URL Tabanlı Zararlı Web Sitesi Tespiti: Derin Öğrenme ve Makine Öğrenimi Yöntemleriyle Bir Yaklaşım

Halil İbrahim Kaya[1], *Ali Çalhan*[2], *Murtaza Cicioğlu*[3]

[1]Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 81620, Düzce, Türkiye

[2] Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 81620, Düzce, Türkiye

[3] Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 16059, Düzce, Türkiye

1 2 3

halil213058@ogr.duzce.edu.tr, alicalhan@duzce.edu.tr

İletişim yazarı telefon no: +90 541 476 37 54

URL-Based Malicious Website Detection: An Approach Using Deep Learning and Machine Learning Methods

Halil İbrahim Kaya[1], *Ali Çalhan*[2], *Murtaza Cicioğlu*[3]

[1]Department of Computer Engineering, Engineering Faculty, Duzce University, 81620, Duzce, Turkiye

[2]Department of Computer Engineering, Engineering Faculty, Duzce University, 81620, Duzce, Turkiye

[3]Department of Computer Engineering, Bursa Uludağ University, 16059, Bursa, Turkiye

1 2 3

halil213058@ogr.duzce.edu.tr, alicalhan@duzce.edu.tr

Phone of contact author: +90 541 476 37 54

URL Tabanlı Zararlı Web Sitesi Tespiti: Derin Öğrenme ve Makine Öğrenimi Yöntemleriyle Bir Yaklaşım

Halil İbrahim Kaya, Ali Çalhan, Murtaza Cicioğlu

Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, 81620, Düzce, Türkiye

Özet

Siber tehditler, artan çeşitlilik ve karmaşıklıkla kullanıcıları hedef alarak maddi ve manevi kayıplara yol açmaktadır. Bu tehditlerin başında gelen zararlı bağlantılar (URL - Uniform Resource Loader), kullanıcıların kişisel ve finansal bilgilerini ele geçirmeye yönelik en yaygın güvenlik ihlallerinden biridir. Literatürde, zararlı URL tespitinde genellikle web sitesi içerikleri, yönlendirmeler, görseller, meta veriler ve site trafiği gibi ek veri kaynakları kullanılırken, bu çalışmada URL'nin yapısal özelliklerine dayalı bir yöntem benimsenmiştir. Yapay Sinir Ağları (ANN), Evrişimsel Sinir Ağları (CNN), Uzun Kısa Vadeli Bellek (LSTM), Kapılı Tekrarlayan Birim (GRU), CatBoost, XGBoost ve Destek Vektör Makineleri (SVM) modelleri kullanılarak zararlı URL tespiti için karşılaştırmalı bir analiz yapılmıştır. Çalışmanın önemli katkılarından biri, yalnızca URL analiziyle **%98,88 doğruluk**, **%98,55 F2 skoru** ve **%98,21 AUC** değeri elde edilmesidir. CatBoost modeli, bu metriklerde en iyi performansı göstermiştir. Ayrıca, çalışmada alan adı yaşı özelliğinin etkisi de incelenmiş ve kritik bir rol oynadığı belirlenmiştir. Alan adı yaşının dahil edildiği modellerde doğruluk oranları önemli ölçüde artarken (CatBoost ile %98,88), bu özellik çıkarıldığında modelin doğruluğu ciddi bir düşüş göstermiştir (CatBoost ile %93,38). Bu durum, alan adı yaşının zararlı URL tespitinde önemli bir özellik olduğunu ortaya koymakta ancak yeni oluşturulmuş alan adlarının yanlışlıkla zararlı olarak sınıflandırılması riskini de beraberinde getirmektedir. Sonuç olarak bu çalışma, yalnızca URL yapısal özelliklerini kullanarak zararlı URL tespitinde yüksek doğruluk sağlayarak literatüre önemli bir katkıda bulunmuş, aynı zamanda alan adı yaşının model performansı üzerindeki etkisini net bir şekilde ortaya koymuştur.

Anahtar Kelimeler: Zararlı URL Tespiti, Makine Öğrenmesi, Derin Öğrenme, Siber Güvenlik.

Abstract

Cyber threats are increasingly diverse and complex, targeting users and causing both material and moral losses. Malicious links (URL - Uniform Resource Loader), one of the most common security breaches aimed at compromising users' personal and financial information, are among the most common threats. While the literature typically utilizes additional data sources such as website content, referrals, images, metadata, and site traffic to detect malicious URLs, this study utilizes a method based on URL structural features. A comparative analysis was conducted for malicious URL detection using Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), CatBoost, XGBoost, and Support Vector Machines (SVM) models. One of the key contributions of this study is the achievement of **98.88% accuracy, 98.55% F2-score, and 98.21% AUC** using URL analysis alone. The CatBoost model demonstrated the best performance in these metrics. The study also examined the impact of the domain age feature and found it to play a critical role. While models incorporating domain age showed significantly higher accuracy (98.88% with CatBoost), removing it significantly decreased model accuracy (93.38% with CatBoost). This demonstrates that domain age is a crucial feature in detecting malicious URLs, but it also introduces the risk of incorrectly classifying newly created domains as malicious. In conclusion, this study makes a significant contribution to the literature by achieving high accuracy in detecting malicious URLs using only URL structural features. It also clearly demonstrates the impact of domain age on model performance.

Keywords: Malicious URL Detection, Machine Learning, Deep Learning, Cyber Security.

1. Giriş

İnternet kullanımının küresel ölçekte yaygınlaşması, bilgiye erişimi kolaylaştırırken siber tehditlerin de çeşitlenmesine ve karmaşıklığının artmasına yol açmaktadır. Bu tehditlerin başında gelen oltalama saldırıları, zararlı URL'ler aracılığıyla kullanıcıları yanıltarak kredi kartı bilgileri, kimlik verileri ve e-posta adresleri gibi hassas bilgileri ele geçirme amacı taşır. Dolayısıyla zararlı URL'lerin erken ve doğru bir biçimde tespiti, bireysel kullanıcı güvenliğinden kurumsal ağ savunmasına kadar tüm siber güvenlik katmanlarında kritik bir gereksinimdir. Nitekim Tian ve arkadaşları (2025), "Zararlı URL'ler siber güvenlik ekosistemini kalıcı olarak tehdit etmekte, kullanıcıların gizli verilerini ifşa etmeye yönlendirmekte veya Zararlı yazılım bileşenleri dağıtarak hedef sistemlere sızmaktadır" tespitinde bulunarak bu tehditlerin sürekliliğine ve ciddiyetine dikkat çekmektedir [1].

Mevcut çalışmalarda söz konusu tehditleri belirlemek için genellikle web sayfası içeriği, yönlendirme zincirleri, görsel benzerlik analizi, meta veriler veya trafik takibi gibi içerik-odaklı ve kaynak-ağır yöntemler tercih edilmektedir. Her ne kadar bu yaklaşımlar yüksek doğruluk sağlayabilse de hem büyük hesaplama maliyeti doğurmakta hem de gerçek zamanlı koruma gerektiren senaryolarda gecikme yaratmaktadır. Makine öğrenmesi ve derin öğrenme tabanlı modeller, son yıllarda bu soruna çözüm sunmak üzere öne çıksa da çoğu çalışma hâlen ek veri kaynaklarına bağımlı kalarak hafif ve hızlı çözümler üretmede sınırlı kalmıştır [3–6].

Bu makalede, yalnızca URL'nin yapısal özelliklerine dayanan, dolayısıyla hem hafif hem de gerçek zamanlı uygulamalara uygun bir zararlı URL tespit yaklaşımı sunulmaktadır. Karakter dağılımı, özel sembol dizileri, alt alan adı yapısı, URL uzunluğu ve gerektiğinde alan adı yaşı gibi özellikler çıkartılarak SVM, ANN, CNN, LSTM, GRU, CatBoost ve XGBoost algoritmalarının performansları sistematik olarak karşılaştırılmıştır. Çalışmanın devamı şu şekilde organize edilmiştir: **2. Bölüm** ilgili literatürü ve mevcut yöntemleri özetlemekte; **3. Bölüm** veri seti, özellik çıkarımı ve model yapılandırmalarını ayrıntılandırmakta**; 4. Bölüm** deneysel tasarım ve elde edilen sonuçları sunmakta**; 5. Bölüm** ise bulguları tartışarak makaleyi sonlandırmaktadır.

2. İlgili Çalışmalar

Phishing (**oltalama**) saldırıları, kullanıcıların internet üzerindeki bilinçsiz davranışlarından ve yetersiz bilgi düzeylerinden faydalanan bir yöntemdir. Örneğin, saldırganlar sahte çevrimiçi işlem siteleri oluşturarak kullanıcıları bu sitelere giriş yapmaya ikna eder ve kredi kartı numaraları, e-posta adresleri gibi kişisel bilgilerini çalarlar. Bu bağlamda, çeşitli çalışmalar oltalama saldırılarının tespit edilmesi için makine öğrenimi ve derin öğrenme tabanlı modeller önermiştir.

R. Jayaraj ve arkadaşlarının çalışmasında, URL özellikleri hibrit bir öznitelik seçimi yöntemi (HEFS, Hybrid Ensemble Feature Selection) kullanılarak çeşitli algoritmaların performansı karşılaştırılmıştır. Önerilen model, %97,8 doğruluk, %98,2 hassasiyet, %98,17 özgüllük ve %97,6 F1 skoru elde etmiş; eğitim süresi 0,416 sn, test süresi ise 0,02 sn olarak raporlanmıştır [8]. Dam Minh Linh ve arkadaşları tarafından yayımlanan çalışmada 651.191 URL içeren büyük bir veri seti kullanılarak CNN tabanlı bir tarayıcı uzantısı geliştirilmiş ve %98,4 doğruluk oranına ulaşılmıştır [9]. Ariyadasa ve arkadaşları tarafından yürütülen çalışmada, URL ve HTML özellikleri birleştirilmiş hibrit modelde %98,34 doğruluk oranı elde edilmiştir [14]. Sindhu ve arkadaşlarının çalışmasının değerlendirmesinde, Random Forest algoritması %96,7 doğruluk oranı ile öne çıkmıştır [19]. Yao ve arkadaşları tarafından önerilen "Deep learning for phishing detection" adlı Faster R-CNN tabanlı yaklaşım, küçük nesne tanıma konusunda odaklanarak URL'lerin meşruiyetini görsel analizlerle değerlendirmiştir. [13].

Sonuç olarak, oltalama saldırılarının tespiti üzerine yapılan çalışmalar hem makine öğrenimi hem de derin öğrenme yaklaşımlarının yüksek doğruluk oranları ile etkili bir şekilde kullanılabileceğini göstermektedir. Ancak, önerilen yöntemlerin eğitim süreleri ve karmaşıklıkları dikkate alınarak uygulamalara özgü modellerin seçilmesi önemlidir. Diğer çalışmalara ait bazı sonuçlar Tablo 1'de verilmiştir.

Tablo 1: Konu ile ilgili diğer çalışmalar

Araştırmacı	Amaçlar	Algoritma	Veri Seti	Doğruluk
Yao et al. 2018 [13]	URL'nin iki boyutlu koddaki meşruiyetini tespit etmek için göreceli bir algılama yöntemi önerdi.	Faster R-CNN	FlickrLogos-32	Küçük boyutlu nesne tanıma konusunda iyi performans.
Ariyadasa et al. 2020 [14]	LSTM Tekrarlayan Sinir Ağları (RNN) kullanarak oltalama web siteleri için yeni bir tespit sistemi tasarlandı.	LSTM - CNN	Zararlı ve legal web siteleri	Doğruluk %98,34
Yang et al. 2019 [15]	Çok boyutlu özelliklere dayalı oltalama tespiti için derin öğrenme (MFPD) önerildi.	CNN - LSTM	Zararlı ve geçerli URL'ler	Doğruluk %98,99
Huang et al. 2019 [16]	Verimli ve etkili bir kapsül tabanlı sinir ağı önerildi.	CNN- Capsule NN	Geçerli ve taranmış zararlı URL'ler	Mükemmel performans elde edildi.

Ali et.al. 2019 [17]	Hibrit bir oltalama web sitesi yaklaşımı önerildi, özellik seçimi ve ağırlıklandırma ile birlikte.	DNN	UCI phishing websites	Doğruluk %89,50
Sahingoz et.al. [18]	Makine öğrenimi tabanlı algoritmalar, Yapay Sinir Ağları (ANN'ler) ve Derin Sinir Ağları (DNN'ler) önerilmiştir.	ANN - DNN	Zararlı ve geçerli URL'ler	ANN için Doğruluk %92 ve DNN için %96

Çalışmamızın ana katkıları şunlardır;

- **Sadece URL Analizi ile Yüksek Performans**: Çalışmamız, yalnızca URL özelliklerini kullanarak zararlı web sitelerini tespit etmede %98'in üzerinde doğruluk oranına ulaşmıştır. Bu, ek veriler (içerik analizi veya görsel özellikler) kullanmadan elde edilen önemli bir başarıdır.
- **Gelişmiş Algoritmaların Kullanımı**: Çalışmamızda ANN, CNN, LSTM, GRU, CatBoost, XGBoost ve SVM gibi hem derin öğrenme hem de ağaç tabanlı gelişmiş makine öğrenmesi algoritmaları kullanılmıştır. Bu sayede geniş bir model yelpazesi ile karşılaştırmalı performans analizi yapılmıştır.
- **Alan Adı Yaşı Olmadan Model Geliştirme**: Literatürde yaygın olarak kullanılan "alan adı yaşı" özelliği, büyük firmaların yeni oluşturulmuş alan adlarını tanıma sorunlarını beraberinde getirmektedir. Çalışmamızda bu özellik kullanılmadan model performansı değerlendirilmiş ve başarılı sonuçlar elde edilmiştir.
- **Yeni URL Özelliklerinin Entegrasyonu**: Mevcut çalışmalardan farklı olarak yeni URL özellikleri eklenerek modelin tespit kapasitesi iyileştirilmiş ve daha hassas sonuçlar elde edilmiştir.
- **Performans**: Kullanılan modeller, kısa eğitim ve test süreleriyle yüksek doğruluk sağlamış ve bu durum, özellikle gerçek zamanlı uygulamalar için modelin etkinliğini göstermiştir.
- **Model Karşılaştırması**: Çalışmamızda farklı öğrenme yöntemlerinin (ANN, CNN, LSTM, GRU, CatBoost, XGBoost ve SVM) performansı detaylı bir şekilde incelenmiş ve hangi modelin en iyi sonucu verdiği belirlenmiştir. Bu durum, ileride yapılacak çalışmalar için rehber niteliğinde bir katkı sağlamaktadır.
- **Gerçek Dünya Uygulama Potansiyeli**: Yüksek doğruluk oranı, kısa işlem süreleri ve sadece URL analizine dayalı yapısı sayesinde çalışmamız, zararlı URL tespiti için pratikte uygulanabilir bir çözüm sunmaktadır.

