**Makine Öğrenme Yöntemleriyle Phishing Website Tahmini**

*Bilgisayar Mühendisliği, Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, Türkiye <23181616010@gazi.edu.tr> - Abdülselam İsmail*

*Bilgisayar Mühendisliği, Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, Türkiye [benes.atay@gazi.edu.tr](mailto:benes.atay@gazi.edu.tr) - Bayram Enes Atay*

*Bilgisayar Mühendisliği, Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, Türkiye [23181616407@gazi.edu.tr](mailto:23181616407@gazi.edu.tr) - Bektur Mukhtarbekov*

**ÖZET**

Phishing (oltalama) saldırıları, bireylerin siber güvenlik konusundaki bilinç eksikliklerinden ve yetersiz eğitimlerinden faydalanarak gerçekleştirilen, kullanıcıları kandırma yoluyla hassas bilgileri ele geçirmeyi amaçlayan bir siber saldırı yöntemidir. Bu saldırılar her geçen günle, özellikle web ortamında, yaygınlaşıyor. Bu saldırılardan korunmak amaçla birçok çeşit çözüm sunulabilir. Bizim yaptığımız çalışmanın amacı makine öğrenme yöntemleri kullanarak oltalama web-sitelerin tespitini kolaylaştırmaktır. Bunun için *ucimlrepo’*dan veri seti seçtik. Veri seti genel olarak *PhishTank archive* ve *MillerSmiles archive’*tantoplanmıştır. Veri seti yanında çalışmamız için uygun yöntem seçmede sıradaki yöntemleri kullanarak sonuçlarını karşılaştırmaya karar verdik: *K-NN, Karar Ağacı, SVM, Random Forest, Naive Bayes, Lineer Regresyon ve* *RF Regresyon* . Çalışma sonucunda %97 doğruluk oranıyla en büyük başarıyı Random Forest yöntemi göstermiştir.

**Anahtar Kelimeler:** Phishing, Makine öğrenme, Tahmin

1. **Giriş**

Phishing saldırıları genellikle kullanıcı isimleri, şifreler, kredi kartı bilgileri, ağ kimlik bilgileri gibi hassas ve gizli bilgilere ulaşmak amacıyla yapılan saldırılardır. Siber saldırganlar, telefon veya e-posta yoluyla normal bir birey veya kurum gibi görünerek mağdurları belirli eylemleri - zararlı bir bağlantıya veya eke tıklamak gibi - gerçekleştirmeleri veya isteyerek gizli bilgileri açıklamaları için manipüle etmek üzere sosyal mühendisliği kullanırlar[1].

*Comcast Business Siber Güvenlik Tehdit Rapor’a*  göre siber saldırıların çoğu phishing saldırısıyla başlarlar[2]. Ayrıca *Cloudflare*, 13 milyar e-postayı analiz ettiği çalışmasında, aldatıcı bağlantıların (%36 ile) en yaygın oltalama yöntemi olduğunu bildirdi[3]*.*

Örneğin Ağustos 2023’te Twilio, bilgisayar korsanlarının müşteri verilerine yetkisiz erişim sağladığını doğruladı. Saldırganlar, iyi organize edilmiş bir SMS oltalama kampanyasıyla Twilio çalışanlarını giriş bilgilerini vermeleri için kandırarak şirketin dahili güvenlik önlemlerini etkili bir şekilde aşmayı başardı. Bu ihlal, büyük teknoloji şirketlerinin karşı karşıya olduğu devam eden güvenlik açıkları ve bu şirketlerin nasıl daha iyi korunabileceği konusunda kritik soruları gündeme getirdi.

Twilio, saldırının şirketin BT departmanını taklit eden bir dizi SMS oltalama mesajıyla başladığını açıkladı. Bu mesajlarda, çalışanların şifrelerinin süresinin dolduğu ya da çalışma programlarının değiştiği öne sürülerek kötü amaçlı bir bağlantıya tıklamaları istendi. Bu bağlantı, çalışanların giriş bilgilerini girmeye teşvik edildiği sahte bir web sitesine yönlendiriyordu.

