri temsil eder. Anahtar (key), tüm girdileri tanımlamak için kullanılırken, değer (value), bu girdilerin gerçek içeriklerini ifade eder. Öncelikle, sorgu ve anahtar girdileri arasındaki benzerlik hesaplanır. Bu benzerlikler, softmax fonksiyonu aracılığıyla normalleştirilerek ağırlıklar elde edilir. Daha sonra, hesaplanan ağırlıklar tüm değerlerle çarpılıp toplanarak dikkat katmanının çıktısı oluşturulur. Sonuç olarak, girdiler (sorgu, anahtar ve değer) işlenerek bir dikkat vektörüne (attention vector) dönüştürülür [44].

Eş. 4.1'de, q, k ve v sembolleri sırasıyla sorgu (query), anahtar (key) ve değer (value) bileşenlerini temsil eder. d_k sembolü, anahtar vektörlerinin boyutunu ifade eder. q ve k arasındaki işlem, nokta çarpımı (dot product) işlemidir. A sembolü, dikkat ağırlıklarının toplamını (attention weight sum) belirtir.

$$A(q,k,v) = softmax\left(\frac{q.k^t}{\sqrt{d_k}}\right) * v \tag{4.1}$$

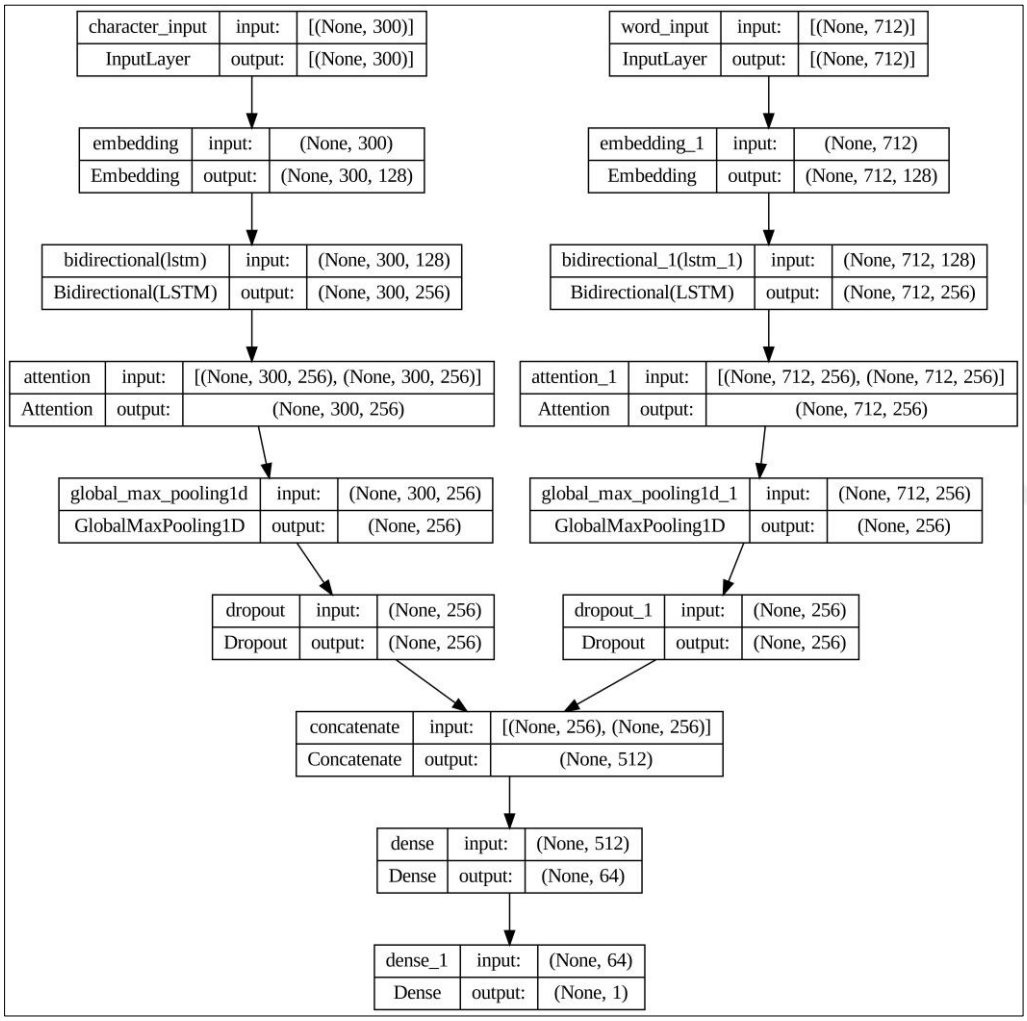
Dikkat katmanının çıkış şekli, karakter tabanlı giriş için (300, 256) ve kelime tabanlı giriş için (max_token_number, 256) şeklindedir. Şekil 4.9'da, GramBeddings veri kümesi kullanıldığı için çıkış şekli (712, 256) olmuştur.

4.3.3. Maksimum havuzlama (Max pooling)

Maksimum havuzlama (Max Pooling) katmanı, önceki katmandaki özniteliklerin en büyük değerlerini seçerek önemli bilgileri vurgular. Giriş verisinin boyutu, çıkış verisinin boyutundan farklıdır. Şekil 4.9 ve Şekil 4.10'da görüldüğü üzere, maksimum havuzlama katmanından çıkan verinin çıkış şekli (256)'dır. Veri boyutu küçültüldüğünden, hesaplama maliyeti azalır ve model daha hızlı çalışır.

4.3.4. Düğüm bırakma (Dropout)

Dropout işlemi sırasında, belirli nöronlar rastgele seçilerek devre dışı bırakılır ve bu nöronlarla herhangi bir bağlantı kurulmaz. Dropout, modelin aşırı öğrenmesini (overfitting) engellemek amacıyla uygulanır. Bu işlemde dropout oranı rastgele 0,2 olarak belirlenmiştir. Şekil 4.9 ve Şekil 4.10'da gösterildiği gibi, Dropout katmanına giriş olarak verilen verinin giriş şekli (256)'dır. İşlem sonrası, çıkış şekli de (256) olarak kalmaktadır.



Şekil 4.10. GramBeddings veri setinde önerilen modelin katmanları

4.3.1. Birleştirme ve sınıflandırma

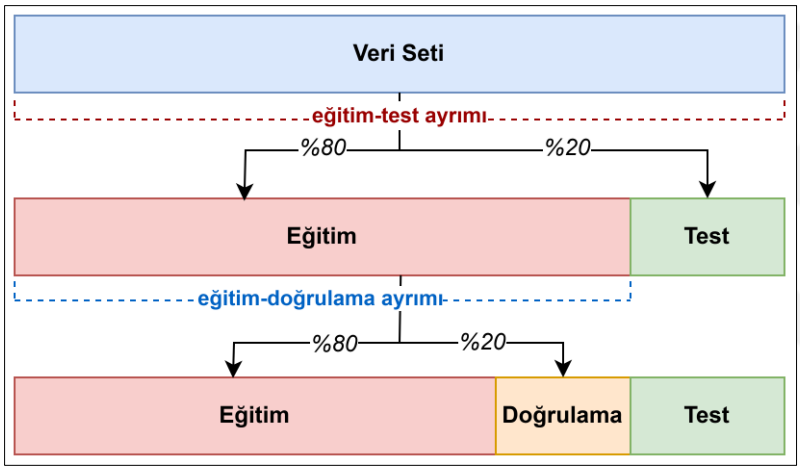
Eğitim süreci, karakter ve kelime tabanlı öznitelikler için bağımsız olarak yürütülmektedir. Karakter vektörleri, karakter tabanlı öznitelikleri içeren yüksek boyutlu vektörlerdir. Karakter gömme (embedding) katmanında, yüksek boyutlu vektörler düşük boyutlu ve

yoğun vektör temsillerine dönüştürülerek, derin öğrenme algoritmalarında sınıflandırma için uygun hale getirilir. Eğitim, BiLSTM, dikkat (attention), maksimum havuzlama (max pooling) ve dropout katmanlarını içeren katmanlar içinde gerçekleştirilir. Her katmanda farklı bir işlem uygulanır. Benzer bir süreç kelime tabanlı öznitelikler için de takip edilir. Kelime gömme (word embedding) katmanı, yüksek boyutlu kelime vektörlerini düşük boyutlu ve yoğun vektör temsillerine dönüştürür. Daha sonra, kelime tabanlı özniteliklerin eğitimi BiLSTM, dikkat (attention), maksimum havuzlama (max pooling) ve dropout katmanlarında gerçekleştirilir. Eğitim tamamlandıktan sonra karakter ve kelime tabanlı modellerin çıktıları birleştirme (concatenate) katmanında birleştirilir. Bu işlem, daha kapsamlı, güçlü ve zengin bir veri temsili oluşturur. Birleştirme katmanında, karakter ve kelime tabanlı öznitelikler bir araya getirilerek çıkış şekli (512) olur. Tam bağlı (fully connected) katman, birleştirilmiş öznitelikleri işleyerek daha üst düzey temsiller elde eder. Bu katmanda nöron sayısı 64 olarak belirlenmiştir ve giriş verisinin şekli (512) iken, bu katmandan sonra (64) olarak dönüşür. Bu katmanda ReLU aktivasyon fonksiyonu (Eş. 3.13) kullanılır. Son aşamada, çıkış katmanında ikili sınıflandırma (binary classification) işlemi gerçekleştirildiğinden, sigmoid aktivasyon fonksiyonu (Eş. 3.5) kullanılarak sınıflandırma yapılır. Bu işlem sonucunda, URL'nin oltalama (phishing) (1) veya yasal (legitimate) (0) olduğu belirlenir. Bu süreçlerin tamamı, Algoritma 1'de (Çizelge 4.4) adım 52 ile 68 arasında açıklanmıştır. Şekil 4.9'da, önerilen model görsel olarak sunulmuştur. Şekil 4.10, GramBeddings veri seti kullanıldığında önerilen modelin katman yapısını gösterir. Ayrıca, Şekil 4.10'da her katmanın giriş ve çıkış şekillerine de yer verilmiştir.

4.4. Eğitim-Doğrulama-Test

K-katlı çapraz doğrulama yöntemi, özellikle GramBeddings veri kümesinin büyük boyutu nedeniyle eğitim sürecinde Rastgele Erişim Belleği (RAM) üzerinde aşırı yüklenmeye sebep olmuştur. Bu yöntem, veri kümesini k alt parçaya ayırarak her bir parçayı eğitim ve doğrulama aşamalarında tekrar kullandığından, her kat için büyük veri parçalarının aynı anda RAM'de depolanmasını gerektirir. Bu durum yetersiz RAM ve performans sorunlarına yol açmıştır. Bu nedenle, veri kümeleri başlangıçta 80-20 oranında rastgele iki farklı alt kümeye bölünmüş, biri eğitim diğeri test için belirlenmiştir. Daha sonra, eğitim veri kümesi yine 80-20 oranında eğitim ve doğrulama alt kümelerine rastgele bölünmüştür (Şekil 4.11). Veri setinin üç farklı kategoriye (eğitim, doğrulama ve test) ayrılmasının arkasındaki mantık, makine ve derin öğrenme modellerinin doğru bir şekilde değerlendirilmesini sağlamak ve

genelleme yeteneklerini geliştirmektir. Deneysel çalışmalar Google Colab üzerinde NVIDIA Tesla T4 GPU kullanılarak gerçekleştirilmiştir. Tüm veri setleri bu ortamda eğitilmiş ve test edilmiştir. Deneyler, yüksek RAM kapasitesine sahip bulut tabanlı Google sunucularında yürütülmüştür. Sunucu altyapısı, 51 GB sistem RAM'i, 15 GB GPU ve 201,2 GB disk belleğinden oluşur. Python 3 programlama dili kullanılmıştır. Keras ve TensorFlow kütüphanelerinden yararlanılmıştır. Bu kütüphanelerin 2.15 sürümü tercih edilmiştir. Eğitim, doğrulama ve test ayrımı için sözde kod, Algoritma 1'de (Çizelge 4.4), karakter tabanlı öznitelikler için 31 ve 32. adımlarda, kelime tabanlı öznitelikler için ise 49 ve 50. adımlarda verilmiştir.



Şekil 4.11. Veri setlerinin eğitim, doğrulama ve test olarak ayrılması

5. DENEYSEL BULGULAR

Bu bölümde, Şekil 4.9'da gösterilen önerilen modelin adım adım geliştirme süreci sunulmuş ve deneysel bulgular neden-sonuç ilişkisi çerçevesinde tartışılmıştır. Deneysel bulgular bölümü, önerilen modelin geliştirme sürecini açıklamak amacıyla soru-cevap yaklaşımını temel alır. Modelin performansını en üst seviyeye çıkarmak ve en iyi sonuçları elde etmek amacıyla çeşitli sorular sorulmuş, deneyler yapılmış ve elde edilen yanıtlar doğrultusunda geliştirme süreci yönlendirilmiştir.