3. Zararlı Web Sitelerinin Tespiti için Mevcut Yöntemler

Zararlı web sitelerinin tespiti, siber güvenlik alanında önemli bir araştırma konusu olmuştur. Geleneksel yöntemler genellikle kara liste (blacklist) kullanımı, imza tabanlı analiz veya manuel inceleme gibi tekniklere dayanır [21]. Ancak bu yöntemler, sürekli değişen tehdit ortamına uyum sağlamakta yetersiz kalmaktadır. Örneğin, zararlı siteler, kara listeleme sistemlerini atlatmak için alan adlarını sıkça değiştirmekte ve bu durum, geleneksel yöntemlerin etkinliğini azaltmaktadır.

Son yıllarda, URL tabanlı analiz teknikleri ve makine öğrenmesi yaklaşımları zararlı web sitelerinin tespitinde önemli bir rol oynamaya başlamıştır [22]. URL tabanlı analiz, bir web sitesinin içeriğine erişmeden yalnızca URL'nin yapısal özelliklerine odaklanarak hızlı ve maliyet etkin bir analiz sunmaktadır. Bu yaklaşımda kullanılan yaygın özellikler arasında karakter uzunluğu, özel sembollerin kullanımı (ör. '-', '_'), büyük/küçük harf dağılımı, alan adı yaşı, popülerlik düzeyi, üst seviye alan adı (.com, .net) gibi faktörler yer almaktadır. Bunun yanı sıra, URL'nin belirli bölümlerinin (ör. path, query string) analiz edilmesiyle, saldırganların oluşturduğu ortak kalıplar tespit edilebilmektedir.

Makine öğrenmesi ve derin öğrenme tabanlı yaklaşımlar ise daha dinamik ve akıllı bir tespit süreci sunmaktadır. Klasik makine öğrenmesi modelleri olan karar ağaçları, destek vektör makineleri (SVM) ve lojistik regresyon gibi yöntemler, URL'nin belirli özelliklerini kullanarak zararlı olup olmadığını sınıflandırmaktadır. Daha gelişmiş teknikler arasında yer alan gelişmiş ağaç tabanlı modeller (ör. Random Forest, XGBoost, CatBoost) yüksek doğruluk oranlarıyla öne çıkmaktadır. Öte yandan, derin öğrenme tabanlı modeller (CNN ve Recurrent Neural Networks - RNN), URL'leri doğrudan analiz ederek özdevimli özellik çıkarımı yapabilmekte ve daha karmaşık saldırı kalıplarını tespit etmede avantaj sağlamaktadır.

Günümüzde yapılan araştırmalar, geleneksel yöntemlerin eksiklerini gidermek adına, URL tabanlı analiz teknikleriyle makine öğrenmesi ve derin öğrenme yöntemlerinin bir araya getirildiği hibrit yaklaşımların daha yüksek başarı sağladığını göstermektedir [23].

4. Yöntem

Bölüm 3'te sunulan yöntemler, zararlı URL'lerin otomatik olarak tespit edilmesinde hem hız hem de esneklik açısından önemli avantajlar sunmaktadır. Bu yaklaşımlar, tehdit ortamındaki dinamik değişimlere uyum sağlama kabiliyetleri sayesinde klasik yöntemlerin ötesine geçmektedir.

Bu çalışmada kullanılan veri seti, tamamen tarafımızdan üretilmiş ve ilk defa bu çalışmada kullanılmıştır[20]. Veri seti, en çok ziyaret edilen güvenli web sitelerine ait URL'ler **Majestic 1 Milyon** listesinden, zararlı URL'ler ise USOM (**Ulusal Siber Olaylara Müdahale Merkezi**) kaynaklarından derlenerek oluşturulmuştur[1-2]. Literatürde oldukça nadir rastlanan bu iki güvenilir kaynağın bir araya getirilmesiyle oluşturulan bu özel veri seti, alanında öncü bir katkı niteliği taşımaktadır. Toplamda 22.708 URL içeren veri setinde, zararlı URL sayısı 9404 iken zararsız URL sayısı 13304 olarak belirtilmiştir. Bu durum, sınıflandırma algoritmalarının dengeli biçimde eğitilmesine olanak tanıyarak model başarımını doğrudan olumlu yönde etkilemektedir.

4.1. Öznitelik Çıkarımı

Bu çalışmada zararlı URL'lerin tespiti amacıyla kullanılan özellikler, literatürde sıkça kullanılan yöntemlerle birlikte bu çalışmaya özgü yeni özellikler barındırmaktadır. Kullanılan özellikler Tablo 2'de verilmiş ve açıklanmıştır.

Tablo 2: Literatürde yaygın olarak kullanılan özellikler

URL Uzunluğu	URL'nin toplam karakter sayısı. Uzun URL'lerin zararlı olma olasılığı üzerine analizler yapılmıştır.
TLD (Üst Seviye Alan Adı)	URL'nin ".com", ".net", ".org" gibi üst seviye alan adlarının sınıflandırılması.
Subdomain Sayısı	URL'deki subdomain sayısının belirlenmesi, zararlı URL'lerde subdomain kullanımının yaygınlığı nedeniyle önemlidir.
Domain Yaşı	Alan adının yaşını gün cinsinden ölçerek, sahte alan adlarının yeni olma olasılığı üzerine odaklanılmıştır.
Özel Karakter Sayısı	URL'deki toplam özel karakter sayısı ve karakterlerin türlerine göre frekansı.
Rakam Sayısı	URL'deki toplam rakam sayısı.
Entropy	URL'nin karmaşıklık düzeyini ölçmek için kullanılmıştır. Daha yüksek karmaşıklık zararlı URL'lerde yaygın bir özelliktir.

Tablo 2'de verilen ve literatürde yaygın olarak kullanılan öznitelikler zararlı URL tespitinde başarılı sonuçlar vermektedir. Ancak bu çalışmada, yalnızca mevcut yöntemlerle yetinilmemiş, daha güçlü modeller geliştirmek ve URL analizine derinlik kazandırmak amacıyla çalışmaya özgü yeni öznitelikler önerilmiştir. Eklenen bu özgün özellikler, zararlı URL'lerde daha sık rastlandığı gözlemlenen fakat önceki çalışmalarda çoğunlukla göz ardı edilen niteliklerdir. Bu sayede, modelin zararlı URL'leri daha hassas biçimde ayırt edebilmesi sağlanmış ve geleneksel yaklaşımların ötesine geçen bir performans elde edilmiştir.

Tablo 3: Bu çalışmaya özgü öznitelikler

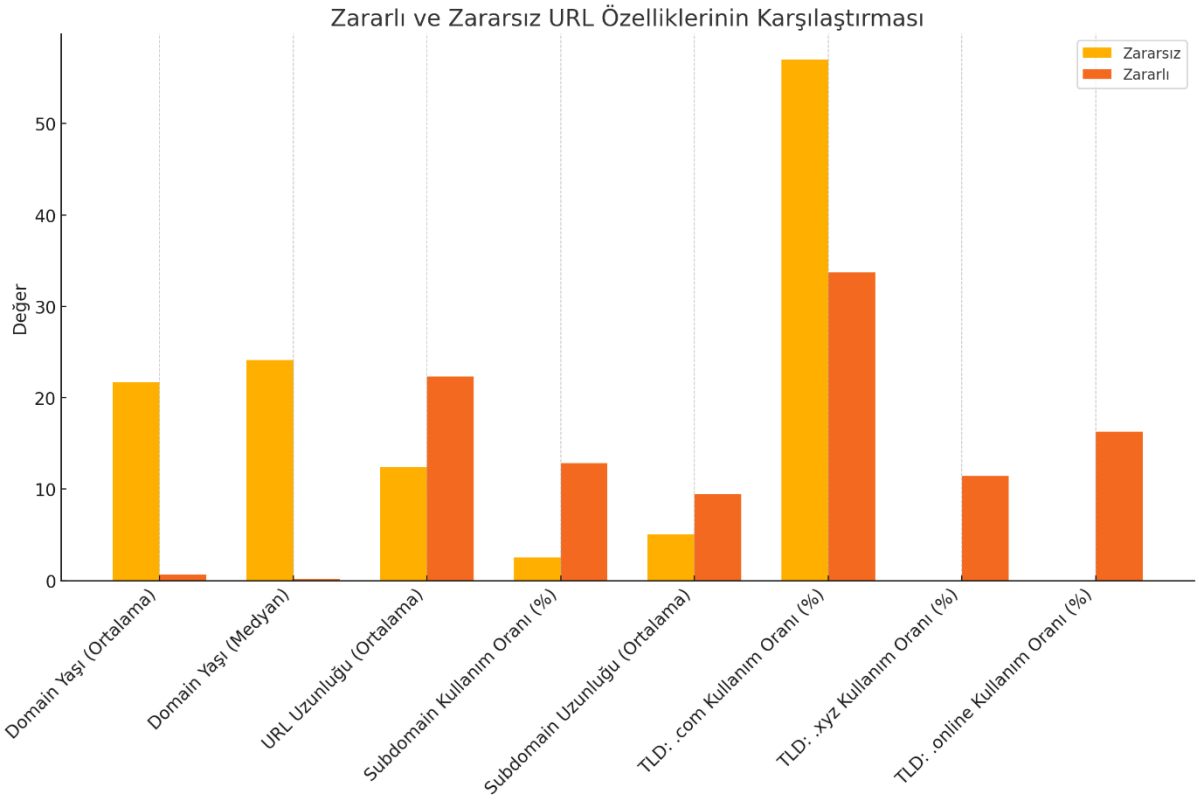
Harf Frekansı	URL'deki her harfin (ör. x, w, q gibi zararlı URL'lerde sıkça rastlanan harfler) toplam sayısı.
Yan Yana Harfler	URL'deki yan yana gelen aynı harflerin toplam sayısı, özellikle zararsız URL'lere göre farklılık gösterebilir.
Sesli ve Sessiz Harf Dizileri	Yan yana gelen sesli ve sessiz harflerin toplam uzunlukları.
Subdomain Uzunluğu	Subdomain'in uzunluğu ve adı üzerinde yapılan analizler, zararlı URL'lerin genellikle daha uzun ve karmaşık subdomain yapılarında olması göz önüne alınarak gerçekleştirilmiştir.
Oranlar	URL içerisindeki harflerin, rakamların ve özel karakterlerin birbirine oranı. **Harf Oranı**: Harflerin toplam uzunluğa oranı. **Rakam Oranı**: Rakamların toplam uzunluğa oranı. **Özel Karakter Oranı**: Özel karakterlerin toplam uzunluğa oranı.

Bu çalışmaya özgü olarak geliştirilen ve yenilikçi nitelik taşıyan özellikler **Tablo 3**'te sunulmaktadır. Literatürde yaygın olarak yer almayan bu özellikler, modelin zararlı URL'leri daha isabetli tespit etmesine katkı sağlamıştır. İlk olarak, her bir harfin

URL içerisinde kaç kez geçtiği sayısal olarak hesaplanmış ve bazı karakterlerin belirli bir eşiğin üzerinde tekrarlanmasının zararlı URL'lerde anlamlı bir ayırt edici özellik taşıdığı gözlemlenmiştir. Özellikle, sahte web sitelerinin orijinal adresleri taklit etmeye çalıştığı durumlarda, yan yana tekrarlanan harf kullanımı yaygın olarak tespit edilmiş ve bu durum model için önemli bir sinyal olarak değerlendirilmiştir. Ayrıca sesli ve sessiz harflerin URL içerisindeki sayısal dağılımı da incelenmiş; bu iki tür harfin oranında gözlenen anormal farklılıkların zararlı URL'lerin belirlenmesinde etkili olduğu sonucuna varılmıştır. URL içerisindeki rakam ve özel karakterlerin toplam karakter sayısına oranı da dikkate alınmış, bu oranların normal harf kullanımına göre olağan dışı seviyelere çıkmasının zararlı URL'ler için belirleyici bir özellik olduğu ortaya konmuştur. Bu özgün özellikler, sadece geleneksel metriklerle değil, aynı zamanda metin içerisindeki yapısal ve semantik anomalileri de dikkate alarak zararlı URL'leri tespit etmede daha güçlü bir yaklaşım sunmuştur.

