NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2025; 14(4), 1498-1508
Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences
Araştırma makalesi / Research article
www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

İki aşamalı kötücül URL tespit mimarisi: Yeni nesil tehdit tanıma ve sınıflandırma yaklaşımı

Two-Stage malicious URL detection architecture: A next-generation threat recognition and classification approach

Durmuş Özkan Şahin[1,*], Sercan Demirci [2], Muhammet Abdullah Şahin [3], Nuri Can Acar [4], Hamit Burak Can Kodal [5]

[1,2,3,4,5] *Ondokuz Mayıs Üniversitesi, Bilgisayar Mühendisliği Bölümü, 55139, Samsun, Türkiye*

Öz

Günümüzde çevrimiçi tehditlerin artması, zararlı URL'lerin tespit edilmesini ve zarar türlerine göre sınıflandırılmasını önemli bir araştırma konusu haline getirmiştir. Bu çalışma, zararlı URL'lerin tespit edilmesi ve zarar türlerinin belirlenmesi amacıyla farklı makine öğrenimi algoritmaları ile özellik seçme yöntemlerinin kombinasyonları üzerine değerlendirme yaparak performanslarını karşılaştırmayı hedeflemektedir. Çalışmada üç farklı veri seti kullanılmıştır. Bunlardan birincisi ISCX-2016 veri seti olup 79 farklı özellik içermektedir. Ham URL'lerden oluşan diğer iki veri setinde bulunan her bir URL için aynı 79 özellik çıkarılmıştır. Modelleme sürecinde, ilk aşamada zararlı URL'lerin doğru bir şekilde sınıflandırılmasına odaklanılmış, ikinci aşamada ise zarar türlerinin belirlenmesinde kullanılan algoritmaların etkinliği karşılaştırılmıştır. Özellik seçimi uygulanmadan kullanılan Rastgele Orman algoritması, tüm özellikler üzerinden değerlendirildiğinde, hem ikili sınıflandırmada (%97 doğruluk) hem de çok sınıflı sınıflandırmada (%98 doğruluk) en yüksek performansa ulaşmıştır. Bu bulgular, zararlı URL tespit sistemlerinin geliştirilmesi açısından önemli bir rehber niteliği taşımakta ve literatüre değerli katkılar sunmaktadır.

Anahtar kelimeler: Bilgisayar güvenliği, Kötü amaçlı yazılım tespiti, Makine öğrenimi algoritmaları, Özellik çıkartma, URL sınıflandırma, Zararlı URL tespiti

Abstract

The increasing prevalence of online threats has made the detection and classification of malicious URLs a critical research topic. This study aims to evaluate and compare the performance of different machine learning algorithms combined with feature selection methods for detecting malicious URLs and determining their attack types. Three different datasets were used in the study. The first dataset, ISCX-2016, contains 79 distinct features. The same 79 features were extracted for each URL in the other two raw URL datasets. In the modeling process, the first phase focused on accurately classifying malicious URLs, while the second phase compared the effectiveness of algorithms in determining attack types. The Random Forest algorithm, when applied without feature selection and evaluated across all features, achieved the highest performance in both binary classification (97% accuracy) and multi-class classification (98% accuracy). These findings serve as a valuable guide for the development of malicious URL detection systems and provide significant contributions to the literature.

Keywords: Cybersecurity, Feature extraction, Machine learning algorithms, Malicious URL detection, Malware detection, URL classification

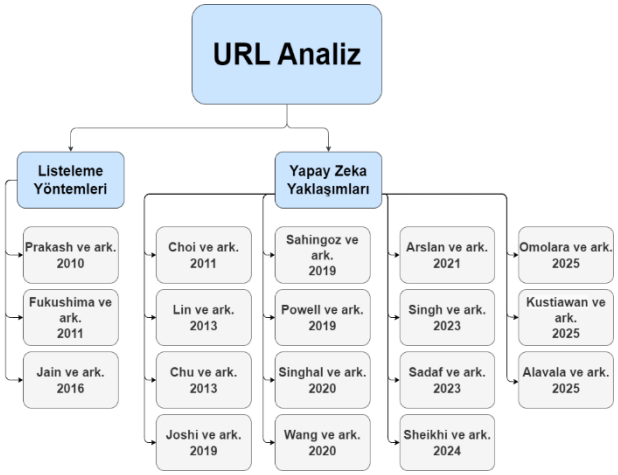
1 Giriş

Birçok kişinin bilgi güvenliği konusunda yeterli bilgiye ve bilince sahip olmadığı görülmektedir. Bu durum, web üzerinde dikkatsiz davranışlar sergilemelerine ve kişisel verilerini istemeden açığa çıkarmalarına yol açabilmektedir. Ayrıca, kişisel donanımları veya iş istasyonları zararlı yazılımlara maruz kalabilir ve istenmeyen reklamlar gibi beklenmedik durumlarla karşılaşmaları kaçınılmaz hale gelebilir. Bu doğrultuda, çalışma kapsamında web üzerindeki sitelerin URL adresleri üzerinden tahmin gerçekleştirilmesi amaçlanmaktadır.

İnternetin yaygınlaşmasıyla birlikte, bu tür zararlı faaliyetler de artmış ve bilgi güvenliği konusu giderek daha kritik bir hale gelmiştir [1]. Bu konuya ilişkin literatürde farklı çalışmalar mevcuttur. İnternet üzerinde vakit geçirirken, farklı alan adlarına ait IP adreslerine Tekdüzen Kaynak Bulucular (URL: Uniform Resource Loader) aracılığıyla erişim sağlanmaktadır. URL'ler, kullanıcıların internetteki kaynaklara kolayca ulaşmasını sağlayan bir köprü işlevi görerek, internet trafiğinin yönlendirilmesinde kritik bir rol üstlenmektedir. Bunlardan bazıları zararlı sınıfına girebilmektedir [2]. Bu çalışmada makine öğrenmesi tabanlı URL analizi yapılması amaçlanmıştır. Bu sayede URL'lerin kötücül veya iyicil olarak ayırt edilmesi, kötücül URL'lerin ise hangi tür olduğunun tespit edilmesi sağlanmıştır.

* Sorumlu yazar / Corresponding author, e-posta / e-mail: durmus.sahin@bil.omu.edu.tr (D. Ö. Şahin)
Geliş / Received: 14.04.2025 Kabul / Accepted: 17.09.2025 Yayımlanma / Published: 15.10.2025
doi: 10.28948/ngumuh.1675785

URL'lerin sınıflarına ayrılması son yıllarda bilgi güvenliği alanında önem arz etmektedir. Bu bağlamda birçok araştırmacı bu alanda başarımı yüksek tespit sistemleri geliştirmektedir. Bu konu üzerine yapılan çalışmalar genel olarak Şekil 1'de gösterildiği gibi sınıflandırılabilmektedir.



Şekil 1. Yapılan çalışmaların taksonomisi.

Listeleme tabanlı yaklaşımlar, beyaz liste (whitelist) ve kara liste (blacklist) sınıflandırma olarak incelenmektedir. Bu yaklaşımı uygulayan Prakash ve ark. [3], kimlik avı saldırılarını engellemeye yönelik kara liste oluşturma üzerinde çalışmışlardır. Üst düzey alan adı filtreleme, IP adresi süzgeci, dizin incelemeleri, sorgu analizi ve marka adlarını araştırarak doğrulama yapılmıştır. Bu süzgeç işlemi sonrasında bu IP adreslerini kara listeye eklemişlerdir. Fukushima ve ark. [4], zararlı URL'leri "exploit" ve "redirect" gibi sözcükleri filtreleyerek ayrıştırmışlardır. Ayrıca saldırganın alan adı değişikliği yapabileceklerine dikkat çekerek, IP adreslere göre sınıflandırılmasını önermişlerdir. Jain ve Gupta [5], kimlik avı saldırılarına karşı otomatik olarak güncellenen bir beyaz liste yapısı önermişlerdir. İki modüllü bir sistem, kullanıcı yeni bir sisteme eriştiğinde bir algoritmanın devreye girmesiyle çalışır. Algoritma, sistemin zararsızlık durumunu analiz eder ve zararsız olduğu tespit edilen sistemleri güvenli listeye ekler. Eğer daha önceden ziyaret edilmiş bir sistem ise buna da Alan Adı Sistemi (DNS: Domain Name System) zehirlenmesi saldırısına yönelik eşleştirme yaparak çalışmaktadır. Önerdikleri bu yöntem hem güvenlik hem de erişim verimliliği sağlamak amacıyla geliştirilmiştir.