5.1. Sınıflandırma Metrikleri

Bir URL adresinin kimlik avı girişimi mi yoksa yasal bir site mi olduğunu belirlemek, doğası gereği bir sınıflandırma problemidir. Bu nedenle, sistemin başarımı sınıflandırma metrikleri kullanılarak değerlendirilmiştir. Sınıflandırma sürecinde, verilerin doğru bir şekilde kategorize edilip edilmediğini analiz etmek amacıyla karışıklık matrisi (confusion matrix) kullanılmıştır. Karışıklık matrisine göre, tahmin edilen ve gerçek değerlerin her ikisinin de pozitif olduğu durumlar doğru pozitifler (True Positive - TP) olarak sınıflandırılır. Buna karşılık, modelin negatif tahminde bulunmasına rağmen, gerçek değer pozitif olduğunda yanlış negatifler (False Negative - FN) oluşur. Benzer şekilde, modelin pozitif tahminde bulunmasına rağmen gerçek değerin negatif olması yanlış pozitifler (False Positive - FP) olarak tanımlanırken, hem tahmin edilen hem de gerçek değerin negatif olduğu durumlar doğru negatifler (True Negative - TN) olarak kabul edilir. Bu kategoriler, modelin performansını ölçmek için kullanılan doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1-skor ve Eğri Altındaki Alan (Area Under Curve - AUC) gibi çeşitli istatistiksel metriklerin hesaplanmasında temel alınmıştır. Doğruluk, bir modelin yaptığı tüm tahminler içindeki doğru tahminlerin oranını ifade eder. Eş. 5.1'de verilen formül kullanılarak hesaplanılır.

$$Doğruluk\ (Accuracy) = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Kesinlik (Precision), model tarafından pozitif olarak tahmin edilen tüm örnekler içinde doğru tahmin edilen pozitif örneklerin oranı olarak tanımlanır. Eş. 5.2 ile hesaplanan bu metrik, modelin pozitif tahminlerinin doğruluğunu gösterir.

$$Kesinlik\ (Precision) = \frac{TP}{TP + FP} \tag{5.2}$$

Duyarlılık (Recall) metriği, modelin pozitif sınıfa ait örnekleri doğru bir şekilde tespit etme oranını ifade eder ve Eş. 5.3 kullanılarak hesaplanır. Genel anlamda, modelin pozitif örnekleri ne kadar başarılı şekilde tanıyabildiğini ölçen bir performans göstergesidir.

$$Duyarlılık\ (Recall) = \frac{TP}{TP + FN} \tag{5.3}$$

F1-skor, kesinlik (precision) ve duyarlılık (recall) değerlerinin harmonik ortalamasını hesaplayan ve bu iki metriği tek bir değerle özetleyen istatistiksel bir ölçüttür. F1-skor, farklı kesinlik ve doğruluk değerleri gösteren modeller arasından en uygun modeli belirlemek için kullanılır. 0 ile 1 arasında bir değere sahiptir; 1 en yüksek, 0 ise en düşük değeri temsil eder. F1-skorun hesaplanması için Eş. 5.4'te verilen denklem kullanılır.

$$F1 - Skor = 2 * \left(\frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık}\right) \tag{5.4}$$

Doğru Pozitif Oranı (True Positive Rate - TPR), modelin gerçek kimlik avı (phishing) URL'lerini doğru tahmin etme oranını gösteren bir metriktir. Yüksek bir TPR değeri, modelin kimlik avı URL'lerini başarıyla tespit edebildiğini gösterir. Bu ölçüm, Eş. 5.5 kullanılarak hesaplanır. TPR, duyarlılık (recall) metriği ile aynı anlama gelmektedir.

$$TPR = \frac{TP}{TP + FN} \tag{5.5}$$

Doğru Negatif Oranı (True Negative Rate - TNR), modelin gerçek yasal (legitimate) URL'leri doğru tahmin etme oranını temsil eder. Bu metrik, Eş. 5.6 kullanılarak hesaplanır. Yüksek bir TNR değeri, modelin yasal URL'leri doğru şekilde ayırt edebilme yeteneğinin güçlü olduğunu ifade eder.

$$TNR = \frac{TN}{TN + FP} \tag{5.6}$$

Yanlış Pozitif Oranı (False Positive Rate - FPR), modelin yasal (legitimate) URL'leri hatalı

bir şekilde kimlik avı (phishing) olarak sınıflandırma oranını ifade eder ve Eş. 5.7 ile hesaplanır. FPR değerinin düşük olması, modelin yasal URL'leri doğru bir şekilde tanıyabildiğini gösterir.

$$FPR = \frac{FP}{FP + TN} \tag{5.7}$$

Yanlış Negatif Oranı (False Negative Rate - FNR), modelin kimlik avı (phishing) URL'lerini hatalı bir şekilde yasal (legitimate) olarak sınıflandırma oranını ifade eder. Bu metrik, Eş. 5.8 kullanılarak hesaplanır. FNR değerinin düşük olması, modelin kimlik avı URL'lerini doğru şekilde ayırt edebildiğini gösterir.

$$FNR = \frac{FN}{FN + TP} \tag{5.8}$$

Alıcı İşletim Karakteristiği (Receiver Operating Characteristic - ROC) eğrisi, x ekseninde Yanlış Pozitif Oranı (FPR) ve y ekseninde Doğru Pozitif Oranı (TPR) arasındaki ilişkiyi görselleştiren bir grafiktir. Modelin performansı, ROC eğrisinin y eksenine yakınlığı ile değerlendirilir. ROC eğrisinin daha yüksek ve merkezi bir yapıya sahip olması, modelin üstün performans sergilediğini gösterir. Eğri Altındaki Alan (AUC), ROC eğrisinin altındaki toplam alanı ifade eder ve modelin genel başarısını özetleyen bir metriktir. Bu değer, modelin kimlik avı (phishing) ve yasal (legitimate) URL'leri ayırt etme yeteneğini ölçer ve 0 ile 1 arasında değişen bir aralığa sahiptir. 1'e yaklaşan AUC değeri, modelin iki sınıfı başarılı bir şekilde ayırt edebildiğini gösterir [29]. Sonuç olarak, yüksek bir TPR, düşük bir FPR ve büyük bir AUC değerine sahip bir ROC eğrisi, modelin kimlik avı URL'lerini doğru şekilde tespit ederken yanlış pozitif oranını en aza indirme yeteneğine sahip olduğunu gösterir.

5.2. Deneyler

Bu bölümde, yapılan deneyler ve elde edilen bulgular detaylı olarak açıklanmıştır. Modelin geliştirilmesi, ayrıntılı deneysel sonuçlara dayandırılmıştır.

5.2.1. En iyi optimizasyon algoritmasının seçimi

URL adreslerinin kimlik avı (phishing) veya yasal (legitimate) olarak sınıflandırılması bir ikili sınıflandırma problemi olduğundan, bu çalışmada ikili çapraz entropi (binary cross-entropy) kayıp fonksiyonu kullanılmıştır. Kayıp fonksiyonu, gerçek ve tahmin edilen sınıflar arasındaki farkı değerlendirerek derin öğrenme modelinin verimliliğini artırır. Derin öğrenme modellerinin eğitimi sırasında, kayıp fonksiyonunu minimize etmek ve modelin sınıflandırma başarımını yükseltmek amacıyla optimizasyon algoritmalarından faydalanılır. Bu doğrultuda, Stokastik Gradyan İnişi (Stochastic Gradient Descent - SGD), Uyarlanabilir Momentum Tahmini (Adaptive Moment Estimation - Adam) ve Karekök Ortalama Yayılımı (Root Mean Square Propagation - RMSprop) en yaygın kullanılan optimizasyon algoritmaları arasındadır [27]. Bu çalışmada, önerilen model için en uygun optimizasyon algoritmasını belirlemek amacıyla SGD, Adam ve RMSprop algoritmaları, Malicious and Benign URLs ile Ebbu2017 Phishing veri setleri üzerinde eğitilmiş ve test edilmiştir. Analiz sürecinde, görece daha küçük boyutlu olmaları nedeniyle bu veri setleri tercih edilmiştir. Sınıflandırma işlemi için LSTM algoritması kullanılmış, model eğitimi 0,01 öğrenme oranı ile 10 epoch boyunca gerçekleştirilmiştir.

Çizelge 5.1. Modelde kullanılacak optimizasyon algoritmasının seçimi

Veri Seti	Optimizasyon Algoritması	Test Doğruluk Değeri	Öğrenme Oranı	Epoch	Sınıflandırma Algoritması
Malicious and Benign URLs Veri Seti [36]	SGD	0,7695	0,01	10	LSTM
	Adam	0,9656	0,01	10	LSTM
	RMSprop	0,7680	0,01	10	LSTM
Ebbu2017 Phishing Veri Seti [6]	SGD	0,5053	0,01	10	LSTM
	Adam	0,9405	0,01	10	LSTM
	RMSprop	0,4967	0,01	10	LSTM

LSTM algoritmasının ilk deneyde tercih edilme nedeni, literatürde karakter dizilerindeki anlamsal ve sıralı bağımlılık özniteliklerini yakalamada yaygın olarak kullanılmasıdır. Deneyden elde edilen test doğruluk oranları karşılaştırmalı bir analizle değerlendirilmiştir. Çizelge 5.1'de gösterildiği üzere, Adam algoritması diğer algoritmalara göre daha yüksek performans göstermiş ve Malicious and Benign URLs veri setinde 0,9656; Ebbu2017 Phishing veri setinde ise 0,9405 doğruluk değeri elde etmiştir. Ayrıca, Wang ve ark. [23],

Afzal ve ark. [24] ve Özcan ve ark. [29] çalışmalarında Adam optimizasyon algoritması ile başarılı sınıflandırma sonuçları elde etmişlerdir. Çizelge 5.1'de sunulan bulgular ve literatürdeki benzer çalışmaların sonuçları göz önünde bulundurularak, bu çalışmada Adam algoritmasının kullanılması tercih edilmiştir.

5.2.2. Karakter tabanlı öznitelikler için algoritma seçimi

Literatürde, kimlik avı saldırılarını sınıflandırmak amacıyla farklı algoritmalar uygulanmıştır. Rao ve ark. [31] LSTM'i kullanırken, Yang ve ark. [27] LSTM ve CNN'i birleştirmişlerdir. Roy ve ark. [33] LSTM ve BiLSTM algoritmalarını kullanmışlardır. Özcan ve ark. [29] ise LSTM ve DNN algoritmalarını uygulamışlardır. Bu çalışmada, karakter tabanlı öznitelikleri eğitmek ve önerilen modeli geliştirmek için en uygun algoritmayı belirlemek amacıyla LSTM, BiLSTM, CNN ve DNN algoritmaları ile deneyler gerçekleştirilmiştir. Eğitim, doğrulama ve test sonuçlarının özeti Çizelge 5.2'de sunulmuştur.

Çizelge 5.2. Karakter tabanlı özniteliklerin eğitimi için algoritma seçimi

Veri Seti	Algoritma	Eğitim Doğruluk Değeri	Doğrulama Doğruluk Değeri	Test Doğruluk Değeri	Eğitim Süresi (s)	Epoch
Malicious and Benign URLs Veri Seti [36]	LSTM	0,9888	0,9869	0,9869	1286,10	10
	BiLSTM	0,9901	0,9886	0,9881	2246,09	10
	CNN	0,9740	0,9722	0,9716	564,50	10
	DNN	0,9660	0,9645	0,9647	435,51	10
Ebbu2017 Phishing Veri Seti [6]	LSTM	0,9354	0,9283	0,9308	326,42	10
	BiLSTM	0,9493	0,9491	0,9474	455,35	10
	CNN	0,8548	0,8361	0,8270	143,08	10
	DNN	0,7443	0,7441	0,7460	83,69	10

Karakter tabanlı özniteliklerin eğitimi için en uygun algoritmayı belirlemek amacıyla, Malicious and Benign URLs ile Ebbu2017 Phishing veri setleri üzerinde eğitim ve test işlemleri gerçekleştirilmiştir. Bu iki veri seti, GramBeddings veri kümesine kıyasla daha küçük boyutlu olmaları ve biri dengeli, diğeri ise dengesiz veri dağılımına sahip olması nedeniyle analiz amacıyla seçilmiştir. Ayrıca, model geliştirme sürecinin sonraki aşamalarında GramBeddings veri kümesi üzerinde de deneyler gerçekleştirilmiştir. Çizelge 5.2'de sunulan sonuçların analizi doğrultusunda, BiLSTM algoritmasının eğitim, doğrulama

ve test doğruluk değerlerinin diğer algoritmalardan daha yüksek olduğu gözlemlenmiştir. BiLSTM algoritması çift yönlü çalıştığından, eğitim süresi diğer algoritmalara kıyasla daha uzun sürmesine rağmen, performans açısından daha başarılı sonuçlar elde edilmiştir. Sonuç olarak, karakter tabanlı özniteliklerin eğitimi için BiLSTM algoritması tercih edilmiştir.