Özelliklerin Genel Özeti:

Öznitelik oluşturma sürecinde, yalnızca literatürde yaygın olarak kullanılan geleneksel URL metrikleriyle sınırlı kalınmamış, modele özgün ve ayırt edici nitelikler kazandıracak yeni öznitelikler de geliştirilmiştir. Öncelikle, geleneksel metrikler olarak URL uzunluğu, özel karakter frekansı, sayı oranı, domain yaşı gibi temel özellikler kullanılmıştır. Bunlara ek olarak, özellikle USOM kaynaklı zararlı URL'lerde sıkça karşılaşılan harf frekansları (örneğin x, q, w gibi harflerin kullanım yoğunluğu), URL içerisindeki tekrarlayan harf sayıları (örn. haliil.com gibi manipüle edilmiş alan adları), subdomain'in varlığı ve sayısı, subdomain uzunluğu, URL içerisindeki sesli/sessiz harflerin ardışık olarak tekrarlanma sıklığı gibi dilsel ve yapısal niteliklere dayalı yeni öznitelikler de modele dahil edilmiştir. Ayrıca, URL'deki harf, sayı ve özel karakterlerin birbirlerine oranları hesaplanarak karakter kompozisyonu hakkında ayrıntılı bilgi sunulmuştur. Entropy hesaplaması ile URL'nin rastgelelik derecesi ölçülmüş ve bu bilgi, modelin ayırt edici gücünü artırmada kullanılmıştır. TLD'ler (örneğin .com, .xyz) one-hot encoding yöntemiyle işlenerek, üst düzey alan adlarının sınıflandırmaya katkısı da değerlendirilmiştir. Bu öznitelikler, yapılan deneylerde model performansına önemli ölçüde katkı sağlamıştır. Örneğin, yalnızca geleneksel özelliklerle %94 civarında seyreden doğruluk oranı, yukarıda belirtilen özgün özniteliklerin dahil edilmesiyle %98 seviyesinin üzerine çıkarılmıştır. Bu sonuç, geliştirilen özellik setinin, zararlı URL tespitinde güçlü bir temsil yeteneğine sahip olduğunu ve modelin genelleme kapasitesini artırdığını göstermektedir.



Şekil 1: URL Öznitelik Karşılaştırması

Veri setimiz, zararlı URL tespiti alanında literatüre önemli katkılar sunacak şekilde, tamamen bu çalışma kapsamında oluşturulmuştur. Bu yönüyle ilk kez kullanılan özel bir veri seti niteliği taşımaktadır. Zararsız URL'ler Majestic Million listesinden, zararlı URL'ler ise USOM'dan alınmıştır. Literatürde yaygın olarak kullanılan açık kaynaklı, homojen ve çoğunlukla İngilizce içerikli veri setlerinin aksine, bu çalışmada kullanılan veriler çok dilli ve Türkiye odaklı kaynaklardan oluşturulmuş olup, zararlı sitelere ait örnekler gerçek dünyadaki saldırıların daha geniş bir yelpazesini temsil etmektedir. Bu sayede, geliştirdiğimiz modellerin pratikte uygulanabilirliği ve gerçek dünya verileri karşısındaki dayanıklılığı artırılmıştır.

Toplam 22.708 URL içeren veri setinde, zararlı ve zararsız örneklerin sayıları birbirine oldukça yakındır, bu da dengeli bir sınıflandırma problemi ortaya koymakta ve model eğitiminin istikrarlı olmasını sağlamaktadır. Veri seti üzerindeki detaylı istatistiksel analizler, zararlı ve zararsız URL'lerin belirli yapısal özellikler açısından ciddi farklılıklar barındırdığını ortaya koymaktadır. Örneğin, Şekil 1'de de görüldüğü üzere domain yaşı bakımından zararsız sitelerin oldukça köklü olduğu görülmektedir: zararsız sitelerin ortalama domain yaşı 21.70 yıl ve medyanı 24.14 yıl iken, zararlı sitelerde bu değer ortalama 0.64 yıl ve medyan 0.15 yıldır. Yani zararlı sitelerin büyük çoğunluğu çok yeni oluşturulmuştur ve geçici süreli saldırı amacıyla kullanılmaktadır. Zararlı sitelerin %91'inden fazlası bir yıldan genç alan adlarına sahiptir. Bu durum, domain yaşı özelliğinin önemli bir ayrıştırıcı sinyal taşıdığını göstermektedir. URL uzunluğu bakımından da dikkat çekici farklar gözlemlenmiştir. Zararlı URL'lerin ortalama uzunluğu 22.33 karakter iken zararsız URL'ler yalnızca ortalama 12.41 karakter uzunluğundadır. Ayrıca, zararlı URL'lerin önemli bir kısmı 21-30 karakter aralığında yoğunlaşırken, zararsız URL'lerin çok büyük bölümü (yaklaşık %94'ü) 10 ila 20 karakter aralığındadır. Bu da, saldırganların kullanıcıyı yanıltmak adına daha karmaşık ve dikkat dağıtıcı URL yapıları oluşturduklarını göstermektedir. TLD dağılımı da bu farkı destekler niteliktedir. Zararsız sitelerde en yaygın görülen TLD'ler .com (%57.01), .org (%12.70), .edu, .net ve .gov gibi kurumsal ve yaygın uzantılardan oluşurken; zararlı sitelerde .com (%33.69), .online (%16.27), .xyz (%11.45), .net ve .com.tr gibi daha az güvenilir ya da ücretsiz temin edilebilen TLD'ler öne çıkmaktadır. Özellikle .online ve .xyz gibi TLD'lerin zararlı URL'lerde sık görülmesi, saldırganların düşük maliyetli veya kolay erişilebilir domain uzantılarını tercih ettiğini göstermektedir. Subdomain analizine bakıldığında da benzer bir ayrışma görülmektedir. Zararlı URL'lerin %12.83'ü subdomain içermekteyken, zararsız URL'lerde bu oran yalnızca %2.51'dir. Saldırganlar, subdomain'leri genellikle kullanıcıda güven hissi yaratacak şekilde biçimlendirmekte ve meşru alan adlarını taklit eden yapılandırmalara yönelmektedir. Ayrıca, zararlı URL'lerdeki subdomain adları daha uzundur (ortalama 9.44 karakter), oysa zararsız sitelerde bu değer ortalama 5.06 karakterdir. Bu fark da sahte subdomain kullanımıyla ilgili bir ayırt edici faktör sunmaktadır. Sonuç olarak, veri setimizin sunduğu bu istatistiksel içgörüler, yalnızca URL analizine dayalı zararlı site tespitinin mümkün ve etkili olduğunu göstermekte; bu alanın sadece klasik domain adı kontrolleri veya içerik analizi ile sınırlı olmadığını ortaya koymaktadır. Geliştirdiğimiz veri seti hem Türkçe içerik barındırması hem de daha önce kullanılmamış kaynakları birleştirmesi açısından özgün olup, gelecek çalışmalara açık ve dengeli bir temel sunmaktadır.

4.2. Kullanılan Modeller

Çalışmada, farklı türden yapay zekâ modelleri kullanılmıştır. Bu modeller: i) Derin Öğrenme Modelleri: LSTM, GRU, CNN ve ANN, ii) Gelişmiş Ağaç Yöntemleri: CatBoost ve XGBoost, iii) Klasik Makine Öğrenmesi Modeli: SVM olarak sıralanmaktadır.

Derin öğrenme modelleri, URL'lerin karakter dizilimlerinden çıkarılan karmaşık desenleri öğrenme yetenekleriyle öne çıkar. ANN, çok katmanlı yapısıyla temel sınıflandırma görevlerinde başarılı sonuçlar sunarken; CNN, karakter seviyesindeki bölgesel örüntüleri yakalayarak URL içindeki belirli harf ve sembol kalıplarını tespit etmede yüksek performans gösterir. Ardışık veri modelleri olan LSTM ve GRU ise URL'in zaman dizinli (sequential) doğasını kullanarak karakterler arasındaki uzun menzilli bağımlılıkları öğrenir; bu sayede, özellikle farklı alan adlarının ve parametrelerin tekrarlama veya manipülasyon kalıplarını başarıyla yakalayabilir.

Gelişmiş ağaç tabanlı yöntemler, gradyan artırma tekniğiyle zenginleştirilmiş karar ağaçlarının bir araya gelmesinden oluşur. XGBoost, hata fonksiyonunu iteratif olarak minimize eden optimizasyon stratejisi sayesinde hızlı eğitim ve yüksek doğruluk sunarken; CatBoost, kategorik değişkenlerin kodlanmasına getirdiği yenilikçi yaklaşım sayesinde özellikle ayrık URL özellikleriyle çalışırken ciddi aşırı öğrenme riskini engeller. Her iki model de URL'den çıkarılan yapısal ve istatistiksel özellikleri kullanarak güvenilir sınıflandırma sonuçları üretir.

Klasik makine öğrenmesi algoritmalarından SVM, özellikle iki sınıfı ayıran en geniş marjini bulma prensibiyle bilinir. URL özelliklerinin yüksek boyutlu uzayda ayrılabilirliğini artırarak, zararlı ve zararsız URL kümeleri arasındaki sınırı net bir şekilde tanımlar. Hafif ve yorumlanabilir bir model olan SVM, sınıflandırma performansını artırmak için çekirdek (kernel) fonksiyonlarıyla esneklik sağlar ve gerçek zamanlı uygulamalarda düşük hesaplama maliyetiyle tercih edilebilir.

4.3. Model Eğitimi ve Doğrulama

Derin öğrenme modelleri (ANN, CNN, LSTM, GRU) için Random Search yöntemiyle geniş çaplı bir hiperparametre optimizasyonu gerçekleştirilmiş; her iterasyonda öğrenme oranı, katman sayısı, birim sayısı ve dropout oranı gibi kritik parametreler test edilerek en iyi konfigürasyonlar belirlenip kaydedilmiştir. Ağaç tabanlı yöntemler CatBoost ve XGBoost'ta ise daha verimli bir arama için Optuna Kütüphanesi kullanılarak Bayes tabanlı hiperparametre optimizasyonu uygulanmıştır. Optuna, klasik genetik algoritmalara benzer şekilde "deneme" (trial) kavramını kullanarak her denemeyi bir popülasyon üyesi gibi değerlendirir; ancak arama stratejisini genellikle Bayesian optimizasyon altında çalışan Tree-structured Parzen Estimator (TPE) yöntemiyle yürütür. Arama uzayını tanımladıktan sonra Optuna, her denemede rastgele seçilen parametre setini eğitip doğrulama maliyeti (validation loss) üzerinden puanlar; başarılı denemelerin parametre kombinasyonlarından yola çıkarak sonraki denemelerde daha iyi adaylar oluşturur. Ayrıca, Optuna'nın otomatik budama mekanizması sayesinde uzun süren ve vaat etmeyen denemeler erken sonlandırılarak kaynak kullanımı optimize edilir. Böylece, CatBoost ve XGBoost için yüzlerce deneme hızlı ve verimli şekilde yürütülmüş, en iyi öğrenme oranı, ağaç derinliği ve regülasyon katsayıları gibi kritik hiperparametreler bulunmuştur. Tüm modeller, doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1 skoru ve F2

Skoru metrikleriyle sınanmış ve gelişmiş ağaç tabanlı yöntemler arasında Catboost'un en yüksek performansı gösterdiği gözlemlenmiştir.

Ağaç tabanlı modellerin eğitiminde model performansının daha güvenilir bir şekilde değerlendirilmesi için çapraz doğrulama (cross validation) yöntemi kullanılmıştır. Bu çalışmada tercih edilen yöntem K-Fold Cross Validation olup, veri kümesi rastgele eşit büyüklükteki k parçaya bölünerek uygulanmaktadır. Her iterasyonda, bir parça test kümesi olarak ayrılırken kalan parçalar eğitim kümesi olarak kullanılır ve süreç k kez tekrarlanır. Bu yöntem sayesinde modeller, tüm veri üzerinde eğitim ve test süreçlerine dahil edilerek, genelleştirme yetenekleri daha doğru ve güvenilir bir biçimde ölçülür.

K-Fold çapraz doğrulama yöntemi, veri setindeki olası önyargıları azaltarak, modelin aşırı öğrenme (overfitting) veya yetersiz öğrenme (underfitting) durumlarının daha iyi tespit edilmesini sağlar. Bu çalışmada veri seti 5 eşit parçaya bölünerek (5-Fold Cross Validation) her model için tüm veri kümesi üzerinde performans değerlendirmesi yapılmıştır. Böylece, model performansları sadece tek bir veri bölünmesine bağlı kalınmadan, daha tutarlı ve güvenilir şekilde belirlenmiştir. Bu kapsamlı değerlendirme yöntemi, özellikle domain yaşı gibi kritik özelliklerin modele etkisinin daha iyi analiz edilmesini sağlamış ve farklı veri alt kümelerindeki performans dalgalanmalarının önüne geçmiştir.

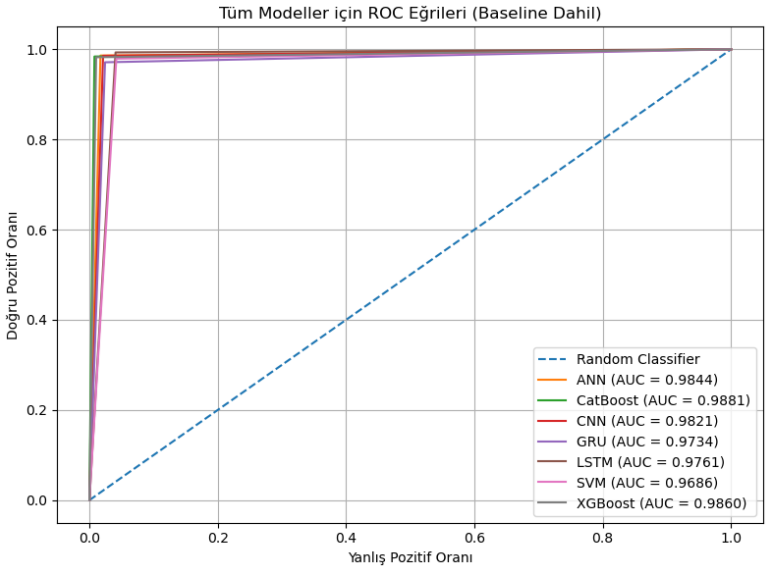
Modelleri eğitirken en büyük zorluklardan biri, domain yaşı küçük olan URL'lerin önyargılı biçimde zararlı sınıflandırılmasıydı. Bu önyargıyı ölçmek adına, domain yaşı içeren ve içermeyen iki ayrı model grubu oluşturduk; her iki grupta da performans karşılaştırmaları yaparak, yaş özelliğinin tespit başarısını ve yeni alan adlarının yanlış sınıflandırılma riskini titizlikle analiz ettik.

