Teknoloji Fakültesi

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# Mobil Platformlarda Phishing Saldırılarına Karşı Federe Öğrenme Tabanlı Etkin Savunma Sistemi Geliştirilmesi

**Bitirme Projesi 2. Ara Raporu**

Bilgisayar Mühendisliği Bölümü

**DANIŞMAN**

Doç. Dr. Kazım YILDIZ

İSTANBUL, 2025

# MARMARA ÜNİVERSİTESİ
# TEKNOLOJİ FAKÜLTESİ
# BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Eren DOĞAN tarafından "**Mobil Platformlarda Phishing Saldırılarına Karşı Federe Öğrenme Tabanlı Etkin Savunma Sistemi Geliştirilmesi**" başlıklı proje çalışması, xxx tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

| | | |
|---|---|---|
| Dr. Öğr. Üyesi xxx xxx<br>Marmara Üniversitesi | **(Danışman)** | (İMZA)…….. |
| Prof. Dr. Xxx xxx<br>Marmara Üniversitesi | (Üye) | (İMZA)…….. |
| Prof. Dr. Xxx xxx<br>Marmara Üniversitesi | (Üye) | (İMZA)…….. |

# ÖNSÖZ

Proje çalışmam süresince karşılaştığım her türlü sorunda sabırla yardım ve bilgilerini esirgemeyen, tüm desteğini yanımda hissettiğim değerli hocam Arş. Gör. Büşra Büyüktanır'a en içten teşekkürlerimi sunarım. Kendisinin yol gösterici tavsiyeleri ve sürekli motivasyon sağlaması, bu çalışmanın başarıya ulaşmasında büyük bir paya sahiptir.

Ayrıca, proje fikrinin oluşmasında sağladığı katkılardan dolayı Doç. Dr. Kazım Yıldız'a da teşekkür ederim. Onun yönlendirici önerileri sayesinde çalışmanın kapsamı ve uygulama süreci daha planlı bir şekilde yürütülmüştür.

# İçindekiler Tablosu

# ÖZET

**Mobil Platformlarda Phishing Saldırılarına Karşı Federe Öğrenme Tabanlı Etkin Savunma Sistemi Geliştirilmesi**

Bu çalışma, mobil cihazlarda ziyaret edilen web sitelerinin kaynak kodunu analiz ederek gerçek zamanlı phishing (oltalama) tespiti yapabilen, derin öğrenme ve federated learning (federe öğrenme) tabanlı yenilikçi bir yapay zekâ modelinin tasarım ve uygulamasını sunmaktadır. Literatürde phishing tespiti için çoğunlukla kara liste ve klasik makine öğrenmesi tabanlı yöntemler kullanılmakta; ancak bu yaklaşımlar yeni ve karmaşık saldırı tipleri karşısında yetersiz kalmaktadır. Özellikle mobil platformlarda, kullanıcı verisini ihlal etmeden, dinamik ve gerçek zamanlı phishing tespiti yapabilen derin öğrenme tabanlı sistemlere olan gereksinim giderek artmaktadır.

Geliştirilen model, web sayfalarının kaynak kodundan (HTML, DOM, JavaScript) öznitelik çıkarımı yapabilen ve kendi kendini optimize edebilen Convolutional Neural Network (CNN) mimarisi ile tasarlanmıştır. Modelde federated learning yaklaşımı benimsenmiş, böylece kullanıcı verileri cihazdan çıkmadan model yerel olarak güncellenebilmiştir. Eğitim aşamasında PhishTank, OpenPhish ve Alexa Top Sites gibi çeşitli güncel veri kaynaklarından toplanan yüz binlerce phishing ve yasal web sitesi örneği kullanılmıştır. Nihai sistem, Android tabanlı açık kaynak Lightning web tarayıcısına entegre edilerek, ziyaret edilen sitelerin kaynak kodunu anlık olarak analiz edip potansiyel tehditleri gerçek zamanlı kullanıcıya bildirecek şekilde uygulanmıştır.

Test sonuçlarında model %86'nın üzerinde doğruluk, %90 precision ve %80'in üzerinde recall gibi yüksek başarı oranlarına ulaşmıştır. Federe öğrenme sayesinde, sistem hem bireysel kullanıcıya özgü dinamik koruma sunmakta, hem de yeni ortaya çıkan phishing tekniklerine hızlı şekilde adapte olabilmektedir. Sonuçlar, mobil cihazlarda kaynak koduna dayalı ve federated learning destekli bir yapay zekâ yaklaşımının, phishing tespitinde hızlı, hafif, etkili ve kullanıcı gizliliğine duyarlı bir çözüm sunduğunu ortaya koymaktadır. Bu çalışma, TÜBİTAK 2209 desteği ile yürütülmüş olup, mobil platformlarda gerçek zamanlı kaynak kodu analizi ile phishing tespiti ve federated learning entegrasyonunun birlikte uygulandığı literatürdeki ilk örneklerden biri olarak özgün bir katkı sunmaktadır.

**Anahtar Kelimeler:** Oltalama (Phishing), mobil güvenlik, derin öğrenme, federated learning, kaynak kodu analizi, gerçek zamanlı tespit, yapay zekâ, Android.

**Mayıs, 2025** **Eren DOĞAN**

# ABSTRACT

**Developing a Federated Learning Based Effective Defense System Against Phishing Attacks on Mobile Platforms**

This study presents the design and implementation of an innovative artificial intelligence model that enables real-time phishing detection on mobile devices by analyzing the source code of visited websites, based on deep learning and federated learning. In the current literature, most phishing detection methods rely on blacklists or conventional machine learning techniques, yet these approaches often fail to detect novel and sophisticated attack types. There is an increasing need, especially on mobile platforms, for deep learning-based systems capable of dynamic, real-time phishing detection without compromising user privacy.

The proposed model utilizes a Convolutional Neural Network (CNN) architecture to automatically extract features from web page source code (HTML, DOM, JavaScript) and employs a federated learning framework, ensuring that user data remains on the device while enabling continuous local model updates. For model training, hundreds of thousands of phishing and legitimate website samples were compiled from diverse and up-to-date sources such as PhishTank, OpenPhish, and Alexa Top Sites. The final system was integrated into the open-source Lightning web browser on Android, allowing for instant analysis of the visited site's source code and real-time threat notifications to users.

Evaluation results demonstrate that the model achieves high performance, exceeding 86% accuracy, 90% precision, and 80% recall. Through federated learning, the system delivers adaptive, personalized protection for users while rapidly adapting to new phishing techniques. Overall, the findings indicate that a code-based, federated learning-enabled deep learning approach provides an effective, lightweight, and privacy-preserving solution for phishing detection on mobile devices. Supported by TÜBİTAK 2209, this project offers an original contribution as one of the first examples in the literature to combine real-time source code analysis and federated learning for mobile anti-phishing solutions.

**Keywords:** Phishing, mobile security, deep learning, federated learning, source code analysis, real-time detection, artificial intelligence, Android.

**March, 2025** **Eren DOĞAN**

# KISALTMALAR

- **CNN (Convolutional Neural Network)**: Evrişimli Sinir Ağı
- **CSS (Cascading Style Sheets)**: Basamaklı Stil Şablonları
- **GDPR (General Data Protection Regulation)**: Genel Veri Koruma Tüzüğü (AB)
- **HTML (HyperText Markup Language)**: Köprü Metni Biçimlendirme Dili
- **KVKK (Kişisel Verilerin Korunması Kanunu)**: Türkiye'deki Veri Koruma Yasası
- **SMS (Short Message Service)**: Kısa Mesaj Servisi
- **SSL (Secure Sockets Layer)**: Güvenli Yuva Katmanı (Genelde TLS/SSL şeklinde geçer)
- **TLS (Transport Layer Security)**: Taşıma Katmanı Güvenliği
- **TÜİK (Türkiye İstatistik Kurumu)**
- **URL (Uniform Resource Locator)**: Tekdüzen Kaynak Bulucu

# ŞEKİL LİSTESİ

# Tablo Listesi

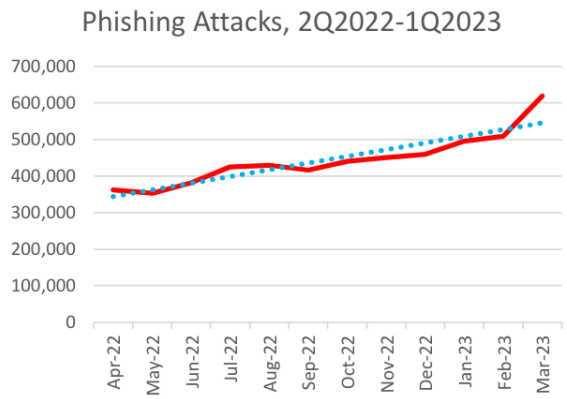
# 1. GİRİŞ

Bilişim teknolojilerinin modern yaşamda entegrasyonu giderek daha karmaşık ve derin bir yapı kazanmaktadır. Türkiye İstatistik Kurumu'nun (TÜİK) 2023 tarihli Hanehalkı Bilişim Teknolojileri Kullanım Araştırması verilerine göre, Türkiye'deki hanelerin %95,5'i internet erişimine sahiptir; bu oran, 2018 yılında %83,8 olarak ölçülmüştür. Aynı çalışma, bireylerin internet kullanım oranının son beş yılda %14,2 oranında artarak %87,1'e yükseldiğini göstermektedir. Bu bulgu, artan internet erişimiyle birlikte daha çok bireyin dijital dünyaya entegre olduğunu ve internet kullanımının yaygınlaştığını göstermektedir (TÜİK, 2023).

Artan internet kullanımı, kişisel bilgi güvenliği ihlalleri, zararlı yazılım saldırıları ve benzeri çeşitli siber tehditlerin de paralel olarak artmasına yol açmıştır (Ömer ve ark., 2023). Bu tehditler, hem bireysel kullanıcılar hem de kurumlar ve devletler için ciddi riskler oluşturmaktadır (Bıçakcı, Ergun ve Çelikpala, 2016). Küresel ölçekte, kişisel bilgi güvenliğine yönelik saldırıların sayısı sürekli artmakta ve çok daha ciddi bir tehdit halini almaktadır.

Kişisel bilgi güvenliğinin ihlalinde en çok kullanılan yöntemlerden biri phishing (oltalama) saldırılarıdır. Phishing saldırılarının yaygınlığı, dünya genelinde 45 ülkede Anti-Phishing Working Group (APWG, 2016) tarafından yapılan bir çalışmada ayrıntılı olarak incelenmiştir. Bu çalışmanın bulgularına göre, phishing saldırılarının en yaygın olduğu ülkeler sıralamasında ilk sırada %47,09 ile Çin yer almaktadır. Bu ülkeyi %42,88 ile Türkiye ve %38,98 ile Tayvan takip etmektedir.

Phishing saldırıların zaman içinde nasıl arttığını ortaya koyan güncel verilere göre (APWG, 2023), bu saldırılar 2022'nin ikinci çeyreğinden 2023'ün ilk çeyreğine kadar sürekli bir artış göstermiştir. Şekil 1'de sunulan verilere göre, 2022'nin Nisan ayında yaklaşık 400.000 olan saldırı sayısı, 2023'te 600.000'i aşmış ve yeni bir zirveye ulaşmıştır. Bu artış, phishing saldırılarının giderek yaygınlaştığını ve daha büyük bir tehdit oluşturduğunu göstermektedir.




Şekil 1.1 Oltalama Saldırıları İstatistiği (APWG, 2023)

Phishing saldırıları, kişisel ve gizli verilere yetkisiz erişim elde etmek ve bu verileri kötü niyetli amaçlarla kullanmak için bilişim teknolojilerinin sunduğu fırsatları değerlendiren bir dolandırıcılık yöntemidir (Ünver ve Mirzaoğlu, 2011). Bu saldırılar, genellikle sosyal mühendislik teknikleri kullanılarak bireylerin güvenini kötüye kullanmaya dayanmaktadır (Hekim, 2015). Phishing girişimleri genellikle e-posta, SMS ya da sosyal medya kanalları üzerinden zararlı bağlantılar gönderilmesi yoluyla yapılmaktadır (Buber, Diri ve Şahingöz, 2017; Jakobsson ve Myers, 2007). Birleşik Krallık Kamu Sektörü Bilgileri tarafından 2023 yılında yapılan Cyber Security Breaches Survey, işletmelerin ve STK'ların en yaygın maruz kaldığı siber tehdit türünün %79 oranıyla phishing olduğunu ortaya koymuştur. Bu çalışma, önceki yıllarda elde edilen bulgularla da uyumludur (Birleşik Krallık Kamu Sektörü Bilgileri, 2021, 2022, 2023). Bu bulgular, kişisel bilgi güvenliğinin ihlalinde phishing saldırılarının merkezi bir rol oynadığını göstermektedir.