5.2.3. Kelime tabanlı öznitelikler için algoritma seçimi

Bu bölümde, kelime tabanlı özniteliklerin eğitimi için en uygun algoritmayı belirlemeye yönelik gerçekleştirilen çeşitli deneylerin sonuçlarına yer verilmiştir. URL adreslerinden elde edilen belirteçleri (token) vektör temsillerine dönüştürmek amacıyla Word2Vec modeli kullanılmıştır. Kelime vektör temsillerinin oluşturulmasında Word2Vec modelinde hangi algoritmanın tercih edilmesi gerektiğini belirlemek için skip-gram ve CBoW algoritmaları üzerinde deneyler yapılmıştır. Skip-gram algoritmasının sonuçları Çizelge 5.3'de, CBoW algoritmasının sonuçları ise Çizelge 5.4'te gösterilmiştir. URL adreslerinden elde edilen belirteçler vektör temsillerine çevrilmiş ve kelime tabanlı özniteliklerin eğitim, doğrulama ve test süreçleri LSTM, BiLSTM, CNN ve DNN algoritmaları kullanılarak uygulanmıştır.

Çizelge 5.3. Word2Vec modelinde skip-gram algoritması kullanılarak kelime tabanlı özniteliklerin eğitimi için algoritma seçimi

Veri Seti	Algoritma	Eğitim Doğruluk Değeri	Doğrulama Doğruluk Değeri	Test Doğruluk Değeri	Eğitim Süresi (s)	Epoch	Word2Vec Algoritması
Malicious and Benign URLs Veri Seti [36]	LSTM	0,7700	0,7704	0,7698	1346,56	10	Skip-gram
	BiLSTM	0,9999	0,9914	0,9918	1839,94	10	Skip-gram
	CNN	0,7679	0,7685	0,7680	508,22	10	Skip-gram
	DNN	0,7679	0,7685	0,7680	442,87	10	Skip-gram
Ebbu2017 Phishing Veri Seti [6]	LSTM	0,5042	0,5041	0,5053	204,37	10	Skip-gram
	BiLSTM	0,9998	0,9765	0,9761	220,99	10	Skip-gram
	CNN	0,7391	0,7640	0,7672	83,32	10	Skip-gram
	DNN	0,5050	0,5041	0,5053	63,21	10	Skip-gram

Çizelge 5.3 ve Çizelge 5.4'te sunulan verilerin incelenmesi sonucunda aşağıdaki gözlemler yapılmıştır:

- Kelime tabanlı özniteliklerin eğitimi için en uygun algoritma BiLSTM algoritmasıdır.
- Word2Vec modelinde skip-gram algoritmasının eğitimi ve testi sonucunda, Malicious and Benign URLs veri setinde 0,9918; Ebbu2017 Phishing veri setinde ise 0,9761 doğruluk oranı elde edilmiştir.
- Word2Vec modelinde CBoW algoritması kullanılarak yapılan eğitim ve test işlemleri sonucunda, Malicious and Benign URLs veri setinde 0,9866; Ebbu2017 Phishing veri setinde ise 0,9827 doğruluk oranı elde edilmiştir.
- Skip-gram algoritması, Malicious and Benign URLs veri setinde daha üstün bir başarı oranı elde etmiştir.
- Ebbu2017 Phishing veri setinde ise CBoW algoritması daha yüksek başarı oranına sahiptir.
- 10 epoch boyunca gerçekleştirilen eğitim sonuçları, skip-gram ve BiLSTM algoritmalarının eğitim süresinin, CBoW ve BiLSTM algoritmalarının eğitim süresine kıyasla daha kısa olduğunu göstermektedir.

Çizelge 5.4. Word2Vec modelinde CBoW algoritması kullanılarak kelime tabanlı özniteliklerin eğitimi için algoritma seçimi

Veri Seti	Algoritma	Eğitim Doğruluk Değeri	Doğrulama Doğruluk Değeri	Test Doğruluk Değeri	Eğitim Süresi (s)	Epoch	Word2Vec Algoritması
Malicious and Benign URLs Veri Seti [36]	LSTM	0,7716	0,7735	0,7728	1644,74	10	CBoW
	BiLSTM	0,9999	0,9863	0,9866	2287,87	10	CBoW
	CNN	0,7679	0,7685	0,7680	571,81	10	CBoW
	DNN	0,7679	0,7685	0,7680	432,48	10	CBoW
Ebbu2017 Phishing Veri Seti [6]	LSTM	0,5030	0,5041	0,5053	171,19	10	CBoW
	BiLSTM	0,9999	0,9795	0,9827	264,97	10	CBoW
	CNN	0,7437	0,7654	0,7598	87,46	10	CBoW
	DNN	0,5056	0,5041	0,5053	82,69	10	CBoW

Malicious and Benign URLs veri setinin boyutu, Ebbu2017 Phishing veri setine göre daha büyüktür. Skip-gram algoritması, Malicious and Benign URLs veri setinde CBoW algoritmasına kıyasla daha iyi performans sergilemiştir. GramBeddings veri kümesi de büyük bir veri kümesi olduğundan, yukarıda belirtilen faktörler göz önünde bulundurulduğunda, bu çalışmada skip-gram algoritmasının seçilmesi daha uygun olacaktır. CBoW algoritması, sözlükte daha sık kullanılan yüksek frekanslı kelimelerle eğitildiği için, bu tür kelimelerde daha başarılı sonuçlar üretir. Buna karşılık, skip-gram algoritması,

sözlükte nadiren bulunan düşük frekanslı kelimelerle eğitildiğinden, bu tür kelimelerde CBoW algoritmasına kıyasla daha başarılı sonuçlar verir. Veri setindeki URL adreslerinden elde edilen belirteçler kullanılarak bir kelime sözlüğü oluşturulmuştur (Çizelge 4.3). Bu sözlükte yer alan her kelime, veri kümesindeki farklı URL'lerde bulunabilir. Bir kelime, hem kimlik avı (phishing) olarak işaretlenmiş bir URL'de hem de yasal (legitimate) olarak etiketlenmiş bir URL'de yer alabilir. Bu nedenle, nadiren kullanılan düşük frekanslı kelimeler büyük önem taşımaktadır. Çünkü düşük frekanslı bir kelime, kimlik avı URL'lerini ayırt edici bir öznitelik olarak işlev görebilir. Çizelge 5.3 ve Çizelge 5.4'te sunulan bulgular, kelime vektör temsillerini oluşturmak için skip-gram algoritmasının uygun bir seçim olduğunu, kelime tabanlı özniteliklerin eğitimi için ise BiLSTM algoritmasının önerilen modelde en iyi performansı sergilediğini göstermiştir.

5.2.4. En uygun URL uzunluğunun belirlenmesi

Kelime tabanlı özniteliklerin eğitim sürecinin başlangıcında, *max_token_number* değeri veri setine bağlı olarak belirlenmiştir. GramBeddings veri seti için maksimum belirteç sayısı 712, Malicious and Benign URLs veri seti için 166 ve Ebbu2017 Phishing veri seti için ise 99 olarak tespit edilmiştir. Karakter tabanlı öznitelikler için en uygun URL uzunluğunu belirlemek amacıyla *max_url_length* değeri için 250, 300 ve 350 olmak üzere üç farklı değer seçilmiş ve ardından deneysel gözlemler yapılmıştır. Bölüm 4, Şekil 4.1, 4.2 ve 4.3'te veri setlerine göre URL uzunluklarının dağılımı gösterilmiştir. Bu dağılımlar göz önünde bulundurularak 250, 300 ve 350 değerleri tercih edilmiştir. Çizelge 5.5, Şekil 4.9'da gösterilen model ve Algoritma 1 (Çizelge 4.4) kullanılarak gerçekleştirilen eğitim, doğrulama ve test süreçlerinin sonuçlarını içerir. Elde edilen bulgulara göre, test doğruluk oranları şu şekilde belirlenmiştir. GramBeddings veri seti için 300 URL uzunluğunda doğruluk oranı 0,9824; Malicious and Benign URLs veri seti için 250 URL uzunluğunda doğruluk oranı 0,9937; Ebbu2017 Phishing veri seti için 350 URL uzunluğunda doğruluk oranı 0,9846 olarak elde edilmiştir.

Şekil 4.1, GramBeddings veri setindeki URL uzunluklarının çoğunlukla 300'ün altında olduğunu gösterir. Şekil 4.2 incelendiğinde, Malicious and Benign URLs veri setindeki URL'lerin büyük bir kısmının 250-270 uzunluk değerlerinin altında olduğu görünür. Şekil 4.3 ise, Ebbu2017 Phishing veri setinin benzer bir eğilim gösterdiğini ve çoğu URL'nin 300-320 uzunluk değerlerinin altında olduğunu ortaya koyar. Şekil 4.1, 4.2, 4.3 ve Çizelge 5.5'te

sunulan sonuçlar tutarlıdır.

Önerilen model için ortalama bir değer alınması önerilir. Bu doğrultuda, karakter tabanlı özniteliklerin eğitim sürecinde kullanılacak maksimum URL uzunluğu olarak 250, 300 ve 350 değerleri arasından 300 seçilmiştir. 250 değerinin seçilmesi, GramBeddings ve Ebbu2017 Phishing veri setlerindeki belirli URL'lerin dışarda kalmasına neden olacaktır. 350 değerinin seçilmesi ise, GramBeddings ve Malicious and Benign URLs veri setleri için eğitim süresinin uzamasına yol açacaktır. Bu faktörler dikkate alındığında, önerilen model için en uygun değer 300 olarak seçilmiştir.

Çizelge 5.5. Veri setlerine göre uygun URL uzunluğunun belirlenmesi

Veri Seti	URL Uzunluğu	Eğitim Doğruluk Değeri	Doğrulama Doğruluk Değeri	Test Doğruluk Değeri	Eğitim Süresi (s)	Epoch
GramBeddings Veri Seti [20]	250	0,9949	0,9818	0,9820	44232,77	30
	300	0,9951	0,9815	0,9824	45100,84	30
	350	0,9926	0,9804	0,9809	47221,18	30
Malicious and Benign URLs Veri Seti [36]	250	0,9985	0,9927	0,9937	10307,31	30
	300	0,9985	0,9935	0,9932	10830,34	30
	350	0,9981	0,9930	0,9927	12299,67	30
Ebbu2017 Phishing Veri Seti [6]	250	0,9984	0,9850	0,9836	1426,76	30
	300	0,9986	0,9845	0,9834	1523,78	30
	350	0,9973	0,9838	0,9846	1766,47	30

5.2.5. BiLSTM nöron sayısının seçimi

Önerilen modelin farklı nöron sayılarıyla etkinliğini değerlendirmek amacıyla, BiLSTM ağı içerisinde 32, 64, 128 ve 256 nöron ile deneyler gerçekleştirilmiştir. Eğitim süreci 20 epoch boyunca gerçekleştirilmiştir. Çizelge 5.6'da, üç farklı veri seti için elde edilen sonuçlar detaylı olarak sunulmuştur. Çizelge 5.6'ya göre, 64 ve 128 nörona sahip BiLSTM ağı, GramBeddings veri setinde test verisiyle 0,9819 sınıflandırma başarısı elde etmiştir. Ayrıca, 128 nörona sahip BiLSTM ağının eğitim, doğrulama ve test doğruluk performansının, diğer nöron sayılarından daha üstün olduğu gözlemlenmiştir. Çizelge 5.6'da sunulan Malicious and Benign URLs veri seti sonuçları incelendiğinde, 32, 64 ve 128 nörona sahip BiLSTM ağlarının test doğruluk oranlarının aynı olduğu ve 0,9933 olarak gerçekleştiği görülmüştür.

Ayrıca, 128 nörona sahip BiLSTM ağı ile elde edilen eğitim, doğrulama ve test sonuçlarının, diğer nöron sayılarıyla elde edilen sonuçlara kıyasla genellikle daha başarılı olduğu tespit edilmiştir. Çizelge 5.6'da gösterildiği gibi, Ebbu2017 Phishing veri seti analiz edildiğinde, 256 nörona sahip BiLSTM ağının eğitim doğruluk oranının daha yüksek olduğu görülür. Ancak, 128 nöronlu BiLSTM ağı, doğrulama ve test doğruluk değerlerinde daha yüksek performans göstermiş olup, bu değerler sırasıyla 0,9834 ve 0,9831 olarak hesaplanmıştır.