5. Yaş Dahil Edilen Modellerin Sonuçları

Bu çalışmada, zararlı URL tespitinde alan adı yaşı özelliğinin etkisini değerlendirmek amacıyla iki ayrı model eğitilmiştir: birincisi alan adı yaşı bilgisi dahil edilerek, diğeri ise bu özellik çıkarılarak oluşturulmuştur. Bu deneysel yaklaşımın temel amacı, URL tabanlı analizlerde sıkça kullanılan ancak her zaman erişilebilir veya güncel olmayan alan adı yaşı bilgisinin model performansı üzerindeki katkısını sayısal olarak ortaya koymaktır. Özellikle gerçek zamanlı tespit sistemlerinde, alan adı yaşına dair verilere erişim her zaman mümkün olmayabileceği gibi, bazı saldırganlar sahte veya önceden oluşturulmuş alan adları kullanarak bu özelliğin güvenilirliğini azaltabilmektedir. Bu nedenle aynı veri seti kullanılarak, önce domain yaşı özelliği dahil edilerek, ardından bu özellik çıkarılarak iki ayrı model eğitilmiş bu değişkenin genel doğruluk, F1 skoru gibi metriklere etkisi karşılaştırmalı olarak analiz edilmiştir. Böylece alan adı yaşı gibi dış kaynaklara bağımlı özelliklerin, modelin güvenilirliğini ve taşınabilirliğini nasıl etkilediği konusunda daha derinlemesine bir değerlendirme yapılması hedeflenmiştir. Ayrıca, modelin her yeni oluşturulan alan adını doğrudan zararlı olarak sınıflandırma eğilimini test etmek ve bu soruna yönelik olası bir çözüm geliştirmek amacıyla da domain yaşı özelliğinin etkisi değerlendirilmiştir. Böylece domain yaşı gibi dış kaynaklara bağımlı özelliklerin, modelin güvenilirliğini ve taşınabilirliğini nasıl etkilediği konusunda daha derinlemesine bir analiz yapılması hedeflenmiştir.

5.1 ROC Eğrileri ve AUC değerleri

Alıcı İşletim Karakteristiği (**ROC**) eğrisi, bir sınıflandırma modelinin farklı eşik değerlerinde doğru pozitif oranı (**TPR**) ile yanlış pozitif oranı (**FPR**) arasındaki ilişkiyi gösterir. TPR, gerçek zararlı URL'lerin model tarafından doğru şekilde tespit edilme oranını ifade ederken, FPR zararsız URL'lerin yanlışlıkla zararlı olarak sınıflandırılma oranını gösterir. ROC eğrisinin grafikteki köşeye yakın, dik çıkışlı yapısı hem yüksek TPR hem de düşük FPR elde edildiğinin işaretidir. Eğrinin altında kalan alan, yani AUC (**Area Under the Curve**) değeri, modelin genel ayrıştırma yeteneğini tek bir skora indirger. AUC, 0.5 ile 1 arasında değişir; 0.5 rastgele sınıflandırıcıyla eşdeğer performansı, 1 ise kusursuz sınıflandırmayı temsil eder. AUC hesaplaması, ROC eğrisi altındaki integralin numerik olarak değerlendirilmesiyle gerçekleştirilir.



Şekil 2: Algoritmaların ROC eğrileri ve AUC değerleri

Şekil 2'de yer alan ROC eğrileri, tüm modellerin zararlı ve zararsız URL'leri ayırt etmedeki başarısını görselleştirmektedir. Derin öğrenme mimarileri (ANN, CNN, LSTM, GRU), Random Search ile gerçekleştirilen geniş çaplı hiperparametre optimizasyonu sonucunda en iyi değerler kullanıldı. ANN için 0.9844, CNN için 0.9821, LSTM için 0.9761 ve GRU için 0.9734 AUC değerleriyle sonuçlanmıştır. Ağaç tabanlı yöntemler (CatBoost, XGBoost, SVM) ise Optuna'nın TPE tabanlı Bayes optimizasyonu ve erken sonlandırma mekanizmalarından yararlanılarak en uygun parametreler belirlendi. En iyi sonuçlar CatBoost'ta 0.9881, XGBoost'ta 0.9860 ve SVM'de 0.9686 AUC değerleri olarak elde edilmiştir. Mavi kesik çizgiyle gösterilen rastgele sınıflandırıcı referans çizgisi, doğru pozitif ve yanlış pozitif oranlarının eşit olduğu durumu simgelerken, tüm model eğrileri bu çizginin çok üzerinde kalarak güçlü ayrıştırma kabiliyetlerini ortaya koymaktadır. Eğrilerin dik ve oyuk görünümü, düşük yanlış pozitif oranlarında bile yüksek doğru pozitif oranlarına ulaşıldığını, yani zararlı URL'lerin neredeyse hiç kaçırılmadığını göstermekte; yüksek AUC değerleri ise hem Random Search hem de Optuna ile elde edilen ayarların, URL tabanlı özelliklere dayanan modelleri tüm eşik değerlerinde güvenilir kıldığını kanıtlamaktadır.

5.2. Genel Performans

Doğruluk, sınıflandırma modelinin doğru olarak sınıflandırdığı URL'lerin tüm tahminlere oranını gösterir. Bu metrik, modelin genel performansını değerlendirirken kullanılan temel ölçüttür. Yüksek doğruluk, modelin zararlı ve zararsız URL'leri genel olarak doğru şekilde sınıflandırdığını gösterirken, düşük doğruluk ise modelin çok sayıda hata yaptığını ifade eder.

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

Kesinlik, modelin zararlı olarak sınıflandırdığı URL'lerin ne kadarının gerçekten zararlı olduğunu ölçer. Yanlış pozitif sonuçları dikkate alarak, modelin tahmin doğruluğunu değerlendirmeye odaklanır. Yüksek kesinlik, modelin zararlı olarak işaret ettiği URL'lerin çoğunun gerçekten zararlı olduğunu gösterirken, düşük kesinlik modelin zararsız URL'leri zararlı olarak yanlış sınıflandırdığı anlamına gelir.

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

Duyarlılık, modelin zararlı URL'leri ne kadar iyi yakalayabildiğini ölçer. Kaçırılan tehditleri (yanlış negatif) dikkate alarak, modelin kapsamlılığını değerlendirir. Yüksek duyarlılık, zararlı URL'lerin büyük çoğunluğunun doğru bir şekilde tespit edildiğini ifade ederken, düşük duyarlılık ise modelin çok sayıda zararlı URL'yi kaçırdığı anlamına gelir.

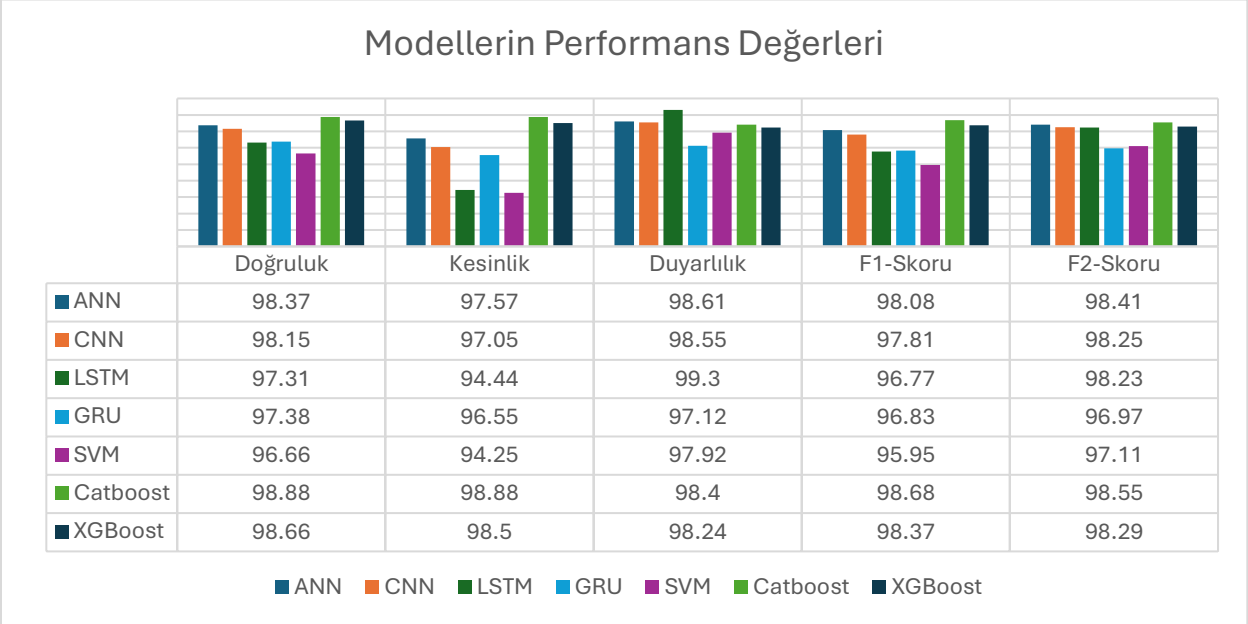
$$Duyarlılık = \frac{TP}{TP+FN} \quad (3)$$

F1 Skoru, kesinlik ve duyarlılık arasında denge sağlayan bir performans ölçütüdür. Bu iki ölçütün harmonik ortalamasını hesaplayarak, modelin hem doğru tahmin yapma yeteneğini hem de zararlı URL'leri yakalama başarısını birlikte değerlendirir. Yüksek F1 skoru, modelin hem az hata yaptığını hem de yüksek oranda tehdit yakaladığını gösterirken, düşük F1 skoru ya çok fazla yanlış alarm verildiğini (düşük kesinlik) ya da tehditlerin önemli bir kısmının kaçırıldığını (düşük duyarlılık) gösterir.

$$F1\ Skoru = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \quad (4)$$

F2 Skoru, duyarlılığa daha fazla ağırlık veren bir performans ölçütüdür. Özellikle zararlı URL tespiti gibi tehditlerin kaçırılmasının ciddi sonuçlar doğurabileceği senaryolarda kullanılır. F2 skoru, yanlış negatif sonuçları minimize etmeye odaklanır ve duyarlılığa kesinliğe oranla daha fazla önem verir. Yüksek F2 skoru, modelin özellikle zararlı URL'leri tespit etme konusunda başarılı olduğunu ve çok az tehdit kaçırdığını gösterir. Düşük F2 skoru ise modelin zararlı URL'leri kaçırma riskinin yüksek olduğunu belirtir.

$$F2\ Skoru = (1 + 2^2) * \frac{Kesinlik * Duyarlılık}{(2^2 * Kesinlik) + Duyarlılık} \quad (5)$$



	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	F2-Skoru
ANN	98.37	97.57	98.61	98.08	98.41
CNN	98.15	97.05	98.55	97.81	98.25
LSTM	97.31	94.44	99.3	96.77	98.23
GRU	97.38	96.55	97.12	96.83	96.97
SVM	96.66	94.25	97.92	95.95	97.11
Catboost	98.88	98.88	98.4	98.68	98.55
XGBoost	98.66	98.5	98.24	98.37	98.29

Şekil 3: Algoritmaların karşılaştırmalı performans sonuçları

Şekil 3'de gösterilen grafik, modellerin doğruluk, kesinlik, duyarlılık, F1 skoru ve F2 skoru açısından performanslarını karşılaştırmaktadır. Genel olarak incelendiğinde, tüm modellerin zararlı URL'leri tespit etmede yüksek performans sergilediği görülmektedir. Özellikle CatBoost modeli, 0.9888doğruluk, 0.9888kesinlik, 0.9884 duyarlılık, 0.9868 F1 skoru ve 0.9855 F2 skoru ile tüm metriklerde üstün performans göstererek diğer modeller arasında en başarılı sonucu vermiştir.

XGBoost ve ANN modelleri de yüksek performans sağlayarak CatBoost modeline oldukça yakın değerler elde etmişlerdir. Özellikle ANN'nin duyarlılık (0.9861) ve F1 skoru (0.9808) açısından dikkat çekici bir başarı sağladığı söylenebilir. CNN, GRU ve LSTM gibi diğer derin öğrenme modelleri de tatmin edici sonuçlar sunmuş fakat CatBoost, XGBoost ve ANN modellerinin biraz gerisinde kalmıştır. Destek Vektör Makineleri (SVM) ise tüm metriklerde diğer modellere kıyasla daha düşük değerler elde ederek görece daha zayıf performans sergilemiştir.

Ayrıca, F2 skorunun genel olarak yüksek olması, modellerin zararlı URL'leri kaçırmama konusunda başarılı olduklarını ortaya koymakta ve tehditleri yakalama konusunda yüksek duyarlılık sergilediklerini ifade etmektedir. CatBoost ve XGBoost'un yüksek kesinlik ve duyarlılık değerleri, hem yanlış pozitif sonuçların minimize edildiğini hem de zararlı URL'lerin etkin biçimde yakalandığını göstermektedir. Bu sonuçlar doğrultusunda, özellikle gelişmiş ağaç tabanlı yöntemlerin URL tabanlı zararlı web sitesi tespitinde son derece başarılı ve güvenilir bir çözüm sundukları söylenebilir.