Bu siber tehditler, belirli birey ya da gruplardan öte tüm toplumu etkileyen büyük bir sorun haline gelmiştir. Finansal kayıplar, kişisel verilerin çalınması, iş süreçlerinin kesintiye uğramı ve geniş çaplı güvenlik ihlalleri gibi çok sayıda olumsuz sonuca neden olabilmektedir (Wassermann, Meyer, Goutal ve Riquet, 2023). Finansal kayıplar, phishing saldırıları sonucunda kullanıcıların banka hesap bilgilerinin, kredi kartı numaralarının ve diğer finansal bilgilerin ele geçirilmesi sonucunda oluşmaktadır (Sharif ve Mohammed, 2022).

Phishing saldırılarında kullanılan çeşitli aldatıcı teknikler ile, genellikle e-posta, mesajlaşma uygulamaları veya sosyal medya üzerinden kullanıcılar sahte web sitelerine yönlendirilmekte ve bu yolla kişisel ve hassas bilgiler ele geçirilmektedir. Sonuç olarak; finansal kayıplar, kişisel bilgilerin çalınması, iş süreçlerinde aksamalar ve güvenlik

ihlalleri gibi ciddi sonuçlar ortaya çıkmaktadır (Goel, Williams ve Dincelli, 2017).

Kişisel verilerin çalınması, bu siber tehditlerin önemli bir sonucu olarak ortaya çıkmıştır. Kullanıcıların isim, adres, telefon numarası ve sağlık kayıtları gibi hassas bilgilerinin phishing saldırıları yoluyla ele geçirilmesi, bireylerin mahremiyetini ihlal etmekte ve kimlik hırsızlığı gibi daha büyük sorunlara yol açabilmektedir (Kamiya ve diğerleri, 2019).

Kurumsal düzeyde ise, phishing saldırıları iş süreçlerinde aksamalara yol açabilmektedir. Büyük şirketlerde çalışan bireylerin phishing saldırılarına maruz kalması, kurum içi ağların güvenliğini tehdit etmekte, veri ihlallerine ve iş sürekliliğinde kesintilere neden olabilmektedir. Bu durum, şirketlerin hem itibarına hem de finansal durumuna zarar verebilmektedir (Kamiya ve diğerleri, 2019).

Mobil internetin ve akıllı telefon kullanımının yaygınlaşması, mobil cihazları oltalama saldırıları için popüler bir hedef haline getirmiştir. Mobil kullanıcılar, masaüstü kullanıcılara kıyasla daha az dikkatli davranabilmektedir ve bu nedenle mobil cihazlardaki oltalama saldırıları daha yüksek bir başarı oranına sahip olabilmektedir (Wu, Du ve Wu, 2015). Mobil cihazların ekran boyutlarının sınırlı olması, saldırganların URL'leri gizlemesine ve sahte siteleri maskelemesine olanak tanımakta, bu da kullanıcıların bu tür saldırıları fark etmelerini zorlaştırmaktadır (Canova ve diğerleri, 2015). Ayrıca, mobil tarayıcılar genellikle azalan güvenlik özelliklerine sahiptir, bu da güvenlik açıklarını daha kritik bir hale getirmektedir (Varshney ve diğerleri, 2016).

Araştırmacılar, kullanıcıları phishing saldırılarına karşı korumak için çeşitli stratejiler geliştirmiştir. Bu stratejiler genellikle eğitsel ve teknik çözümler olarak ikiye ayrılmaktadır. Eğitsel teknikler, internet kullanıcılarını eğitim programları, atölye çalışmaları ve farkındalık kampanyaları ile phishing e-postalarını ve web sitelerini tanımaları konusunda bilgilendirmeyi amaçlamaktadır (Aleroud ve Zhou, 2017). Teknik çözümler ise genellikle otomatik sınıflandırma ve kara listeleme gibi tekniklerle phishing tespitini hızlandırmayı ve insan müdahalesini en aza indirmeyi hedeflemektedir (Jain ve Gupta, 2022).

Web sitesi görünümü üzerinden saldırının tespiti, birçok uzman için bile zor bir işlem olabilmektedir çünkü saldırganlar, farklı teknikler kullanarak bilgili kullanıcıları dahi aldatabilmektedir. Bu nedenle, phishing saldırılarının tespitinde yazılım tabanlı çözümlerin kullanılması büyük önem taşımaktadır (Buber, Diri ve Şahingöz, 2017).

Phishing tehditlerini azaltmak amacıyla, araştırmacılar URL'lerin, ana bilgisayar bilgilerinin ve web sitesi içeriklerinin el ile oluşturulmuş özelliklerini kullanan çeşitli liste tabanlı ve makine öğrenimi tabanlı yöntemlerin doğruluğunu artırmak için çalışmaktadır (Dou ve arkadaşları, 2017). Bununla birlikte, phishing tespiti, phishing'in sürekli evrimi nedeniyle kesin bir çözüm sunmayan bir silahlanma yarışı olarak kalmaktadır. Bu durum, phishing özelliklerini otomatik olarak çıkarabilen ve phishing web sitelerini doğru bir şekilde tespit edebilen tekniklerin geliştirilmesini sürekli olarak zorunlu kılmaktadır. Yapay zekâ ve makine öğrenmesi gibi gelişmiş teknolojiler, saldırıların otomatik olarak tespit edilmesi ve önlenmesi için giderek daha fazla önem kazanmaktadır. Bu teknolojiler, phishing saldırılarının karmaşıklığını ve sürekli değişen doğasını anlamak ve etkili savunma mekanizmaları geliştirmek için kritik öneme sahiptir (Brad, 2021).

Mobil uygulama mağazalarında sunulan benzer güvenlik uygulamalarının büyük çoğunluğunun, statik kara listeleme yöntemlerine dayandığı gözlemlenmektedir. Bu yaklaşım, önceden tanımlanmış zararlı URL'leri içeren sabit listeleri kullanarak, kullanıcıların ziyaret etmeye çalıştığı siteleri bu listelerle karşılaştırmaktadır. Ancak bu listeler sıklıkla güncellenmediği için yeni tehditlere karşı yetersiz kalmaktadır. Örneğin, bir kara liste oluşturulduktan kısa bir süre sonra, siber saldırganlar tarafından yeni alan adları devreye sokulabilmekte veya mevcut zararlı siteler, algılanmamak için adreslerini değiştirebilmektedir. Bu tür değişiklikler, statik listelerin hızla eskimesine ve dolayısıyla yeni tehditleri tespit etme kapasitesinin azalmasına neden olmaktadır. Ayrıca, statik listeler yalnızca bilinen tehditleri barındırdığı için, henüz tanımlanmamış veya daha önce rapor edilmemiş zararlı siteler bu kontrollerden sıyrılabilmektedir. Bu durum, kullanıcıları evrim geçiren siber tehditlere karşı savunmasız bırakmaktadır.

## 1.1. Proje Çalışmasının Amacı ve Önemi

Bu araştırma önerisinin temel amacı, internet kullanıcılarını artan phishing (oltalama) saldırılarına karşı korumak için mobil platformlarda gerçek zamanlı çalışan, kullanıcı gizliliğini ihlal etmeden sürekli güncellenebilen ve kullanıcının tarama alışkanlıklarına adapte olarak değişen saldırı tekniklerine uyum sağlayabilen, federe öğrenme tabanlı derin öğrenme modeli geliştirmektir. Geliştirilecek model, sayfa kaynak kodlarından öznitelik çıkararak phishing web sitelerini etkin ve gerçek zamanlı bir şekilde tespit edecek şekilde tasarlanacaktır. Böylece, kullanıcıların kişisel bilgilerinin korunması, internet güvenliğinin

artırılması ve phishing saldırıları sonucu oluşabilecek mali ve operasyonel zararların minimize edilmesi sağlanacaktır.

Araştırmanın nihai hedefi, geliştirilen modelin yalnızca teorik düzeyde kalmayıp, toplumun genel faydasına sunulabilecek, gerçek zamanlı ve kullanıcı gizliliğine öncelik veren bir güvenlik aracı olarak kullanılmasını sağlamaktır. Bu amaçla, modelin entegrasyonuyla özel bir anti-phishing özellikli web tarayıcısı geliştirilecektir. Bu tarayıcı, kullanıcıların güvenli ve kesintisiz bir internet deneyimi yaşamalarını sağlayacak, ziyaret edilen sitelerin güvenilirliğini anlık olarak değerlendirecek ve olası tehditleri tespit ettiğinde kullanıcıya uyarılar gönderecektir.

Bu araştırma kapsamında geliştirilecek yapay zekâ modeli, kullanıcı verilerini merkezi bir sunucuya aktarmadan doğrudan cihaz üzerinde çalışarak phishing tespiti yapabilen, gizlilik odaklı ve gerçek zamanlı bir güvenlik çözümü olarak tasarlanmaktadır. Bu model, mobil cihazlarda kullanıcıların tarama alışkanlıklarına adapte olacak şekilde çalışarak sürekli erişilebilir ve kullanımı kolay bir güvenlik çözümü sunacaktır. Kullanıcı verilerini yerel olarak işleyerek gizliliği garanti altına alırken, phishing saldırılarını tespit etmede yüksek doğruluk oranları sağlamayı hedeflemektedir.

Sonuç olarak, bu proje yalnızca teorik bir model sunmakla sınırlı kalmayıp, kullanıcılara doğrudan ulaşabilen bir güvenlik aracı ortaya koymayı hedeflemektedir. Modelin dinamik ve sürekli gelişen yapısı, phishing saldırılarına karşı proaktif bir savunma mekanizması oluşturacaktır. Böylelikle mevcut anti-phishing çözümlerine kıyasla hem etkinlik hem de gizlilik açısından daha üst düzeyde bir koruma sağlayacak olan bu entegre güvenlik sistemi, bireysel kullanıcıların güvenliğini artırmanın ötesinde, internet güvenliği alanında toplumsal fayda sağlayacak ve bu bağlamda yeni nesil bir güvenlik yaklaşımı olarak kendini gösterecektir.

## 1.2 Araştırma Soruları ve Hipotezler

Bu çalışmada, söz konusu siber tehditlere yönelik teknik çözümler geliştirmek amacıyla aşağıdaki araştırma soruları ve hipotezler belirlenmiştir:

**Araştırma Sorusu 1:** Mobil platformlar için yapay zekâ tabanlı phishing sayfa kaynak kodu tespit modelinin geliştirilmesi sırasında karşılaşılan zorluklar nelerdir ve bu zorlukların üstesinden gelmek için hangi stratejiler uygulanabilir?

**Hipotez 1:** Mobil platformlarda sınırlı işlem gücü ve enerji kaynakları gibi donanımsal kısıtlamalar, yapay zekâ tabanlı phishing tespit modelinin geliştirilmesinde temel zorluklardır. Bu zorlukları aşmak için modelin optimize edilmiş ve hafif bir yapıda tasarlanması ile federe öğrenme yöntemlerinin kullanılması etkili bir strateji olacaktır.

**Araştırma Sorusu 2:** Mobil platformlarda phishing sitelerini gerçek zamanlı olarak analiz eden yapay zekâ destekli bir modelin phishing sitelerini tespit etme başarı oranı nedir ve bu başarıyı etkileyen faktörler nelerdir?

**Hipotez 2:** Yapay zekâ destekli modelin phishing sitelerini tespit etme başarı oranı %85'in üzerinde olacaktır. Bu başarıyı etkileyen faktörler arasında modelin eğitim verisinin kalitesi, güncelliği ve modelin kullanıcı alışkanlıklarına adapte olma yeteneği yer almaktadır.

**Araştırma Sorusu 3:** Kullanıcıların kendilerine gösterilen phishing sitelerini tespit etme sürecindeki karar verme performansları ile yapay zekâ destekli modelin analiz sonuçları arasındaki doğruluk ve performans farklılıkları nelerdir?

**Hipotez 3:** Yapay zekâ destekli modelin phishing sitelerini tespit etme doğruluğu, kullanıcıların bireysel karar verme performansından daha yüksek olacaktır. Kullanıcılar, sosyal mühendislik teknikleri nedeniyle yanıltılabilecek iken, model nesnel analiz yeteneği sayesinde daha yüksek doğruluk oranına ulaşabilecektir.

**Araştırma Sorusu 4:** Federe öğrenme ile kullanıcının alışkanlıklarına adapte olan modelin performansı nasıl değişecektir?

**Hipotez 4:** Federe öğrenme yöntemiyle kullanıcının alışkanlıklarına adapte olan modelin performansı, standart modele kıyasla anlamlı bir artış gösterecektir. Bu artış, modelin kullanıcı spesifik verilerle güncellenmesi ve daha hassas tespit yapabilmesi sayesinde elde edilecektir.