Saldırganlar, “SSO” (tek oturum açma) gibi sektöre özgü terimler ve şirket içi dil kullanarak mesajlara inandırıcılık katmış ve çalışanların bu tuzağa düşme olasılığını artırmıştı. Bu strateji, saldırganların güvenilir ve gerçekçi görünmesi için özel olarak hazırlanmış mesajlar kullandığı hedefli oltalama (spear phishing) türünün tipik bir örneğidir.[4]

Bu yanıltıcı web-sitelerin tehlikesini ve onlardan korunmamız ne kadar önemli olduğunu gösterir. Bir şirketin personelini bu tür saldırılara karşı eğitmek o güvenliğin olmazsa olmazı. Fakat bu yöntem şirketi sadece bir yere kadar koruyabilir. İnsan hataları hep olan ve olacak bir şeydir. Bu tür problemlere özel hazırlanmış araçlar kullanması insanların yapabileceği hata payını azaltmakta yardımcı olabilir.

Bu tür araçların *Makine öğrenmesi* yöntemleriyle oluşturulması son dönemlerin konusu olmuştur. *Makine öğrenmesi* yapay zekanın (AI) bir alt kümesidir. Bilgisayarları verilerden öğrenmeyi öğretmeye ve bunu yapmak için açıkça programlanmak yerine deneyimle geliştirmeye odaklanır. Makine öğrenmesinde algoritmalar büyük veri kümelerinde desenleri ve korelasyonları bulmak ve bu analize dayalı en iyi kararları ve tahminleri yapmak için eğitilir.[5]

1. **Materyal ve Yöntem**

**2.1. Veri Seti**

Bu çalışma için *UC Irvine Machine Learning Repository’*den *Phishing Websites*[6]adlı hazır veri setini kullandık. Veri seti Rami M. Mohammad, Fadi Thabtah, Lee McCluskey tarafından hazırlanmış ve bu veri seti ağırlıklı olarak PhishTank arşivi, MillerSmiles arşivi ve Google'ın arama operatörlerinden toplanmıştır. Veri seti 30 özellik ve 1 hedeften oluşmaktadır bunlar Tablo 1’deki gibidir.