Çizelge 5.6. Önerilen model için BiLSTM nöron sayısının belirlenmesi

Veri Seti	BiLSTM Nöron Sayısı	Eğitim Doğruluk Değeri	Doğrulama Doğruluk Değeri	Test Doğruluk Değeri	Eğitim Süresi (s)	Test Süresi (s)	Epoch
GramBeddings Veri Seti [20]	32	0,9874	0,9801	0,9805	20395,64	131,06	20
	64	0,9914	0,9811	0,9819	23805,56	141,42	20
	128	0,9936	0,9820	0,9819	29663,86	169,61	20
	256	0,9929	0,9816	0,9814	50799,30	293,31	20
Malicious and Benign URLs Veri Seti [36]	32	0,9978	0,9929	0,9933	6985,26	62,92	20
	64	0,9973	0,9931	0,9933	7591,15	82,63	20
	128	0,9978	0,9932	0,9933	8190,86	82,45	20
	256	0,9977	0,9915	0,9915	12073,76	84,68	20
Ebbu2017 Phishing Veri Seti [6]	32	0,9933	0,9806	0,9802	1048,38	5,95	20
	64	0,9948	0,9818	0,9822	1085,79	6,13	20
	128	0,9968	0,9834	0,9831	1229,67	7,15	20
	256	0,9972	0,9803	0,9807	1698,19	9,87	20

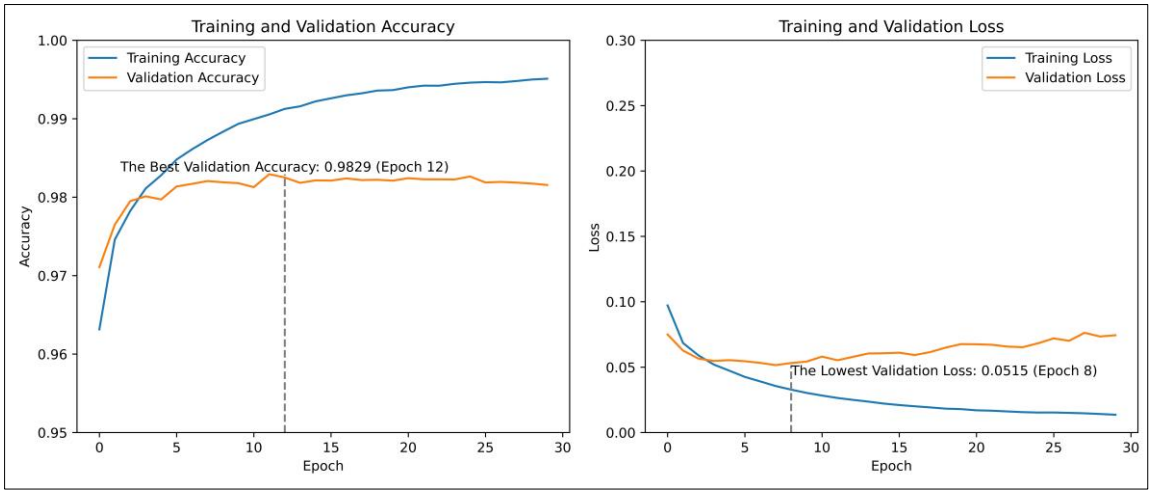
Çizelge 5.6 tüm veri kümeleri için analiz edildiğinde, 128 nöronlu BiLSTM ağının sınıflandırma performansı ve genelleme yeteneğinin daha iyi olduğu görülmüştür. Çizelge 5.6, aynı zamanda veri kümelerinin eğitim ve test sürelerini de içerir. BiLSTM ağındaki nöron sayısının artmasıyla birlikte, eğitim ve test sürelerinin de uzadığı tespit edilmiştir. Test verileri ise modelin eğitim aşamasında hiç karşılaşmadığı verilerdir. Gerçek dünya senaryolarında, kimlik avı tespitinde yanıt süresi kritik öneme sahiptir, çünkü ilk kez karşılaşılan bir URL adresinin yasal mı yoksa kimlik avı mı olduğunu belirlemek gerekir. Bu nedenle, test süreleri de değerlendirilmiştir. Gerçek zamanlı kullanımda test süreleri doğrudan yanıt süresi hakkında kesin bir kanıt sağlamasa da, genel bir öngörü sunabilir. Önerilen modelin gerçek zamanlı ortama uyarlanması durumunda, 256 nöronlu modelin yanıt süresinin daha uzun olacağı tahmin edilmektedir. 32 ve 64 nöronlu modeller, hızlı yanıt

verme avantajına sahip olsa da, sınıflandırma başarıları 128 nöronlu modele kıyasla daha düşük seviyede kalmaktadır. Bu değerlendirmeler doğrultusunda, önerilen model için en uygun nöron sayısı, hem sınıflandırma başarısı hem de yanıt süresi açısından 128 olarak belirlenmiştir.

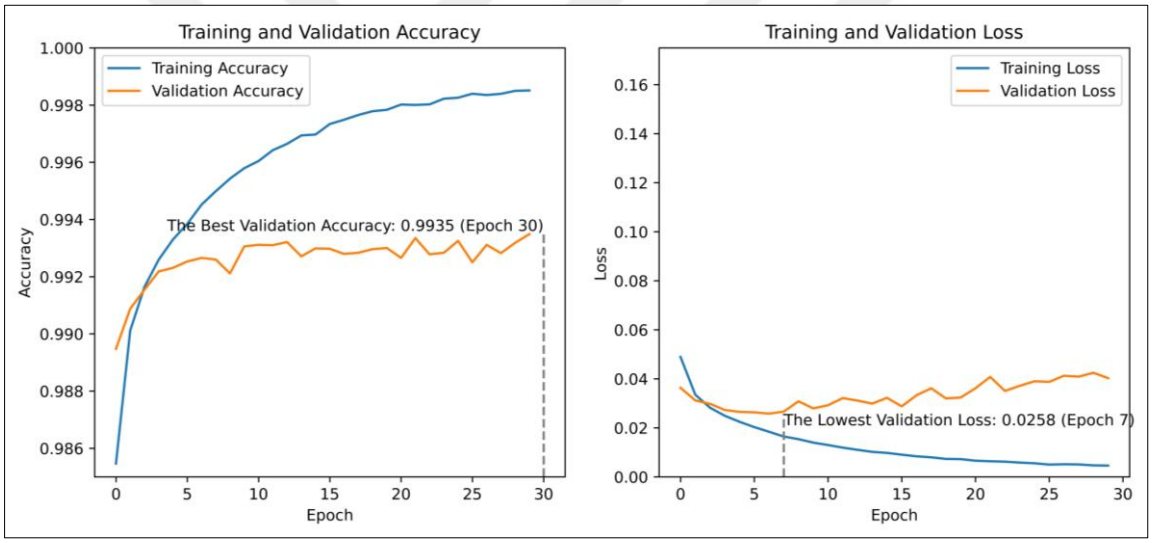
5.2.6. Sınıflandırma metriklerinin sonuçları

Deneysel gözlemler sonucunda, önerilen model için en uygun parametreler belirlenmiştir. Optimizasyon algoritması olarak Adam tercih edilmiş, karakter ve kelime tabanlı özniteliklerin eğitimi için BiLSTM kullanılmış ve Word2Vec modeli için skip-gram algoritması seçilmiştir. BiLSTM ağındaki nöron sayısı 128, karakter tabanlı öznitelikler için belirlenen maksimum URL uzunluğu ise 300 olarak belirlenmiştir. Önerilen modelde, BiLSTM algoritması dikkat (attention), maksimum havuzlama (max pooling) ve dropout katmanları ile desteklenmiştir ve Şekil 4.9 ile Şekil 4.10'da gösterildiği gibi yapılandırılmıştır. Önerilen model, GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setleri kullanılarak eğitilmiş, doğrulama ve test aşamalarından geçirilmiştir. Eğitim süreci 30 epoch boyunca gerçekleştirilmiştir. Öncelikle, veri seti %80 eğitim ve %20 test olacak şekilde bölünmüştür. Ardından, eğitim verisi %80 eğitim ve %20 doğrulama alt kümelerine ayrılmıştır. Bu yaklaşımın temel amacı, aşırı öğrenmeyi önlemek ve modelin genelleme yeteneğini artırarak ilk kez karşılaştığı bir URL adresini doğru şekilde sınıflandırmasını sağlamaktır. Her veri seti için ayrı eğitim-doğrulama doğruluk (train-validation accuracy) ve eğitim-doğrulama kayıp (train-validation loss) grafikleri oluşturulmuştur. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerine ait bu grafiklerin sonuçları, sırasıyla Şekil 5.1, 5.2 ve 5.3'te gösterilmiştir.

Şekil 5.1, 5.2 ve 5.3'te gösterilen grafikler, önerilen modelin üç veri kümesinde de aşırı öğrenme (overfitting) veya yetersiz öğrenme (underfitting) gibi sorunlar yaşamadığını ortaya koymuştur. Şekil 5.1'de gösterildiği gibi, GramBeddings veri seti ile eğitilen modelde, en yüksek doğrulama doğruluk değeri 12. epoch'ta 0,9829; en düşük doğrulama kayıp değeri ise 8. epoch'ta 0,0515 olarak elde edilmiştir. Şekil 5.2'de gösterildiği gibi, Malicious and Benign URLs veri seti ile eğitilen modelde, en yüksek doğrulama doğruluk değeri 30. epoch'ta 0,9935; en düşük doğrulama kayıp değeri ise 7. epoch'ta 0,0258 olarak kaydedilmiştir.



Şekil 5.1. Önerilen modelin GramBeddings veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri

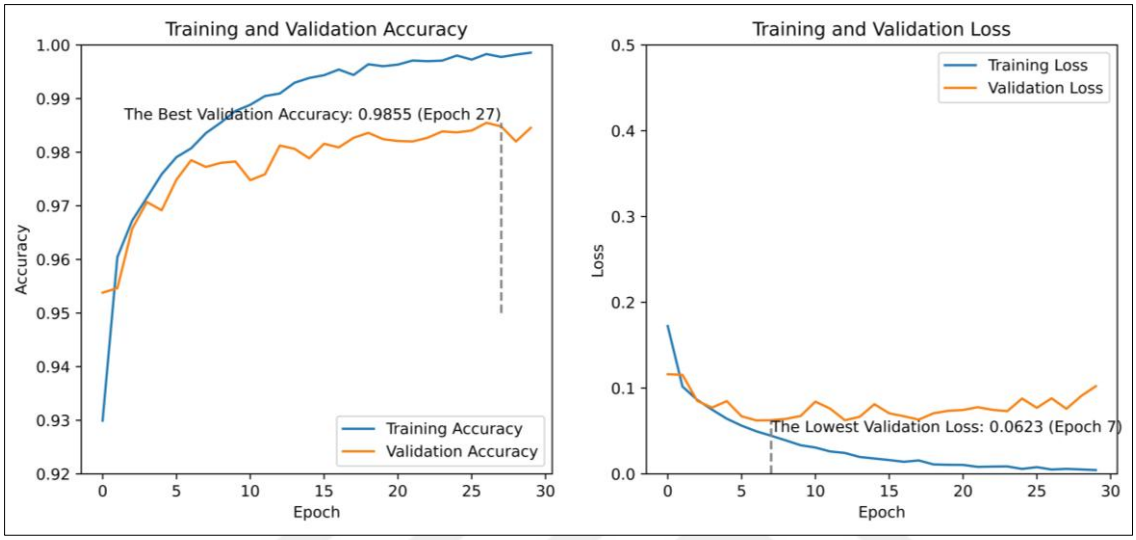


Şekil 5.2. Önerilen modelin Malicious and Benign URLs veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri

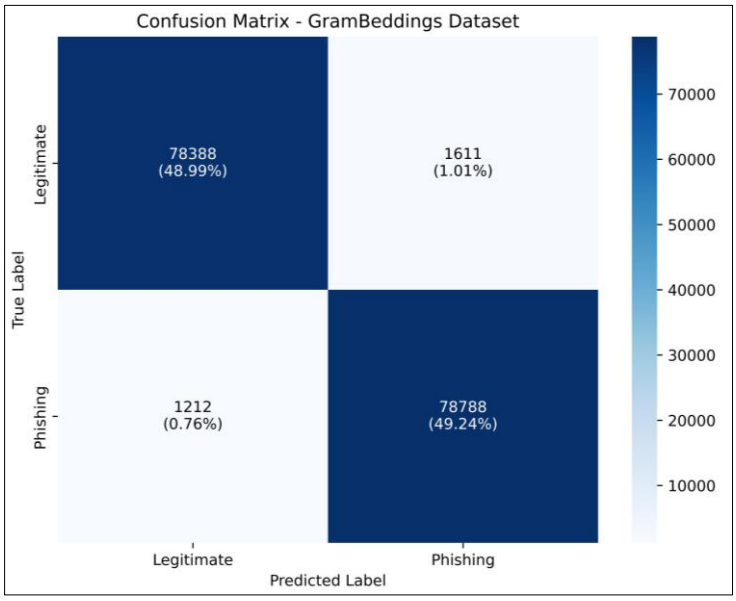
Şekil 5.3'te gösterildiği gibi, Ebbu2017 Phishing veri seti ile eğitilen modelde, en yüksek doğrulama doğruluk değeri 27. epoch'ta 0,9855; en düşük doğrulama kayıp değeri ise 7. epoch'ta 0,0623 olarak elde edilmiştir.