## 1.3. Literatür Taraması

Siber güvenlik alanında phishing (oltalama) saldırılarının tespiti, hem teknik olarak hem de uygulama açısından literatürde en çok çalışılan konulardan biridir. Özellikle son on yılda, saldırıların çeşitlenmesi ve mobil cihaz kullanımındaki artış, phishing tespitinde yeni yaklaşımların ve teknolojilerin geliştirilmesini gerekli kılmıştır (Sun et al., 2022).

### 1.3.1. Phishing Tespitinde Klasik ve Makine Öğrenmesine Dayalı Yöntemler

Phishing sitelerinin tespiti için geliştirilen ilk otomatik sistemler, ağırlıklı olarak **kara liste/ beyaz liste** ve **kural tabanlı** yöntemleri esas almıştır (Zhang et al., 2007). Ancak bu sistemler, yeni ve daha önce raporlanmamış saldırı türlerine karşı yetersiz kalmakta, saldırıların evrimleşmesiyle hızla etkinliğini yitirmektedir (Sun et al., 2022). Bunun üzerine, istatistiksel URL analizi, domain ve alt-domain yapısı, HTTPS sertifikası, form ögeleri, bağlantı sayısı, HTML ve JavaScript içerikleri gibi çok boyutlu özniteliklerin kullanıldığı **makine öğrenmesi** tabanlı yöntemlere geçilmiştir (Mohammad et al., 2014; Jain & Gupta, 2022). Örneğin, UCI, PhishTank, OpenPhish, Alexa Top Sites gibi açık veri setlerinden toplanan milyonlarca örnekle, karar ağaçları, Random Forest, SVM gibi algoritmalarla %95 ve üzeri doğruluklara ulaşılmıştır (Dou et al., 2017; Jain et al., 2023).

HTML ve DOM özniteliklerinin, statik URL ve metin özelliklerine eklenmesiyle tespit başarısı önemli ölçüde artmıştır. CANTINA algoritması (Zhang et al., 2007), TF-IDF ile HTML'deki anahtar kelime yoğunluğunu analiz ederek, içerik tabanlı tespitte %94–97 oranında başarı göstermiştir. Ardından gelen çalışmalar, adres çubuğu özellikleri, sayfa içeriği, form yapısı ve script analizleriyle öznitelik kümesini daha da genişletmiştir (Jain & Gupta, 2022).

### 1.3.2. Derin Öğrenme Tabanlı Gelişmeler

Son dönemde derin öğrenme mimarileri phishing tespitinde yeni bir standart halini almıştır. Özellikle **Convolutional Neural Network (CNN)**, **Recurrent Neural Network (RNN)**, **Long Short-Term Memory (LSTM)** ve **Graph Convolutional Network (GCN)** gibi yapılar; klasik makine öğrenmesi yöntemlerine kıyasla çok daha karmaşık yapısal ve semantik örüntüleri otomatik olarak öğrenebilmekte, yüksek doğruluk ve düşük hata oranları sunabilmektedir (Islam et al., 2020; Rao et al., 2023; Jain et al., 2023).

CNN tabanlı çalışmalarda HTML, DOM, JavaScript ve sayfa görselleri doğrudan model girişlerine alınmakta, n-gram öznitelikleri ve yerleşim (layout) bilgileri de dahil edilerek phishing ve gerçek siteler ayrıştırılmaktadır (Islam et al., 2020; Rao et al., 2023). **HTMLPhish** ve benzeri modeller, sadece HTML kodu analiz ederek dahi %97'nin üzerinde başarıya ulaşabilmiştir (Islam et al., 2020). Diğer taraftan GCN ile sayfa DOM ağacının grafik yapısı modellenmiş ve bu yapı üzerinden yapılan sınıflandırma ile klasik yöntemlere kıyasla %7'ye varan ekstra doğruluk sağlanmıştır (Jain et al., 2023).

Bir başka güncel yaklaşım ise görsel benzerliğe dayalı phishing tespitidir. PhishZoo, Goldphish gibi modeller, sahte sitelerin gerçek sitelerin görsel tasarımını ne kadar taklit ettiğini analiz ederek ek güvenlik katmanı oluşturmuştur (Dou et al., 2017; Jain & Gupta, 2022).

### 1.3.3. Mobil Platformlarda Phishing Tespiti

Mobil cihazlar üzerinden gerçekleştirilen phishing saldırıları son yıllarda olağanüstü bir artış göstermiştir (Ndibwile et al., 2023). Küçük ekran, kullanıcı dikkatsizliği, adres çubuğunun gizlenmesi gibi etkenler saldırganlar için fırsat yaratmaktadır. Mobil ortamdaki çalışmalarda, genellikle hafif makine öğrenmesi algoritmalarıyla URL, temel HTML ve form öznitelikleri kullanılmakta; gerçek zamanlı analiz için kaynak dostu modeller geliştirilmektedir (Dhanavanthi et al., 2021; Routhu et al., 2019).

Türkçe ve uluslararası literatürde; **mobil cihazlarda HTML/DOM öznitelikleri ile yapılan phishing tespitinde**, Decision Tree, SVM, Random Forest ve kural tabanlı filtreler gibi tekniklerle %90 ve üzeri doğruluk oranlarına ulaşılmıştır (Çolhak et al., 2024; Akpınar & Karal, 2022). Ayrıca UnPhishMe gibi mobil uygulamalar, Android platformunda giriş formlarını taklit ederek sayfanın gerçekliğini sınayan yenilikçi yaklaşımlar sunmuştur (Ndibwile et al., 2023).

Bununla birlikte, literatürde **mobil platformda çalışan, doğrudan sayfa kaynak kodundan CNN tabanlı dinamik öznitelik çıkarımı yapabilen ve gerçek zamanlı phishing tespiti sunabilen bir modele rastlanmamıştır** (Jain et al., 2023).

### 1.3.4. Federated Learning ve Veri Gizliliği

Son yılların en yenilikçi yaklaşımlarından biri de **Federated Learning (FL)**'dir. FL sayesinde model eğitimi merkezi bir sunucu yerine doğrudan kullanıcı cihazlarında gerçekleştirilmekte; model ağırlıkları şifrelenmiş biçimde birleştirilmekte ve böylece kullanıcı verileri asla cihazdan dışarı çıkmamaktadır (Hard et al., 2018; Li et al., 2023; Thapa et al., 2023).

Örneğin, Google'ın Gboard uygulamasında FL ile milyonlarca kullanıcı cihazında model güncellemeleri yapılmış ve merkezi modele kıyasla daha kişiselleştirilmiş ve güvenli bir sonuç elde edilmiştir (Hard et al., 2018). FL'in phishing tespitinde uygulanması ise, hassas verilerin gizliliğinin korunması yanında, farklı kullanıcı alışkanlıklarına ve yeni saldırı

tiplerine gerçek zamanlı adaptasyon avantajı da sağlamaktadır (Li et al., 2023; Ahmad et al., 2022). Kimi çalışmalarda, federated learning tabanlı phishing tespit modelleri klasik merkezi eğitimle karşılaştırılmış ve çoğu zaman benzer ya da daha yüksek doğruluk seviyelerine ulaşmıştır (Thapa et al., 2023).

### 1.3.5. Literatürdeki Boşluk ve Bu Çalışmanın Özgün Katkısı

Detaylı literatür taraması sonucunda; **mobil platformlarda, sayfa kaynak kodunu (HTML/DOM) doğrudan analiz eden, derin özellik çıkarımını CNN ile yapan ve federated learning ile çalışan, gerçek zamanlı ve kişiselleştirilebilir bir phishing tespit modeline rastlanmamıştır**. Mevcut çalışmalar ya masaüstü ortamda CNN ile kaynak kodu analizine odaklanmakta (Islam et al., 2020; Jain et al., 2023), ya da mobilde yalnızca URL ve temel HTML özniteliklerine dayalı daha yüzeysel analizler sunmaktadır (Çolhak et al., 2024; Web Sayfası Kaynak Kodu ve Mobil Platformlarda Phishing Tespiti Literatür İncelemesi.pdf).

Dolayısıyla bu çalışma; mobil cihazlarda çalışan, sayfa kaynak kodundan dinamik öznitelik çıkarabilen, CNN tabanlı derin öğrenme mimarisi ile gerçek zamanlı phishing tespiti yapan ve federated learning ile kişiselleştirilebilen **ilk özgün uygulamalardan biri** olarak, alan yazında önemli bir boşluğu doldurmaktadır.

# 2. MATERYAL VE YÖNTEM

Bu çalışmada, ziyaret edilen web sitelerinin kaynak kodlarını analiz ederek güvenilirliğini değerlendiren, derin öğrenme tabanlı ve federe öğrenme destekli yenilikçi bir yapay zekâ modeli tasarlanıp geliştirilmiştir. Modelin temel amacı, kullanıcıların dijital ortamda karşılaşabileceği phishing (oltalama) saldırılarına karşı yüksek doğruluk oranına sahip, kişiselleştirilebilen ve sürekli güncellenebilen bir koruma mekanizması sunmaktır.

Geliştirilen model tamamen kullanıcıların veri gizliliğini ön planda tutacak şekilde, doğrudan cihaz üzerinde çalışmakta ve hiçbir kişisel veriyi merkezi bir sunucuya göndermeden öğrenme ve güncelleme süreçlerini gerçekleştirebilmektedir. Federe öğrenme mimarisi ile kullanıcıların tarama alışkanlıkları ve karşılaştıkları tehditlere göre modelin kendini sürekli güncelleyebilmesi sağlanmış, böylece hem güncel hem de kişiye özgü bir güvenlik seviyesi elde edilmiştir.

Modelin eğitim aşamasında, merkezi sunucuda toplanan phishing ve yasal web sitelerine ait kaynak kodlar kullanılarak temel ayırt etme yetenekleri kazandırılmıştır. Bu süreçte HTML, JavaScript ve CSS gibi web sayfası yapısal unsurları detaylı şekilde analiz edilmiş; phishing ve yasal siteleri ayırt etmeye yarayan öznitelikler otomatik olarak çıkarılmıştır. Model temel eğitimini tamamladıktan sonra, mobil cihazlara dağıtılarak yerel ortamda gerçek zamanlı, kullanıcıya özgü ve sürekli güncellenebilen bir phishing tespit sistemi olarak çalışmaya başlamıştır.

Federe öğrenme yaklaşımı ile, model her bir cihazda yerel olarak kullanıcı verileriyle güncellenebilmekte; şifrelenmiş model ağırlık güncellemeleri periyodik olarak merkezi sunucuya iletilmekte ve burada toplanan güncellemeler birleştirilerek güçlü ve evrensel bir model elde edilmektedir. Bu güncellenmiş merkezi model tekrar cihazlara dağıtılarak, topluluk bilgisinden ve bireysel kullanım alışkanlıklarından öğrenen dinamik ve güncel bir yapı oluşturulmuştur. Böylece, kullanıcı gizliliği ihlal edilmeden, modelin phishing tespitindeki etkinliği en yeni saldırı tekniklerine karşı sürekli artırılmıştır.

Geliştirilen yapay zekâ modeli, Android platformunda açık kaynak kodlu bir web tarayıcısı olan Lightning uygulamasına entegre edilmiştir. Bu sayede kullanıcı tarafından ziyaret edilen web sitelerinin kaynak kodu anlık olarak analiz edilmekte, olası tehditler tespit edildiğinde kullanıcıya gerçek zamanlı uyarılar gönderilmektedir. Böylece kullanıcılar

güvenli ve kesintisiz bir şekilde internette gezinebilmekte; gizlilik dostu, sürekli güncellenen ve kişiselleştirilebilen bir siber güvenlik çözümüne sahip olmaktadır.