5.3 Model Parametreleri

Yaş dahil edilen modelin en iyi sonuçlarını elde etmek için Random Search yöntemiyle optimize edilen **ANN** modelinin parametreleri Tablo 4 ve Tablo 5'te verilmiştir. Tablo 4'te sunulan ANN modelinde, her katmanda kullanılan birim sayısı (nöron sayısı), aktivasyon fonksiyonu, L2 regülasyonu ve dropout oranları görülmektedir. Birim sayıları, katmanların karmaşıklığı ve öğrenme kapasitesini belirler; genelde daha fazla birim, daha karmaşık örüntüleri yakalama imkânı sunar. Aktivasyon fonksiyonu olarak ReLU seçilmesi, ağın doğrusal olmayan karmaşık ilişkileri etkili şekilde öğrenmesini sağlar. L2 regülasyonu, aşırı öğrenme (overfitting) riskini azaltmak amacıyla kullanılır ve bu tabloda farklı katmanlarda değişen değerlere sahiptir; daha yüksek değerler, daha güçlü bir regularizasyon anlamına gelir. Dropout oranları (%10-%35 arasında) ise rastgele birimlerin eğitim sırasında devre dışı bırakılmasıyla modelin genelleme performansını artırır. Çıkış katmanında Sigmoid

fonksiyonu kullanılmıştır, bu da modelin binary (ikili) sınıflandırma yaptığını ifade eder. Bu parametre kombinasyonuyla oluşturulan model, güçlü öğrenme kapasitesini, aşırı öğrenmeye karşı alınan önlemlerle dengeleyerek yüksek performans sağlamayı amaçlamaktadır.

Tablo 4: ANN Modeli

	Birim Sayısı	Aktivasyon Fonksiyonu	L2 Regülasyonu	Dropout Oranı
İlk Katman	416	ReLU	0.025	0.2
İkinci Katman	384	ReLU	0.0002	0.35
Üçüncü Katman	304	ReLU	0.028	0.25
Dördüncü katman	32	ReLU	0.005	0.35
Beşinci katman	320	ReLU	0.0015	0.2
Altıncı katman	128	ReLU	0.0003	0.1
Yedinci katman	240	ReLU	0.008	0.3
Sekizinci katman	432	ReLU	0.002	0.15
Çıkış katmanı	1	Sigmoid		

Tablo 5'te modele ait diğer parametreler verilmiştir. Bu modelin ağırlık güncellemeleri RMSprop optimizasyon algoritmasıyla gerçekleştirilmiştir. RMSprop, parametre başına karekökü alınmış gradyan karelerinin üstel ortalamasıyla ölçeklenmiş adımlar kullanarak dengesiz gradyanlarda istikrarlı ve hızlı yakınsama sağlar. İlk öğrenme oranı 0,0037 olarak belirlenmiş olup bu değer, ağın hızlı fakat kararlı bir şekilde öğrenmesine imkân tanır. Momentum katsayısının 0,1 seçilmesi, geçmiş gradyan bilgisini kısmen güncel adıma ekleyerek optimizasyon yolunu pürüzsüzleştirir ve küçük yerel minimumlara takılma olasılığını azaltır. Rho parametresi 0,88 değerindedir; bu katsayı, gradyan karelerinin üstel hareketli ortalamasında güncel ve geçmiş bilginin dengeli biçimde harmanlanmasını sağlar. Sayısal kararlılığı artırmak için paydada 3×10^{-8} büyüklüğünde çok küçük bir epsilon sabiti kullanılmıştır. Ek olarak, Beta1 ve Beta2 değerlerinin sırasıyla 0,97 ve 0,999 olarak ayarlanması, RMSprop'a momentum benzeri bir bellek bileşeni kazandırarak hem ortalama gradyanı hem de gradyan karelerinin ortalamasını daha uzun süreli birikimle dengelemekte ve bu sayede ağırlık güncellemelerinin kararlılığını pekiştirmektedir. Tüm bu hiperparametre değerleri geniş bir arama uzayında random search yöntemi ile belirlenmiştir. Bu hiperparametre kombinasyonu, öğrenme sürecini hızlandırırken aşırı sapmaları ve sayısal dengesizlikleri en aza indirerek güvenilir bir yakınsama elde etmeyi amaçlar.

Tablo 5: Optimizasyon ve Öğrenme Parametreleri

Optimizasyon Fonksiyonu	RMSprop
Momentum	0.1
İlk Öğrenme Oranı	0.0037
Rho	0.88
Epsilon	3.0000000004e-08
Beta1	0.97
Beta2	0.999

Yaş parametresi dahil edilerek yapılan Random Search ile optimize edilen **CNN** modeline ait parametreler Tablo 6'da ve Tablo 7'de verilmiştir. Modelin parametreleri Tablo 6'da verilmiştir. Bu model, beş konvolüsyonel katman ve üç tam bağlantılı (dense) katmandan oluşan tipik bir CNN mimarisidir. İlk beş katman, (3 × 1) çekirdekli ReLU etkin filtreler, ardışık batch-normalization, belirli katmanlarda (2 × 1) maksimum havuzlama (max pooling) ve %10–%25 aralığında nöron atma (dropout) içererek karakter düzeyindeki kalıpları yakalar. Konvolüsyonel bloktan sonra gelen vektöre dönüştürme (flatten katmanı), çok boyutlu özellik haritalarını tek boyutlu vektöre dönüştürür. Bunu izleyen üç dense katman sırasıyla 128, 96 ve 224 nöronlu ReLU etkinimlidir; her biri toplu normalizasyon (batch-normalization) ve %10–40 arasında değişen nöron atma ile desteklenerek aşırı öğrenme engellenir. Tek nöronlu sigmoid çıkış katmanı, URL'nin zararlı olma olasılığını üretir. Bu katman dizilimi hem yerel karakter örüntülerini öğrenir hem de yüksek düzeyli temsiller oluşturarak ikili sınıflandırma için dengeli bir mimari sunar.

Tablo 6: CNN modeli parametreleri

	Filtre/Birim sayısı	Kernel Boyutu	Aktivasyon Fonksiyonu	Batch Normalization	Maxpooling	Dropout
Konvolüsyonel Katman 1	160	(3,1)	ReLU	Eklendi	(2,1)	0.25
Konvolüsyonel Katman 2	144	(3,1)	ReLU	Eklendi		0.15
Konvolüsyonel Katman 3	16	(3,1)	ReLU	Eklendi	(2,1)	0.1
Konvolüsyonel Katman 4	240	(3,1)	ReLU	Eklendi		0.15
Konvolüsyonel Katman 5	96	(3,1)	ReLU	Eklendi	(2,1)	0.1
Flatten Katmanı						
Tam Bağlantılı Katman 1	128		ReLu	Eklendi		0.2
Tam Bağlantılı Katman 2	96		ReLU	Eklendi		0.4
Tam Bağlantılı Katman 3	224		ReLU	Eklendi		0.1
Çıkış Katmanı	1		Sigmoid			

Optimizasyon ve öğrenme parametreleri Tablo 7'de verilmiştir. Modelin ağırlık güncellemeleri Adam optimizasyon algoritmasıyla gerçekleştirilmiştir. Başlangıç öğrenme oranı 0,009 olarak belirlenmiş, böylece ağın parametreleri hızlı bir şekilde ayarlanırken kararlılık korunmuştur. Sayısal taşmaları önlemek için çok küçük bir epsilon (6×10^{-8}) değeri kullanılmıştır. Momentum benzeri bellek katsayıları olan $\beta_1 = 0{,}909$ ve $\beta_2 = 0{,}9998$, sırasıyla ortalama gradyan ile gradyan karelerinin üstel hareketli ortalamalarını uzun süreli birikimle dengeleyerek daha pürüzsüz ve güvenilir bir yakınsama sağlamıştır.

Tablo 7: Optimizasyon ve Öğrenme Parametreleri

Optimizer	Adam
İlk Öğrenme Oranı	0.009
Epsilon	6e-08
Beta1	0.909
Beta2	0.9998

Yaş parametresi dahil edilerek yapılan Optuna ile optimize edilen **Catboost** modeline ait parametreler Tablo 8'de verilmiştir. Optuna ile hiperparametre araması sonucunda elde edilen CatBoost modeli, 1515 deneme ardından en uygun ayarlarla yapılandırılmıştır. Öğrenme hızı 0,12 olarak seçilerek modelin her ağaç eklemesinde parametre güncellemelerini dengeli şekilde gerçekleştirmesi sağlanmıştır. Ağaç derinliği 5 seviyede tutulmuş, böylece karmaşık örüntüler yakalanırken aşırı öğrenme riski sınırlanmıştır. L2 düzenleştirme katsayısının 12,03'e ayarlanması, yaprak skorlarına güçlü bir ceza uygulayarak modelin genelleme kabiliyetini artırmıştır. Rastgelelik düzeyi için rastgelelik gücü 3,66 ve bagging temperature 3,78 değerleri kullanılmış; bu iki parametre birlikte her iterasyonda farklı örnek ve özellik alt kümeleri seçerek çeşitliliği yükseltmiş ve stabil performans elde edilmesine katkı sağlamıştır. Son olarak, sınır sayısı 254 ile sayısal özelliklerin daha ince aralıklara bölünmesine izin verilmiş, bu da özellikle domain yaşı gibi sürekli değişkenlerin hassas eşiklerde değerlendirilmesini mümkün kılmıştır. Bu parametre bileşimi, yaş bilgisi dâhil edildiğinde CatBoost'un zararlı URL ayırımında yüksek doğruluk ve sağlamlık sergilemesine olanak tanımıştır.

Tablo 8: Model Parametreleri

İterasyon	1515
Öğrenme Oranı	0.12
Ağaç Derinliği	5
L2 Regülasyonu	12.03
Rastgelelik Gücü	3.66

Bagging Temperature	3.78
Sınır Sayısı	254

Yaş parametresi dahil edilerek yapılan Optuna ile optimize edilen **XGBoost** modeline ait parametreler Tablo 9'da verilmiştir. Seçilen XGBoost yapılandırması, modelin hem öğrenme kapasitesini hem de genelleme yeteneğini dengeleyen optimize edilmiş hiperparametrelere sahiptir. 1 385 ağaç içeren topluluk, karmaşık örüntüleri yakalamak için yeterli derinliğe sahiptir; her ağacın maksimum derinliği 10 olarak sınırlandığından aşırı uyum riski azaltılmıştır. 0,067'lik öğrenme oranı, her iterasyondaki güncellemeleri ölçülü tutarak stabil bir yakınsama sağlar. Min. Yaprak Ağırlığı 7 ve gamma değeri 0,589 değerleri, dallanma kararlarını kısıtlayarak yalnızca istatistiksel olarak güçlü bölünmelerin yapılmasına izin verir. Alt Örnek Oranı (0,829) ve Öznitelik Alt Örneği (0,785) oranları, her ağacı farklı örnek ve öznitelik alt kümeleriyle eğitip topluluğa rastgelelik katarak genelleme performansını artırır. Son olarak, L1 (0,924) ve L2 (8,44) düzenleştirme katsayıları, yaprak ağırlıklarını cezalandırarak modelin karmaşıklığını kontrol eder. Bu parametre seti, zararlı URL tespitinde yüksek doğruluk elde ederken aşırı öğrenmeyi önlemeye yönelik dengeli bir yapı sunmaktadır.

Tablo 9: Model Parametreleri

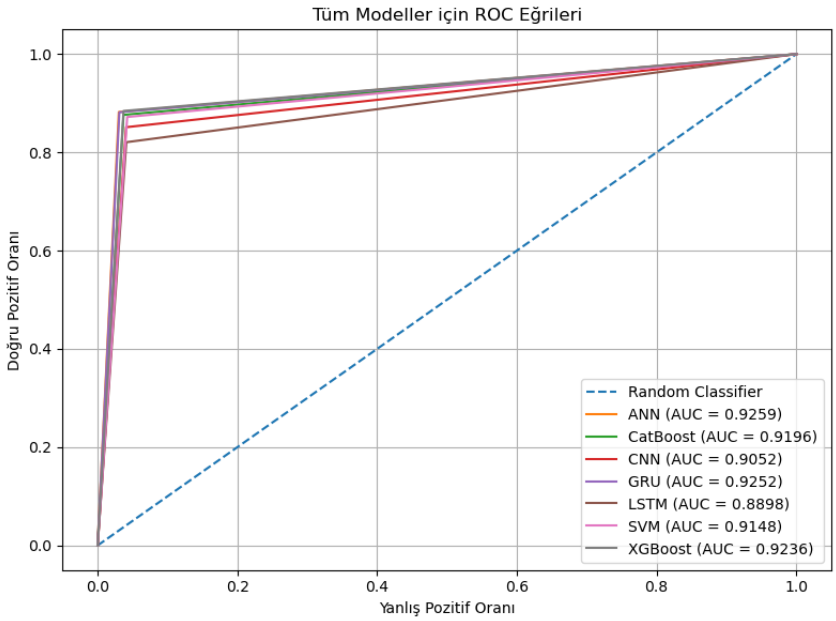
Ağaç Sayısı	1385
Öğrenme Oranı	0.06
Maksimum Derinlik	10
Min. Yaprak Ağırlığı	7
Gamma Değeri	0.58
Alt Örnek Oranı	0.82
Öznitelik Alt Örneği	0.78
L1 Regülasyonu	0.92
L2 Regülasyonu	8.44

6 Yaş Dahil Edilmeyen Modellerin Sonuçları

Domain yaşı özelliği çıkarılarak eğitilen model, yalnızca yapısal ve dilsel URL özniteliklerine dayanarak sınıflandırma yaptığı için, dış veri kaynaklarına bağımlılığı ortadan kaldırmakta ve gerçek zamanlı sistemler için daha taşınabilir, hızlı ve güvenilir bir alternatif sunmaktadır. Bu yaklaşım sayesinde, domain yaşı bilgisine erişimin mümkün olmadığı senaryolarda dahi etkin bir zararlı URL tespiti gerçekleştirilebilmiştir. Ayrıca, domain yaşı kullanılmadığında modelin, yalnızca yeni oluşturulmuş olması nedeniyle zararsız URL'leri hatalı biçimde "zararlı" olarak sınıflandırma eğilimi de azalmış; böylece hem meşru sitelere yönelik yanlış pozitiflerin önüne geçilmiş hem de sahte domain yaşlarıyla yapılan manipülasyonlara karşı daha dayanıklı bir yapı elde edilmiştir.