Şekil 5.4, 5.5 ve 5.6'da, önerilen modelin karışıklık matrislerine yer verilmiştir. Şekil 5.4 GramBeddings veri setine ait sonuçları; Şekil 5.5, Malicious and Benign URLs veri setine ait sonuçları; Şekil 5.6 ise Ebbu2017 Phishing veri setine ait sonuçları gösterir. Karışıklık matrislerinde TP (Doğru Pozitif), TN (Doğru Negatif), FP (Yanlış Pozitif) ve FN (Yanlış Negatif) değerleri hem sayısal hem de yüzdelik olarak sunulmuştur. Şekil 5.4, 5.5 ve 5.6'da

yer alan karışıklık matrisleri incelendiğinde, GramBeddings ve Ebbu2017 Phishing veri setlerinin dengeli oranlara sahip olduğu görülmüştür. Buna karşılık, Malicious and Benign URLs veri setinde değerlerin dengesiz bir dağılım sergilediği gözlemlenmiştir. Bu durum, veri setlerinde yer alan URL adreslerinin etiket dağılımlarından kaynaklanır.



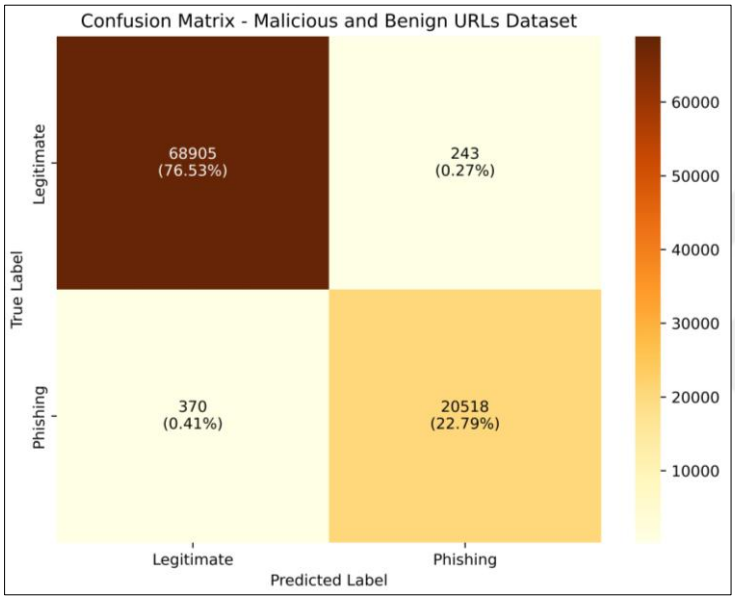
Şekil 5.3. Önerilen modelin Ebbu2017 Phishing veri seti üzerindeki eğitim-doğrulama doğruluk ve eğitim-doğrulama kayıp grafikleri



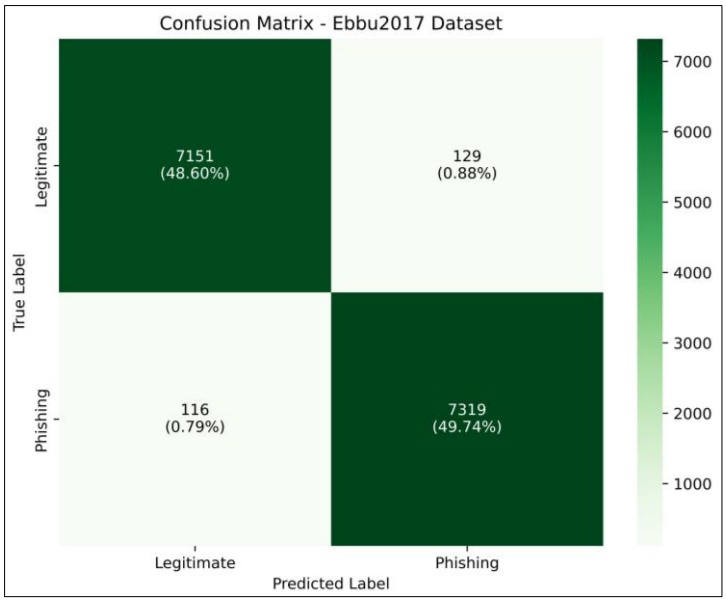
Şekil 5.4. GramBeddings veri seti karışıklık matrisi

Çizelge 5.7, her veri kümesi için karışıklık matrislerinden elde edilen sonuçları gösterir. Bu tabloda kesinlik (precision), duyarlılık (recall), F1-skoru ve AUC değerleri yer alır. GramBeddings veri setinde, maksimum URL uzunluğu 300 olan modelde 0,9824 test

doğruluk oranı ile en yüksek değer elde edilmiştir. Aynı veri kümesi için AUC değeri 0,9977 olarak hesaplanmıştır. Malicious and Benign URLs veri setinde, maksimum URL uzunluğu 250 olarak belirlenen modelde, 0,9937 test doğruluk oranı ile en yüksek başarı elde edilmiştir. Bu veri kümesinde, 250, 300 ve 350 URL uzunluklarına sahip tüm modellerde AUC değeri 0,9986 olarak hesaplanmıştır. Ebbu2017 Phishing veri setinde, maksimum URL uzunluğu 350 olan modelde 0,9846 test doğruluk oranı ve 0,9984 AUC değeri ile en yüksek sonuçlar elde edilmiştir.



Şekil 5.5. Malicious and Benign URLs veri seti karışıklık matrisi



Şekil 5.6. Ebbu2017 Phishing veri seti karışıklık matrisi

Çizelge 5.7, veri setinde kullanılan etiketleri (sınıf sütununu) gösterir. 0, yasal (legitimate) öğeleri, 1 ise kimlik avı (phishing) öğelerini temsil etmektedir. Her iki sınıf için hesaplanan kesinlik (precision), duyarlılık (recall) ve F1-skor değerlerine ait sınıflandırma metriği detayları, Çizelge 5.7'de verilmiştir.

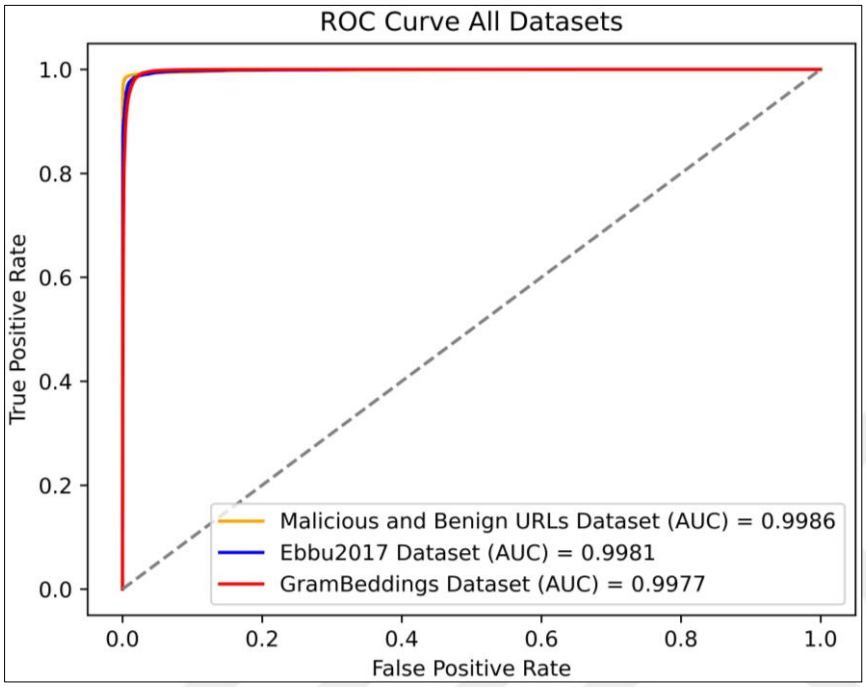
Çizelge 5.7. Önerilen modelin farklı URL uzunluklarındaki sınıflandırma metrik değerleri

Veri Seti	URL Uzunluğu	Test Doğruluk Değeri	Sınıf	Kesinlik	Duyarlılık	F1-Skor	AUC
GramBeddings Veri Seti [20]	250	0,9820	0	0,9838	0,9802	0,9820	0,9977
			1	0,9802	0,9838	0,9820	
	300	0,9824	0	0,9848	0,9799	0,9823	0,9977
			1	0,9800	0,9849	0,9824	
	350	0,9809	0	0,9825	0,9792	0,9809	0,9975
			1	0,9793	0,9826	0,9809	
Malicious and Benign URLs Veri Seti [36]	250	0,9937	0	0,9949	0,9969	0,9959	0,9986
			1	0,9898	0,9831	0,9864	
	300	0,9932	0	0,9947	0,9965	0,9956	0,9986
			1	0,9883	0,9823	0,9853	
	350	0,9927	0	0,9949	0,9957	0,9953	0,9986
			1	0,9856	0,9831	0,9843	
Ebbu2017 Phishing Veri Seti [6]	250	0,9836	0	0,9832	0,9835	0,9834	0,9983
			1	0,9839	0,9836	0,9837	
	300	0,9834	0	0,9840	0,9823	0,9832	0,9981
			1	0,9827	0,9844	0,9835	
	350	0,9846	0	0,9834	0,9854	0,9844	0,9984
			1	0,9857	0,9837	0,9847	

Çizelge 5.7'de gösterildiği gibi, tüm veri kümelerinde maksimum URL uzunluğu 300 olan modelin eğitimi ve testi sonucunda elde edilen ROC eğrileri ve AUC değerleri, Şekil 5.7'de aynı grafik üzerinde gösterilmiştir.

Şekil 5.7'de görüldüğü gibi, üç veri kümesinin ROC eğrileri, sola yakın ve yukarı doğru bir

eğilim sergilemektedir. ROC eğrileri, önerilen modelin farklı veri kümeleri üzerinde yüksek sınıflandırma performansı sergilediğini gösterir. AUC değerleri 0,9 ile 1,0 arasında değişir, bu da önerilen modelin yüksek doğrulukla çalıştığını göstermiştir.



Şekil 5.7. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerine ait ROC eğrileri ve AUC değerleri

5.3. URL Kısaltma

Kısaltılmış URL, uzun ve karmaşık URL'lerin daha kısa ve kullanımı daha kolay hale getirilmiş versiyonudur. Bu, çevrimiçi hizmetler aracılığıyla gerçekleştirilir ve uzun URL'lerin daha kısa sürümlerini oluşturmak için kullanılır. Örneğin, bir URL kısaltma servisi, *https://www.abc.com/1234?xyz=1a3xpNOQ1xm3* gibi uzun bir adresi *https://tinyurl.com/3b4tf5* formatında daha kısa bir bağlantıya dönüştürebilir. Kısaltılmış URL'ler, kullanıcıları zararlı veya kötü amaçlı web sitelerine yönlendirme riski taşıyabilir. Önerilen modelin kısaltılmış URL'ler üzerindeki performansını değerlendirmek için, üç farklı veri seti üzerinde istatistiksel bir analiz yapılmıştır. Bu veri setlerindeki kısaltılmış URL'lerle ilgili istatistiksel bilgiler Çizelge 5.8'de gösterilmiştir. GramBeddings veri setindeki kısaltılmış URL oranı %5,23; Malicious and Benign URLs veri setindeki kısaltılmış URL oranı %5,90; Ebbu2017 Phishing veri setindeki kısaltılmış URL oranı ise %6,01'dir. Bu üç veri setindeki tüm kısaltılmış URL adresleri tek bir veri setinde

birleştirilmiş ve bu küme ‘Kısaltılmış URL Veri Seti’ (Shortened URL Dataset) olarak adlandırılmıştır. Kısaltılmış URL veri setinde toplam kısaltılmış URL sayısı 72 854, kimlik avı (phishing) URL sayısı 22 626 ve yasal (legitimate) URL sayısı 50 228’dir. Bu veri kümesindeki dengesizlik, gerçek internet kullanımındaki durumu yansıtmaktadır. Önerilen model, bu veri kümesiyle eğitilip test edilmiş ve elde edilen sonuçlar, modelin kısaltılmış URL adreslerini %92,16 doğruluk oranı ile sınıflandırabildiğini ortaya koymuştur. Çizelge 5.9, kısaltılmış URL veri setiyle gerçekleştirilen eğitim ve test sürecinden elde edilen sınıflandırma metriklerini gösterir. Bu veri setindeki maksimum belirteç sayısı 471 olarak belirlenmiştir. Modelin karakter ve kelime tabanlı yaklaşımı, özniteliklerdeki kendine özgü kalıpları belirlemeyi sağlamış ve kısaltılmış URL adreslerinin sınıflandırılması için etkili bir çözüm sunmuştur. Literatürde, kısaltılmış URL adresleriyle ilgili herhangi bir eğitim ve test işlemi gerçekleştirilmediğinden, bu çalışmanın sonuçlarının doğrudan diğer çalışmalarla karşılaştırılması mümkün değildir. Ayrıca, bu çalışmada kullanılan kısaltılmış URL adreslerinden elde edilen veri kümesi benzersizdir.