Yapay zekâ yaklaşımları ile yapılan çalışmalarda, Choi ve ark. [6], zararlı URL tespitinde Destek Vektör Makinesi (SVM: Support Vector Machine), tür tespitinde ise Rastgele K-Etiket Setleri (RAKEL: Random K-Labelsets) ve Çoklu Etiket K-En Yakın Komşular (MLKNN: Multi Label K-Nearest Neighbors) algoritmalarını kullanmıştır. Çalışmada, 82,000 zararlı URL üzerinde sözcüksel özellikler çıkartılarak Yahoo, Web spam, PhishTank, DNS-BH veri setleri ile %98 doğruluk oranı ve %93 saldırı türü tespiti başarımına ulaşılmıştır. Chu ve ark. [7], kimlik avı URL'leri üzerinden sözcüksel ve alan adı özelliklerini kullanarak Taobao-phishin URLs ve Yahoo veri setleri ve SVM algoritmasıyla %99'un üzerinde tespit sağlayan yöntem ortaya koymuştur. Lin ve ark. [8], sözcüksel özellikleri filtreleyerek Online Learning, Batch Learning, Passive-Aggressive, Confidence Weighted algoritmalarının bir güvenlik şirketinden aldığı veri seti üzerinde kullanarak algoritmaların başarımlarını incelemiştir. Joshi ve ark. [9], Openphish, Alexa, FireEye ile elde edilen veri setlerini, Rastgele Orman, Gradient Arttırma, AdaBoost, Lojistik Regresyon ve Naive Bayes algoritmaları karşılaştırılarak incelenmiştir. En yüksek sonucu ise Rastgele Orman ile %92 olarak elde edilmiştir. Powell ve ark. [10], NSL-KDD, ISCX-URL-2016, CICIDS-2017 veri setleri üzerinde özellik seçilerle ve Lojistik Regresyon, K-En Yakın Komşu, Karar Ağacı ve Rastgele Orman algoritmalarını kullanarak karşılaştırma sonucu elde edilmiştir. En iyi sonucu ise Rastgele Orman kullanarak %99'un üzerinde görülmüştür. Şahingöz ve ark. [11], kimlik avı tespitinde Rastgele Orman, Native Bayes vb. 7 farklı algoritmayı karşılaştırmıştır. Wang ve ark. [12], Naive Bayes sınıflandırıcı ile ISCX-URL-2016 veri seti üzerinde özellikleri değerlendirerek %90'a varan doğruluk oranı görülmüştür. Signhal ve ark. [13], kavram kaymasına değinerek PhishTank, Malware Domain List, Majestic elde edilen veri setleri üzeninde Gradient Arttırma algoritmasıyla %96'ya yaklaşan sonuç elde ettiği görülmüştür. Arslan [14], Doc2Vec ağında DM ve DBOW algoritmalarını kullanarak URL'lerden özellik vektörü çıkarmıştır. ISCX-URL-2016 veri seti üzerinde Lojistik Regresyon ile %99'un üzerinde başarı elde ettiği görülmüştür. Singh ve ark. [15], ISCX-URL2016 veri kümesini üzerinde HistGradBoost, XGBoost, CatBoost, Ekstra Ağaç vb. birçok performansını değerlendirmişlerdir. Ayrıca bu algoritmalar sınıflandırıcılar ile kombinasyonlanarak doğruluk oranının yükseltilmesi sağlanmıştır. En yüksek sonuç tüm ana sınıflandırıcılar ile meta sınıflandırıcı olarak XGBoost algoritması birleştirilerek elde edilmiştir. Bu sonuç doğruluk metriğine göre %98'in üzerine çıkmıştır. Sadaf [16], UCI makine öğrenmesi deposundan alınan iki farklı veri seti üzerinde yaptığı çalışmada, XGBoost ve Catboost sınıflandırıcılarının performansını hiperparametre ayarı olmaksızın değerlendirmiştir. Çalışmanın sonucunda, XGBoost'un %96.79 doğruluk oranı ile Catboost'a kıyasla marjinal de olsa daha iyi bir sonuç elde ettiğini ortaya koymuştur. Sheikhi ve Kostakos [17], ISCX-URL2016 veri setini kullanarak çalışmalarında, Firefly tabanlı özellik seçimi ve Parçacık Sürü Optimizasyonu (PSO) ile optimize edilmiş XGBoost algoritmasını değerlendirmiştir. Çalışmada, Firefly algoritması ile en uygun özellikler belirlenmiş ve ardından PSO-XGBoost modeli kullanılarak sınıflandırma gerçekleştirilmiştir. Çok sınıflı sınıflandırmada ise %98'in üzerinde doğruluk elde edilmiştir. Omolara ve Alawida [18], ISCX-URL2016 veri seti üzerinde geliştirdikleri DaE2 (Diverse and Efficient Ensemble) yaklaşımıyla AdaBoost, Bagging, Stacking ve Voting topluluk algoritmalarını karşılaştırmış ; en yüksek başarıyı %98.5 doğruluk oranıyla AdaBoost modelinin elde ettiğini göstermişlerdir. Kustiawan ve Ghauth [19], PhishOFE veri seti üzerinde yaptıkları çalışmada, URL ve HTML tabanlı özellikler ile bunlardan türetilmiş sentetik özelliklerin oltalama tespitindeki etkisini incelemişlerdir. CatBoost, XGBoost ve LightGBM gibi topluluk öğrenmesi algoritmalarını karşılaştırdıkları

analizlerinde, tüm özellikleri bir arada kullanan CatBoost modeli ile %99.45 doğruluk oranı elde ederek en yüksek başarıma ulaşmışlardır. Alavala ve ark. [20], 10.000 URL'den oluşan derleme bir veri seti üzerinde, geleneksel (Random Forest, SVM), gelişmiş (XGBoost, CatBoost) ve derin öğrenme (LSTM, CNN) modellerinin performansını karşılaştırmış; en yüksek başarıyı %91.85 doğruluk oranıyla CatBoost modelinin elde ettiğini göstermişlerdir.

İncelenen çalışmaların büyük çoğunluğunun kimlik avı saldırılarının tespiti üzerine yoğunlaştığı, bazı çalışmaların ise kısıtlı veri kümeleriyle gerçekleştirildiği görülmüştür. Ayrıca, URL'lerin sözcüksel özelliklerine dayalı ayrıştırma işleminin yapılan çalışmalarda yer almadığı tespit edilmiştir. Veri kümesi çeşitliliği açısından değerlendirildiğinde, kimlik avı saldırılarına odaklanan çalışmaların ağırlıkta olduğu, ancak kötü amaçlı yazılım barındıran veya sahte siteler aracılığıyla yönlendirme yapan zararlı alan adları gibi diğer tehdit türlerine yönelik araştırmaların sınırlı kaldığı gözlemlenmiştir. Bu durum, geliştirilen modellerin farklı tehdit türleri karşısındaki performansını değerlendirme açısından önemli bir eksiklik oluşturmaktadır.

Özellikle modern siber saldırılar, yalnızca kimlik avı teknikleriyle sınırlı kalmayıp çok katmanlı ve gelişmiş tehdit unsurlarını içermektedir. Mevcut çalışmalar incelendiğinde, kullanılan veri setlerinin genellikle sınırlı sayıda URL içerdiği görülmektedir. Bu nedenle, çalışmada veri seti genişletilerek 1 milyondan fazla URL üzerinde analiz gerçekleştirilmiştir. Önerilen model, iki aşamalı bir tespit yöntemi ile geliştirilmiştir. İlk aşamada bir URL'nin zararlı olup olmadığı belirlenirken, ikinci aşamada zararlı olarak sınıflandırılan URL'lerin türü analiz edilmiştir. Sonuç olarak, geniş ölçekli bir veri kümesi üzerinde test edilen bu yaklaşım hem zarar tespiti hem de zarar türü açısından daha güvenilir ve kapsamlı bir analiz sunmaktadır.

Literatür taraması sonucunda, yapay zekâ yaklaşımlarının geniş veri setleri ile kullanıldığında en verimli ve geliştirilmeye açık yöntem olduğu tespit edilmiştir. Bu nedenle, çalışma kapsamında büyük ölçekli veri setleri kullanılarak çeşitli algoritmalar karşılaştırılıp değerlendirilecektir. Mevcut çalışmalar incelendiğinde, eğitim sürecinin genellikle tek aşamalı olarak gerçekleştirildiği görülmüştür. Bu çalışmada ise; ilk aşamada, URL'lerin zararlı olup olmadığına yönelik ikili sınıflandırma yapılmış, ikinci aşamada ise zararlı olarak belirlenen URL'lerin zarar türleri tespit edilerek modelin ikinci aşaması oluşturulmuştur. Bu yapı sayesinde daha kapsamlı bir analiz gerçekleştirilmiştir.