Tablo 1. Phishing websites veri setinin özellikleri

| **No** | **Özellik Adı** | **Türü** | **Açıklama** | **Tipi** |  |
| --- | --- | --- | --- | --- | --- |
| 1 | having\_ip\_address | Integer | URL IP adresi içeriyor mu | Özellik |  |
| 2 | url\_length | Integer | URL'nin uzunluğu | Özellik |  |
| 3 | shortining\_service | Integer | URL kısaltma servisi kullanılmış mı | Özellik |  |
| 4 | having\_at\_symbol | Integer | '@' sembolü içeriyor mu | Özellik |  |
| 5 | double\_slash\_redirecting | Integer | Çift eğik çizgi ile yönlendirme yapılıyor mu | Özellik |  |
| 6 | prefix\_suffix | Integer | Alan adı tire (-) içeriyor mu | Özellik |  |
| 7 | having\_sub\_domain | Integer | Alt alan adı içeriyor mu | Özellik |  |
| 8 | sslfinal\_state | Integer | SSL sertifikası geçerli mi | Özellik |  |
| 9 | domain\_registration\_length | Integer | Alan adı uzun süreli mi kayıtlı | Özellik |  |
| 10 | favicon | Integer | Site simgesi orijinal mi | Özellik |  |
| 11 | port | Integer | Standart dışı port kullanılmış mı | Özellik |  |
| 12 | https\_token | Integer | URL'de https ifadesi yanlış yerde mi | Özellik |  |
| 13 | request\_url | Integer | Sayfadaki içerik başka sitelerden mi alınmış | Özellik |  |
| 14 | url\_of\_anchor | Integer | Bağlantıların güvenilir olup olmadığı | Özellik |  |
| 15 | links\_in\_tags | Integer | Script/meta etiketlerinde dış bağlantılar var mı | Özellik |  |
| 16 | sfh | Integer | Formların yönlendirildiği sunucu güvenli mi | Özellik |  |
| 17 | submitting\_to\_email | Integer | Form sonuçları e-posta ile mi gönderiliyor | Özellik |  |
| 18 | abnormal\_url | Integer | URL anormal mi | Özellik |  |
| 19 | redirect | Integer | Yönlendirme sayısı yüksek mi | Özellik |  |
| 20 | on\_mouseover | Integer | Mouse-over ile içerik gizleniyor mu | Özellik |  |
| 21 | rightclick | Integer | Sağ tıklama engellenmiş mi | Özellik |  |
| 22 | popupwindow | Integer | Açılır pencere kullanılmış mı | Özellik |  |
| 23 | iframe | Integer | Sayfada iframe kullanılmış mı | Özellik |  |
| 24 | age\_of\_domain | Integer | Alan adı ne kadar süredir var | Özellik |  |
| 25 | dnsrecord | Integer | DNS kaydı mevcut mu | Özellik |  |
| 26 | web\_traffic | Integer | Web sitesi trafiği | Özellik |  |
| 27 | page\_rank | Integer | Sayfanın Google’daki sıralaması | Özellik |  |
| 28 | google\_index | Integer | Google’da indekslenmiş mi | Özellik |  |
| 29 | links\_pointing\_to\_page | Integer | Sayfaya gelen bağlantı sayısı | Özellik |  |
| 30 | statistical\_report | Integer | URL hakkında olumsuz istatistiksel rapor var mı | Özellik |  |
| 31 | result | Integer | Sonuç (1 = Güvenli, -1 = Phishing/sahte) | Hedef |  |

Her bir özellik bize modelimizin doğruluğunu artırabilecek potansiyelde olduğu için bunlardan hiçbirini silmemeye karar verdik. Veri setinde 11055 örnek girişi vardı ve hiç birinde aykırı değer yoktu. Ayrıca her örnek her özellik için {-1,0,1} değerleri alıyordu. Bunlar hepsi bizim veri önişleme aşamasını kolaylaştırmasını sağladı. O açıdan veri seti çok iyiydi.

**2.2. Kullanılan Makine Öğrenmesi Algoritmaları**

Bu çalışmada, farklı sınıflandırma problemleri için yaygın olarak kullanılan bazı denetimli makine öğrenmesi algoritmaları uygulanmıştır.

**2.2.1. K-En Yakın Komşu (K-Nearest Neighbors, KNN)** algoritması, sınıflandırma sürecinde eğitim verisindeki örnekler arasındaki mesafeleri esas alarak, test örneğinin en yakın k komşusuna göre sınıf etiketini belirler. Parametrik olmayan yapısıyla dikkat çeken bu algoritma, özellikle düşük boyutlu veri kümelerinde etkili sonuçlar verebilmektedir.

**2.2.2. Karar Ağacı (Decision Tree)** algoritması, veri kümesini öznitelik değerlerine göre bölerek ağaç yapısı oluşturur; her iç düğüm bir özniteliği, her dal bir karar kuralını ve yaprak düğümler sınıf etiketlerini temsil eder. Bu yapı, yüksek yorumlanabilirlik ve hızlı karar verme avantajı sunar.

**2.2.3. Destek Vektör Makineleri (Support Vector Machines, SVM)** ise sınıflar arası en geniş marjini sağlayacak hiper düzlemi bulmayı amaçlayan güçlü bir sınıflandırma yöntemidir. Bu çalışmada hem lineer çekirdek kullanılarak doğrusal ayrılabilir veriler için hem de RBF (Radial Basis Function) çekirdeği kullanılarak doğrusal olmayan durumlar için SVM modeli eğitilmiştir.