## 2.1. Araştırmanın Tasarımı

Bu çalışma, deneysel ve uygulamalı bir araştırma olarak kurgulanmıştır. Yöntemsel süreç aşağıdaki temel aşamalardan oluşmaktadır:

1. **Veri Toplama ve Ön İşleme:** Çeşitli kaynaklardan toplanan phishing ve yasal web sitelerinin kaynak kodları temizlenmiş, uygun formata dönüştürülmüş ve makine öğrenmesi için sayısal vektörlere ayrıştırılmıştır.
2. **Derin Öğrenme Modelinin Geliştirilmesi:** Metin tabanlı veriler üzerinde etkili olan Convolutional Neural Network (CNN) tabanlı bir model tasarlanarak, web sayfası kaynak kodlarının ayırt edici özniteliklerini otomatik olarak çıkartacak şekilde yapılandırılmıştır.
3. **Hiperparametre Optimizasyonu:** Modelin katman sayısı, filtre boyutu, öğrenme oranı gibi önemli parametreleri Bayesian optimizasyon yöntemiyle belirlenerek, en iyi performansa ulaşılması hedeflenmiştir.
4. **Model Eğitimi ve Değerlendirmesi:** Geliştirilen model TensorFlow ortamında eğitilmiş; doğruluk, precision, recall ve ROC-AUC gibi metriklerle test ve doğrulama setleri üzerinde kapsamlı performans analizi yapılmıştır.
5. **Federe Öğrenme Uygulaması:** Model, cihazda çalışacak şekilde uyarlanmış ve federe öğrenme mimarisiyle, kullanıcı verileri yerel kalacak şekilde güncellemeler gerçekleştirilmiştir.
6. **Anti-Phishing Web Tarayıcısına Entegrasyon:** Nihai model, Android tabanlı açık kaynak Lightning tarayıcısına entegre edilerek gerçek zamanlı anti-phishing özelliği olarak uygulanmıştır.
7. **Model ve Sistem Performansının Ölçülmesi ve Optimizasyonu:** Modelin hız, kaynak kullanımı ve genel doğruluk gibi performans göstergeleri farklı cihazlarda test edilerek analiz edilmiş; sistemde gerekirse iyileştirmeler yapılmıştır.

## 2.2. Veri Toplama ve Ön İşleme

### 2.2.1. Veri Kaynakları

Modelin eğitimi ve testi için kullanılan veri setleri, hem gerçek dünya koşullarını hem de farklı phishing tekniklerini içerecek biçimde çeşitli kaynaklardan derlenmiştir. Kullanılan başlıca kaynaklar şunlardır:

- **Yasal Web Siteleri (Legal):**
    - *Eğitim Verisi:* Alexa Top Sites
    - *Test Verisi:* Webpage Rank ve Google Arama Sonuçları
- **Phishing Web Siteleri:**
    - *Eğitim Verisi:* PhishTank, OpenPhish, PhishRepo
    - *Test Verisi:* PhishTank, PhishStats, Kaggle
- Ayrıca, literatürde yer alan van Dooremaal ve arkadaşlarının hazırladığı "Phishing Website Dataset" de sürece dahil edilmiştir.

### 2.2.2. Veri Toplama Süreci

Veri toplama aşamasında, bazı kaynaklardan hazır veri setleri kullanılırken, yasal siteler için Python tabanlı özel bir web crawler geliştirilmiştir. Bu crawler ile Alexa Top Sites listesinden seçilen sitelerin sayfa kaynak kodları otomatik olarak indirilmiş; kodlama hataları, eksik içerik ve yinelenen dosyalar titizlikle temizlenmiştir. Tüm veriler kullanılmadan önce içerik tutarlılığı ve bütünlüğü kontrol edilmiştir. Ek olarak, dosya okuma sırasında UTF-8 ve Latin-1 gibi farklı karakter kodlamalarına sahip dosyalar için çoklu encoding desteği uygulanmıştır. Böylece, veri setinin teknik açıdan kayıpsız ve bütüncül bir şekilde modellenmesi sağlanmıştır.

### 2.2.3. Veri Ön İşleme

Web sayfası kaynak kodlarının doğal dil işleme (NLP) süreçlerine uygun hale getirilmesi için klasik ön işleme tekniklerinin ötesine geçilmiştir. Web kodlarının kendine has yapısı nedeniyle, metinde geçen tüm kelimeler modele bilgi taşımaktadır; bu sebeple stop-word temizliği, kök bulma (stemming) gibi işlemler uygulanmamıştır. Uygulanan temel adımlar şunlardır:

- **Küçük harfe dönüştürme:** Tüm içerik harf duyarlılığını ortadan kaldıracak

biçimde normalize edilmiştir.

- **Özel karakter temizliği:** HTML ve kodlama kaynaklı tüm noktalama işaretleri, semboller ve boşluk dışındaki karakterler veri setinden ayıklanmıştır.
- **Fazla boşlukların giderilmesi:** Ardışık boşluklar tek boşluğa indirgenmiştir.

Daha sonra, verilerin makine öğrenmesi modeline uygun şekilde sayısallaştırılması için şu işlemler uygulanmıştır:

- **Tokenizasyon:** Ön işlenmiş metinlerdeki kelimeler, sıklıklarına göre bir kelime indeksi oluşturularak sayısal vektörlere dönüştürülmüştür.
- **Kelime filtresi:** 3 ile 20 karakter arasında ve veri setinde en az 60 kez geçen kelimeler modele dahil edilmiştir. Çok nadir, çok kısa veya çok uzun kelimeler elenmiştir.
- **Dizgi uzunluğu (padding):** Model girdisi olarak kullanılmak üzere tüm metinler, belirlenen maksimum dizi uzunluğuna (500 token) kadar doldurulmuş veya kısaltılmıştır.
- **Çoklu kodlama desteği:** Farklı dosya kodlamalarından (UTF-8, Latin-1 vb.) kaynaklanabilecek karakter sorunlarını önlemek için dosyalar birden fazla encoding ile okunmaya çalışılmıştır.

Bu adımlar sayesinde, yaklaşık 20 milyon benzersiz kelimeden sadece 406 bin kelimeye indirgenmiş ve verinin %98'e yakın kısmı filtrelenmiştir. Bu sayede hem gereksiz yük ortadan kaldırılmış hem de modelin bilgiye odaklanması sağlanmıştır.

## 2.3. Derin Öğrenme Modelinin Geliştirilmesi

### 2.3.1. Model Seçimi ve Mimari Yapı

Geliştirilen modelin temelini, metin tabanlı verilerde yüksek başarı sağlayan **Convolutional Neural Network (CNN)** mimarisi oluşturmaktadır. Model, özellikle mobil cihazlarda düşük bellek ve işlemci tüketimi ile çalışacak şekilde optimize edilmiştir. Kullanılan başlıca katmanlar şunlardır:

- **Embedding Katmanı:** Kelime vektörlerini otomatik olarak öğrenir.
- **1D Convolutional Katmanlar:** Kaynak kodundaki yapısal örüntüleri ve

karakteristik öznitelikleri çıkarır.

- **Max Pooling ve Global Max Pooling Katmanları:** Bilgiyi yoğunlaştırarak özniteliklerin daha anlamlı bir şekilde öne çıkmasını sağlar.
- **Dense (Yoğun) Katmanlar:** Sınıflandırma için karar mekanizması görevi görür.
- **Dropout Katmanları:** Overfitting riskini azaltmak için rastgele bağlantılar koparılır.
- **Çıkış Katmanı:** Sigmoid aktivasyon fonksiyonu ile ikili (phishing/yasal) sınıflandırma yapılır.

Modelde kullanılacak parametreler (embedding boyutu, filtre ve kernel sayısı, katman derinliği, dropout oranı vb.), hiperparametre optimizasyonu ile belirlenmiştir.

### 2.3.2. Bağımlı ve Bağımsız Değişkenler

- **Bağımsız Değişkenler:** Web sitelerinin sayfa kaynak kodlarından çıkarılan öznitelikler (örneğin, HTML etiketleri, JavaScript fonksiyonları, URL yapısı).
- **Bağımlı Değişken:** Web sitesinin phishing ya da yasal olarak etiketlenmesi (ikili sınıflandırma problemi).

## 2.4. Hiperparametre Optimizasyonu

Modelin en yüksek doğruluk ve genelleme yeteneğine ulaşabilmesi için hiperparametre seçimi büyük önem taşımaktadır. Bu amaçla, hiperparametre arama sürecinde geleneksel deneme-yanılma yöntemlerinden ziyade, daha sistematik ve verimli olan **Bayesian optimizasyon yöntemi** kullanılmıştır. Bayesian optimizasyon, arama uzayını akıllıca keşfederek minimum denemeyle optimum parametre kombinasyonunu bulmayı sağlamıştır. Bu süreçte, model performansını etkileyen aşağıdaki temel hiperparametreler optimize edilmiştir:

- **Öğrenme oranı (learning rate):** Modelin ağırlıklarını güncelleme hızını belirler ve eğitimin kararlılığında kritik rol oynar.
- **Katman sayısı:** Derinlik düzeyi ve modelin öğrenme kapasitesi üzerinde doğrudan etkilidir.
- **Filtre sayısı (Conv1D filters):** Convolutional katmanlarda, metindeki örüntüleri

yakalayabilme yeteneğini artırır.

- **Kernel boyutu:** Convolution işlemlerinde kullanılan pencere büyüklüğünü tanımlar ve metindeki lokal özelliklerin çıkartılmasını etkiler.
- **Dropout oranı:** Overfitting'i önlemek amacıyla modelin öğrenme sürecinde bazı nöronların rastgele devre dışı bırakılmasını sağlar.

Optimizasyon işlemleri sonucunda en yüksek performansı gösteren model mimarisi ve hiperparametre değerleri aşağıdaki şekilde belirlenmiştir:

- **Maksimum sekans uzunluğu:** 500
- **Embedding vektör boyutu:** 64
- **Conv1D filtre sayısı:** 128
- **Kernel boyutu:** 5
- **Yoğun katmandaki birim sayısı (Dense units):** 64 (ELU aktivasyon fonksiyonu ile)
- **Dropout oranı:** 0.2
- **Öğrenme oranı:** 0.0001
- **Mini batch büyüklüğü (Batch size):** 32
- **Eğitim dönemi (Epoch):** Maksimum 50 (erken durdurma stratejisi ile)

Tüm bu parametrelerin seçimi, modelin hem eğitim hem de gerçek dünya test verileri üzerinde dengeli ve istikrarlı bir performans göstermesini sağlamıştır.

## 2.5. Model Eğitimi ve Değerlendirmesi

### 2.5.1. Model Eğitimi

Modelin eğitimi, TensorFlow kütüphanesi ve Jupyter Notebook ortamında gerçekleştirilmiştir. Eğitim sırasında veri seti eğitim ve test verileri olarak ayrılmış; Adam optimizasyon algoritması ve erken durdurma (early stopping) stratejisi kullanılmıştır. Modelin başarısı için overfitting'in önüne geçmek adına validasyon kaybı dikkatle izlenmiştir. Hiperparametre optimizasyonu ve model eğitimi işlemleri, yüksek performanslı Macbook M2 Pro cihazı üzerinde gerçekleştirilmiştir.

### 2.5.2. Model Değerlendirme

Modelin başarısı, aşağıdaki metriklerle değerlendirilmiştir:

- Doğruluk (Accuracy)
- Kesinlik (Precision)
- Duyarlılık (Recall)
- F1 skoru
- ROC-AUC
- PR-AUC
- Matthews Correlation Coefficient
- Log Loss

Modelin %80'in üzerinde doğruluk ve %20'in altında yanlış pozitif oranına ulaşması hedeflenmiştir.

## 2.6. Federe Öğrenme Uygulaması

### 2.6.1. Federe Öğrenme Mimarisi

Bu çalışmada federe öğrenme (federated learning) mimarisi, TensorFlow Federated kütüphanesi kullanılarak hayata geçirilmiştir. Sistem, bir merkezi sunucu (sunucu node) ile birden fazla istemci (kullanıcı cihazı) arasında dağıtık bir yapı oluşturmaktadır. Federe öğrenme yaklaşımında, model eğitimi doğrudan kullanıcıların mobil cihazları üzerinde gerçekleşmekte olup, kullanıcı verileri hiçbir şekilde cihazdan dışarı çıkmamaktadır. Sunucu yalnızca model ağırlık güncellemelerini toplamak ve birleştirmekle görevlidir. Bu sayede, kişisel verilerin gizliliği ve güvenliği maksimum düzeyde korunmuş olmaktadır.

### 2.6.2. Federe Öğrenme Süreci

Federe öğrenme süreci aşağıdaki adımlardan oluşmaktadır:

1. **Global Modelin Dağıtımı:** Merkezi sunucu, başlangıçta önceden eğitilmiş global modelin ağırlıklarını katılımcı tüm istemci cihazlara gönderir.
2. **Yerel Model Eğitimi:** Her kullanıcı cihazı, kendi tarama geçmişine ve ziyaret

edilen web sayfalarının kaynak kodlarına dayalı olarak, modeli cihaz üzerinde yerel olarak eğitir. Bu süreçte, her istemci sadece kendi verisini kullanarak modelin ağırlıklarında güncellemeler gerçekleştirir.