Önerilen model, gelecekte bir araca entegre edilerek kullanıcı deneyimine sunulabilecektir. Ayrıca, çalışmada URL'lerin sözcüksel özelliklerini çıkarmaya yönelik bir ayrıştırma aracı geliştirilmiştir. Bu araç, gelecekte oluşturulacak yeni veri setleriyle birlikte kullanılabilecek ve farklı veri setleri üzerinde ortak çalışmalar yürütülmesine olanak sağlayacaktır.

Giriş bölümünde, zararlı URL tespiti ve zarar türlerine göre sınıflandırmanın önemi ele alınarak çalışmanın amacı ve katkıları açıklanmıştır. Literatür taraması kısmında, bu alanda kullanılan makine öğrenimi ve özellik seçme algoritmalarına ilişkin mevcut çalışmalar incelenmiştir. Materyal ve metot bölümünde, kullanılan veri setleri, özellik çıkarımı süreçleri ve algoritmalar detaylı şekilde anlatılmıştır. Ayrıca bu sürecin nasıl işlediğine dair açıklamalar verilmiştir. Bulgular ve tartışma bölümünde, gerçekleştirilen deneyler ve modellerin karşılaştırmalı analizleri sunularak performans değerlendirmeleri yapılmıştır. Son olarak, sonuç bölümünde elde edilen bulgular tartışılmış ve araştırmanın geliştirilebilecek yönleri üzerinde durulmuştur.

2 Materyal ve metot

Bu bölüm üç alt bölümden oluşmaktadır. İlk alt bölümde çalışmada kullanılan veri kümeleri tanıtılacaktır. İkinci alt bölümde çalışmada takip edilen veri ön işleme adımlarına değinilecektir. Üçüncü alt bölümde kullanılan sınıflandırma algoritmaları verilecektir. Son olarak dördüncü alt bölümde ise sınıflandırıcı performanslarını ölçmede kullanılan metriklere değinilecektir.

2.1 Kullanılan veri kümeleri

Çalışmada kullanılan veri kümeleri; ISCX-URL-2016 [21], Kaggle Malicious URLs Dataset [22] ve Kaggle URL Dataset [23] veri kümeleridir. Bu veri kümeleri, eğitim aşamasının iki farklı bölümünde değerlendirilmiştir. Veri setlerinin URL'lerin türlerine göre dağılımı kapsamlı bir şekilde, Tablo 1'de gösterilmektedir. Tablo 1, her bir veri setinin toplam URL sayısını, zararsız URL'lerin miktarını ve zararlı URL'lerin türlerine göre dağılımını içermektedir. Çalışmada, veri kümelerinin %70'i eğitim, %30'u ise test seti olarak ayrılmıştır. Modellerin tutarlılığını sağlamak ve her birinin aynı eğitim ve test kümesi üzerinde işlem yapmasını temin etmek amacıyla, sklearn kütüphanesindeki random_state parametresi 42 değeriyle sabitlenmiştir. Bu şekilde, veri kümesinin rastgele bölünmesi her seferinde aynı şekilde gerçekleştirilmiş ve elde edilen sonuçların karşılaştırılabilirliği sağlanmıştır. İki aşamalı modelde hangi veri kümelerinin hangi aşamada kullanıldığı Tablo 1'de gösterilmektedir.

ISCX- URL-2016 veri seti içerisinde toplamda 114250 adet URL içermekte olup, bu URL'ler zararsız (35300), tahrif (45450), kötü amaçlı yazılım (11500), kimlik avı (10000) ve istenmeyen (12000) kategorileri altında sınıflandırılmıştır. Veri seti, her bir URL için 79 özellik ve 1 etiket içermektedir. URL'lerde bulunan özelliklerin sayısı oldukça fazladır ve bu veri setinde ham URL tanımlaması yapılmamış; bunun yerine, URL'lerin özellikleri sayısal değerler aracılığıyla zararlı ya da zararsız olarak sınıflandırılmıştır. Bu özelliklerden bazılarına, Tablo 2'de yer verilmektedir. ISCX- URL-2016 veri seti Tablo 1'de belirtildiği üzere, zararlılık tespiti ve zarar türü tespitinde kullanılmıştır.

Kaggle Malicious URLs Dataset, Kaggle platformunda, CC0: Public Domain lisansı ile paylaşılan bir veri setidir. Bu veri seti, toplamda 651191 URL içermekte olup, zararsız ve zararlı URL'lerin sınıflandırılmış biçimde sunulduğu kapsamlı bir kaynaktır. Veri setinde 428103 zararsız URL, 96457 tahrif URL, 94111 kimlik avı URL ve 32520 kötü amaçlı yazılım içeren URL yer almaktadır. Kaggle Malicious URLs veri seti, Tablo 1'de belirtildiği üzere, zararlılık tespiti ve zarar türü tespitinde kullanılmıştır.

Tablo 1. Veri seti içerikleri.

Veri Seti	Zararsız	Zararlı				Toplam URL	Zararlılık Tespiti	Zarar Türü Tespiti
		Tahrif	Kimlik Avı	Kötü Amaçlı Yazılım	İstenmeyen			
ISCX-URL-2016	35300	45450	10000	11500	12000	114250	✓	✓
Kaggle Malicious URLs Dataset	428103	96457	94111	32520		651191	✓	✓
Kaggle URL Dataset	344821			**75643**		420464	✓	×
Toplam	**808224**	**141907**	**104111**	**44020**	**12000**	**1185905**		

Kaggle URL Dataset, Kaggle platformunda CC0: Public Domain lisansı ile paylaşılan bir veri setidir. Bu veri seti, toplamda 420464 URL içermekte olup, zararsız ve zararlı URL olarak iki sınıfa ayrılmıştır. Veri setinde 344821 zararsız URL ve 75643 zararlı URL bulunmaktadır. Kaggle URL veri seti Tablo 1'de belirtildiği üzere, sadece zararlılık tespitinde kullanılmıştır.

2.2 Veri ön işleme

Kaggle Malicious URLs Dataset ve Kaggle URL Dataset veri setleri içerisindeki URL'lerin sözcüksel özellikleri ISCX-URL-2016 veri seti referans alınarak çıkartılmıştır. 79 özellik ve 1 etiket şeklinde diğer veri kümelerinde bulunan URL'ler için özellik çıkarımı yapılmıştır.

Bu kapsamda, URL'lerin yapısal, içerik tabanlı ve istatistiksel özellikleri belirlenmiş, zararlı ve zararsız URL'ler arasındaki farklılıkları tespit etmek amacıyla veri madenciliği ve analiz teknikleri uygulanmıştır.

Tablo 2. ISCX-URL-2016 özellik seti.

Özellik Adı	Tanım	Veri Tipi
Querylength	Sorgu (query) dizisinin karakter uzunluğu	Numerik
domain_token_count	Domain adının kaç parçaya (token) bölündüğü	Numerik
tld	Top-level domain (ör. "com", "org") ifadesi sayısal karşılığı.	Numerik
+76 özellik	…	…
URL_Type_obf_Type	URL'in zarar türü etiketi	Kategorik

ISCX-URL-2016 veri setindeki benzersiz URL'ler belirlenmiş olup, toplamda 36707 URL elde edilmiştir. Ardından, veri setinde yer alan boş değer içeren satırlar, Python programlama dili ile pandas kütüphanesi kullanılarak veri setinden çıkarılmıştır. Bu işlem sonucunda, tüm veri setlerinde toplam 1107364 veriden hatasız olarak 1089639 veri elde edilmiştir. ISCX-URL-2016 veri setinde bulunan 17725 boş değer içeren kayıt silinirken, diğer veri setlerinde yer alan hatalı veriler de çıkartılmıştır. Kaggle URL Dataset üzerinde hatalı bulunan 1 veri ve Kaggle Malicious URLs Dataset üzerinde de 903 hatalı veri çıkartılmıştır. Bu veri kümeleri üzerinde sözcüksel özelliklere ayrıştırma işlemleri kontrollü bir şekilde gerçekleştirildiğinden herhangi bir boş değer veya benzeri bir veri sorunu ile karşılaşılmamıştır.

2.3 Kullanılan sınıflandırma algoritmaları

Bu bölümde, özellik seçimi ve zararlılık-zarar türü tespiti için kullanılacak makine öğrenimi algoritmalarından bahsedilmiştir.