**2.2.4. Rastgele Orman (Random Forest)** algoritması, birden fazla karar ağacının rastgele örneklenmiş veri alt kümeleri ve özniteliklerle eğitilmesi ile oluşan bir topluluk (ensemble) yöntemidir. Ağaçların oylamasıyla sonuç üretir ve aşırı öğrenmeye karşı dirençlidir.

**2.2.5. Naive Bayes (Naif Bayes)** algoritması, Bayes teoremi temelinde çalışan olasılıksal bir sınıflandırma yöntemidir. Öznitelikler arasında koşulsal bağımsızlık varsayımı yapar; bu basitleştirici varsayıma rağmen birçok gerçek dünya problemlerinde yeterli doğrulukla sonuçlar üretebilmektedir. Naive Bayes’in özellikle yüksek boyutlu ve metin tabanlı veri kümelerinde etkin olduğu bilinmektedir.

Bu çalışmada, sayısal veri üzerinde tahminleme yapmak amacıyla iki farklı regresyon yöntemi kullanılmıştır: Lineer Regresyon ve Random Forest Regresyon. Her iki yöntem de farklı varsayımlara ve işleyiş mekanizmalarına sahiptir ve bu yönleriyle birbirlerini tamamlayıcı niteliktedir.

**2.2.6. Lineer regresyon**, bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi doğrusal bir model aracılığıyla ifade eden temel bir regresyon yöntemidir. En küçük kareler yöntemi (Ordinary Least Squares - OLS) ile modelin hata karelerinin toplamı minimize edilerek en uygun regresyon doğrusu elde edilir. Modelin matematiksel gösterimi aşağıdaki gibidir:

yazı tipi, metin, tipografi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 1. Lineer regresyon formülü

y: Bağımlı değişken (tahmin edilmek istenen çıktı),

𝑥𝑖: Bağımsız değişkenler (girdi verileri),

𝛽𝑖​: Regresyon katsayıları (modelin öğrendiği parametreler),

ϵ: Hata terimi (gürültü).

Lineer regresyon, özellikle değişkenler arasında doğrusal bir ilişki bulunduğu durumlarda yüksek performans gösterir. Ancak çok değişkenli veya doğrusal olmayan ilişkilerin bulunduğu veri kümelerinde sınırlı kalabilir.

**2.2.7. Random Forest regresyon**, çok sayıda karar ağacının (decision trees) bir araya gelmesiyle oluşturulan ansambl (ensemble) tabanlı bir makine öğrenme yöntemidir. Model, her ağacın tahminini alarak bunların ortalamasını hesaplar ve bu sayede yüksek doğruluk ve genelleme kabiliyeti sağlar. Rastgele örnekleme (bootstrap) ve rastgele değişken seçimi gibi teknikler sayesinde aşırı öğrenmenin (overfitting) önüne geçilir.

Random Forest regresyonun temel avantajları:

Karmaşık, doğrusal olmayan ilişkileri başarıyla modelleyebilir.

Aykırı değerlerden (outlier) ve eksik verilerden daha az etkilenir.

Özelliklerin önem derecesini belirleyerek yorumlama kolaylığı sağlar.

Ancak, modelin yorumlanabilirliği düşüktür ve eğitim süresi, lineer modellere göre daha uzundur. Bu yöntem, özellikle büyük ve değişken sayısı fazla olan veri setlerinde güçlü bir alternatif sunar.

**2.3. Kullanılan Ölçütler**

Makine öğrenmesi modellerinin başarımını objektif bir şekilde değerlendirmek, doğru sonuçlar elde etmek ve modelin genel performansını analiz edebilmek için çeşitli değerlendirme parametreleri (evaluation parameters) kullanılır. Bu parametreler, modelin doğruluğunu, güvenilirliğini ve etkinliğini belirlemek amacıyla kullanılır.