6.1 ROC Eğrileri ve AUC değerleri

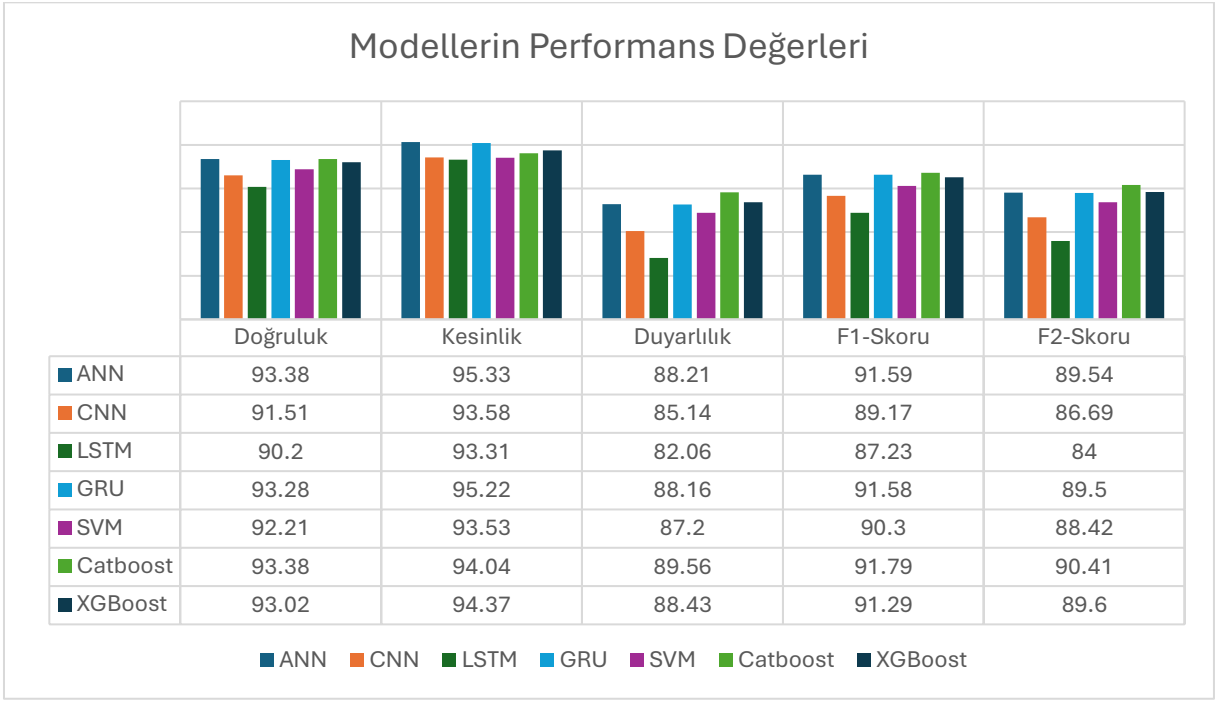
Şekil 4'de alan yaş bilgisi kullanılmadan eğitilen modellerin ROC eğrileri karşılaştırılmıştır. Tüm eğrilerin, rastgele sınıflandırıcıyı temsil eden kesikli referans çizgisinin üzerinde yer alması, modellerin zararlı ve zararsız URL'leri ayırt etme noktasında temel düzeyin belirgin biçimde üzerine çıktığını göstermektedir. Ancak alan adı yaşı devre dışı bırakıldığında ayrıştırma gücünün bir miktar azaldığı da dikkat çekmektedir. En yüksek ayırt edici performans ANN (0,9259) ve GRU (0,9252) modellerinde gözlenirken, XGBoost (0,9236) yakın bir performans sergilemiştir. CatBoost (0,9196) ile SVM (0,9148) orta sıralarda yer almış, CNN (0,9052) ve özellikle LSTM (0,8898) daha düşük ayırıcı güçle grafikte geride kalmıştır. Eğrilerin rastgele sınıflandırıcı çizgisine eski senaryoya kıyasla biraz daha yaklaşması, domain yaşı özelliğinin çıkarılmasının modellerin karar sınırlarını daralttığını ve bazı zararlı URL'lerin kaçırılma riskini arttırdığını teyit etmektedir. Bununla birlikte, en iyi üç modelin hâlâ %0,92'nin üzerinde AUC değerlerine ulaşması, yalnızca URL yapısal karakteristiklerine dayanarak bile kabul edilebilir düzeyde güvenilirlik sağlanabildiğini ortaya koymaktadır.



Şekil 4: Algoritmaların ROC eğrileri ve AUC değerleri

6.2 GENEL PERFORMANS

Şekil 5'de, alan adı yaşı özelliği çıkarıldıktan sonra eğitilen modellerin temel performans metrikleri karşılaştırılmaktadır. Yaş bilgisi olmaksızın en dengeli sonuçları veren yöntemlerin CatBoost ve ANN olduğu görülmektedir. CatBoost, %93,38 doğruluk ve %90,41 F2 skoru ile listedeki en yüksek duyarlılığı (%89,56) koruyarak zararlı URL'leri yakalama konusunda öne çıkmıştır; ANN ise benzer doğruluk (%93,38) düzeyine ulaşırken kesinlikte (%95,33) liderliği paylaşmış ve yüksek F1 (%91,59) değeriyle dengeli bir performans sergilemiştir. Ağaç tabanlı rakibi XGBoost da %93,02 doğruluk ve %89,60 F2 skoru ile CatBoost'a yakın bir başarı elde etmiştir.



	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	F2-Skoru
ANN	93.38	95.33	88.21	91.59	89.54
CNN	91.51	93.58	85.14	89.17	86.69
LSTM	90.2	93.31	82.06	87.23	84
GRU	93.28	95.22	88.16	91.58	89.5
SVM	92.21	93.53	87.2	90.3	88.42
Catboost	93.38	94.04	89.56	91.79	90.41
XGBoost	93.02	94.37	88.43	91.29	89.6

Şekil 5: Algoritmaların genel performansları

Derin öğrenme ailesinde GRU, %93,28 doğruluk ve %91,58 F1 skoru ile CNN ve LSTM'yi geride bırakmıştır; CNN'in duyarlılıktaki (%85,14) görece düşük değeri, bazı zararlı URL'leri kaçırma eğilimine işaret ederken, LSTM'nin duyarlılığı (%82,06) ve F2 skoru (84) bu özellik olmaksızın sınırlı kaldığını göstermektedir. SVM, doğruluk ve kesinlik metriklerinde orta sıralarda yer almakla birlikte, %88,42 F2 skoru ile derin öğrenme modellerine göre daha tutarlı bir denge yakalamıştır.

Genel eğilim, domain yaşı olmaksızın kesinlik değerlerinin yüksek kaldığını, ancak duyarlılığın özellikle CNN ve LSTM'de anlamlı biçimde düştüğünü ortaya koymaktadır. Bu sonuç, yaş bilgisinin bazı algoritmalar için kritik bir ayırt edici etmen olduğunu, fakat CatBoost, XGBoost ve ANN gibi modellerin yalnızca URL yapısal özellikleriyle de rekabetçi bir başarı sağlayabildiğini göstermektedir.

6.3 Model Parametreleri

Yaş parametresi dahil edilmeden eğitilen Random Search ile optimize edilen ANN modelinin parametreleri Tablo 10'da ve Tablo 11'de verilmiştir. Tablo 10'da verilen yapay sinir ağı mimarisi, altı gizli katman ve bir sigmoid çıkış katmanından oluşan derin, ancak sıkı şekilde düzenlenmiş bir yapıya sahiptir. Her gizli katmanda ReLU etkinleştirmesi kullanılarak doğrusal olmayan örüntülerin öğrenilmesi kolaylaştırılmıştır. Katman genişlikleri (128 – 448 nöron aralığında) dönüşümlü olarak yüksek ve orta seviyelerde tutulmuş; böylece ağ önce karmaşık temsiller üretip ardından bunları sıkıştırarak bilgi kaybını en aza indiren bir "sık–geniş" akış oluşturmuştur. L2 regülasyon katsayıları 0,0002 – 0,01 aralığında değişmekte olup darbeli değerler (örneğin ikinci katmandaki 0,01) belirli katmanlarda daha sıkı ağırlık cezaları uygulayarak aşırı öğrenmeyi baskılar. Dropout oranları da katman derinliğine göre ayarlanmıştır: orta katmanlarda %15–25 civarında hafif dropout tercih edilirken dördüncü katmanda %50'ye varan agresif dropout kullanılarak en yoğun parametre bloğunun aşırı uyum riskine karşı korunması sağlanmıştır. Son olarak, tek nöronlu sigmoid çıkış katmanı modelin çıktısını 0–1 aralığında zararlı-zararsız olasılığına çevirerek ikili sınıflandırmayı tamamlar. Bu dengeli katman boyutları, değişken L2 cezaları ve katman-özgü dropout oranları birleşerek hem öğrenme kapasitesini yüksek tutan hem de genelleme kabiliyetini koruyan bir ağ mimarisi ortaya koyar.

Tablo 10: ANN modeli parametreleri

	Birim Sayısı	Aktivasyon Fonksiyonu	L2 Regülasyonu	Dropout
İlk Katman	384	ReLU	0.001	0.15
İkinci Katman	128	ReLU	0.01	0.3
Üçüncü Katman	384	ReLU	0.001	0.15
Dördüncü Katman	272	ReLU	0.0002	0.5
Beşinci Katman	448	ReLU	0.001	0.1
Altıncı Katman	384	ReLU	0.0002	0.25
Çıkış Katmanı	1	Sigmoid		

Tablo 11'de görüldüğü üzere bu modelin ağırlık güncellemeleri RMSprop tabanlı bir optimizatörle gerçekleştirilmiştir. Başlangıç öğrenme oranı 0,0037, parametrelerin yeterince hızlı ancak kararlı biçimde güncellenmesine olanak tanır. Sayısal kararlılığı sağlamak için paydada $\varepsilon = 3 \times 10^{-8}$ değeri kullanılmıştır; bu küçük ekleme, çok küçük gradyan karelerinde bölme işlemini dengeler. Momentum (0,1), geçmiş gradyan bilgisini sınırlı ölçüde güncel adıma dahil ederek optimizasyon yolunu yumuşatır ve yerel minimumlarda takılmayı azaltır. Rho (0,88), gradyan karelerinin üstel hareketli ortalamasında geçmişe ait bilgiyle güncel bilgiyi dengeler; daha yüksek rho, daha uzun "hafıza" anlamına gelir. Ek olarak, $\beta_1 = 0{,}97$ ve $\beta_2 = 0{,}99$ değerleri, RMSprop'a Adam benzeri bir moment unsuru ekleyerek hem ortalama gradyanı hem de gradyan karelerinin ortalamasını uzun vadede izler; bu da daha istikrarlı ve hızlı yakınsama sağlar. Bu hiperparametre kombinasyonu, öğrenme sürecini hızlandırırken aşırı salınımları önleyerek modelin güvenilir şekilde yakınsamasını hedefler.

Tablo 11: Optimizasyon ve Öğrenme Parametreleri

Optimizer	RMSprop
İlk Öğrenme Oranı	0.0037
Epsilon	3e-08
Momentum	0.1
Rho	0.88
Beta 1	0.97

Beta 2	0.99

Yaş parametresi dahil edilmeden yapılan Random Search ile optimize edilen CNN modelinin kullanılan katman parametreleri Tablo 12'de ve Tablo 13'de verilmiştir.

Tablo 12'de verilen ve yaş bilgisi olmadan eğitilen bu CNN, yedi ardışık konvolüsyonel katman ile karakter dizilimlerindeki yerel örüntüleri kademeli olarak yakalayan derin bir yapıya sahiptir. İlk katmandaki 192 filtre, ham URL girişinden zengin özellik haritaları çıkarırken, takip eden katmanlar filtre sayısını 16'ya kadar azaltarak özellikleri giderek yoğunlaştırır. Her konvolüsyon bloğunda ReLU etkinleştirme ve batch-normalization kullanılmış; seçili katmanlarda (2 × 1) max-pooling uygulanarak uzamsal boyut daraltılmış ve bilgi kaybı en aza indirilmiştir. %10–%25 aralığındaki dropout oranları, modelin yalnızca URL karakteristiklerine dayalı öğreniminde aşırı uyumu engeller. Konvolüsyonel bloktan sonra gelen flatten katmanı, özellik haritalarını tek boyutlu vektöre dönüştürür; bunu 256 ve 64 nöronlu iki tam bağlantılı katman izler. Bu katmanlar ReLU, batch-normalization ve hafif dropout kombinasyonuyla yüksek düzeyli temsiller oluşturur. Tek nöronlu sigmoid çıkış katmanı ise URL'nin zararlı olma olasılığını üretir. Domain yaşı bilgisinden yoksun olmasına rağmen, katman derinliği, düzenli normalizasyon ve dropout kullanımı sayesinde model, URL'nin yalnızca yapısal işaretlerini dikkate alarak kayda değer bir ayırt etme performansı sergilemektedir.