Tablo 3. Kullanılan algoritmalar

Aşama	Algoritmalar
Özellik Seçimi	SelectKBest – SelectPercentile – Ekstra Ağaç
Tespit Algoritmaları	AdaBoost, CatBoost, Karar Ağacı, Gradient Arttırma, Naive Bayes, Rastgele Orman, XGBoost

Bu çalışma kapsamında kullanılan algoritmalar Tablo 3'te verilmiştir. Tablo 3, literatürdeki ilgili çalışmalarda zararlı URL tespiti konusunda yüksek başarı gösterdiği kanıtlanmış yöntemler arasından özenle seçilerek oluşturulmuştur.

Bu doğrultuda özellik seçimi aşamasında, Singh ve ark. [15] tarafından da kullanılan Ekstra Ağaç (Extra Trees) algoritmasının yanı sıra, alanda sıkça başvurulan SelectKBest ve SelectPercentile yöntemleri tercih edilmiştir.

Bu çalışmada kullanılan ve Tablo 3'te belirtilen özellik seçme yöntemleri ve sınıflandırıcılar, herhangi bir hiperparametre optimizasyonu yapılmadan uygulanmıştır. ExtraTreesClassifier yalnızca random_state=42 parametresi ile sabitlenmiş, diğer tüm hiperparametreler scikit-learn kütüphanesinin varsayılan değerleriyle kullanılmıştır. SelectKBest ve SelectPercentile yöntemlerinde ise yalnızca seçilecek özellik sayısı veya yüzdelik dilim belirtilmiş, istatistiksel skor fonksiyonu f_classif varsayılan ayarlarıyla çalıştırılmıştır.

Tespit modelleri belirlenirken, özellikle CatBoost ve XGBoost algoritmalarının seçimi, bu modellerin güncel literatürde en yüksek başarım oranlarını sunmasına dayanmaktadır. Nitekim Alavala ve ark. [20] ile Kustiawan ve Ghauth [19], yaptıkları karşılaştırmalı analizlerde en yüksek doğruluk oranlarını CatBoost ile elde ederken; Sadaf [16] ise XGBoost'un marjinal bir üstünlük sağladığını göstermiştir. Bu durum, her iki algoritmanın da güncel ve en etkili yaklaşımlar olarak çalışmaya dahil edilmesini gerekli kılmıştır. Benzer şekilde, bir diğer topluluk öğrenmesi yöntemi olan AdaBoost, Omolara ve Alawida [18] tarafından %98.5 gibi çok yüksek bir doğruluk oranıyla öne çıkarıldığı için seçilmiştir. Bunların yanı sıra, Joshi ve ark. [9] ve Powell ve ark. [10] gibi çalışmalarda yüksek performans sergileyen Rastgele Orman ve Karar Ağacı [10] gibi temel sınıflandırıcılar da kullanılmıştır. Son olarak, Joshi ve ark. [9] ve Singhal ve ark. [13] tarafından başarıyla uygulanan Gradient Arttırma ile literatürde sıkça temel model olarak karşılaştırılan Naive Bayes [12] algoritmaları da model setine eklenmiştir. Bu seçimler, hem klasik ve temel hem de modern topluluk öğrenmesi yaklaşımlarını kapsayarak kapsamlı bir karşılaştırma ve değerlendirme yapmayı amaçlamaktadır.

Bu çalışma kapsamında, özellik seçimi metodolojileri ve özellik seçimi uygulanmayan durumlar için yedi farklı makine öğrenmesi sınıflandırıcısı kullanılarak toplam 28 model geliştirilmiş ve karşılaştırılmıştır. Modellerin tekrarlanabilirliğini ve tutarlılığını sağlamak amacıyla, random_state=42 parametresi Decision Tree, Gradient Boosting, Random Forest ve XGBoost gibi rassallık içeren tüm algoritmalarda sabitlenmiştir. Ensemble yöntemlerinden AdaBoost, 100 tahminci (n_estimators=100) ve 'SAMME' algoritması ile yapılandırılmıştır. Benzer şekilde, Rastgele Orman modeli de 100 tahminci ile eğitilmiştir. CatBoost modelinde 1000 iterasyon, 0.1 öğrenme oranı (learning_rate) ve 6 ağaç derinliği (depth) parametreleri tercih edilmiştir. XGBoost sınıflandırıcısı için ise 100 tahminci, 0.1 öğrenme oranı ve 3 maksimum derinlik (max_depth) ayarları kullanılmış; ikili sınıflandırma görevlerinde eval_metric olarak "logloss", çoklu sınıflandırma görevlerinde ise "mlogloss" belirlenmiştir. Naive Bayes sınıflandırıcısı ise herhangi bir hiperparametre optimizasyonu yapılmadan, GaussianNB implementasyonunun varsayılan (default) parametreleri ile modellere dahil edilmiştir. Bu parametre setleri, tüm özellik seçimi senaryolarında tutarlı bir şekilde uygulanarak, sadece özellik setinin model performansı üzerindeki etkisinin gözlemlenmesi hedeflenmiştir.

2.4 Değerlendirme metrikleri

Bu çalışmada, modellerin başarılarını değerlendirmek için doğruluk (accuracy-Acc), hassasiyet (precision-P), duyarlılık (recall-R), F1 skoru ve pozitif/negatif sınıflandırma analizinde kullanılan Doğru Pozitif (True Positive-TP), Yanlış Pozitif (False Positive-FP), Doğru Negatif (True Negative-TN), Yanlış Negatif (False Negative-FN) metrikleri kullanılmıştır. Kullanılan değerlendirme metrikleri, modellerin performansını çok boyutlu bir şekilde analiz etmeye olanak tanır.

Doğruluk değeri eşitlik Denklem (1), hassasiyet eşitlik Denklem (2), duyarlılık eşitlik Denklem (3) ve F1-Değeri eşitlik Denklem (4) ile hesaplanmıştır.

Doğruluk, doğru tahminlerin (TP + TN) tüm tahminlere (TP + FP + TN + FN) oranıdır. Denklem (1) ile gösterilmiştir.

Hassasiyet, modelin pozitif tahminlerinin doğruluk oranıdır. Yanlış pozitiflerin etkisini ölçmek için kullanılır. Denklem (2) ile gösterilmiştir.

Duyarlılık, gerçek pozitiflerin model tarafından ne kadar doğru yakalandığını ölçer. Denklem (3) ile gösterilmiştir.

F1-Değeri, Hassasiyet ve Duyarlılık arasında bir denge sağlar. Harmonik ortalama ile hesaplanır. Denklem (4) ile gösterilmiştir Bu hesaplamalar çoklu sınıflandırma için genişletilerek hesaplanmıştır.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

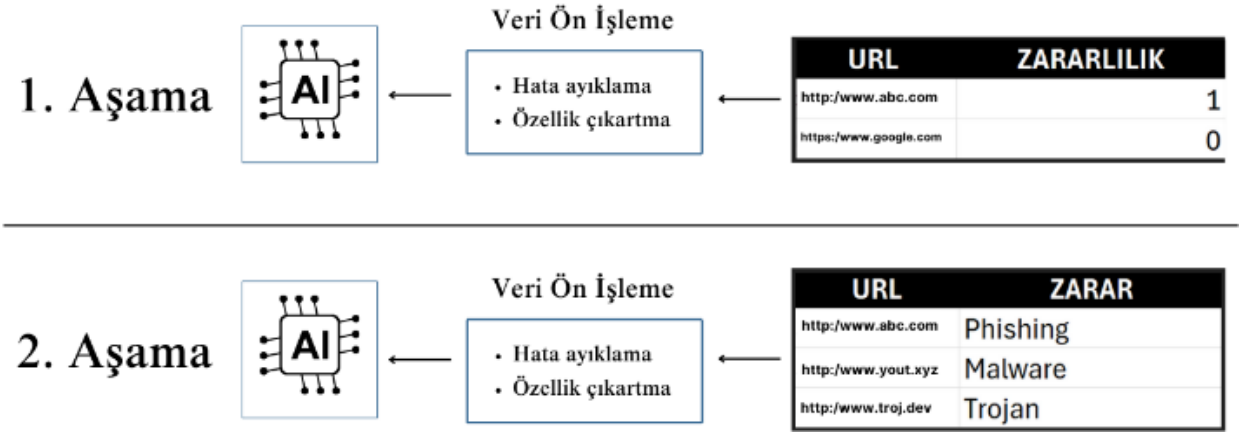
$$F1 = 2 * \frac{P * R}{P + R} \quad (4)$$