En yaygın kullanılan değerlendirme parametreleri aşağıda açıklanmıştır:

**2.3.1. Doğruluk (Accuracy)**, doğru sınıflandırılmış örneklerin tüm örneklere oranıdır. Genellikle sınıflandırma problemlerinde en yaygın kullanılan parametrelerden biridir. Ancak, dengesiz veri kümelerinde yanıltıcı olabilir, çünkü nadir sınıfları göz ardı edebilir. Formülü şu şekildedir:

metin, yazı tipi, çizgi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 2. Doğruluk formülü

**2.3.2. Hassasiyet (Precision)**, doğru şekilde pozitif olarak sınıflandırılan örneklerin, modelin pozitif olarak tahmin ettiği tüm örneklere oranıdır. Özellikle yanlış pozitiflerin (false positives) önemli olduğu durumlarda kullanılır. Formülü şu şekildedir:

metin, yazı tipi, çizgi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 3. Hassasiyet formülü

**2.3.3. Duyarlılık (Recall) / Tespit Oranı (Sensitivity)**, modelin tüm gerçek pozitif örnekleri doğru şekilde sınıflandırma oranıdır. Yüksek duyarlılık, özellikle yanlış negatiflerin (false negatives) önemli olduğu durumlarda tercih edilir. Formülü şu şekildedir:

metin, yazı tipi, çizgi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 4. Duyarlılık formülü

**2.3.4. F1 Skoru (F1 Score)**, hassasiyet ve duyarlılığın harmonik ortalamasıdır ve dengeli bir değerlendirme sağlar. Özellikle dengesiz veri setlerinde, her iki parametreyi dengeleyerek modelin genel başarımını ölçmeye yardımcı olur. Formülü şu şekildedir:

metin, yazı tipi, beyaz, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 5. F1 skor formülü

**2.3.5. Mean Squared Error (MSE - Ortalama Kare Hatası)**, Regresyon modelin yaptığı tahminlerle gerçek değerler arasındaki farkların karelerinin ortalamasını ifade eder. Her tahmin hatasının karesi alındığı için, büyük hatalar daha çok cezalandırılır. Değer sıfıra ne kadar yakınsa, modelin performansı o kadar iyidir.

yazı tipi, metin, beyaz, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 6. MSE formülü

Düşük MSE: Daha doğru tahminler.

Yüksek MSE: Model, gerçek değerlere uzak tahminler yapıyor olabilir.

**2.3.6. R-squared (R² - Determinasyon Katsayısı)** skoru, Regresyon modelin bağımlı değişkendeki (hedef değişken) değişkenliği ne kadar iyi açıkladığını gösterir. 0 ile 1 arasında bir değer alır:

R² = 1: Model veriye mükemmel uyum sağlamış.

R² = 0: Model, hedef değişkenin varyansını hiç açıklayamıyor (yani rastgele tahmin yapıyor gibi).

Negatif R²: Model, sabit ortalama tahmininden bile daha kötü sonuç veriyor.

**ROC Eğrisi (Receiver Operating Characteristic Curve)**

Makine öğrenmesi modellerinin performansını değerlendirmede kullanılan önemli araçlardan biri de ROC eğrisidir (Receiver Operating Characteristic Curve). ROC eğrisi, sınıflandırma modelinin farklı eşik değerleri altındaki başarı düzeyini görselleştirir ve özellikle ikili sınıflandırma (binary classification) problemlerinde sıkça tercih edilir.

ROC eğrisi, yatay eksende Yanlış Pozitif Oranı (False Positive Rate - FPR) ve dikey eksende Doğru Pozitif Oranı (True Positive Rate - TPR) gösterilerek çizilir:

Doğru Pozitif Oranı (TPR), yani Recall (Duyarlılık): metin, yazı tipi, çizgi, beyaz içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 7. TPR formülü

Yanlış Pozitif Oranı (FPR):

metin, yazı tipi, çizgi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 8. FPR formülü

ROC eğrisi, modelin tüm eşik değerleri boyunca nasıl performans gösterdiğini sunarak, sadece tek bir eşik değere bağlı kalmaksızın genel yeteneğini analiz etme imkânı sağlar. Bu eğrinin altında kalan alan, yani AUC (Area Under the Curve), modelin ayırt edici gücünü özetleyen tek bir değer olarak kullanılır.