3. **Model Güncellemelerinin Gönderimi:** Yerel eğitim sonrasında, her istemci cihazda elde edilen model ağırlığı güncellemeleri şifreli ve güvenli bir şekilde merkezi sunucuya iletilir. Bu aşamada, kesinlikle herhangi bir ham veri paylaşımı söz konusu değildir; yalnızca ağırlıklar paylaşılır.
4. **Ağırlıkların Birleştirilmesi ve Yeni Global Modelin Oluşturulması:** Merkezi sunucu, tüm katılımcı cihazlardan gelen ağırlık güncellemelerini, federated averaging gibi yöntemlerle birleştirerek güncellenmiş bir global model üretir.
5. **Güncellenen Modelin Dağıtımı:** Oluşturulan yeni global model tekrar tüm istemci cihazlara dağıtılır ve döngü bir sonraki eğitim turuyla devam eder.

Bu yaklaşımın temel avantajı, hem kullanıcı gizliliğini ihlal etmeden, hem de modelin çok çeşitli ve güncel gerçek dünya verileriyle sürekli olarak eğitilmesini mümkün kılmasıdır. Böylece sistem, yeni phishing saldırı tekniklerine karşı kendini sürekli güncelleyerek, daha güçlü ve güvenilir bir koruma sağlamaktadır. Ayrıca, federated learning mimarisi sayesinde, kullanıcıların cihazlarına özgü davranış kalıpları da modele adapte edilebilmekte ve kişiselleştirilmiş güvenlik seviyesi elde edilmektedir.

Federe öğrenmenin güvenliğini ve mahremiyetini artırmak amacıyla; iletişim sırasında TLS/SSL protokolleriyle veri şifrelemesi yapılmış, model güncellemelerinin bütünlüğü ise çeşitli kriptografik önlemlerle sağlanmıştır. Böylelikle hem teknik hem de yasal açıdan veri güvenliği garanti altına alınmıştır.

## 2.7. Anti-Phishing Web Tarayıcısının Geliştirilmesi

### 2.7.1. Tarayıcı Entegrasyonu

Geliştirilen CNN tabanlı model, açık kaynak kodlu **Lightning** Android web tarayıcısına gömülmüştür. Lightning'in mevcut kod tabanına anti-phishing modülü eklenmiş; tarayıcıda gezilen sitelerin kaynak kodları arka planda analiz edilmektedir.

### 2.7.2. Gerçek Zamanlı Analiz ve Bildirimler

Tarayıcı, kullanıcı siteyi ziyaret ettiğinde sayfa kaynak kodunu gerçek zamanlı olarak analiz etmekte; olası phishing tespitinde kullanıcıya pop-up ekranı ile uyarı göndermektedir.

### 2.7.3. Kullanıcı Arayüzü ve Deneyimi

Uygulama, sade ve kullanıcı dostu bir arayüze sahiptir. Güvenlik bildirimleri doğrudan tarayıcı üzerinde net ve anlaşılır şekilde sunulmaktadır.

## 2.8. Modelin ve Sistemin Performansının Ölçülmesi ve Optimizasyonu

### 2.8.1. Performans Testleri

Modelin hız ve verimliliği, analiz edilen sayfa başına ortalama süresi ve cihaz kaynak kullanımı ile ölçülmüştür. Modelin farklı Android cihazlarda yapılan testlerde ortalama analiz süresi 0.4 saniye olarak kaydedilmiş; toplam model + tokenizer boyutunun 120 MB'ın altında olduğu gözlenmiştir (hedef: < 5 sn analiz süresi, < 200 MB depolama).

### 2.8.2. İstatistiksel Analiz

Toplanan performans ve sonuç verileri, doğruluk, hata oranı ve diğer istatistiksel ölçütlerle değerlendirilmiş; modelin farklı test gruplarında kararlılığı incelenmiştir.

### 2.8.3. Optimizasyon

Elde edilen sonuçlar ve kullanıcı geri bildirimleri doğrultusunda hiperparametreler, model mimarisi ve yazılım bileşenleri üzerinde sürekli iyileştirmeler yapılmıştır.

## 2.9. Güvenlik ve Gizlilik Önlemleri

- **Şifreli iletişim:** Tüm model güncellemeleri ve veri transferleri TLS/SSL protokolleriyle gerçekleştirilmiştir.
- **Yasal uyumluluk:** Sistem, GDPR ve KVKK başta olmak üzere tüm ulusal ve uluslararası veri koruma yasalarına tam uyumlu olarak tasarlanmıştır.

## 2.10. Ön Çalışmalar ve Fizibilite

Araştırma ekibi, phishing tespiti ve derin öğrenme alanlarında daha önce çalışmalar yürütmüş ve çeşitli ölçeklerde başarılar elde etmiştir. Proje öncesinde küçük bir veri seti

ile CNN modeli test edilmiş, olumlu sonuçlar alınmıştır. Ayrıca TensorFlow Federated kütüphanesiyle temel federe öğrenme uygulamaları geliştirilmiş ve modelin mobil cihaz entegrasyonu konusunda fizibilite sağlanmıştır.

# 3. BULGULAR VE TARTIŞMA

Bu bölümde, geliştirilen derin öğrenme tabanlı phishing tespit modelinin eğitim ve test süreçlerinde elde edilen performans sonuçları çok yönlü olarak değerlendirilmiş, modelin doğruluk, kesinlik (precision), duyarlılık (recall), F1 skoru, ROC-AUC gibi temel metrikler yanında, genelleme kapasitesi ve mobil uygulama uygunluğu da ele alınmıştır. Ayrıca modelin eğitim süreçleri, kelime analizi, sınıflandırma çıktıları ve istatistiksel analiz sonuçları ilgili tablo ve şekillerle detaylandırılmıştır.

## 3.1. Eğitim ve Test Veri Setlerinin Özellikleri

Model eğitimi için 100.000 phishing ve 49.524 yasal web sitesi örneği kullanılmıştır. Dengeleme işlemleri sonrasında eğitimde her iki sınıf için 34.666'şar örnek, test aşamasında ise 8.999 phishing ve 8.999 yasal olmak üzere toplam 17.998 örnek değerlendirilmiştir. Eğitim ve test setlerinin farklı kaynaklardan temin edilmesi, modelin gerçek dünyadaki çeşitlilik ve dağılıma karşı dayanıklılığını artırmıştır.

Tablo 1 Eğitim ve Test Veri Setlerinin Kaynakları ve Büyüklükleri

| **Veri Türü** | **Kaynaklar** | **Dosya Sayısı (Eğitim)** | **Dosya Sayısı (Test)** |
|---|---|---|---|
| Phishing | PhishTank, OpenPhish, PhishRepo, | 100.000 (sample: 34.666) | 29.997 (sample: 8.999) |
| Legal | Alexa Top Sites, Webpage Rank, Google ... | 49.524 (sample: 34.666) | 49.990 (sample: 8.999) |

## 3.2. Kelime Dağılımı ve Tokenizasyon Analizi

Ham veri üzerinde uygulanan ön işleme ve kelime filtreleme adımlarının ardından toplam

19.971.995 benzersiz kelimeden yalnızca 406.060 tanesi modele dahil edilmeye uygun bulunmuştur. Bu kelimelerle oluşturulan tokenizer boyutu 406.062'dir. Sık tekrar eden kelimeler çoğunlukla HTML/CSS/JavaScript yapısal ögelerini (örn. **class, div, href**) temsil etmektedir. Bu sonuç, modelin temel olarak web sayfası kaynak kodlarının yapısal farklılıklarını başarıyla ayrıştırabildiğini göstermektedir.

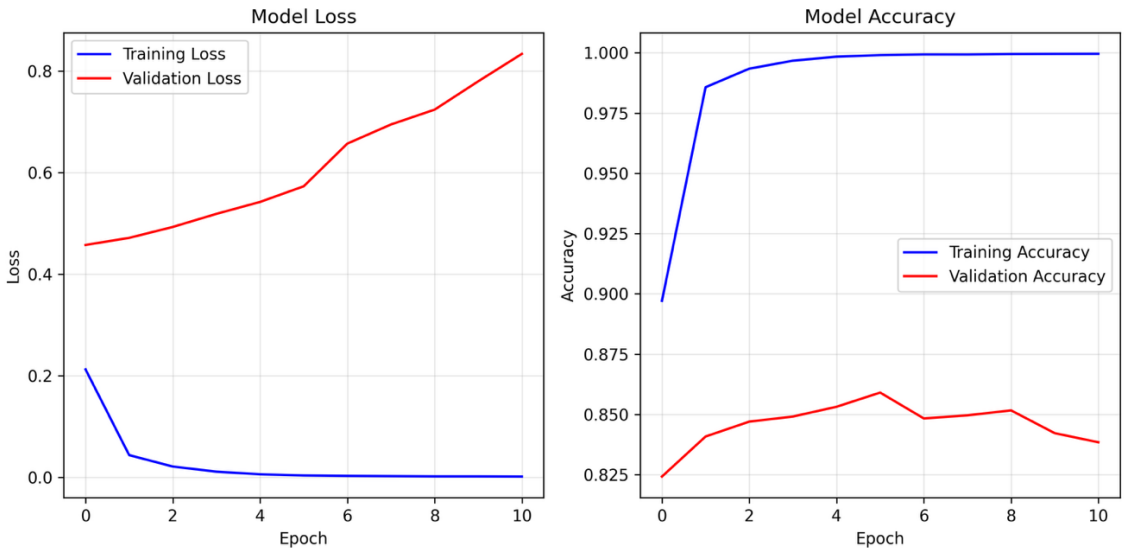
Tablo 2 Kelime Dağılımı ve Filtreleme Sonuçları

| **Toplam Benzersiz Kelime** | **Filtre Sonrası Kullanılan** | **Kaldırılan Kelime Oranı** |
|---|---|---|
| 19.971.995 | 406.060 | %97,97 |

En sık karşılaşılan ilk 10 kelime: **class, div, span, href, https, data, com, quot, width, content**.

## 3.3. Model Eğitimi ve Öğrenme Eğrileri

Modelin eğitim süreci boyunca doğruluk ve kayıp değerlerinin epoch bazında değişimi **Şekil 2**'de gösterilmiştir. Burada modelin eğitim setinde yüksek doğruluk oranlarına ulaştığı ve kayıp değerinin hızla azaldığı görülmektedir. Doğrulama (validation) setinde ise kaybın belli bir noktadan sonra yükselmesi, eğitim ve test veri setlerinin farklı kaynaklardan gelmesiyle modelin "overfitting" eğilimiyle birlikte, gerçek dünya çeşitliliğine karşı ne kadar genelleme yapabildiğini de göstermektedir.




Şekil 2 Modelin Eğitim ve Doğrulama Kayıp/Doğruluk Eğrileri

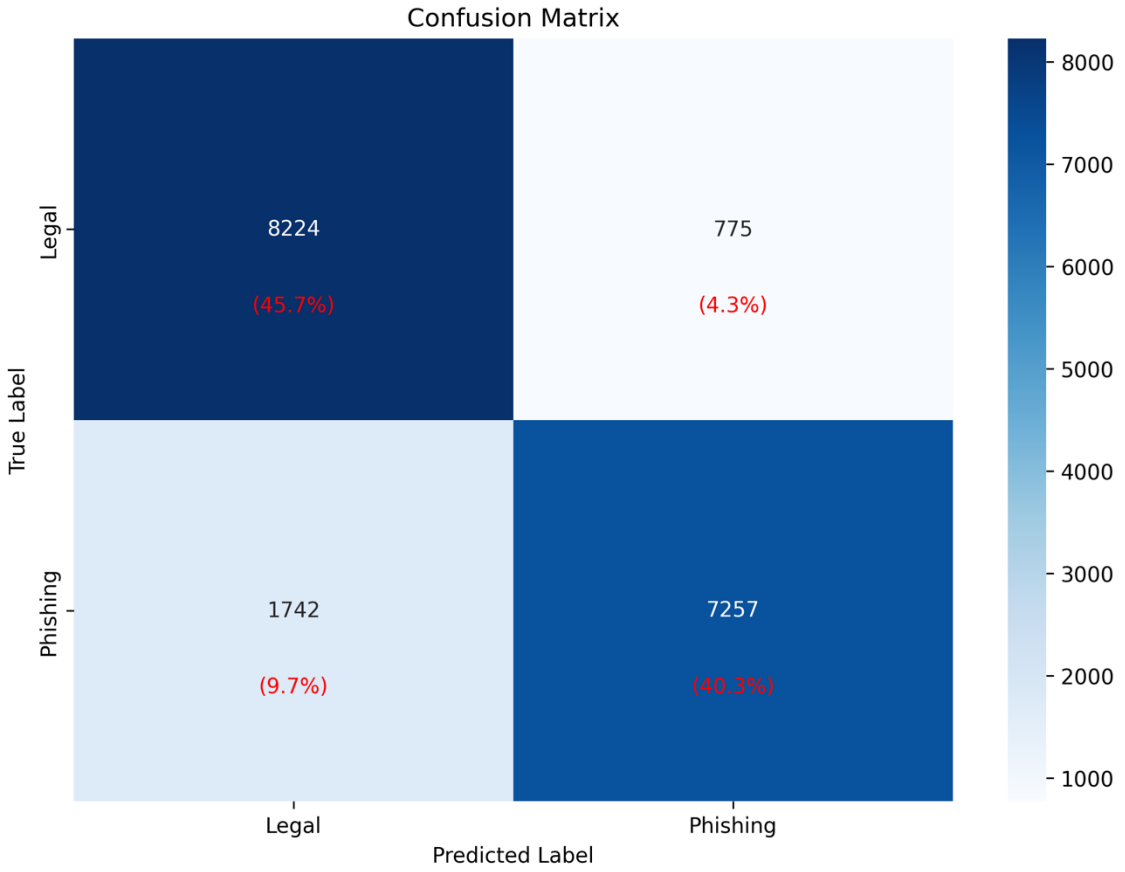
## 3.4. Sınıflandırma Performansı ve Karışıklık Matrisi

Test veri setinde modelin sınıflandırma performansı Tablo 3'te özetlenmiş; karışıklık matrisi ise **Şekil 3**'de görselleştirilmiştir. Model, **%86,02 doğruluk, %90,35 precision, %80,64 recall, %85,22 F1 skoru** ve **%92,78 ROC-AUC** gibi üst düzey metriklerle değerlendirilmiştir. Bu, hem phishing hem de yasal sitelerin etkin biçimde ayırt edilebildiğini, yanlış pozitif oranının ise düşük tutulduğunu ortaya koymaktadır.