Çizelge 5.8. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerindeki kısaltılmış URL sayılarına dair istatistiksel veriler

	GramBeddings Veri Seti [20]	Malicious and Benign URLs Veri Seti [36]	Ebbu2017 Phishing Veri Seti [6]
Toplam kısaltılmış URL sayısı	41 849	26 582	4423
Kısaltılmış kimlik avı (phishing) URL sayısı	15 790	5044	1792
Kısaltılmış yasal (legitimate) URL sayısı	26 059	21 538	2631
Kısaltılmış URL oranı	%5,23	%5,90	%6,01
Kimlik avı (phishing) oranı	%37,73	%18,98	%40,52
Yasal (legitimate) oranı	%62,27	%81,02	%59,48

Çizelge 5.9. Önerilen modelin Kısaltılmış URL veri seti üzerindeki sınıflandırma metrik değerleri

Veri Seti	URL Uzunluğu	Test Doğruluk Değeri	Sınıf	Kesinlik	Duyarlılık	F1-Skor	AUC	Epoch
Kısaltılmış URL Veri Seti	300	0,9216	0	0,9589	0,9261	0,9422	0,9750	30
			1	0,8476	0,9118	0,8785		

Çizelge 5.9, önerilen modelin Kısaltılmış URL veri seti ile yapılan sınıflandırmada %92,16

doğruluk oranına ulaştığını gösterir. Yasal (legitimate) URL'ler için, kesinlik oranı %95,89, duyarlılık oranı %92,61 ve F1-skor oranı %94,22 olarak hesaplanmıştır. Kimlik avı (phishing) URL'ler için, kesinlik oranı %84,76 duyarlılık oranı %91,18 ve F1-skor oranı %87,85'dir. Modelin ayırt edici gücünün yüksek olduğunu gösteren AUC değeri 0,9750 olarak hesaplanmıştır. Bu sonuçlar, önerilen modelin kısaltılmış URL'leri etkili ve güvenilir bir şekilde sınıflandırma kapasitesine sahip olduğunu göstermiştir.

5.4. Birleşik Tokenizasyon

GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setlerinden elde edilen belirteçler, ortak bir kelime sözlüğü oluşturmak amacıyla birleştirilmiştir. Bu süreçte, yinelenen belirteçler kaldırılmış ve her bir belirteç yalnızca bir kez olacak şekilde düzenlenmiştir. Sonuç olarak, oluşturulan sözlük toplam 1 320 097 kelime içermektedir. Bu ortak sözlük, eğitim ve test aşamalarında önerilen model ile üç veri setinin değerlendirilmesi için kullanılmıştır. Çalışmada, üç veri setine ayrı ayrı tokenizasyon işlemi uygulanarak farklı kelime sözlükleri elde edilmiştir. Ancak, bu durum modellerin ve gerçek dünya senaryolarının adil bir şekilde değerlendirilmesi açısından ek açıklamalar gerektirebilir. Bu sorunu gidermek için, tüm veri setlerindeki belirteçleri içeren birleşik bir kelime sözlüğü oluşturulmuş ve model, bu ortak sözlük kullanılarak yeniden eğitilip test edilmiştir. Bu yaklaşım, farklı veri kaynakları arasında daha tutarlı ve genellenebilir bir performans sunarak modelin etkinliğini ve çeşitli veri setlerine uyum yeteneğini güçlendirmiştir. Çizelge 5.10, üç veri seti için birleştirilmiş tokenizasyon yöntemiyle elde edilen sınıflandırma metriklerini göstermektedir.

Çizelge 5.10'a göre, GramBeddings veri setinin test doğruluğu %97,95 iken AUC değeri 0,9974'tür. Malicious and Benign URLs veri seti için test doğruluğu %99,16, AUC değeri 0,9983 olarak hesaplanmıştır. Ebbu2017 Phishing veri setinde ise test doğruluğu %97,86, AUC değeri 0,9975 olarak elde edilmiştir. Her iki sınıfa ait diğer sınıflandırma metrikleri Çizelge 5.6'da ayrıntılı şekilde sunulmuştur. Bu sonuçlar, önerilen yöntemin genelleştirilebilir ve pratik bir yaklaşım olduğunu ortaya koymaktadır.

Çizelge 5.10. Birleşik tokenizasyon yöntemi kullanılarak elde edilen sınıflandırma metrik değerleri

Veri Seti	Test Doğruluk Değeri	Sınıf	Kesinlik	Duyarlılık	F1-Skor	AUC	Epoch
GramBeddings Veri Seti [20]	0,9795	0	0,9856	0,9733	0,9794	0,9974	20
		1	0,9737	0,9858	0,9797		
Malicious and Benign URLs Veri Seti [36]	0,9916	0	0,9942	0,9949	0,9946	0,9983	20
		1	0,9832	0,9809	0,9820		
Ebbu2017 Phishing Veri Seti [6]	0,9786	0	0,9813	0,9754	0,9784	0,9975	20
		1	0,9761	0,9818	0,9789		

5.5. Maliyet Analizi

Çizelge 5.11'de, toplam, eğitilebilir ve eğitilemez parametre sayıları ile veri kümeleri için bellek kullanım değerleri gösterilmiştir. Eğitilebilir parametre sayısı ve bellek kullanımı, tüm veri setleri için aynıdır. Geliştirilen derin öğrenme modeli, her veri seti üzerinde aynı mimariyi kullanır. Model mimarisi, eğitilebilir parametreleri belirler ve veri setlerinin boyutu veya türü bu parametreleri etkilemez.

Çizel 5.11. Veri setlerine göre model parametreleri ve bellek kullanımı için maliyet analizi

Veri Seti	Toplam Parametre		Eğitilebilir parametreler		Eğitilebilir olmayan parametreler	
	Parametre sayısı	Model depolama boyutu (MB)	Parametre sayısı	Model depolama boyutu (MB)	Parametre sayısı	Model depolama boyutu (MB)
GramBeddings Veri Seti [20]	110 525 313	421,62	578 433	2,21	109 946 880	419,41
Malicious and Benign URL Veri Seti [36]	6 3491 713	242,20	578 433	2,21	62 913 280	240,00
Ebbu2017 Phishing Veri Seti [6]	9 316 225	35,54	578 433	2,21	8 737 792	33,33
Mendeley Data Web Page Phishing Detection Veri Seti [37]	3 358 593	12,81	578 433	2,21	2 780 160	10,61

Word2Vec gibi önceden eğitilmiş gömme katmanları, modelin bellek kullanımını önemli ölçüde artırırken, Çift Yönlü LSTM ve dikkat katmanları hesaplama gereksinimlerini ve

işlem maliyetlerini yükseltmiştir. Ayrıca, eğitim sürecinde gerçekleştirilen metrik değerlendirmeleri, işlem süresini uzatarak enerji tüketiminin artmasına neden olmuştur. Buna rağmen, karakter ve kelime düzeyindeki özniteliklerin birleşimi, kimlik avı saldırılarının tespitinde doğruluk ve güvenliği artırarak önemli bir avantaj sağlamıştır. Elde edilen bu önemli kazanımlar göz önüne alındığında, yüksek maliyetin finans ve güvenlik gibi hassas alanlar için kabul edilebilir olduğu değerlendirilmektedir.

5.6. Çapraz Doğrulama

Önerilen modelde k-katlı çapraz doğrulama yöntemi, büyük veri setlerinde RAM aşırı yüklenme sorununa neden olduğu için, daha küçük ölçekli Mendeley Data Web Page Phishing Detection veri seti kullanılarak test edilmiştir. Doğrulama k = 3, 5 ve 7 ile gerçekleştirilmiş, en yüksek performans k=7 ile elde edilmiştir (Çizelge 5.12).

Çizelge 5.12. Önerilen modelin çapraz doğrulama ve tokenizasyon yöntemleri ile karşılaştırılması

	k-katlı çapraz doğrulama			Karakter ve kelime tokenizasyonu	Karakter ve bayt çifti kodlama
	k=3	k=5	k=7		
Test Doğruluk Değeri	0,9072	0,9076	0,9120	0,9033	0,9212
AUC	0,9708	0,9711	0,9682	0,9694	0,9720
Eğitim Süresi (s)	316,65	318,95	461,23	306,52	597,29
Test Süresi (s)	2,63	2,66	2,62	1,93	1,91
Epoch	30	30	30	30	30

Bayt Çifti Kodlama (Byte Pair Encoding - BPE) yöntemi, en sık tekrar eden karakter çiftlerini birleştirerek metnin alt birimlerini oluşturan bir tekniktir ve kelime tabanlı yönteme kıyasla anlamsal ilişkiler üzerine daha az odaklanır. Bu çalışmada, URL adreslerindeki anlamsal bağlantıları yakalamaya yönelik öznitelikleri çıkarmak için kelime tabanlı tokenizasyon yöntemi kullanılmıştır.

6. TARTIŞMA

Çizelge 6.1 ve Çizelge 6.2'de, bu çalışmada önerilen model, literatürdeki diğer kimlik avı sınıflandırma yöntemleriyle karşılaştırılmıştır. Karşılaştırma, bu çalışmada kullanılan veri setleri temel alınarak gerçekleştirilmiştir. Bölüm 2.2'deki ilgili çalışmalar kısmında, Çizelge 6.1 ve Çizelge 6.2'de görüldüğü üzere, web sayfalarının URL adreslerinin sınıflandırılması üzerine yapılan çalışmalar başarılı sonuçlar vermiştir. Çizelge 6.1 yakından incelendiğinde Bozkır ve ark. [20]'nın GramBeddings veri setini literatüre kazandırarak çalışmalarında kullandıkları ve %98,27 doğruluk ve 0,9973 AUC değerine sahip bir model önerdikleri görülmektedir. Tez kapsamında önerilen modelin aynı veri kümesinde elde ettiği sonuçlar %98,24 doğruluk ve 0,9977 AUC değeridir. Bozkır ve ark. [20]'nın çalışmaları incelendiğinde önerilen modelin 0,9935 eğitim doğruluğu sergilediği görülmektedir. Önerilen modelin eğitim veri seti üzerinde ulaştığı doğruluk değeri 0,9951 olarak hesaplanmıştır (Çizelge 5.5, max_url_length=300). Bozkır ve ark. [20] tarafından geliştirilen modelin, test veri seti üzerindeki doğruluk değeri daha yüksek olmasına rağmen, tez kapsamında önerilen modelin eğri altında kalan alan değeri daha yüksektir. Bu sonuçlar, önerilen modelin GramBeddings veri setindeki kimlik avı URL'lerini sınıflandırmada etkili, tutarlı ve karşılaştırılabilir bir çözüm sunduğunu ve önceki çalışmaların bulgularıyla uyumlu olduğunu ortaya koymuştur.

Çizelge 6.1. Önerilen modelin literatürdeki mevcut çalışmalarla karşılaştırılması

Veri Seti	İlgili Çalışmalar	Algoritma	Test Doğruluk Oranı (%)	AUC
GramBeddings Veri Seti [20]	Bozkır ve ark. [20]	CNN-BiLSTM	%98,27	0,9973
	Önerilen Model	BiLSTM	%98,24	0,9977
Malicious and Benign URLs Veri Seti [36]	Roy ve ark. [33]	BiLSTM	%99,00	N/A
	Afzal ve ark. [24]	LSTM	%98,30	0,9935
		K-Means Clustering	%99,70	N/A
	Önerilen Model	BiLSTM	%99,32	0,9986
Ebbu2017 Phishing Veri Seti [6]	Şahingöz ve ark. [6]	Random Forest	%97,98	N/A
	Önerilen Model	BiLSTM	%98,34	0,9981
Common Crawl, Alexa, Yandex, PhishTank	Aslam ve ark. [34]	LSTM and XGBoost	%96,04	0,9581
Mendeley Data	Zaimi ve ark. [35]	XGBoost	%97,82	N/A

Roy ve ark. [33] ile Afzal ve ark. [24] tarafından yapılan çalışmaların her ikisinde de Malicious and Benign URLs veri seti kullanılmıştır. Roy ve ark. [33] tarafından geliştirilen ve BiLSTM algoritmasını kullanan modelin sonuçları ile karşılaştırıldığında, tez kapsamında önerilen modelin sonuçları daha iyi performans göstermiştir. Roy ve ark. [33], %99,00 doğruluk oranına ulaşırken, önerilen model aynı veri setinde %99,32 doğruluk elde etmiştir. Araştırmacılar çalışmalarında AUC değerine yer vermedikleri için AUC açısından karşılaştırma yapmak mümkün olmamıştır.