AUC değeri 1’e yaklaştıkça, modelin sınıflar arasında ayırt ediciliği yüksek demektir.

AUC değeri 0.5’e yakınsa, modelin ayırt etme yeteneği şansa yakındır.

1. **Literatür Taraması**

Bu tür çalışmalar daha önce yapılmıştı ve onlara örnek getirmek gerekirse:

Abdul Karim vd.in “Phishing Detection System Through Hybrid Machine Learning Based on URL” makalesinde toplanan veri seti üzerinde çeşitli makine öğrenme yöntemlerini kullanarak oluşturulan modellerinin sonuçları karşılaştırıldı. Çalışmalarında kullanılan veri seti, araştırma ve karşılaştırmalı analizler için sıklıkla başvurulan bir kaynak olan Kaggle veri havuzundan temin edilmiştir. Veri seti, 11.000’den fazla web sitesinden elde edilen ve siber güvenlik açısından anlamlı 33 öznitelik içeren toplam 11.054 örnekten (kayıttan) oluşmaktadır. Sonuç olarak en iyi performansı %98,12 doğruluk oranıyla Lineer Regresyon, SVC ve Karar Ağacı yöntemlerin birleşmesiyle oluşturdukları model verdi. Bu yöntem en iyi sonucu verse bile daha geleneksel yöntemlerden biri Rastgele Orman yöntemi %96.8 doğruluk oranıyla çok geride kalmadı.[7]

Mehmet Korkmaz vd.in yaptıkları çalışmada 3 farklı veri seti kullanıldı. Birinci veri setinde, zararsız web siteleri Alexa veri tabanından, phishing (oltalama) siteleri ise PhishTank platformundan elde edilmiştir. İkinci veri setinde, zararsız siteler Common Crawl veri kümesinden, phishing siteleri ise yine PhishTank’tan temin edilmiştir. Üçüncü veri setinde ise zararsız URL’ler hem Common Crawl hem de Alexa kaynaklarından alınmış, phishing URL’leri yalnızca PhishTank veri kaynağından toplanmıştır. Her veri seti için 48 özellik olmak üzere sırayla 83.857, 82.888 , 126.077 örnek bulunmakta. XGBoost (Aşırı Gradyan Artırma), Rastgele Orman(Random Forest), Lojistik Regresyon, En Yakın Komşular (KNN), Destek Vektör Makineleri (SVM), Karar Ağacı, Yapay Sinir Ağı ve Naive Bayes yöntemleriyle oluşturdukları modellerde Rastgele Orman yöntemi üç veri setinde de en iyi sonucu gösterdi. Doğruluk oranları sırayla %94,59; %90,5; %91,26’dır.[8]

Ankit Kumar Jain vd. hazırladıkları makalede mobil cihazlar üzerindeki web oltalama saldırılara karşı makine öğrenme yöntemleri kullanma hakkında çalışmalar yürüttüler. Veri seti olarak Android mobil tarayıcı kullanılarak Alexa’dan popüler web siteleri elde edilerek manuel olarak 2000 zararsız (benign) URL toplanmıştır. Kötü amaçlı URL’ler için ise, mobil odaklı 2000 kötü amaçlı URL OpenPhish ve PhishTank kaynaklarından toplanmıştır. Toplam 4000 örnek olmak üzere 34 farklı özellik tanımlandı. Yaptıkları çalışmalarda Rastgele Orman yöntemi %93,58 doğruluk oranıyla diğer yöntemlerden daha üstün sonuçlar elde etti.[9]

Yukarıdaki örnek çalışmaları baz alarak bu problemde Rastgele Orman yöntemi güzel sonuçlar sunması bekleniyor. Bizim yaptığımız çalışmada, bu hipoteze uygun şekilde, Rastgele Orman yöntemi en güzel sonuçlar gösterdi.