2.5 Yöntem

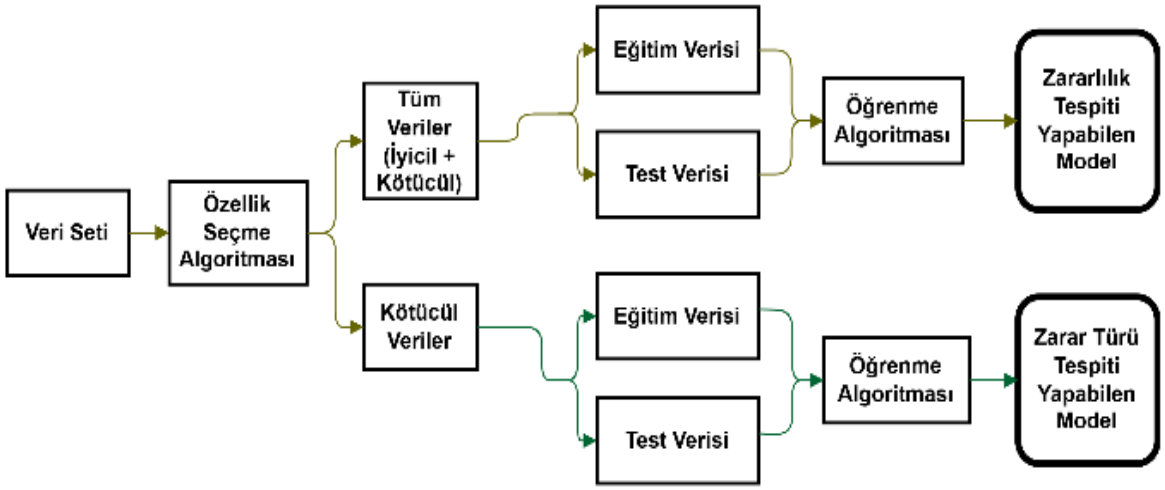
Bu çalışmada, zararlı URL'lerin tespit edilmesi ve zarar türlerinin belirlenmesi için iki aşamalı bir model yapısı uygulanmıştır. Modelin aşamaları Şekil 2'de gösterilmiştir. Her iki aşamada da ham URL şeklinde olan verilerin ortak bir formata getirilmesi amacıyla özellik çıkarma yöntemleri kullanılmış ve veriler, 79 özelliği çıkartılarak analize uygun hale getirilmiştir.

Modelin ilk aşaması, Zararlılık Tespiti üzerine diğer adıyla ikili sınıflandırmaya odaklanmaktadır. Bu aşamada, veri setindeki URL'ler iyicil (benign) ve kötücül (malicious) olarak etiketlenmiş, ardından bu veriler kullanılarak bir sınıflandırma modeli eğitilmiştir. Model, bir URL'nin zararlı olup olmadığını tespit edebilme yeteneğine sahip olacak şekilde eğitilmiştir. Modelin ikinci aşaması ise Zarar Türü Tespitine diğer adıyla çoklu sınıflandırmaya yöneliktir. Bu aşamada yalnızca kötücül URL'ler işleme alınmış, bu veriler üzerinden ayrı bir sınıflandırma modeli eğitilerek her bir zararlı URL'nin türünü belirlemeye yönelik bir model oluşturulmuştur. Bu model, zararlı URL'leri kimlik avı (phishing), kötü amaçlı yazılım (malware), tahrif (defacement) ve istenmeyen (spam) gibi farklı sınıflara ayırma yeteneğine sahiptir.

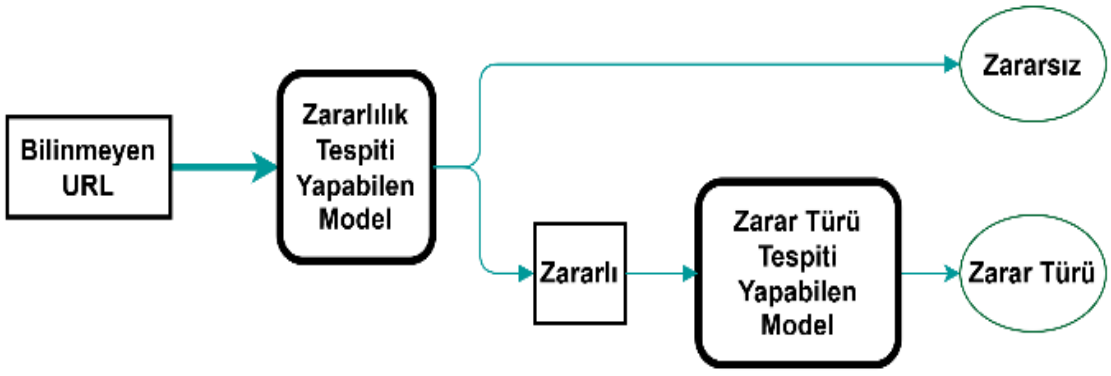
Modelin eğitim süreci Şekil 3'te gösterildiği gibi planlanmıştır. Öncelikle, 79 özellik ve 1 etiketten oluşan veri



Şekil 2. Algoritma eğitim şeması



Şekil 3. Veri setleri eğitim kullanımları



Şekil 4. Çalışma şeması

seti kullanılarak herhangi bir özellik seçme algoritması uygulanmadan 7 farklı makine öğrenimi algoritması kullanılarak modeller eğitilmiştir. Model eğitimi ilk paragrafta belirtildiği gibi iki aşamalı olarak gerçekleştirilmiştir İlk aşamada, tüm veri seti kullanılarak zararlılık tespiti için eğitim yapılmıştır ve ikili sınıflandırma yapabilen model oluşturulmuştur. Bu model, URL'leri zararsız (benign) veya zararlı (malicious) olarak ayırma yeteneğine sahiptir.

İkinci aşamada ise yalnızca zararlı olarak etiketlenmiş ve zarar türü belirlenmiş URL'ler kullanılarak çoklu sınıflandırma yapabilen model eğitilmiştir. Bu aşamada zararsız URL'ler kullanılmamış olup, model yalnızca zararlı türleri belirlemeye yönelik olarak çalıştırılmıştır. Çoklu

sınıflandırma modeli, zararlı URL'leri zarar türlerine ayırmak üzere eğitilmiştir. Bu süreç, toplamda 7 farklı makine öğrenimi algoritması için tekrar edilerek her biri

üzerinde ayrı ayrı değerlendirme yapılmıştır. Aynı algoritma hem ikili hem de çoklu sınıflandırmada kullanılmıştır.

Bir sonraki aşamada, veri setine Tablo 3'te belirtilen üç farklı özellik seçme algoritması uygulanarak 79 özellikten 16 özellik seçilmiştir. Seçilen bu 16 özellik ve 1 etiketten oluşan yeni veri setleri, önce zararlılık tespiti ardından zarar türü tespiti için, özellik seçimsiz yapılan eğitimler gibi, eğitilmiştir. Bu işlem, 7 farklı makine öğrenimi algoritması ile birleştirilerek toplamda 21 farklı model elde edilmiştir. Buna ek olarak, özellik seçimi uygulanmadan eğitilen 7 model ile birlikte toplamda 28 farklı model oluşturulmuştur.

Bu uygulamaların sonucunda Şekil 4'te gösterildiği gibi çalışan, iki aşamalı bir model ortaya çıkmıştır.

Bilinmeyen bir URL modele verildiğinde, öncelikle modele göre 79 özelliğin tamamı ya da özellik seçimiyle belirlenen 16 özellik çıkarılır. Ardından bu özellikler, Zararlılık Tespiti Modeli tarafından analiz edilir. URL zararsız olarak sınıflandırılırsa sistem herhangi bir ek işlem yapmadan sonuç verir. Eğer URL zararlı olarak sınıflandırılırsa, Zarar Türü Tespiti Modeli devreye girer ve URL'nin zarar türü belirlenerek nihai sınıflandırma sonucu oluşturulur. Bu çift aşamalı yaklaşım, sistemin hem zararlı URL'leri tespit etmesini hem de bunları spesifik türlere ayırmasını sağlayarak güvenlik analizi için daha detaylı bilgiler sunmasına olanak tanımaktadır.