Tablo 12: CNN modeli parametreleri

	Filtre/Unit	Kernel Boyutu	Aktivasyon Fonksiyonu	Batch Normalization	Maxpooling	Dropout
Konvolüsyonel Katman 1	192	(3,1)	ReLU	Eklendi	(2,1)	0.25
Konvolüsyonel Katman 2	144	(3,1)	ReLU	Eklendi		0.15
Konvolüsyonel Katman 3	112	(3,1)	ReLU	Eklendi	(2,1)	0.15
Konvolüsyonel Katman 4	64	(3,1)	ReLU	Eklendi		0.1
Konvolüsyonel Katman 5	32	(3,1)	ReLU	Eklendi	(2,1)	0.1
Konvolüsyonel Katman 6	32	(3,1)	ReLU	Eklendi		0.1
Konvolüsyonel Katman 7	16	(3,1)	ReLU	Eklendi	(2,1)	0.1
Flatten Katmanı						
Full Connected Katman 1	256		ReLU	Eklendi		0.1
Full Connected Katman 2	64		ReLU	Eklendi		0.15
Çıkış Katmanı	1		Sigmoid			

Tablo 13'de verilen bu CNN modelinin ağırlık güncellemeleri Adam optimizatörüyle gerçekleştirilmiştir. Başlangıç öğrenme oranının 0,003 olarak seçilmesi, parametrelerin dengeli ve istikrarlı biçimde güncellenmesine imkân tanırken, $\varepsilon = 3 \times 10^{-8}$ değeri numerik kararlılığı garanti ederek çok küçük gradyanlarda bölme hatalarını önler. Momentum katsayısı $\beta_1 = 0{,}93$, geçmiş gradyan bilgisini güncel adıma %93 oranında taşıyarak yakınsamayı pürüzsüzleştirir; $\beta_2 = 0{,}999$ ise gradyan karelerinin uzun vadeli ortalamasını tutarak büyük ve dengesiz gradyan değerlerinin etkisini dengeler. Bu hiperparametre seti, yalnızca URL özelliklerine dayalı öğrenme senaryosunda modelin hızlı fakat güvenilir biçimde yakınsamasını sağlar.

Tablo 13: Optimizasyon ve Öğrenme Parametreleri

Optimizer	Adam
İlk Öğrenme Oranı	0.003
Epsilon	3e-08
Beta1	0.93
Beta2	0.999

Yaş parametresi dahil edilerek yapılan Optuna ile optimize edilen **Catboost** modeline ait parametreler Tablo 14'de verilmiştir. Tablo 14'te sunulan hiperparametreler, domain yaşı kullanılmadan optimize edilen CatBoost modelinin en iyi konfigürasyonunu göstermektedir. 1808 iterasyon, modelin yeterli sayıda ağaç oluşturmasına olanak tanırken, 0,13'lük öğrenme oranı her ek ağaçtaki güncellemeyi dengede tutarak yakınsamanın hem hızlı hem de kararlı gerçekleşmesini sağlar. Ağaç derinliğinin 7 olarak sınırlandırılması, daha karmaşık örüntüleri yakalamaya imkân verirken aşırı uyum riskini kontrol altında tutar. L2 regülasyonu (9,39), yaprak skorlarına uygulanan kuvvetli ceza ile model karmaşıklığını azaltarak genelleme yeteneğini artırır. Rastgelelik gücü (4,15) ve bagging temperature (7,60) değerleri, her iterasyonda örnek ve özellik çeşitliliğini yükseltip ağaçlar arası korelasyonu düşürerek topluluğun dayanıklılığını güçlendirir. Son olarak, sınır sayısının 38 olarak ayarlanması, sayısal özelliklerin sınırlı fakat yeterli aralıklara bölünmesini sağlayarak hesaplama maliyetini düşük tutarken ayrım gücünü korur. Bu hiperparametre bileşimi, CatBoost'un yalnızca URL yapısal özellikleriyle yüksek ayırt etme performansı sergilemesine katkıda bulunmuştur.

Tablo 14: Model Parametreleri

İterasyon	1808
Öğrenme Oranı	0.13
Ağaç Derinliği	7
L2 Regülasyonu	9.39
Rastgelelik Gücü	4.15
Bagging Temperature	7.60
Sınır Sayısı	38

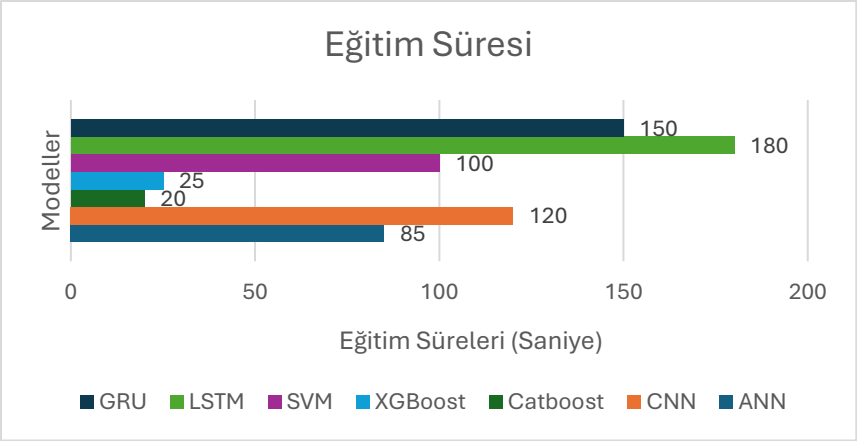
Yaş parametresi dahil edilerek yapılan Optuna ile optimize edilen **XGBoost** modeline ait parametreler Tablo 15'te verilmiştir. Tablo 15'teki hiperparametreler, alan adı yaşı dâhil edilmeden optimize edilen XGBoost modelinin 751 ağaçlık en iyi konfigürasyonunu göstermektedir. Öğrenme oranı 0,13, her ağacın katkısını kontrollü tutarak hızlı ama dengeli bir yakınsamayı mümkün kılar. Maksimum derinlik 6 ile sınırlı tutulmuş; böylece model, karmaşık kalıpları yakalayabilecek esnekliğe sahipken aşırı öğrenme riski azaltılmıştır. Min-child-weight 4 ve gamma 0,37, yalnızca istatistiksel olarak anlamlı bölünmelerin yapılmasına izin vererek ağaç sayısını yönetilebilir düzeyde tutar. Rasgelelik parametreleri olan subsample 0,88 ve colsample_bytree 0,50, her iterasyonda farklı örnek ve özellik alt kümeleriyle eğitim yapıp model çeşitliliğini artırır. Düzenleştirme katsayıları L1 (reg_alpha) 2,75 ve L2 (reg_lambda) 7,67, yaprak ağırlıklarını cezalandırarak modelin genelleme yeteneğini güçlendirir. Bu dengeli hiperparametre seti, yalnızca URL yapısal özelliklerine dayanarak XGBoost'un yüksek doğruluk ve sağlamlıkla zararlı URL'leri tespit etmesini sağlamıştır.

Tablo 15: Model Parametreleri

Ağaç Sayısı	751
Öğrenme Oranı	0.13
Maksimum Derinlik	6
Min. Yaprak Ağırlığı	4
Gamma Değeri	0.37
Alt Örnek Oranı	0.88
Öznitelik Alt Örneği	0.50
L1 Regülasyonu	2.75
L2 Regülasyonu	7.67

6.4 Model Eğitim Süreleri

Eğitim süresi, kullanılan modellerin performans değerlendirmesinde önemli bir kriterdir. Her bir modelin eğitim süresi, özellikle gerçek zamanlı uygulamalar için dikkate alınması gereken bir faktördür. **Şekil 6**, çalışmamızda kullanılan farklı modellerin her bir modeldeki eğitim sürelerini (saniye cinsinden) göstermektedir. Bu süreler, modelin karmaşıklığı, katman sayısı ve parametre boyutuna bağlı olarak değişiklik göstermektedir. Daha karmaşık modeller, özellikle derin öğrenme yaklaşımları, eğitim için daha uzun süre gerektirmiştir. Bu durum, performans ve eğitim süresi arasında bir denge kurulmasını gerektirebilir.

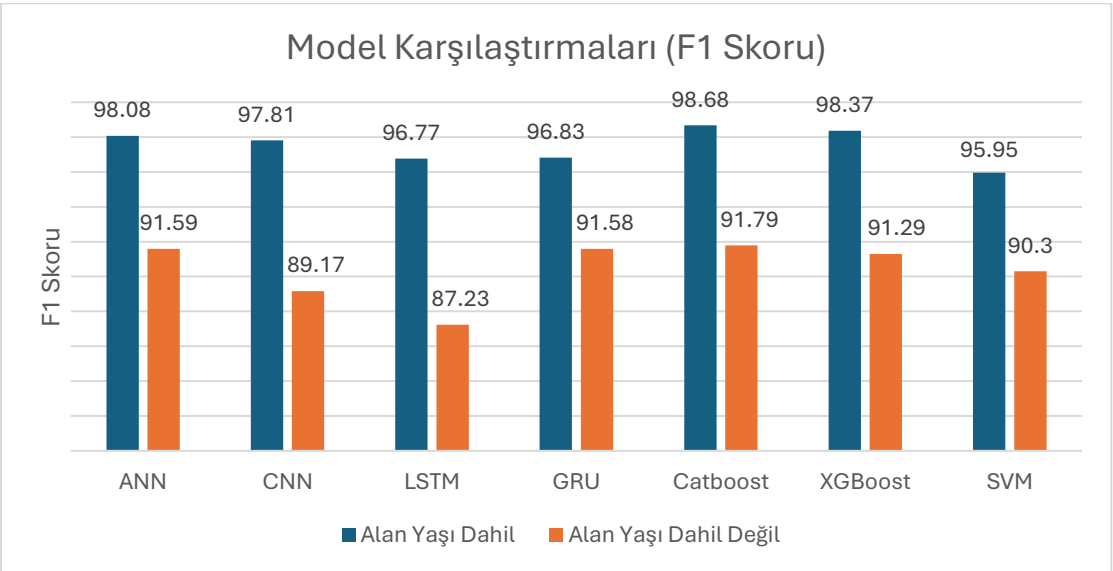


Şekil 6: Algoritmaların eğitim süreleri

Şekil 6, hiperparametre araması sonrasında elde edilen en iyi konfigürasyonlar için modellerin ortalama eğitim sürelerini özetlemektedir. 22 708 URL'lik dengeli veri kümesi her modelde 5-katlı çapraz doğrulama ile eğitildiğinden, toplam işlem yükü katlanarak artsa da süreler kabul edilebilir düzeyde kalmıştır. Ağaç tabanlı yöntemler en kısa süreleri vermiştir: CatBoost yaklaşık 20 saniye, XGBoost ise 25 saniye ile eğitimini tamamlamıştır. Derin öğrenme tarafında ANN ortalama 85 saniye ve CNN 120 saniye sürede yakınsarken, ardışıl mimarilerin parametre sayısı ve zaman adımlı işlem doğası nedeniyle GRU 150 saniye ve LSTM 180 saniye ile en uzun süreleri gerektirmiştir. Klasik yöntemlerden SVM'nin 100 saniyelik ortalaması, kernel hesaplamalarının beklenenden daha maliyetli olabildiğini göstermektedir. Süreler, Random Search (derin öğrenme) ve Optuna (ağaç tabanlı) optimizasyon döngülerine rağmen ölçülmüş olup her bir modelin "tek kat" eğitimi baz alınarak raporlanmıştır. Bu sonuçlar, yalnızca URL yapısal özellikleriyle çalışan sistemin, yüksek doğruluk yanında gerçek-zamanlı uygulamalara uygun hızda eğitilebildiğini ortaya koymaktadır.

7. Model Değerlendirmesi ve Sonuçlar

Şekil 7, alan adı yaşı özniteliğinin dâhil edilmesinin F1 skoru üzerindeki etkisini görsel olarak ortaya koymaktadır. Mavi sütunlar (yaş dâhil) ile turuncu sütunlar (yaş hariç) arasındaki fark, tüm modellerde belirgin bir performans artışına işaret etmektedir. Yaş bilgisi eklendiğinde CatBoost ve XGBoost sırasıyla 98,68 ve 98,37 F1 skorlarıyla başı çekerken ANN de 98,08 değeriyle derin öğrenme grubunun en başarılısı olmuştur. CNN, GRU ve LSTM'de gözlenen artışlar da dikkat çekicidir; özellikle LSTM'nin F1 skoru yaş bilgisi olmadan 87,23 iken yaş eklendiğinde 96,77'ye yükselerek dokuz puandan fazla iyileşme sağlamıştır. SVM dâhil tüm yöntemlerde en az beş puanlık artış yaşanması, domain yaşının modeller için tutarlı biçimde ayırt edici bir özellik olduğunu teyit etmektedir. Bununla birlikte turuncu sütunlar, yaş özelliği çıkarıldığında CatBoost ve XGBoost'un hâlâ 91–92 bandında kalabildiğini, ANN ve GRU'nun ise %91 civarında tutarlı sonuç verdiğini göstermektedir. Dolayısıyla alan adı yaşı, bütün modellerde performansı yükseltse de temel URL özellikleri tek başına kabul edilebilir bir seviye sunmakta; yaş bilgisi eklenerek F1 skorlarında 6–10 puanlık ek bir kazanç elde edilmektedir.