1. **Bulgular ve Tartışma**

**4.1 KNN Yöntemi sonuçları**

KNN yöntemi için farklı komşu sayılarını belirleyerek sonuçlarını karşılaştırdık. Tablo 2’de gördüğünüz gibi komşu sayısı 3 olunca doğruluk oranı %94,33 olmak üzere bize en iyi sonucu verdi. Şekil 9’u incelesek komşu sayısı 3’ken AUC değerinin 0.97508 olduğunu görüyoruz.

Tablo 2. KNN algoritmasının komşu sayısına(n) göre bulunan ölçü sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n değeri | Accurracy | Recall | Precision | F1 score |
| **3** | **%94,33** | **%95,39** | **%94,69** | **%95,04** |
| 4 | %93,10 | %91,95 | %95,76 | %94,35 |
| 5 | %93,55 | %94,65 | %94,06 | %94,35 |
| 6 | %92,46 | %91,48 | %95,10 | %93,25 |
| 7 | %92,95 | %94,02 | %93,62 | %93,82 |
| 8 | %92,79 | %92,54 | %94,69 | %93,60 |
| 9 | %93,28 | %94,34 | %93,89 | %94,11 |

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 9. KNN algoritmasının ROC eğrisi

**4.2 Karar Ağacı Yönteminin Sonuçları**

Karar ağacı yönteminde Gini Algoritması kullanara model oluşturduk. Ayrıca derinlik değerleri değiştirerek farklı sonuçları karşılaştırdık. Tablo 3’te olduğu gibi derinlik 20 olunca doğruluk oranı en iyi sonuçlar veriyor. Şekil 10’da derinlik 20 olunca AUC değeri 0.97019 olduğunu görebiliriz.

Tablo 3. Karar Ağacı algoritmasının derinliğe göre verdiği ölçü sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Derinlik | Accurracy | Recall | Precision | F1 score |
| 10 | %94,45 | %94,71 | %95,52 | %95,11 |
| **20** | **%95,72** | **%96,72** | **%95,80** | **%96,26** |
| 30 | %95,63 | %96,45 | %95,89 | %96,17 |
| 40 | %95,66 | %96,40 | %95,99 | %96,20 |

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 10. Karar Ağacı Algoritmasının ROC eğrisi

**4.3 SVM Yönteminin Sonuçları**

Bu yöntemle Lineer ve RBF çekirdek kullanarak iki farklı model oluşturduk. Tablo 4’teki değerlere baksak RBF çekirdek Lineer çekirdeğe göre daha iyi sonuç verdiğini görebiliriz. Aynı zamanda Şekil 11 ve Şekil 12’de AUC değerleri gözlendiğinde Lineer çekirdeğin değeri 0.97723’ken RBF’nın 0.98718 olmak üzere daha iyi sonuç gösterdi.

Tablo 4. SVM algoritmasının çekirdek(kernel) türüne göre verdiği ölçü sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel | Accurracy | Recall | Precision | F1 score |
| Lineer | %92,40 | %94,02 | %92,74 | %93,38 |
| RBF | %95,24 | %96,66 | %95,05 | %95,85 |

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 11. Lineer çekirdekli SVM algoritmasının ROC eğrisi

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 12. RBF çekirdekli SVM algoritmasının ROC eğrisi

**4.4 Rastgele Orman Yönteminin Sonuçları**

Rastgele Orman modeli için en iyi parametreleri seçmek için sklearn.model\_selection kütüphanesinden GridSearchCV sınıfını kullandık ve parametreleri şu şekilde belirledik:

Max derinlik => Yok,

Ağaç Sayısı => 150.