3 Bulgular ve tartışma

Gerçekleştirilen deneylerde, ikili ve çoklu sınıflandırma problemlerinde en yüksek başarı, Tablo 4'te görüldüğü gibi, Rastgele Orman algoritmasıyla elde edilmiştir (Tablodaki değerler, kütüphane içerisindeki ***ağırlıklı ortalama*** ile hesaplanmıştır.). Özellikle, özellik seçimi yapılmadan tüm 79 özelliğin kullanıldığı senaryoda Rastgele Orman, her iki problem türünde de en iyi doğruluk (%97 ve %98), hassasiyet (%97 ve %98), duyarlılık (%97 ve %98) ve F1-Değeri (%97 ve %98) değerlerini sağlamıştır. Bu sonuç, tüm özelliklerin kullanıldığı kombinasyonun, zararlı URL tespiti ve sınıflandırmasında en etkili yöntem olduğunu açıkça ortaya koymaktadır.

Rastgele Orman'ın yüksek karmaşıklığa sahip veri kümelerinde güçlü bir sınıflandırıcı olarak performans gösterdiğini ve modelin daha fazla bilgiyle beslenmesinin, özellikle tüm özelliklerin kullanıldığı senaryoda, performansı önemli ölçüde artırdığını kanıtlamaktadır.

Özellik seçimi algoritmalarının uygulandığı senaryolarda da dikkate değer sonuçlar elde edilmiştir. Özellikle, Ekstra Ağaç ve SelectKBest algoritmalarının kullanıldığı kombinasyonlar hem ikili hem de çoklu sınıflandırma görevlerinde öne çıkmıştır. Ancak, bu senaryolarda elde edilen doğruluk değerleri, tüm 79 özelliğin kullanıldığı senaryonun altında kalmıştır. Bu durum, doğru özellik seçim algoritmalarının daha az özellik kullanarak işlem maliyeti düşürdüğünü, ancak performans açısından sınırlı bir avantaj sunduğunu göstermektedir.

Özellik seçimi genellikle, özellikle yüksek boyutlu verisetlerinde, sınıflandırma modelinin performansını artırma amacı taşır. Ancak, özellik seçiminde elde edilen bu etki, bazı durumlarda gözle görülür bir fark yaratmayabilir. Tablo 4'deki sonuçlar, bu durumu yansıtmaktadır. Bunun ana sebeplerinden biri, kullanılan özelliklerin zaten sınıflandırma modeline olan katkılarının büyük ölçüde optimize edilmiş ve yeterli bilgi içeriyor olmasıdır. Başka bir deyişle, veri setindeki özellikler, modelin sınıflandırma başarımını artıracak şekilde yeterince ayırt edici olabilir, dolayısıyla bazı özelliklerin seçilmesinin ek bir katkı sağlamadığı gözlemlenmiştir.

Diğer algoritmalarla karşılaştırıldığında, özellikle çoklu sınıflandırma probleminde CatBoost ve Karar Ağacı algoritmaları belirgin bir şekilde iyi sonuçlar elde etmiş olsalar da Rastgele Orman'ın sonuçlarıyla rekabet edememiştir. Bu durum, Rastgele Orman'ın daha başarılı sonuçlar üretebilme kapasitesinin bir kanıtıdır.

Şekil 5'te Rastgele Orman algoritması kullanılarak elde edilen özellik önem skorları karşılaştırmalı olarak **<Özellik, İkili Sınıflandırma Önem Sırası, Çoklu Sınıflandırma Önem Sırası>** şeklinde sunulmaktadır. Grafik, ikili sınıflandırma ve çoklu sınıflandırma senaryolarında, aynı özelliklerin tahmin sürecine katkı düzeylerini ve önem sıralarını göstermektedir. Özellik önemleri, algoritmanın kütüphane içerisinde sunduğu "feature_importances" metriği üzerinden hesaplanmıştır.

Grafikte görüldüğü üzere, bazı özellikler her iki modelde de benzer derecede kritik rol oynamaktadır. Örneğin IsPortEighty ve Delimiter_Count özellikleri her iki senaryoda da üst sıralarda yer almakta ve bu özelliklerin hem ikili hem de çoklu sınıflandırmada tutarlı belirleyiciler olduğunu göstermektedir. Buna karşın, NumberOfDotsInURL gibi bazı özelliklerde ise ciddi farklılıklar ortaya çıkmıştır; ikili sınıflandırmada üst sıralarda yer alan bu özellik, çoklu sınıflandırmada oldukça düşük bir önem skoruna sahip olmuştur.

Bu farklılığın temel nedeni sınıflandırma probleminin yapısından kaynaklanmaktadır. İkili sınıflandırma senaryosunda model yalnızca "zararlı" ve "zararsız" URL'leri ayırt etmeyi amaçlamaktadır. Bu nedenle bazı yapısal özellikler zararlı ile zararsız arasındaki temel farkı yakalamada güçlü bir ayırt edici rol üstlenmektedir. Çoklu sınıflandırmada ise modelin görevi farklı zarar türlerinin birbirinden ayrıştırılmasıdır. Bu durumda, ikili sınıflandırmada yüksek önem taşıyan bazı özellikler, zararlı sınıflar kendi aralarında benzer değerler taşıyorsa çoklu sınıflandırmada belirleyiciliğini kaybedebilmektedir.

Dolayısıyla, özelliklerin önem sıralamasındaki değişiklikler modelin öğrenme hedefinin farklılaşmasından kaynaklanmaktadır. İkili sınıflandırmada model zararlı ile zararsız arasındaki belirgin farkları yakalarken, çoklu sınıflandırmada zararlı türlerinin birbirinden ayrılması için daha ince ayrımlara ihtiyaç duymakta ve bu da farklı özelliklerin öne çıkmasına neden olmaktadır.

Bu analizler, Rastgele Orman modelinin URL yapısını nasıl değerlendirdiğini ve hangi özniteliklerin model tahminlerini yönlendirdiğini ortaya koymaktadır. Elde edilen sonuçlar, özellikle veri setindeki özniteliklerin dağılımına ve

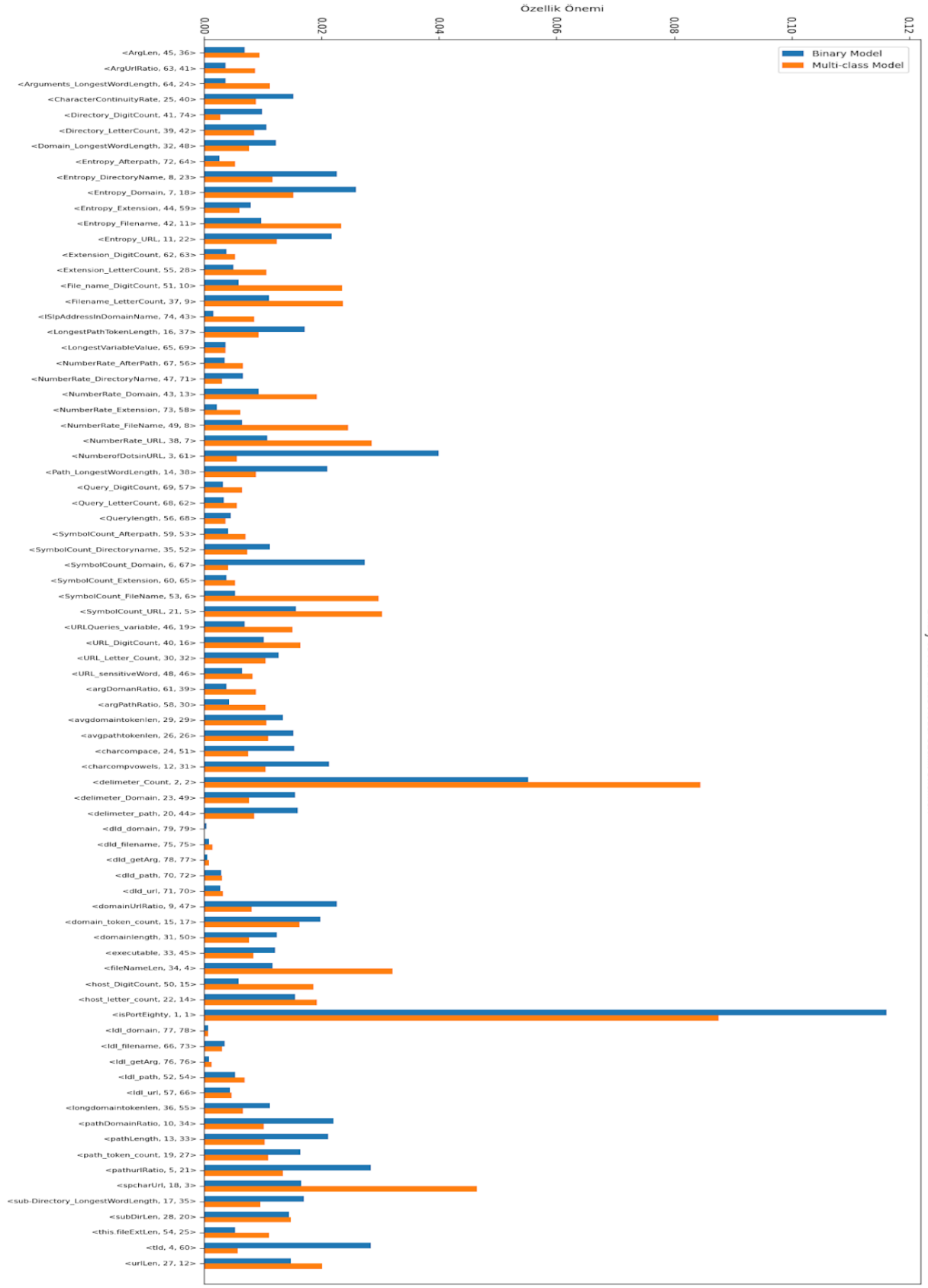
modelin karar mekanizmasına ışık tutarak, gelecekteki benzer çalışmalarda dikkate alınması gereken faktörleri gözler önüne sermektedir.