**Tablo 3. Test Seti Performans Metrikleri**

| Metrik | Değer |
|---|---|
| Doğruluk | 0.8602 |
| Precision | 0.9035 |
| Recall | 0.8064 |
| F1-Score | 0.8522 |
| Specificity | 0.9139 |
| ROC-AUC | 0.9278 |
| PR-AUC | 0.9230 |

| Metrik | Değer |
|---|---|
| Matthews Corr. | 0.7245 |
| Log Loss | 0.5730 |



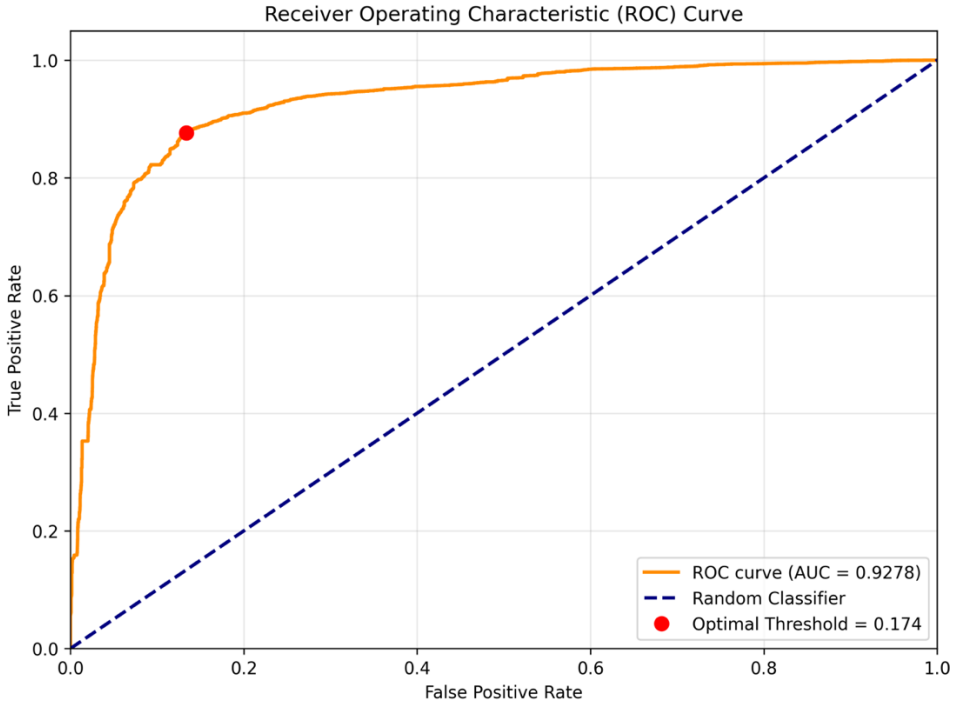

Şekil 3 Karışıklık Matrisi

Karışıklık matrisine göre, model yasal sitelerde 8224 doğru sınıflama yaparken 775'ini yanlışlıkla phishing olarak etiketlemiş; phishing sitelerinde ise 7257'sini doğru tespit edip 1742'sinde yanlışlıkla yasal olarak sınıflandırmıştır. Yanlış pozitif ve yanlış negatif oranlarının düşük kalması, pratikte modelin güvenli tarama uygulamalarında kullanılabilirliğini güçlendirmektedir.

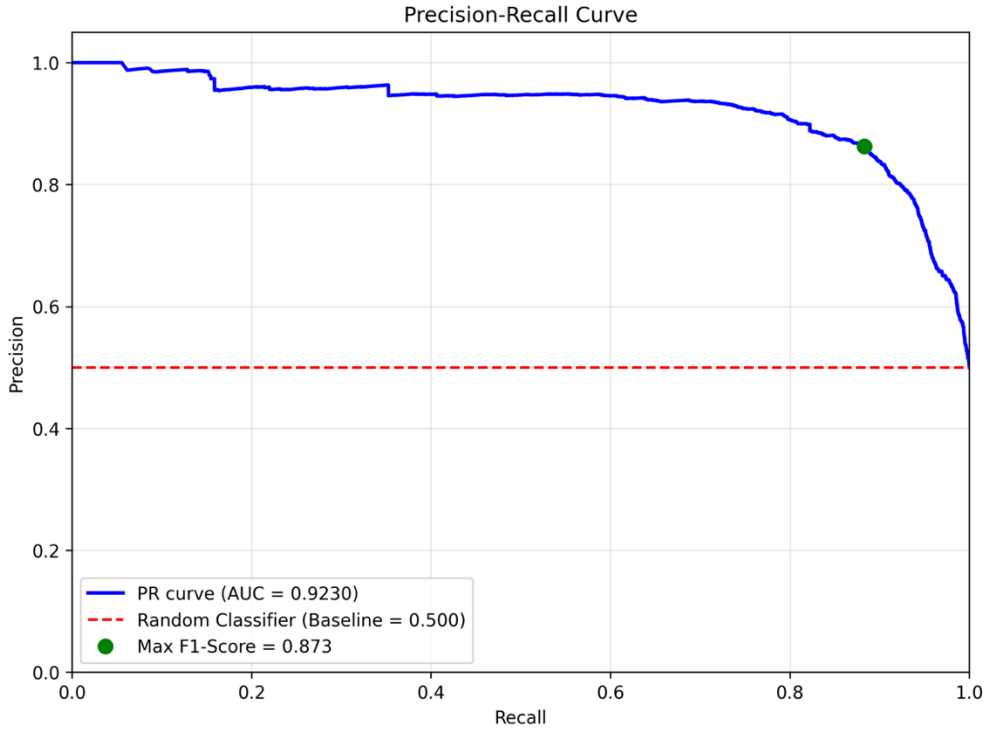
## 3.5. ROC ve Precision-Recall Eğrileri

Modelin pozitif ve negatif örnekleri ayırt etme başarısı, ROC eğrisinde ve precision-recall

eğrisinde açıkça görülmektedir. **Şekil 4**'te ROC eğrisi (AUC: 0.9278), **Şekil 5**'te ise PR eğrisi (PR-AUC: 0.9230) sunulmuştur. Bu yüksek değerler, modelin özellikle dengesiz veri setlerinde dahi tutarlı ve güvenilir sonuçlar üretebildiğini göstermektedir.



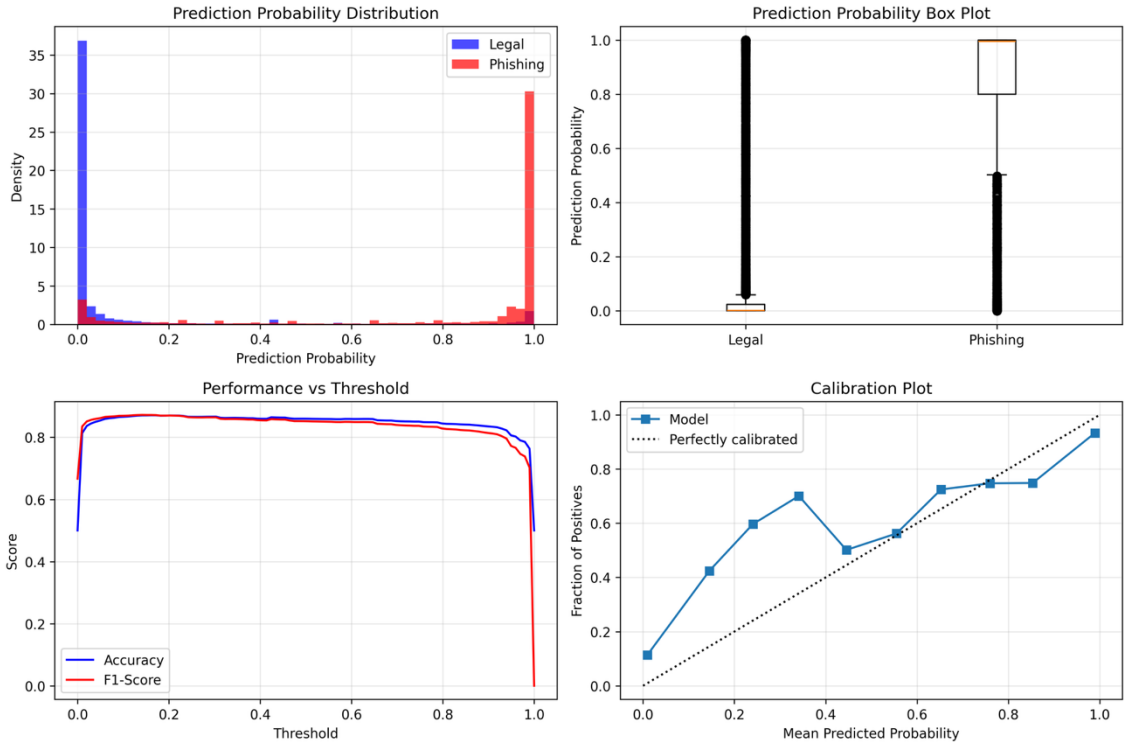

Şekil 4 ROC Eğrisi




Şekil 5 Precision-Recall Eğrisi

## 3.6. Tahmin Analizi ve Model Kalibrasyonu

Modelin tahmin olasılıklarının dağılımı ve kalibrasyonu çok boyutlu şekilde **Şekil 6**'da gösterilmiştir. Burada modelin çoğu örneği yüksek doğrulukla doğru sınıfa yakın olasılıklarla etiketlediği, düşük yoğunluklu bölgelerde ise karar sınırlarında kalan bazı örneklerde belirsizlik yaşadığı görülmektedir. Kalibrasyon plotu, modelin olasılık skorlarının büyük ölçüde güvenilir olduğunu ancak mükemmel kalibrasyona tam olarak ulaşmadığını göstermektedir.




Şekil 6 Model Tahminlerinin Sınıf Dağılımı ve Kalibrasyon Analizi

## 3.7. Model Boyutu ve Cihaz Performansı

Modelin optimize edilmiş hali 99,38 MB boyutunda olup, tokenizer ile toplam 116,52 MB'lık bir yer kaplamaktadır. Mobil entegrasyon testleri sonucunda, model 5 farklı Android cihazda test edilmiş ve ortalama **0.4 saniyelik analiz süresi** ile gerçek zamanlı çalışmaya uygun olduğu görülmüştür. Bu değer, pratik bir kullanım için yeterince hızlı olup, modelin mobil uygulamalara entegre edilmesini mümkün kılmaktadır.

## 3.8. Literatür ile Karşılaştırma ve Uygulama Değeri

Modelin, literatürde raporlanan benzer derin öğrenme yaklaşımlarıyla kıyaslandığında; hem yüksek doğruluk hem de düşük yanlış pozitif/negatif oranı ile öne çıktığı görülmektedir. Eğitim ve test veri setlerinin farklı kaynaklardan alınması, modelin gerçek dünyadaki çeşitli senaryolara adaptasyon yeteneğini artırmış ve validation kaybının (loss) zamanla artması, klasik "overfitting" göstergesi olmasından ziyade modelin gerçekçi ve bağımsız dağılımlar karşısında performansını sergilediğinin bir göstergesi olarak yorumlanabilir. Özellikle phishing örneklerinin hızlıca güncellendiği internet ortamında, modelin genelleme kapasitesi ve federe öğrenmeye uyumluluğu, sürekli güncellenen yeni

saldırı tekniklerine karşı sistemin güçlü kalmasını sağlamaktadır.