Çizelge 6.2. Önerilen modelin literatürdeki mevcut çalışmalarla karşılaştırılması

Veri Seti	İlgili Çalışmalar	Algoritma	Sınıf	Kesinlik	Duyarlılık	F1-Skor
GramBeddings Veri Seti [20]	Bozkır ve ark. [20]	CNN-BiLSTM	-	0,9894	0,9759	0,9826
	Önerilen Model	BiLSTM	0	0,9848	0,9799	0,9823
			1	0,9800	0,9849	0,9824
Malicious and Benign URLs Veri Seti [36]	Roy ve ark. [33]	BiLSTM	0	0,9900	0,9900	0,9900
			1	0,9900	0,9900	0,9900
	Afzal ve ark. [24]	LSTM	-	0,9900	0,9770	0,9830
		K-Means Clustering	-	0,9700	0,9570	0,9680
	Önerilen Model	BiLSTM	0	0,9947	0,9965	0,9956
			1	0,9883	0,9823	0,9853
Ebbu2017 Phishing Veri Seti [6]	Şahingöz ve ark.[6]	Random Forest	-	0,9700	0,9900	0,9800
	Önerilen Model	BiLSTM	0	0,9840	0,9823	0,9832
			1	0,9827	0,9844	0,9835

Afzal ve ark. [24], Malicious and Benign URLs veri setini kullanarak yaptıkları çalışmada, LSTM algoritması kullanarak önerdikleri modelde %98,30 doğruluk değeri ve 0,9935 AUC değeri elde etmişlerdir. Aynı veri kümesinde, tez kapsamında önerilen modelin doğruluk değeri %99,32 ve AUC değeri ise 0,9986'dır. Elde edilen sonuçlar, önerilen modelin sınıflandırma başarısının daha yüksek olduğunu ortaya koymuştur. Benzer şekilde Afzal ve ark. [24] aynı veri seti üzerinde K-Means kümeleme algoritması ile %99,70 doğruluk değeri elde etmişler ancak K-Means kümeleme ile elde ettikleri AUC değerini çalışmalarına dahil etmemişlerdir. K-Means kümeleme ile önerdikleri modelin kesinlik değeri 0,9700, duyarlılık değeri 0,9570 ve F1-skor değeri 0,9680'dir. Tez kapsamında önerilen modelin

maksimum 300 URL uzunluğunda elde ettiği kesinlik değeri yasal URL'ler için 0,9947, oltalama URL'leri için 0,9883'tür. Yasal URL'ler için duyarlılık değeri 0,9965, kimlik avı URL'leri için ise 0,9823 olarak hesaplanmıştır. F1-skoru, yasal URL'ler için 0,9956, kimlik avı URL'leri için ise 0,9853 olarak elde edilmiştir (Çizelge 6.2). Kesinlik, duyarlılık ve F1-skor metrikleri açısından, önerilen modelin daha yüksek sınıflandırma başarısı sergilediği gözlemlenmiştir.

Şahingöz ve ark. [6] Ebbu2017 Phishing veri setini literatüre kazandırmış ve Rastgele Orman algoritmasını kullanarak %97,98 doğruluk oranı elde etmişlerdir. Bu çalışmada önerilen model, Ebbu2017 Phishing veri seti ile %98,34 doğruluk oranına ulaşmıştır. Önerilen modelin sınıflandırma başarısının daha yüksek olduğu gözlemlenmiştir. Şahingöz ve ark. [6] çalışmalarında AUC değerine yer vermedikleri için bu metrikle ilgili herhangi bir karşılaştırma yapılamamıştır.

Aslam ve ark. [34] tarafından yapılan çalışmada kullanılan veri setleri, kamuya açık platformlardan elde edilmesine rağmen, araştırmacıların çalışmalarında kullandıkları biçimde tam olarak erişilebilir değildir. Sonuç olarak, bu çalışmada önerilen modeli kullanarak deneysel gözlemler yapmak mümkün olmamıştır. Benzer şekilde, Zaimi ve ark. [35] çalışmalarında, öznitelikleri önceden çıkarılmış bir veri seti kullanmışlardır. Ancak, Zaimi ve ark. [35]'nın kullandıkları veri setinde özniteliklerin çıkarıldığı URL'ler ham veri olarak yer almadığından ve tez kapsamında önerilen modelin doğrudan ham veriden öznitelik çıkarmak üzere tasarlandığından, önerilen modeli bu veri kümesi üzerinde test etmek mümkün olmamıştır. Buna ek olarak, önerilen modelin sonuçları, literatürde yaygın olarak kullanılan diğer veri setleri üzerinde değerlendirildiğinde, modelin [34] ve [35]'teki çalışmalara kıyasla daha iyi performans sergilediği görülmektedir.

Aynı veri setlerini kullanan önceki çalışmalar, kimlik avı URL sınıflandırma probleminin çözümünde önerilen modelin etkinliğini ortaya koymuştur. Önerilen model, kimlik avı URL'lerini tespit etmede oldukça başarılı bir çözüm sunmaktadır.

Bu çalışmada önerilen model, karakter ve kelime tabanlı öznitelik çıkarma yöntemlerini birleştirerek ve doğruluğu artırmak için BiLSTM ve dikkat mekanizmalarını kullanarak kimlik avı saldırılarını tespit etmek için yenilikçi bir yaklaşım sunmaktadır. Sezgisel yöntemler, sabit kurallara dayanması nedeniyle, gelişen ve karmaşıklaşan kimlik avı

tekniklerine uyum sağlamakta yetersiz kalmaktadır. Buna karşılık, öğrenme tabanlı yapıya sahip olan önerilen model, büyük veri setleri üzerinde eğitildiğinde yüksek genelleme performansı gösterme eğilimindedir. Geleneksel makine öğrenimi yöntemleri (örneğin, Naive Bayes, Random Forest vb.) genellikle manuel olarak seçilen özniteliklere dayanır, bu durum ise modellerin performansını sınırlayabilir. Önerilen model ise, karakter ve kelime tabanlı öznitelikleri otomatik olarak çıkarıp birleştirerek bu sınırlamayı ortadan kaldırmaktadır. Çizelge 6.1'de gösterildiği gibi, önerilen model K-Means kümeleme [24] ve Random Forest [6] gibi geleneksel yöntemlerden daha iyi performans göstermiştir. Bu sonuçlar, önerilen modelin geleneksel yöntemlere göre daha üstün bir performans gösterdiğini kanıtlamıştır.

Çizelge 6.2, önerilen modelin mevcut literatürdeki çalışmalarla kıyaslandığında farklı veri setleri üzerindeki etkinliğini ortaya koymuştur. GramBeddings, Malicious and Benign URLs ve Ebbu2017 Phishing veri setleri ile yapılan karşılaştırmalar, önerilen modelin genel olarak daha yüksek test kesinliği, duyarlılık ve F1-skor değerleri sunduğunu göstermiştir. Önerilen modelin sonuçları, CNN-BiLSTM, LSTM ve Random Forest gibi farklı algoritmalar ve metodolojiler kullanan önceki çalışmalarla karşılaştırıldığında, modelin genellikle üstün veya benzer performans sergilediği gözlemlenmiştir. Bu bulgular, modelin etkinliğini ve rekabetçi yapısını destekleyen güçlü kanıtlar sunmaktadır. Bölüm 5'teki deneysel bulgular ve önerilen modelin literatürdeki mevcut çalışmalarla karşılaştırılması, aşağıdaki sonuçların çıkarılmasını sağlamıştır.

1. Karakter ve kelime tabanlı öznitelik çıkarma yöntemlerinin birleştirilmesiyle elde edilen derin öğrenme modeli, kimlik avı URL'lerinin sınıflandırılması için etkili bir çözüm sunmuştur.
2. Karakter ve kelime düzeyinde öznitelik çıkarma yöntemlerinin birlikte kullanılması ve elde edilen özniteliklerin birleştirilmesi, daha kapsamlı ve zengin bir öznitelik kümesi oluşturulmasını sağlamıştır.
3. Karakter tabanlı öznitelik çıkarımı, karakterlerin yapısal analizine odaklandığı için anlamsal ilişkilerin dikkate alınması öncelikli bir hedef değildir. Buna karşılık, kelime tabanlı öznitelik çıkarımı, kelimeler arasındaki anlamsal ilişkilerin incelenmesine imkan tanımıştır.
4. Önerilen yöntem, dengeli (GramBeddings ve Ebbu2017 Phishing Veri Setleri) ve dengesiz (Malicious and Benign URLs Veri Seti) veri setleri de dahil olmak üzere farklı

URL adreslerini içeren veri setleri üzerinde yüksek sınıflandırma performansı göstermiştir.

5. Kısaltılmış URL adreslerinden oluşan özgün bir veri seti kullanılmıştır. Modelin karakter ve kelime tabanlı yaklaşımı, bu URL'leri sınıflandırmak için etkili bir çözüm sağlamıştır.
6. Gerçek dünya uygulamalarına uyarlanabilir bir model sağlamak için, üç farklı veri setindeki belirteçleri içeren birleşik bir sözlük oluşturulmuştur. Elde edilen bulgular, önerilen modelin başarılı ve etkili olduğunu ortaya koymuştur.
7. Mendeley Data Web Page Phishing Detection veri seti üzerinde yapılan deneyler, bellek sorunlarının olmadığı durumlarda çapraz doğrulama yöntemi kullanılarak eğitim ve doğrulamanın başarıyla gerçekleştirilebileceğini göstermiştir.

6.1. Modelin Güçlü Yönleri, Sınırlamaları ve Gelecek Çalışmalar

Önerilen model, karakter ve kelime tabanlı öznitelik çıkarma yöntemlerini birleştirerek URL adreslerinin sınıflandırılmasında yüksek performans göstermiştir. Karakter tabanlı öznitelikler yüzeysel özelliklere odaklanırken, kelime tabanlı öznitelikler kelimeler arasındaki ilişkileri analiz ederek daha derin bir bağlamsal anlayış sunmaktadır. Bu iki farklı öznitelik kümesinin bir araya getirilmesi, URL'lerin hem yapısal hem de anlamsal özelliklerini yakalayan kapsamlı bir analiz sağlamaktadır. Model, her URL'yi karakter ve kelime düzeyinde ayrı ayrı ele alarak iki farklı öznitelik seti oluşturur. Karakter tabanlı analiz, URL'de yer alan bireysel karakterleri değerlendirirken, kelime tabanlı analiz kelimeler arasındaki bağlamsal bağlantıları inceler. Son aşamada, bu iki öznitelik kümesi modelin birleştirme katmanında entegre edilerek tek bir öznitelik kümesi oluşturulur ve sınıflandırma sürecinde kullanılır. Bu yaklaşım bağımsız öznitelik analizini destekler ve farklı öznitelik türlerinin güçlü yanlarını koruyarak daha etkili bir sınıflandırma sağlar. Önerilen model, GramBeddings, Malicious and Benign URL, Ebbu2017 Phishing ve Mendeley Data Web Page Phishing Detection gibi farklı veri setleri üzerinde test edilerek değerlendirilmiştir. Sonuçlar, modelin ölçeklenebilirlik ve performans açısından karşılaşılan zorluklara karşı dayanıklılık sergilediğini göstermektedir.

Modelin güçlü yönlerinden biri, URL'leri hem karakter hem de kelime düzeyinde analiz edebilme kabiliyetine sahip olmasıdır. Bu iki seviyenin ayrı ayrı işlenmesi, her yöntemin eksikliklerini dengeleyerek daha kapsamlı bir değerlendirme sunar. Örneğin, karakter tabanlı öznitelikler anlamsal ilişkileri dikkate almazken, kelime tabanlı öznitelikler bu

bağlantıları ön plana çıkarır. Ayrıca, kelime tabanlı öznitelik çıkarma yöntemi, "." ve "/" gibi karakterlere dayalı olarak belirteçler oluşturur, ancak bu karakterlerin kimlik avı özelliklerini gösterip göstermediğini değerlendirmez. Buna karşın, karakter tabanlı öznitelikler bu karakterleri açık bir şekilde dikkate alarak bir yöntemin kaçırdığı özelliklerin diğer yöntem tarafından yakalanmasını sağlar. Bu iki katmanlı analiz, modelin karmaşık URL desenlerini belirleme yeteneğini önemli ölçüde artırır. Güçlü yönlerine rağmen, modelin bazı sınırlamaları ve zayıf yönleri de vardır. Bunlardan biri, kelime uzunluklarının farklılık göstermesi ve bu durumun sınıflandırma performansı üzerinde etkili olabilmesidir. URL'lerin "." ve "/" gibi karakterler temel alınarak belirteçlere ayrılması, tek karakterden çok karakterli kelimelere kadar değişen uzunluklarda belirteçlerin oluşmasına yol açar. Kitaplar veya makaleler gibi yazılı metinlerde belirteçler genellikle boşluklarla ayrılırken, URL'lerin birleştirilmiş yapısı, kelime uzunluklarında doğal bir değişkenlik oluşturmuştur. Özellikle tanımlı bir kelime uzunluğu aralığının bulunmaması bir zayıflık olarak değerlendirilir. Buna ek olarak, Word2Vec, Bag of Words (BoW) ve Term Frequency-Inverse Document Frequency (TF-IDF) gibi frekans tabanlı yöntemlerle oluşturulan temsiller, sözlükteki belirteç sayısının fazla olması nedeniyle bellek aşırı yüklenmesi ve kapasite sorunlarına yol açabilir. Bu durum, bellek verimliliğini artırmak için optimizasyon stratejilerine duyulan ihtiyacı açıkça ortaya koymaktadır. Ayrıca, bu çalışmada modelin gerçek zamanlı ortamlardaki performansı değerlendirilmemiştir, çünkü bu konu araştırmanın kapsamı dışında kalmaktadır.