Tablo 5, bu çalışmanın literatürdeki diğer önemli araştırmalarla karşılaştırmasını sunmaktadır. Bu karşılaştırmada, her bir çalışmanın kullandığı algoritma, veri seti ve elde ettiği başarı oranı (doğruluk) gösterilmektedir. Tabloya göre, incelenen çalışmaların büyük bir kısmı, bu çalışmada kullanılan veri setlerine kıyasla daha küçük veya daha az çeşitli veri kümeleriyle gerçekleştirilmiştir. Örneğin, bazı çalışmalar spesifik veri setlerine (PhishTank, UCI) veya sınırlı sayıda URL'e (10.000) odaklanırken, bu çalışma üç farklı ve büyük veri setini (ISCX-URL-2016, Kaggle Malicious URLs Dataset, Kaggle URL Dataset) birleştirerek 1 milyondan fazla URL üzerinde analiz yapmıştır.

Algoritma performansı açısından bakıldığında, literatürdeki çeşitli çalışmalar Rastgele Orman, XGBoost, CatBoost ve AdaBoost gibi algoritmalarla yüksek başarı oranları elde etmiştir. Ancak, bu çalışmada tüm özelliklerin kullanıldığı senaryoda Rastgele Orman algoritmasının %97-%98 gibi yüksek bir başarı oranına ulaşması, algoritmanın büyük ölçekli ve çeşitli veri setleriyle bile son derece etkin olduğunu kanıtlamaktadır. Bu sonuç, Rastgele Orman'ın veri setinin karmaşıklığına ve büyüklüğüne rağmen genellenebilir ve güvenilir bir model ortaya koyduğunu göstermektedir. Dolayısıyla, bu çalışma, geniş veri kümeleri üzerinde iki aşamalı bir model kullanarak elde ettiği sonuçlarla, literatürdeki en başarılı yaklaşımlarla rekabet edebildiğini ve zararlı URL tespitinde kapsamlı bir çözüm sunduğunu vurgulamaktadır.

Tablo 4. Elde Edilen test başarımları

	Özellik Seçimi Olmadan		**Ekstra Ağaç**		**SelectKBest**		**SelectPercentile**	
Sınıflandırıcı Algoritmalar	İkili sınıflandırma	Çoklu Sınıflandırma	İkili Sınıflandırma	Çoklu Sınıflandırma	İkili Sınıflandırma	Çoklu Sınıflandırma	İkili Sınıflandırma	Çoklu Sınıflandırma
AdaBoost								
Doğruluk	0.88	0.87	0.87	0.82	0.86	0.82	0.86	0.82
Hassasiyet	0.88	0.87	0.87	0.82	0.86	0.82	0.86	0.82
Duyarlılık	0.88	0.87	0.87	0.82	0.86	0.82	0.86	0.82
F1-Değeri	0.88	0.87	0.85	0.82	0.86	0.81	0.86	0.81
CatBoost								
Doğruluk	**0.95**	**0.98**	0.93	0.97	0.92	0.96	0.92	0.96
Hassasiyet	**0.95**	**0.98**	0.93	0.97	0.92	0.96	0.92	0.96
Duyarlılık	**0.95**	**0.98**	0.93	0.97	0.92	0.96	0.92	0.96
F1-Değeri	**0.95**	**0.98**	0.93	0.97	0.92	0.96	0.92	0.96
Karar Ağacı								
Doğruluk	0.96	0.97	0.95	0.96	0.94	0.96	0.94	0.96
Hassasiyet	0.95	0.97	0.95	0.96	0.94	0.96	0.94	0.96
Duyarlılık	0.96	0.97	0.95	0.96	0.94	0.96	0.94	0.96
F1-Değeri	0.95	0.97	0.95	0.96	0.94	0.96	0.94	0.96
Gradient Arttırma								
Doğruluk	0.91	0.96	0.90	0.92	0.89	0.93	0.89	0.93
Hassasiyet	0.91	0.96	0.90	0.92	0.89	0.93	0.89	0.93
Duyarlılık	0.91	0.96	0.90	0.92	0.89	0.93	0.89	0.93
F1-Değeri	0.91	0.95	0.90	0.92	0.88	0.92	0.88	0.92
Naive Bayes								
Doğruluk	0.79	0.81	0.83	0.80	0.82	0.77	0.82	0.77
Hassasiyet	0.78	0.82	0.82	0.84	0.81	0.86	0.81	0.86
Duyarlılık	0.79	0.81	0.83	0.80	0.82	0.77	0.82	0.77
F1-Değeri	0.76	0.81	0.83	0.80	0.81	0.79	0.81	0.79
Rastgele Orman								
Doğruluk	**0.97**	**0.98**	0.96	0.98	0.95	0.98	0.95	0.98
Hassasiyet	**0.97**	**0.98**	0.96	0.98	0.95	0.98	0.95	0.98
Duyarlılık	**0.97**	**0.98**	0.96	0.98	0.95	0.98	0.95	0.98
F1-Değeri	**0.97**	**0.98**	0.96	0.98	0.95	0.98	0.95	0.98
XGBoost								
Doğruluk	0.91	0.94	0.90	0.91	0.88	0.91	0.88	0.91
Hassasiyet	0.91	0.95	0.90	0.91	0.89	0.91	0.89	0.91
Duyarlılık	0.91	0.94	0.90	0.91	0.88	0.91	0.88	0.91
F1-Değeri	0.90	0.94	0.89	0.91	0.88	0.91	0.88	0.91



Şekil 5. İkili ve çoklu sınıflandırma modellerine ait özellik ve önem sıralaması

Tablo 5. Önceki çalışmalar ile kıyaslama

Atıf	Algoritma	Veri Seti	Başarı Oranı
Bu çalışma	***Rastgele Orman***	***ISCX-URL-2016, Kaggle Malicious URLs Dataset, Kaggle URL Dataset***	***0.98***
Choi ve ark. [6]	SVM ve ML-kNN	Yahoo, Web spam, PhishTank, DNS-BH	0.93
Lin ve ark. [8]	Güven Ağırlıklı ve PA-I algoritmaları	Belirtilmeyen bir güvenlik şirketi	0.90
Chu ve ark. [7]	SVM	Taobao-phishin URLs ve Yahoo	0.99
Joshi ve ark. [9]	Rastgele Orman	Openphish, Alexa, FireEye	0.92
Şahingöz ve ark. [11]	Rastgele Orman	PhishTank ve Yandex	0.98
Powell ve ark. [10]	Karar Ağacı ve Rastgele Orman	NSL-KDD, ISCX-URL-2016, CICIDS-2017	1.00
Singhal ve ark. [13]	Gradyan Artırma	PhishTank, Malware Domain List, Majestic	0.96
Wang ve ark. [12]	Native Bayes	ISCX-URL-2016	0.90
Arslan [14]	Lojistik Regresyon	ISCX-URL-2016	0.99
Singh ve ark. [15]	XGBoost	ISCX-URL-2016	0.98
Sadaf [16]	XGBoost	UCI Veri Seti	0.97
Sheikhi ve Kostakos [17]	PSO-XGBoost	ISCX-URL-2016	0.98
Omolara ve Alawida [18]	AdaBoost	ISCX-URL2016	0.99
Kustiawan ve Ghauth [19]	CatBoost	PhishOFE	0.99
Alavala ve ark. [20]	CatBoost	Derleme Veri Seti (10.000 URL)	0.92