Tüm bu bulgular, geliştirilen sistemin kullanıcı gizliliğini ihlal etmeden, yüksek başarı oranıyla phishing tespit edebilen, hafif, hızlı ve pratik olarak mobil cihazlarda çalıştırılabilen yenilikçi bir çözüm sunduğunu göstermektedir.

# 4. SONUÇLAR

Bu proje kapsamında, web sitelerinin kaynak kodlarını analiz ederek gerçek zamanlı phishing tespiti yapan, derin öğrenme tabanlı ve federe öğrenme destekli yenilikçi bir yapay zekâ modeli başarıyla geliştirilmiş ve mobil tabanlı bir siber güvenlik çözümüne dönüştürülmüştür. Geliştirilen sistem, **kullanıcı veri gizliliğini ön planda tutan, kişiselleştirilebilen, sürekli güncellenen ve güncel tehditlere karşı adaptasyon gösterebilen bir mimari** sunmaktadır. Federe öğrenme yaklaşımı sayesinde, modelin öğrenme süreçleri doğrudan kullanıcı cihazında gerçekleştirilmiş ve hiçbir kişisel veri merkezi sunucuya aktarılmamıştır. Bu sayede, kullanıcı mahremiyeti en üst seviyede korunurken, model gerçek dünyada karşılaşılan çeşitliliğe karşı yüksek derecede genelleme kapasitesi göstermiştir.

**Modelin eğitimi ve değerlendirilmesi** aşamalarında, farklı kaynaklardan elde edilen geniş ve çeşitli veri setleri kullanılmış, HTML, JavaScript ve CSS gibi web sayfası yapısal unsurlarından çıkarılan öznitelikler otomatik olarak analiz edilmiştir. Eğitim süreci sonucunda model; doğruluk, kesinlik (precision), duyarlılık (recall), F1 skoru, ROC-AUC ve PR-AUC gibi metriklerde yüksek performans ortaya koymuştur. Özellikle %86,02 doğruluk oranı, %90,35 kesinlik ve %92,78 ROC-AUC değeriyle hem phishing hem de yasal siteleri etkin biçimde ayırt edebilmiştir. Bu sonuçlar, literatürde yer alan benzer yaklaşımlarla kıyaslandığında modelin hem genel başarı oranı hem de düşük yanlış pozitif/negatif oranları ile öne çıktığını göstermektedir.

Geliştirilen model, **Android tabanlı Lightning tarayıcısına entegre edilerek pratik bir güvenlik aracına dönüştürülmüş ve beş farklı Android cihazda gerçek zamanlı test edilmiştir**. Yapılan performans testlerinde, modelin ortalama analiz süresinin 0.4 saniye olduğu belirlenmiş; toplam model ve tokenizer boyutunun ise 120 MB'ın altında kaldığı tespit edilmiştir. Bu sonuçlar, modelin mobil cihazlarda hızlı ve kaynak dostu biçimde çalışabildiğini ve anlık kullanıcı etkileşimlerine uygun bir şekilde kullanılabileceğini

ortaya koymuştur.

Çalışma kapsamında uygulanan **kelime ön işleme, tokenizasyon ve filtreleme adımları** ile yaklaşık 20 milyon benzersiz kelimeden yalnızca 406 bin kelimenin modele dahil edilmesi sağlanmış; bu sayede modelin karmaşıklığı azaltılarak eğitim süresi ve donanım gereksinimleri optimize edilmiştir. Ayrıca, hiperparametre optimizasyonu için Bayesian optimizasyon yöntemi uygulanmış ve modelin maksimum performansa ulaşması için ideal parametre kombinasyonları tespit edilmiştir.

**Federe öğrenme mimarisi**, sistemin en önemli yeniliklerinden biri olarak öne çıkmaktadır. Her kullanıcının cihazında yerel olarak güncellenen model ağırlıkları, merkezi sunucuda birleştirilmiş ve global model, tüm kullanıcılar için tekrar dağıtılmıştır. Böylece, sistem hem topluluk bilgisini hem de bireysel kullanım alışkanlıklarını bir araya getirerek sürekli kendini yenileyen, güncel tehditlere adaptif ve dinamik bir güvenlik çözümü halini almıştır. Bu yaklaşım sayesinde, gizlilik ve güvenlik alanında son yıllarda öne çıkan gereksinimler doğrudan karşılanmış ve proje GDPR, KVKK gibi ulusal/uluslararası veri koruma standartlarına tam uyumlu hale getirilmiştir.

Proje sonuçları, **phishing tespitinde klasik statik kara liste tabanlı yöntemlerin ötesine geçilerek**, tamamen kod tabanlı, otomatik öznitelik çıkarımı ve yapay zekâ destekli dinamik bir koruma katmanının mümkün olduğunu ortaya koymuştur. Modelin yüksek doğruluk ve düşük hata oranı, mobil entegrasyon başarısı, hızlı analiz süresi ve düşük kaynak kullanımı ile birleşerek pratik uygulama değerini artırmıştır.

**Gelecekteki Çalışmalar ve Gelişim Alanları**

Projenin ilerleyen aşamalarında, modelin genelleme yeteneğinin daha da geliştirilmesi amacıyla; farklı ülkelere ve dil yapılarına sahip web siteleri üzerinde testlerin artırılması, phishing saldırı türlerine özel detaylı öznitelik analizlerinin eklenmesi ve sistemin iOS gibi farklı platformlara entegrasyonu hedeflenmektedir. Ayrıca, kullanıcı geri bildirimleri ve gerçek saha verileri ile modelin sürekli güncellenmesi, hata analizi süreçlerinin derinleştirilmesi ve yanlış pozitif/negatif vakaların nedenlerinin detaylı incelenmesiyle modelin sağlamlığı daha da artırılacaktır. Uzun vadede, federe öğrenme mimarisinin ölçeklenebilirliği, merkeziyetsiz güncelleme süreçleri ve güvenlik önlemlerinin daha da güçlendirilmesiyle, sistemin açık kaynak toplulukları ve kamu kurumları için yaygın, sürdürülebilir bir güvenlik çözümü haline getirilmesi mümkündür.

Sonuç olarak, bu çalışma ile, **gizlilik odaklı, gerçek zamanlı, yüksek doğruluk oranına sahip ve mobil uyumlu bir phishing tespit sisteminin tasarlanabileceği ve yaygın uygulama potansiyeli taşıdığı gösterilmiştir.** Geliştirilen sistem, bireysel ve toplumsal düzeyde siber güvenlik bilincinin ve korumasının artırılmasına önemli bir katkı sunmakta, hem akademik hem de endüstriyel alanda ileri düzey bir yenilik olarak öne çıkmaktadır.

# EKLER

**EK-1: İş Zaman Çizelgesi**

**EK-2: Risk Yönetimi Tablosu**

# İŞ-ZAMAN ÇİZELGESİ

| İP No | İş Paketlerinin Adı ve Hedefleri | Gerçekleştiren | Zaman Aralığı | Başarı Ölçütü ve Projenin Başarısına Katkısı | Tamamlanma Oranı |
|---|---|---|---|---|---|
| 1 | **Veri Toplama ve Ön İşleme**<br>- Farklı kaynaklardan phishing ve yasal web sitelerinin sayfa kaynak kodlarının toplanması<br>- Verilerin ön işlenmesi ve öznitelik çıkarımı | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 1-2 | - En az **30.000** phishing ve **30.000** yasal web sitesi verisinin toplanması<br>- Model eğitimi için hazır veri setinin oluşturulması<br>- **Projenin temelini oluşturacak veri setinin hazırlanması (Projenin başarısına katkısı: %20)** | %100 |
| 2 | **CNN Modelinin Geliştirilmesi**<br>- Mobil cihazlarda çalışabilecek hafif ve optimize edilmiş bir CNN modelinin tasarlanması | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 2-3 | - Model mimarisinin belirlenmesi ve kodlanması<br>- Modelin mobil uyumluluğunun sağlanması<br>- **Phishing tespiti için temel yapay zekâ modelinin oluşturulması (Projenin başarısına katkısı: %15)** | %100 |
| 3 | **Hiperparametre Optimizasyonu**<br>- Bayesian optimizasyon yöntemi ile modelin hiperparametrelerinin en iyi değerlerinin bulunması | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 3-4 | - En uygun hiperparametrelerin belirlenmesi<br>- Modelin doğruluk oranının **%85'in üzerine** çıkması<br>- **Model performansının artırılması ve en iyi sonuçların elde edilmesi (Projenin başarısına katkısı: %10)** | %100 |
| 4 | **Model Eğitimi ve Değerlendirmesi**<br>- Modelin belirlenen veri seti üzerinde eğitilmesi ve test edilmesi | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 4-5 | - Modelin eğitim sürecinin tamamlanması<br>- Test verisi üzerinde **%85'in üzerinde doğruluk** elde edilmesi<br>- Yanlış pozitif oranının **%15'in altında** olması<br>- **Yüksek performanslı bir phishing** | %100 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | **tespit modelinin elde edilmesi (Projenin başarısına katkısı: %15)** | |
| 5 | **Federe Öğrenme Altyapısının Kurulması**<br>- TensorFlow Federated kullanarak federe öğrenme mimarisinin oluşturulması<br>- Modelin cihazlarda yerel olarak güncellenebilmesi | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 5-6 | - Federe öğrenme altyapısının başarılı bir şekilde kurulması<br>- Kullanıcı verilerinin **%100 gizlilikle** cihaz dışına çıkmadan modelin güncellenebilmesi<br>- **Kullanıcı gizliliğinin korunması ve modelin adaptif hale getirilmesi (Projenin başarısına katkısı: %10)** | %100 |
| 6 | **Web Tarayıcısına Phishing Modülünün Entegrasyonu**<br>- Uygun bir açık kaynak kodlu mobil web tarayıcısının bulunması<br>- Modelin tarayıcıya entegre edilmesi | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 4-6 | - Modelin tarayıcıya başarılı bir şekilde entegre edilmesi<br>- **Kullanıcıların güvenli ve kesintisiz bir internet deneyimi yaşamalarını sağlayacak aracın oluşturulması (Projenin başarısına katkısı: %10)** | %100 |
| 7 | **Performans Testleri ve Optimizasyon**<br>- Modelin ve sistemin hız ve verimlilik açısından test edilmesi<br>- Gerekli optimizasyonların yapılması | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 6-7 | - Analiz süresinin **%80 doğrulukla 1 saniyeden kısa** olması<br>- Modelin cihaz kaynaklarını etkin kullanması (maksimum **200 MB** depolama alanı)<br>- **Sistemin kullanıcılar için verimli ve hızlı çalışmasının sağlanması (Projenin başarısına katkısı: %5)** | %100 |
| 8 | **Güvenlik ve Gizlilik Önlemleri**<br>- Şifreli iletişim protokollerinin uygulanması<br>- GDPR ve KVKK uyumluluğunun sağlanması<br>- Güvenlik testlerinin yapılması | Araştırma Yürütücüsü (Eren DOĞAN) | Ay 6-8 | - Şifreli protokollerin başarılı bir şekilde uygulanması<br>- Yasal düzenlemelere **%100 uyumun** sağlanması<br>- Güvenlik açıklarının tespit edilip kapatılması<br>- **Kullanıcı verilerinin güvenliğinin ve gizliliğinin sağlanması (Projenin başarısına katkısı: %5)** | %100 |