Gelecekteki çalışmalar, bu sınırlamaları hedef alarak iyileştirme stratejilerine odaklanmalıdır. Bellek ve performans sorunlarını aşmak için kuantizasyon ve budama gibi tekniklerden yararlanılabilir. Ayrıca, önceden işlenmiş URL'ler için önbellekleme stratejilerinin uygulanması ve farklı öncelik seviyelerine sahip URL'ler için uygun modellerin atanması, maliyet-performans dengesini sağlamaya yardımcı olabilir. Kelime uzunluğunu kontrol edebilen bir algoritmanın geliştirilmesi, modelin sağlamlığını daha da artırabilir. Ancak, böyle bir algoritmanın geleneksel karakter veya kelime tabanlı n-gram yöntemlerinden farklı olması gerekmektedir. Karakter n-gramları, sabit sayıda önceki veya sonraki karakterleri temel alırken, kelime n-gramları belirli sayıdaki önceki veya sonraki kelimeleri dikkate almaktadır. Ancak, URL belirteçleme süreçlerinde doğal olarak değişkenlik gösterdiğinden, geleneksel n-gram yaklaşımları her zaman ideal bir çözüm sunmayabilir. Bu nedenle, URL'lerin kendine özgü yapısına uygun özel ve yenilikçi yaklaşımların geliştirilmesi gerekmektedir.

Gerçek zamanlı tespit, gelecekteki araştırmalar için önemli bir çalışma alanı olmaya devam etmektedir. Modelin gerçek zamanlı ortamlarda nasıl performans gösterdiğini ve tepki süresini analiz etmek, daha önce karşılaşılmamış URL'lere nasıl tepki verdiğini anlamak açısından büyük önem taşımaktadır. Örneğin, öncelikli URL'ler için daha karmaşık modeller atanabilirken, düşük risk seviyesine sahip URL'ler için daha hafif ve basit modeller tercih edilebilir. Bu tür stratejiler, modelin büyük veri setleriyle çalışabilme yeteneğini artırarak gerçek zamanlı uygulamalara daha uygun hale gelmesini sağlayabilir. Buna ek olarak, bu tür iyileştirmeler, sürekli değişen tehdit ortamında kimlik avı saldırılarının tespit edilmesine yönelik modelin uyarlanabilirliğini ve etkinliğini güçlendirecektir.

7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, kimlik avı saldırılarını tespit edebilen bir derin öğrenme modeli önerilmiştir. Model, kimlik avı ve yasal adresleri doğru bir şekilde ayırt edebilmek için URL adreslerini karakter ve kelime tabanlı öznitelik çıkarma yöntemleriyle incelemiştir. GramBeddings, Malicious and Benign URLs ile Ebbu2017 Phishing veri setleri üzerinde gerçekleştirilen deneyler, modelin hem dengeli hem de dengesiz veri setlerinde yüksek doğruluk, kesinlik ve AUC değerleri elde ettiğini göstermiştir. Bunun yanı sıra, Mendeley Data Web Page Phishing Detection veri seti üzerinde yapılan testler, modelin küçük ölçekli veri kümeleri üzerindeki başarısını ortaya koymuş ve bellek kısıtlamaları giderildiğinde çapraz doğrulamanın uygulanabileceğini göstermiştir. Model, URL'lerin yapısal detaylarını yakalamak için karakter tabanlı öznitelik çıkarma yöntemlerini, anlamsal içeriğini değerlendirmek için ise kelime tabanlı öznitelik çıkarma tekniklerini kullanmıştır. Bu iki yönlü yaklaşım sayesinde model, kimlik avı saldırılarını hem karakter hem de kelime düzeyinde analiz ederek, bir yöntem tarafından gözden kaçabilecek saldırıların diğer yöntemle tespit edilmesine imkan tanımaktadır. Böylece, hata toleransı artırılarak karmaşık kimlik avı stratejilerinin algılanma yeteneği güçlendirilmiştir. BiLSTM algoritması ve dikkat mekanizması ile desteklenen model, yüksek boyutlu öznitelikleri anlamlı ve yoğun vektörlere dönüştürerek öğrenme sürecini iyileştirir ve manuel özellik mühendisliği gereksinimini ortadan kaldırır. Dil bağımsızlığı sayesinde, farklı dil ve karakter yapısına sahip URL'leri analiz ederken herhangi bir üçüncü taraf hizmetine ihtiyaç duymadan etkili bir şekilde çalışabilir. Aynı zamanda, yüksek boyutlu karakter ve kelime vektörlerini düşük boyutlu temsillere indirerek hem depolama maliyetlerini hem de eğitim süresini azaltır. Deneysel bulgular, önerilen modelin kimlik avı URL tespitinde mevcut yöntemlerden daha başarılı olduğunu ve yüksek kesinlik ile duyarlılık oranlarına ulaştığını kanıtlamıştır. Model, kimlik avı saldırılarına karşı güçlü bir savunma mekanizması sunarak internet güvenliğini ve kullanıcıların korunmasını desteklemekte, sürekli değişen kimlik avı tehditlerine karşı sürdürülebilir ve etkili bir çözüm sunmaktadır.

KAYNAKLAR

1. Özkaya, E., Sarıca, R. ve Durmaz, Ş. (2019). *Siber Güvenlik: Saldırı & Savunma Stratejileri* (1. Baskı). Türkiye: Buzdağı Yayınevi, 117-146.

2. İnternet: We Are Social, Digital 2024 Global Overview Report. (2024). Web: https://wearesocial.com/uk/blog/2024/01/digital-2024-5-billion-social-media-users/. Son Erişim Tarihi: 04.03.2025.

3. Do, N.Q., Selamat, A., Krejcar, O., Herrera-Viedma, E. ve Fujita, H. (2022). Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. *IEEE Access*, 10, 36429-36463.

4. Abdillah, R., Shukur, Z., Mohd, M. ve Murah, T.M.Z. (2022). Phishing Classification Techniques: A Systematic Literature Review. *IEEE Access*, 10, 41574-41591.

5. İnternet: Federal Bureau of Investigation, Internet Crime Complaint Centre Internet Crime 2023 Report. (2024). Web: https://www.ic3.gov/AnnualReport/Reports/2023_IC3Report.pdf. Son Erişim Tarihi: 04.03.2025.

6. Sahingoz, O.K., Buber, E., Demir, O. ve Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345-357.

7. Jain, A.K. ve Gupta, B.B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, 68(4), 687-700.

8. Catal, C., Giray, G., Tekinerdogan, B., Kumar, S. ve Shukla, S. (2022). Applications of deep learning for phishing detection: a systematic literature review. *Knowledge and Information Systems*, 64(6), 1457-1500.

9. Chanti, S. ve Chithralekha, T. (2022). A literature review on classification of phishing attacks. *International Journal of Advanced Technology and Engineering Exploration*, 9 (89), 446-476.

10. İnternet: Anti-Phishing Working Group, Phishing Activity Trends Report (2024). Web: https://apwg.org/trendsreports/. Son Erişim Tarihi: 04.03.2025.

11. Jafari, S. ve Aghaee-Maybodi, N. (2024). Detection of phishing addresses and pages with a data set balancing approach by generative adversarial network (GAN) and convolutional neural network (CNN) optimized with swarm intelligence. *Concurrency and Computation-Practice & Experience*, 36(11).

12. Marimuthu, S.K., Gopalasamy, S.K. ve Ben-Othman, J. (2022). Intelligent antiphishing framework to detect phishing scam: A hybrid classification approach. *Software-Practice & Experience*, 52(2), 459-481.

13. Aljofey, A., Jiang, Q.S., Qu, Q., Huang, M.Q. ve Niyigena, J.P. (2020). An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. *Electronics*, 9(9).

14. Zhang, X., Habibi Lashkari, A. ve Ghorbani, A.A. (2020). Classifying and clustering malicious advertisement uniform resource locators using deep learning. *Computational Intelligence*, 37(1), 511-537.

15. Azeez, N.A., Misra, S., Margaret, I.A., Fernandez-Sanz, L. ve Abdulhamid, S.M. (2021). Adopting automated whitelist approach for detecting phishing attacks. *Computers & Security*, 108.

16. Gupta, B.B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A. ve Chang, X.J. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, 175, 47-57.

17. Zieni, R., Massari, L. ve Calzarossa, M.C. (2023). Phishing or Not Phishing? A Survey on the Detection of Phishing Websites. *IEEE Access*, 11, 18499-18519.

18. Tang, L. ve Mahmoud, Q.H. (2021). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. *Machine Learning and Knowledge Extraction*, 3(3), 672-694.

19. Salloum, S., Gaber, T., Vadera, S. ve Shaalan, K. (2022). A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. *IEEE Access*, 10, 65703-65727.

20. Bozkir, A.S., Dalgic, F.C. ve Aydos, M. (2023). GramBeddings: A New Neural Network for URL Based Identification of Phishing Web Pages Through N-gram Embeddings. *Computers & Security*, 124, 102964.

21. Ahmetoglu, H. ve Das, R. (2022). A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions. *Internet of Things*, 20, 100615.

22. Zheng, F., Yan, Q., Leung, V.C.M., Yu, F.R. ve Ming, Z. (2022). HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection. *Computers & Security*, 114, 102584.

23. Wang, W.P., Zhang, F., Luo, X. ve Zhang, S.G. (2019). PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks. *Security and Communication Networks*, 2019.

24. Afzal, S., Asim, M., Javed, A.R., Beg, M.O. ve Baker, T. (2021). URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models. *Journal of Network and Systems Management*, 29(3).

25. Elsadig, M., Ibrahim, A.O., Basheer, S., Alohali, M.A., Alshunaifi, S., Alqahtani, H., Alharbi, N. ve Nagmeldin, W. (2022). Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction. *Electronics*, 11(22), 3647.

26. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D. ve Rodríguez-Galán, G. (2023). A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning. *Applied Sciences*, 13(9), 5275.

27. Yang, P., Zhao, G. ve Zeng, P. (2019). Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. *IEEE Access*, 7, 15196-15209.

28. Alshehri, M., Abugabah, A., Algarni, A. ve Almotairi, S. (2022). Character-level word encoding deep learning model for combating cyber threats in phishing URL detection. *Computers & Electrical Engineering*, 100, 107868.

29. Ozcan, A., Catal, C., Donmez, E. ve Senturk, B. (2023). A hybrid DNN-LSTM model for detecting phishing URLs. *Neural Computing and Applications*, 35(7), 4957-4973.

30. Rasymas, T. ve Dovydaitis, L. (2020). Detection of Phishing URLs by Using Deep Learning Approach and Multiple Features Combinations. *Baltic Journal of Modern Computing*, 8(3), 471-483.

31. Rao, R.S., Vaishnavi, T. ve Pais, A.R. (2019). PhishDump: A multi-model ensemble based technique for the detection of phishing sites in mobile devices. *Pervasive and Mobile Computing*, 60, 101084.

32. Xiao, X., Xiao, W.T., Zhang, D.Y., Zhang, B., Hu, G.W., Li, Q. ve Xia, S.T. (2021). Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets. *Computers & Security*, 108, 102372.

33. Roy, S.S., Awad, A.I., Amare, L.A., Erkihun, M.T. ve Anas, M. (2022). Multimodel Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models. *Future Internet*, 14(11), 340.

34. Aslam, S., Aslam, H., Manzoor, A., Chen, H. ve Rasool, A. (2024). AntiPhishStack: LSTM-Based Stacked Generalization Model for Optimized Phishing URL Detection. *Symmetry*, 16(2), 248.

35. Zaimi, R., Hafidi, M. ve Lamia, M. (2024). A deep learning mechanism to detect phishing URLs using the permutation importance method and SMOTE-Tomek link. *The Journal of Supercomputing*, 80(12), 17159-17191.

36. İnternet: Malicious and Benign URLs. (2019). Web: https://www.kaggle.com/datasets/siddharthkumar25/malicious-and-benign-urls. Son Erişim Tarihi: 13 May 2024.

37. Hannousse, A., Yahiouche, S., 2021. Web page phishing detection, Mendeley Data. V3. 10.17632/c2gw7fy2j4.3

38. Asiri, S., Xiao, Y., Alzahrani, S., Li, S.H. ve Li, T.S. (2023). A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks. *IEEE Access*, 11, 6421-6443.

39. Chollet, F. (2021). *Python ile Derin Öğrenme* (2. Baskı). Ankara: Buzdağı Yayınevi, 48-169.

40. Wang, H.H., Yu, L., Tian, S.W., Peng, Y.F. ve Pei, X.J. (2019). Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network. *Applied Intelligence*, 49(8), 3016-3026.

41. Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K. ve Faheem, M.H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics*, 12(1), 232.

42. Ali, W. ve Ahmed, A.A. (2019). Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting. *IET Information Security*, 13(6), 659-669.

43. Zhu, E.Z., Yuan, Q.X., Chen, Z.L., Li, X.J. ve Fang, X.Y. (2023). CCBLA: A Lightweight Phishing Detection Model Based on CNN, BiLSTM, and Attention Mechanism. *Cognitive Computation*, 15(4), 1320-1333.

44. Xiao, X., Zhang, D.Y., Hu, G.W., Jiang, Y. ve Xia, S.T. (2020). CNN-MHSA: A Convolutional Neural Network and multi-head self-attention combined approach for detecting phishing websites. *Neural Networks*, 125, 303-312.



Gazili olmak ayrıcalıktır