4 Sonuçlar

Yapılan çalışmada zararlı URL tespiti ve zarar türlerine göre sınıflandırma problemine yönelik farklı makine öğrenimi modellerini karşılaştırarak en iyi performans gösteren modeli belirlemeyi amaçlamıştır. 1 milyondan fazla veri ile yapılan deneyler, veri setlerinin çeşitliliği ve özellik seçimi yöntemlerinin model başarısına etkisini ortaya koymuştur. Deneysel sonuçlar, bazı durumlarda özellik seçiminin model başarısını artırdığını, ancak her senaryoda mutlak bir iyileşme sağlamadığını göstermektedir. Özellikle, bazı özellik seçme algoritmalarının daha az sayıda özellik kullanarak işlem maliyetini düşürdüğü, ancak doğruluk açısından sınırlı veya değişken etkiler sunduğu gözlemlenmiştir. Bu durum, özellik seçim stratejilerinin veri setinin yapısına ve kullanılan makine öğrenimi algoritmasına bağlı olarak farklı sonuçlar verebileceğini göstermektedir. Rastgele Orman algoritması hem tüm özelliklerin kullanıldığı senaryoda hem de belirli özellik seçimi uygulamalarıyla en iyi sonuçları vermiştir. Bununla birlikte, bazı durumlarda CatBoost ve Karar Ağacı gibi algoritmaların da çoklu sınıflandırmalarda görevlerinde gayet yüksek bir başarı oranına sahip olduğu görülmüştür. Tüm bu bulgular, zararlı URL tespiti süreçlerinde kapsamlı bir model karşılaştırması yapmanın ve farklı özellik seçim stratejilerinin etkilerini detaylı bir şekilde analiz etmenin önemini ortaya koymaktadır.

Gelecekte, zararlı URL tespitinde yalnızca URL özelliklerinin değil, aynı zamanda sayfa içeriğinin de değerlendirilmesi önerilmektedir. Sayfa içeriğine yönelik analizlerin algoritmalara entegre edilmesi, özellikle metin tabanlı dolandırıcılık ve zararlı içerik tespitinde önemli bir avantaj sağlayacaktır. Bu kapsamda doğal dil işleme (NLP) tekniklerinin kullanılması, sayfa içeriklerindeki semantik ilişkilerin anlaşılmasına ve özelliklerin daha anlamlı hale gelmesine olanak tanıyabilir. Bu tür bir yaklaşım, sistemin zararlı URL tespiti ve zarar türü sınıflandırmasında daha yüksek doğruluk oranları elde etmesini ve literatüre daha kapsamlı katkılar sunmasını sağlayacaktır.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %17

Kaynaklar

[1] A. Sertçelik, Siber Olaylar Ekseninde Siber Güvenliği Anlamak. Medeniyet Araştırmaları Dergisi, 2 (3), 25-42, 2015.

[2] RFC 1738, Uniform resource locators (URL), T. Berners-Lee, L. Masinter, & M. McCahill, (1994). https://doi.org/10.17487/rfc1738.

[3] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, PhishNet: Predictive Blacklisting to Detect Phishing Attacks. 2010 Proceedings IEEE INFOCOM, pp. 1-5, San Diego, CA, USA, 2010. https://doi.org/10.1109/INFCOM.2010.5462216

[4] Y. Fukushima, Y. Hori and K. Sakurai, Proactive Blacklisting for Malicious Web Sites by Reputation Evaluation Based on Domain and IP Address Registration. 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 352-361, Changsha, China, 2011. https://doi.org/10.1109/TrustCom.2011.46

[5] A. K. Jain and B. B. Gupta, A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP Journal on Information Security, 2016, 1-11, 2016. https://doi.org/10.1186/s13635-016-0034-3

[6] H. Choi, B. B. Zhu and H. Lee, Detecting Malicious Web Links and Identifying Their Attack Types. 2nd USENIX Conference on Web Application Development (WebApps 11), Berkeley, USA, 2011.

[7] W. Chu, B. B. Zhu, F. Xue, X. Guan and Z. Cai, Protect Sensitive Sites from Phishing Attacks Using Features Extractable from Inaccessible Phishing URLS. 2013 IEEE International Conference on Communications (ICC), pp. 1990-1994, Budapest, Hungary, 2013. https://doi.org/10.1109/ICC.2013.6654816

[8] M. S. Lin, C. Y. Chiu, Y. J. Lee and H. K. Pao, Malicious URL Filtering – A Big Data Application. 2013 IEEE international conference on big data, pp. 589-596, Santa Clara, CA, USA, 2013. https://doi.org/10.1109/BigData.2013.6691627

[9] A. Joshi, L. Lloyd, P. Westin and S. Seethapathy, Using lexical features for malicious URL detection--a machine learning approach. arXiv preprint arXiv:1910.06277, 2019.

[10] A. Powell, D. Bates, C. Van Wyk and A. D. de Abreu, A cross-comparison of feature selection algorithms on multiple cyber security data-sets. Proceedings of the FAIR 2019 Workshop, pp. 196-207, Cape Town, South Africa, 2019.

[11] O. K. Sahingoz, E. Buber, O. Demir and B. Diri, Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357, 2019. https://doi.org/10.1016/j.eswa.2018.09.029.

[12] S. Wang, Y. Wang and M. Tang, Auto Malicious Websites Classification Based on Naive Bayes Classifier. 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), pp. 443-447, Dalian, China, 2020. https://doi.org/10.1109/ICISCAE51034.2020.9236912

[13] S. Singhal, U. Chawla and R. Shorey, Machine Learning & Concept Drift based Approach for Malicious Website Detection. 2020 12th International Conference on Communication Systems & Networks (COMSNETS), pp. 582-585, Bangalore, India, 2020. https://doi.org/10.1109/COMSNETS48256.2020.9027485

[14] R. S. Arslan, Kötücül Web Sayfalarının Tespitinde Doc2Vec Modeli ve Makine Öğrenmesi Yaklaşımı. Avrupa Bilim ve Teknoloji Dergisi, (27), 792-801, 2021. https://doi.org/10.31590/ejosat.981450.

[15] S. S. K. Singh, V. Menon, S. A. Sajidha, V. M. Nisha, A. Sheik Abdullah, M. Nivedita and A. Mairaj, Meta Learning for Enhanced Web Security Against Malicious URLs. Research Square, 2023. https://doi.org/10.21203/rs.3.rs-3626868/v1

[16] K. Sadaf, Phishing Website Detection using XGBoost and Catboost Classifiers. 2023 International Conference on Smart Computing and Application (ICSCA), pp. 1-6, Bali, Indonesia, 2023. https://doi.org/10.1109/ICSCA57840.2023.10087829

[17] S. Sheikhi and P. Kostakos, Safeguarding cyberspace: Enhancing malicious website detection with PSO-optimized XGBoost and firefly-based feature selection. Computers & Security, 142, 103885, 2024. https://doi.org/10.1016/j.cose.2024.103885.

[18] A. E. Omolara and M. Alawida, DaE2: Unmasking malicious URLs by leveraging diverse and efficient ensemble machine learning for online security. Computers & Security, 148, 104170, 2025. https://doi.org/10.1016/j.cose.2024.104170

[19] Y. A. Kustiawan and K. I. Ghauth, Evaluating the Impact of Feature Engineering in Phishing URL Detection: A Comparative Study of URL, HTML, and Derived Features. IEEE Access, 13, 126756-126768, 2024. https://doi.org/10.1109/ACCESS.2025.3579223

[20] H. R. Alavala, S. Singh, P. Joshi and S. Basavaraju, Enhancing Malicious URL Detection with Advanced Machine Learning Techniques. 2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT), pp. 151-156, Bengaluru, India, 2025. https://doi.org/10.1109/CE2CT64011.2025.10939290

[21] Canadian Institute for Cybersecurity. ISCX-URL-2016 dataset. https://www.unb.ca/cic/datasets/url-2016.html, Accessed 28 December 2024

[22] Siddhartha, M. Malicious URLs dataset. https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset, Accessed 28 December 2024

[23] TeseRact. URL dataset. https://www.kaggle.com/datasets/teseract/urldataset, Accessed 28 December 2024

Ekler

Projeye ait tüm kodlar ve kullanım yönergeleri GitHub platformunda paylaşılmıştır. Algoritmaların koşulduğu depo:https://github.com/SahinMuhammetAbdullah/malicious-url-detection. Veri kümelerinin bulunduğu depo: https://github.com/SahinMuhammetAbdullah/urldataset