Şekil 7: Algoritmaların alan yaşı dahil olan ve olmayan şekilde karşılaştırılması

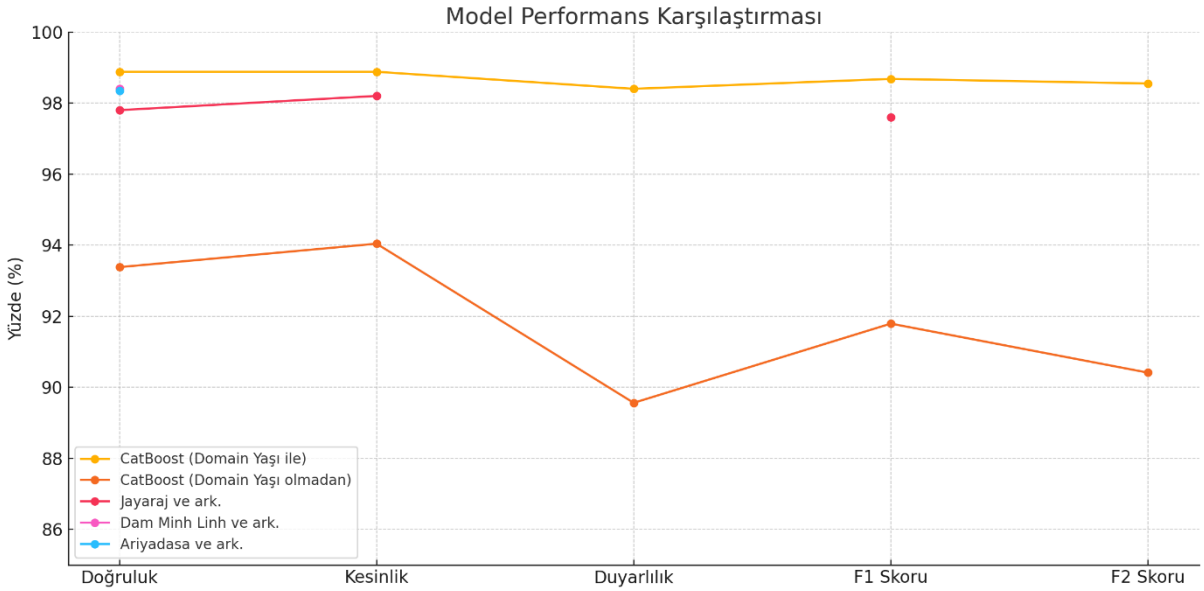
Yalnızca URL'nin yapısal özellikleri kullanılarak eğitilen modeller 5-katlı çapraz doğrulama sonunda ortalama %93-94 doğruluk aralığında performans göstermiştir. Bu senaryoda en yüksek başarı, CatBoost (%93,38 doğruluk, %90,41 F_2) ve ANN (%93,38 doğruluk, %89,54 F_2) modellerinde elde edilmiştir; GRU ve XGBoost da benzer doğruluk düzeylerinde tutarlı sonuçlar vermiştir. URL öznitelikleriyle sınırlı kalınmasına karşın, söz konusu metrikler bu modellerin zararlı URL'leri pratik olarak ayırt edebildiğini göstermektedir.

Domain yaşı parametresi eklendiğinde performans belirgin biçimde yükselmiş; CatBoost %98,88 doğruluk ve %98,55 F_2 skoruyla en üst düzeye çıkmış, XGBoost ve ANN benzer şekilde %98'in üzerinde doğruluk değerlerine ulaşmıştır. Alan adı yaşının bu ölçüde katkı sağlaması, zararlı sitelerin kısa ömürlü olma eğilimini doğrular niteliktedir.

Bununla birlikte, domain yaşı kullanımı yeni tescil edilmiş ancak zararsız siteler için yanlış-pozitif riskini artırmaktadır. Özellikle yaş bilgisi dâhil edildiğinde model, düşük yaşlı tüm alan adlarını zararlı olarak işaretlemeye yatkın hâle gelmiştir; yaş özelliği kaldırıldığında doğruluk yaklaşık beş puan düşerken bu önyargı da ortadan kalkmıştır. Dolayısıyla, domain yaşı güçlü bir ayırt edici faktör olsa da modelin aşırı önyargılı davranmaması için hassas biçimde dengelenmesi gerekmektedir.

Sonuç olarak, çalışmamız domain yaşı olmaksızın dahi rekabetçi doğruluk sağlayan hafif URL-tabanlı modeller ortaya koyarken, yaş bilgisinin eklenmesiyle %98,7'ye varan doğruluk düzeylerine ulaşılabileceğini göstermiştir. Pratik uygulamalarda, yanlış pozitifleri sınırlamak adına domain yaşının ağırlığı dikkatle ayarlanmalı veya ek doğrulama adımlarıyla desteklenmelidir.

Bu çalışmada önerilen CatBoost modeli, hem domain yaşı içeren hem de içermeyen deneysel senaryolarda en yüksek performansı sergilemiştir. Domain yaşı özelliği modele dahil edildiğinde, CatBoost %98.88 doğruluk, %98.88 kesinlik, %98.40 duyarlılık, %98.68 F1 skoru ve %98.55 F2 skoru ile en başarılı sonuçlara ulaşmıştır. Domain yaşı bilgisi hariç tutulduğunda dahi model, %93.38 doğruluk, %94.04 kesinlik, %89.56 duyarlılık, %91.79 F1 skoru ve %90.41 F2 skoru ile diğer modellere kıyasla üstünlüğünü korumuştur. **Şekil 8'de** görülebileceği üzere, CatBoost modeli her iki senaryoda da literatürdeki öne çıkan çalışmalarla karşılaştırıldığında oldukça rekabetçi sonuçlar vermektedir. Örneğin, Jayaraj ve arkadaşlarının HEFS temelli hibrit modeli %97.8 doğruluk, %98.2 hassasiyet ve %97.6 F1 skoru elde etmişken; CatBoost bu üç metrikte de daha yüksek performans göstermiştir. Benzer şekilde, Dam Minh Linh ve arkadaşları tarafından geliştirilen CNN tabanlı sistem %98.4 doğruluk raporlamış, ancak F1 skoru belirtilmemiştir; buna karşın CatBoost'un %98.68 F1 skoru, genel başarım açısından üstünlüğünü ortaya koymaktadır. Ayrıca Ariyadasa ve arkadaşlarının URL+HTML birleşimi hibrit modelinde elde edilen %98.34 doğruluk değeri de CatBoost tarafından aşılmıştır. Sonuç olarak, önerilen CatBoost tabanlı yaklaşım yalnızca yüksek doğruluk sağlamakla kalmayıp, aynı zamanda F1 ve F2 gibi denge metriklerinde de literatürdeki modellere üstünlük sağlamaktadır. Bu yönüyle model, hem veri kalitesi hem de öznitelik çeşitliliğine dayanan yapısı sayesinde oltalama saldırılarının tespitinde etkili ve pratik bir çözüm sunmaktadır.



Şekil 8: Diğer çalışmalarla karşılaştırma

8. Sonuç

Bu çalışma, zararlı web sitelerini yalnızca URL'lerin yapısal özniteliklerinden yararlanarak tespit etmeye odaklanan, makine öğrenmesi ve derin öğrenme modellerinin kapsamlı bir karşılaştırmasını sunmaktadır. Majestic Top-1 Million listesindeki güvenli alan adları ile USOM'dan derlenen zararlı alan adlarının birleştirilmesiyle oluşturulan 22708 örnekten oluşan dengeli veri kümesi üzerinde beş-kat çapraz doğrulama uygulanmış, hiperparametre ayarları derin öğrenme mimarilerinde Random Search, CatBoost ve XGBoost modellerinde ise Optuna-TPE aracılığıyla optimize edilmiştir. Yalnızca URL karakteristiklerine dayalı temel özellikler kullanıldığında modeller ortalama %93-94 doğruluk bandına ulaşmış; CatBoost ve ANN, %93,38 doğruluk ve CatBoost %90,41 ve ANN %89,54 F2 skorlarıyla öne çıkmıştır. Domain yaşı özelliği eklendiğinde sınıflandırma gücü kayda değer biçimde artmış; CatBoost %98,88 doğruluk, %98,68 F1 ve %98,55 F_2 değerleriyle en yüksek performansı sergilerken, XGBoost ve ANN de %98'in üzerinde doğruluk elde etmiştir. Domain yaşı, zararlı URL'lerin kısa ömürlü olma eğilimi sayesinde güçlü bir ayırt edici unsur olsa da yeni tescilli güvenli sitelerde yanlış-pozitif oranını yükseltebilmektedir. Gerçekten de yaş bilgisi kaldırıldığında doğruluğun beş-altı puan düşmesi bu özelliğin önemini teyit ederken, aşırı önyargıyı azaltmak için yaşa duyarlı eşik ayarlarının gerekli olduğu görülmüştür. Optuna'nın erken budama (pruning) yeteneği sayesinde yüzlerce deneme dakikalar içinde tamamlanmış; buna rağmen eğitim süreleri CatBoost için ≈ 20 s, XGBoost ≈ 25 s, ANN ≈ 85 s, CNN ≈ 120 s, GRU ≈ 150 s ve LSTM ≈ 180 s civarında kalarak gerçek-zamanlı güvenlik çözümlerine entegre edilebilir düzeyde kalmıştır. Elde edilen bulgular, içerik analizi veya trafik takibi gibi ağır işlemlere gerek duymadan, salt URL dizgileriyle %98,7'ye varan doğruluk elde edilebileceğini göstermektedir. Optuna-temelli CatBoost ve XGBoost yapılandırmalarının saniyeler mertebesinde eğitilebilmesi, geniş ölçekli dağıtık savunma sistemlerinde pratik uygulanabilirliği artırmaktadır. Önerilen modeller, tarayıcı eklentileri, ağ geçidi filtreleri ve e-posta tarama sistemleri gibi erken uyarı katmanlarına kolayca entegre edilerek siber güvenliğin proaktif savunma hattında etkin rol oynayabilir. Gelecekte, veri kümesinin farklı dönemlerden toplanacak URL'lerle periyodik olarak güncellenmesi, modellerin zaman içindeki dayanıklılığını ve yeni saldırı kalıplarına uyum kabiliyetini ölçmek açısından önemlidir. Domain yaşı küçük alan adlarında ortaya çıkan yanlış-pozitifleri sınırlamak amacıyla dinamik eşikleme veya ek risk skoru gibi yaşa duyarlı düzeltme teknikleri geliştirilebilir. Yalnızca karakter dizilerine odaklanmanın ötesine geçmek için URL yolu (path) ve sorgu (query) segmentlerini içeren dizi-tabanlı dikkat (attention) mekanizmaları denenebilir; ayrıca TabPFN gibi Transformer temelli ön-eğitimli modellerle performans karşılaştırmaları yapılarak model çeşitliliği artırılabilir. Son olarak, sunucu taraflı entegrasyonlarla gerçek trafik akışında anlık tespit performansı ve alarm gecikmesi test edilerek, modelin operasyonel ortamlarda ölçeklenebilirliği ve tepkime süresi netleştirilebilir.

Kaynakça

1) Ye Tian, Yanqiu Yu, Jianguo Sun and Yanbin Wang, (2025). From Past to Present: A Survey of Malicious URL Detection Techniques, Datasets and Code Repositories.

2) Majestic. (2024). *Majestic Million: Top 1 Million Websites by Traffic*.

3) USOM. (2024). *USOM Zararlı URL Listesi*. https://www.usom.gov.tr/

4) Yao, W., Ding, Y., & Li, X. (2018, December). Deep learning for phishing detection. Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, 645–650. IEEE.

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

6) Yerima, S. Y., & Alzaylaee, M. K. (2020, April 8). High Accuracy Phishing Detection Based on Convolutional Neural Networks (arXiv preprint). arXiv. https://doi.org/10.48550/arXiv.2004.03960.

7) National Cyber Security Centre (NCSC). (2024). *Phishing and Malicious Websites: How to Protect Yourself*.

8) Jayaraj, R., Pushpalatha, A., Damodaran, D., Sangeetha, K., Kamaleshwar, T., & Udhaya Shree, S. (2024). *Intrusion detection based on phishing detection with machine learning*.

9) Linh, D. M., Hung, H. D., Chau, H. M., Vu, Q. S., & Tran, T.-N. (2021). *Real-time phishing detection using deep learning methods by extensions*. International Journal of Electrical and Computer Engineering, 14(3), 3021-3035.

10) Vijayalakshmi, M., Shalinie, S. M., Yang, M. H., & Meenakshi, U. R. (2020). *Web phishing detection techniques: A survey on the state-of-the-art, taxonomy and future directions*. IET Networks, 9(5), 235-246.

11) Abdulrahman, L. M., Ahmed, S. H., Rashid, Z. N., Jghef, Y. S., Ghazi Sami, T. M., & Jader, U. H. (2023). *Web phishing detection using web crawling, cloud infrastructure and deep learning framework*. Computers, Materials & Continua, 64(3), 1529-1544.

12) Yi, P., Guan, Y., Zou, F., Yao, Y., Wang, W., & Zhu, T. (2018). Web phishing detection using a deep learning framework. Wireless Communications and Mobile Computing, Article 4678746.

13) Yao, W., Ding, Y., & Li, X. (2018, December 11-13). Deep learning for phishing detection. In Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (pp. 645-650)

14) Ariyadasa, S., Fernando, S., & Fernando, S. (2020). Detecting phishing attacks using a combined model of LSTM and CNN. International Journal of Advanced and Applied Sciences, 7(7), 56-67.

15) P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," IEEE Access, 7, 15196-15209.

16) Y. Huang, J. Qin, and W. Wen, "Phishing URL Detection Via Capsule Based Neural Network," in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2019, pp. 22-26.

17) W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," IET Information Security, vol. 13, pp. 659-669, 2019.

18) O. K. Sahingoz, S. I. Baykal, and D. Bulut, "Phishing detection from URLs by using neural networks," Computer Science & Information Technology (CS & IT), pp. 41-54, 2018.

19) S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and M. S. AN, "Phishing detection using random forest, SVM and neural network with backpropagation," in 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 391-394.

20) https://github.com/halilkkaya/url_phishing

21) Meng W., Li W., Lam For Kwok, “EFM: Enhancing the Performance of Signature-based Network Intrusion Detection Systems Using Enhanced Filter Mechanism” 2014

22) Ye Tiana, Yanqiu Yua, Jianguo Suna and Yanbin Wanga, “From Past to Present: A Survey of Malicious URL Detection Techniques, Datasets and Code Repositories” 2025

23) Mehmet Korkmaz, Emre Kocyigit, Ozgur Koray Sahingoz , Banu Diri “A Hybrid Phishing Detection System Using Deep Learning-based URL and Content Analysis” 2022