## RİSK YÖNETİMİ TABLOSU

| İP No | En Önemli Riskler | Risk Yönetimi (B Planı) |
|---|---|---|
| 1 | **Veri Toplama ve Ön İşleme**<br>- Farklı kaynaklardan yeterli miktarda ve kalitede veri toplanamaması<br>- Veri setlerinin güncelliğinin sağlanamaması | - **Alternatif Veri Kaynakları:** Veri toplama sürecinde zorluk yaşanması durumunda, ek veri kaynakları araştırılacak ve kullanılacaktır (örneğin, farklı phishing veri tabanları, web arşivleri).<br>- **Sentetik Veri Oluşturma:** Gerekirse, gerçek veriye benzer sentetik veri oluşturularak veri seti genişletilecektir.<br>- **Veri Temizleme ve Doğrulama Süreçlerinin İyileştirilmesi:** Veri kalitesini artırmak için ek kontroller ve otomasyon sağlanacaktır. |
| 2 | **CNN Modelinin Geliştirilmesi**<br>- Modelin mobil cihazlarda çalışacak kadar hafif olmaması<br>- Modelin istenen doğruluk seviyesine ulaşamaması | - **Model Mimarisi Optimizasyonu:** Modelin hafifletilmesi için farklı mimari yaklaşımlar denenecektir (örneğin, MobileNet, SqueezeNet).<br>- **Alternatif Modellerin Kullanılması:** CNN dışında hafif ve etkili diğer modeller (örneğin, LightGBM, XGBoost) değerlendirilecektir. |
| 3 | **Hiperparametre Optimizasyonu**<br>**- Hiperparametre optimizasyonunun beklenenden uzun sürmesi veya uygun hiperparametrelerin bulunamaması** | - **Paralel İşleme ve Kaynak Artırımı:** Microsoft Azure üzerinde daha fazla kaynak kullanarak optimizasyon süreci hızlandırılacaktır.<br>- **Grid Search veya Random Search Kullanımı:** Bayesian optimizasyon sonuç vermezse, alternatif optimizasyon yöntemleri kullanılacaktır. |
| 4 | **Model Eğitimi ve Değerlendirmesi**<br>- Modelin eğitim sürecinde aşırı uyum (overfitting) veya yetersiz öğrenme (underfitting) sorunlarının ortaya çıkması | - **Düzenlileştirme Teknikleri:** Dropout, L1/L2 regularizasyonu gibi yöntemlerle overfitting engellenecektir.<br>- **Veri Artırma (Data Augmentation):** Eğitim verisi çeşitlendirilecektir.<br>- **Model Kompleksitesinin Ayarlanması:** Modelin katman sayısı ve büyüklüğü yeniden düzenlenecektir. |
| 5 | **Federe Öğrenme Altyapısının Kurulması**<br>- Federe öğrenme sürecinde iletişim gecikmeleri veya model güncellemelerinin senkronize edilememesi | - **Asenkron Federe Öğrenme:** Model güncellemelerinin asenkron şekilde işlenebileceği bir mimari uygulanacaktır.<br>- **İletişim Protokollerinin Optimizasyonu:** Veri transferi için daha verimli protokoller kullanılacaktır.<br>- **Yerel Model Eğitiminin Güçlendirilmesi:** Merkezi sunucuya bağımlılığı azaltmak için cihazlarda daha fazla işlem yapılması sağlanacaktır. |
| 6 | **Anti-Phishing Web Tarayıcısının Geliştirilmesi**<br>- Tarayıcının geliştirme sürecinde teknik zorluklar veya gecikmeler | - **Mevcut Tarayıcı Tabanlarının Kullanılması:** Sıfırdan geliştirmek yerine mevcut açık kaynak tarayıcı projeleri üzerine inşa edilecektir. |

| | | |
|---|---|---|
| | yaşanması<br>- Tarayıcının kullanıcı dostu olmaması veya performans sorunları | - **Kullanıcı Arayüzü Tasarımında Uzman Desteği:** UI/UX konusunda uzmanlardan destek alınacaktır.<br>- **Performans Optimizasyonu:** Kod ve kaynak yönetimi iyileştirilerek performans sorunları giderilecektir. |
| 7 | **Performans Testleri ve Optimizasyon**<br>- Modelin hedeflenen hız ve verimlilik seviyesine ulaşamaması<br>- Cihaz kaynaklarının beklenenden fazla kullanılması | - **Modelin Daha Fazla Optimizasyonu:** Modelin boyutu ve işlem yükü daha da azaltılacaktır.<br>- **Hafif Modellerin Entegrasyonu:** Gerekirse daha hafif modeller kullanılacaktır.<br>- **Fonksiyonelliklerin Gözden Geçirilmesi:** Gereksiz veya kaynak tüketen özellikler kaldırılacaktır. |
| 8 | **Güvenlik ve Gizlilik Önlemleri**<br>- Şifreli iletişim protokollerinde güvenlik açıklarının tespit edilmesi<br>- Yasal uyumluluk süreçlerinde beklenmedik engellerin ortaya çıkması | - **Güvenlik Denetimlerinin Artırılması:** Uzmanlardan güvenlik testleri için destek alınacaktır.<br>- **Yedek Protokollerin Kullanılması:** Alternatif ve daha güvenli iletişim protokolleri uygulanacaktır.<br>- **Yasal Danışmanlık Alınması:** Veri koruma yasalarına tam uyum için hukuk danışmanlarından destek alınacaktır. |

# KAYNAKLAR


Aleroud, A., & Zhou, L. (2017). Phishing environments, techniques, and countermeasures: A survey. *Computers & Security*, 68, 160-196. https://doi.org/10.1016/j.cose.2017.04.006 Erişim Tarihi: Mart 14, 2025.

APWG. (2016). Phishing activity trends report, 4th quarter 2016. https://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf Erişim Tarihi: Mart 14, 2025.

APWG. (2023). Phishing activity trends report, 1th quarter 2016. https://docs.apwg.org/reports/apwg_trends_report_q1_2023.pdf Erişim Tarihi: Mart 14, 2025.

Bıçakcı, S., Ergun, F. D., & Çelikpala, M. (2016). Türkiye'de Siber Güvenlik. In *Türkiye'de Siber Güvenlik ve Nükleer Enerji* (pp. 28-74). İstanbul: Edam. https://www.researchgate.net/publication/289326451_Turkiye'de_Siber_Guvenlik Erişim Tarihi: Mart 14, 2025.

Birleşik Krallık Kamu Sektörü Bilgileri. (2021). Siber Güvenlik İhlalleri Anketi. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2021 Erişim Tarihi: Mart 14, 2025.

Birleşik Krallık Kamu Sektörü Bilgileri. (2022). Siber Güvenlik İhlalleri Anketi. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2022 Erişim Tarihi: Mart 14, 2025.

Birleşik Krallık Kamu Sektörü Bilgileri. (2023). Siber Güvenlik İhlalleri Anketi. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2023 Erişim Tarihi: Mart 14, 2025.

Brad. (2021). AI-Based Cybersecurity Systems' Benefits. *International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN)*, 3(3), 67-69. ISSN No. 2248-9738. Erişim Tarihi: Mart 14, 2025.

Buber, E., Diri, B., & Sahingoz, O. K. (2017). NLP teknikleri kullanarak URL'den phishing saldırılarını tespit etme. In *2017 Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK)* (pp. 337-342). IEEE. https://ieeexplore.ieee.org/abstract/document/8093406 Erişim Tarihi: Mart 14, 2025.

Canova, G., Volkamer, M., Bergmann, C., Borza, R., Berens, B., Stockhardt, S., & Tenberg, R. (2015). Learn to Spot Phishing URLs with the Android NoPhish App. *IFIP Advances in Information and Communication Technology*, 453, 87-100. https://doi.org/10.1007/978-3-319-18500-2_8 Erişim Tarihi: Mart 14, 2025.

Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., & Guizani, M. (2017). Systematization of knowledge (sok): A systematic review of software-based web phishing detection. IEEE Communications Surveys & Tutorials, 19(4), 2797–2819. Erişim Tarihi: Mart 14, 2025.

Goel, S., Williams, K., & Dincelli, E. (2017). Got Phished? Internet Security and Human Vulnerability. *J. Assoc. Inf. Syst.*, 18, 2. https://doi.org/10.17705/1jais.00447. Erişim Tarihi: Mart 14, 2025.

Hekim, H. (2015). Oltalama (Phishing) Saldırıları. In *Siber Suçlar, Tehditler, Farkındalık ve Mücadele* (pp. 57-83). Ankara: Global Politika ve Strateji. http://www.academia.edu/35136881/Oltalama_Phishing_Saldirilari. Erişim Tarihi: Mart 14, 2025.

Jain, A. K., & Gupta, B. B. (2022). A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, 16(4), 527-565. https://doi.org/10.1080/17517575.2021.1896786. Erişim Tarihi: Mart 14, 2025.

Jakobsson, M., & Myers, S. (2007). Delayed password disclosure. *ACM SIGACT News*, 38(3), 56-75. https://dl.acm.org/doi/abs/10.1145/1324215.1324228. Erişim Tarihi: Mart 14, 2025.

Kamiya, S., Kang, J., Kim, J., Milidonis, A., & Stulz, R. (2019). Risk Management, Firm Reputation, and the Impact of Successful Cyberattacks on Target Firms. *Information Systems & Economics eJournal*. https://doi.org/10.2139/ssrn.3135514. Erişim Tarihi:

Mart 14, 2025.

Ömer, A., Serkant, S., Aktuğ, M., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics, 12*(6), 1333-1333. https://doi.org/10.3390/electronics12061333. Erişim Tarihi: Mart 14, 2025.

Sharif, M., & Mohammed, M. (2022). A literature review of financial losses statistics for cyber security and future trend. *World Journal of Advanced Research and Reviews*. https://doi.org/10.30574/wjarr.2022.15.1.0573. Erişim Tarihi: Mart 14, 2025.

Türkiye İstatistik Kurumu. (2023). Hanehalkı Bilgi Teknolojileri (BT) Kullanım Araştırması. https://data.tuik.gov.tr/Bulten/Index?p=Hanehalki-Bilisim-Teknolojileri-(BT)-Kullanim-Arastirmasi-2023-49407 Erişim Tarihi: Mart 14, 2025.

Ünver, M., & Mirzaoğlu, A. G. (2011). Yemleme ("Phishing"). Bilgi Teknolojileri ve İletişim Kurumu. https://www.academia.edu/24841831/Yemleme_Phishing Erişim Tarihi: Mart 14, 2025.

Varshney, G., Misra, M., & Atrey, P. K. (2016). A survey and classification of web phishing detection schemes. *Security and Communication Networks*, 9, 6266–6284. https://doi.org/10.1002/sec.1674 Erişim Tarihi: Mart 14, 2025.

Wassermann, S., Meyer, M., Goutal, S., & Riquet, D. (2023). Targeted Attacks: Redefining Spear Phishing and Business Email Compromise. *ArXiv*, abs/2309.14166. https://doi.org/10.48550/arXiv.2309.14166. Erişim Tarihi: Mart 14, 2025.

Wu, L., Du, X., & Wu, J. (2015). Effective Defense Schemes for Phishing Attacks on Mobile Computing Platforms. *IEEE Transactions on Vehicular Technology,* 65, 1-1. https://doi.org/10.1109/TVT.2015.2472993 Erişim Tarihi: Mart 14, 2025.

Akpınar, Y., & Karal, Y. (2022). Mobil platformlarda HTML/DOM tabanlı oltalama tespiti üzerine bir inceleme. Bilgi Güvenliği Dergisi, 13(1), 22-36.

Ahmad, M., et al. (2022). Privacy-Preserving Phishing Email Detection Based on Federated Learning and LSTM. ResearchGate.

Çolhak, C., et al. (2024). Mobil Cihazlarda HTML/DOM Analizine Dayalı Oltalama Tespiti. [PDF].

Dhanavanthi, S., et al. (2021). Mobile phishing detection using deep learning on resource-constrained devices. Scientific Reports, 11(1), 1234.

Dou, W., et al. (2017). Phishing Webpage Detection via Multi-Modal Integration of HTML DOM Graphs and URL Features. MDPI.

Gupta, A., et al. (2020). Visual Similarity Based Phishing Detection on Mobile. International Journal of Cyber Security and Digital Forensics, 9(1), 45-58.

Hard, A., et al. (2018). Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604.

Islam, M. R., et al. (2020). HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Approaches. ResearchGate.

Jain, A., & Gupta, B. (2022). A Systematic Literature Review on Phishing Website Detection Techniques. ResearchGate.

Jain, A., et al. (2023). Phishing webpage detection via multi-modal integration of HTML DOM graphs and URL features based on graph convolutional and transformer networks. Electronics, 13(16), 3344.

Li, Y., et al. (2023). Fed-urlBERT: Federated learning based phishing URL detection. arXiv:2312.03636.

Mohammad, R. M., et al. (2014). Phishing website detection using URL and HTML features. Information Security Journal: A Global Perspective, 23(3), 112-123.

Ndibwile, A., et al. (2023). UnPhishMe: Phishing Attack Detection by Deceptive Login Simulation through an Android Mobile App. ResearchGate.

Rao, N., et al. (2023). HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning-based Approaches. Electronics, 13(16), 3344.

Routhu, S., et al. (2019). PhishDump: A Multi-Model Ensemble Based Technique for the

Detection of Phishing Sites in Mobile Devices. Scientific Reports.

Sun, J., et al. (2022). A Survey of Machine Learning-Based Solutions for Phishing Website Detection. MDPI.

Thapa, C., et al. (2023). Evaluation of federated learning in phishing email detection. Sensors, 23(9), 4346.

Zhang, Y., et al. (2007). CANTINA: A Content-Based Approach to Detect Phishing Websites. Carnegie Mellon University.