Bu parametrelerle f1 skoru göz önümüzde bulundurarak en iyi sonuçları üreten model eğittik. Tablo 5’te bu sonuçları gözlemleyebiliriz. Ayrıca Şekil 13’e baktığımızda modelin ROC eğrisinde AUC değeri 0.99508’dir. Bu diğer yöntemlere kıyaslı en iyi sonuçtur.

Tablo 5. Rastgele Orman algoritmasının ölçü sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ağaç Sayısı | Accurracy | Recall | Precision | F1 score |
| 150 | %96,71 | %98,04 | %96,26 | %97,14 |

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 13. Rastgele Orman algoritmasının ROC eğrisi

**4.5 Naive Bayes Yönteminin Sonuçları**

Naive Bayes yönteminde istediğimiz sonuçları maalesef alamadık. Tablo 6’da gördüğümüz gibi %59 doğruluk oranını başardık. Bu da tüm yöntemlerden en kötü sonucu veren olmuştur. Bunun olası nedeni veri setinin dengesiz olması olabilir. Bir “1” sınıfında ağırlık olduğu için Pozitif dediği sonuçlardan kesin emin. Bu precision’un yüksek olmasını sağlıyor. Fakat fazla pozitif değerleri bulmuyor olabilir. O da recall’ın düşük olmasına neden olur. Şekil 14’te de AUC değerinin 0.96471 olduğunu görebiliyoruz.

Tablo 6. Basit Bayes algoritmasının ölçü sonuçları

|  |  |  |  |
| --- | --- | --- | --- |
| Accurracy | Recall | Precision | F1 score |
| %59,00 | %28,16 | %99,44 | %43,89 |

metin, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 14. Basit Bayes algoritmasının ROC eğrisi

**4.6 Regresyon Yönteminin Sonuçları**

Regresyon yöntemi için Lineer Regresyon ve Rastgele Orman Regresyon Yöntemlerini kullandık. Genel olarak Rastgele Orman Regresyon yöntemi Lineer Regresyon’a göre daha iyi sonuçlar gösterdi. Tablo 7’de doğruluk oranı RF Regresyon’da %96,71(Rastgele Orman Sınıflandırıcı ile aynı değerler) iken Lineer Regresyon’da %91,86’dır.

Tablo 7. Regresyon türüne göre çıka ölçü sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Regresyon türü | Accurracy | Recall | Precision | F1 score |
| Lineer | %91,86 | %94,18 | %91,75 | %92,95 |
| RF | %96,71 | %98,04 | %96,26 | %97,14 |

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 15. Lineer regresyonun ROC eğrisi

Fakat Regresyon sonuçları karşılaştırdığımızda asıl bakmamız geren ölçüt R2 ve MSE’dır. Şekil ‘e baktığımızda RF Regresyon yöntemi daha iyi sonuçlar verdi. Lineerin Regresyon modelinde R2 = 0,6969 ve MSE = 0,2979’ken RF Regresyon modelinde R2 = 0,8989 ve MSE = 0,0990’dır. Şekil 16 ve Şekil 17‘de farkları daha iyi gözlemleyebiliriz.

Tablo 8. Regresyon türlerinin R2 ve MSE değerleri

|  |  |  |
| --- | --- | --- |
| Regresyon türü | R2 | MSE |
| Lineer | 0.6969 | 0,2979 |
| RF | 0,8989 | 0,0990 |

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 16. Regresyonların R2 değerleri

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 17. Regresyonların MSE değerleri

**5.** **Sonuç ve Öneriler**

Yaptığımız çalışmalar sonucunda tablo 9’da gördüğümüz gibi %96,71 ile doğruluk oranını en büyük veren yöntem Rastgele Orman Yöntemi oldu. Ayrıca %97,14 ile f1 skoru en yüksek model yine Rastgele Orman yöntemi oldu. Bu da bu problem üzerinde yapılan başka çalışmalardaki sonuçlarla örtüşüyor. Veri seti örnek sayısı ve çeşitliliği artırılsa daha iyi sonuçlar almamız mümkün, özellikle Naive Bayes için sınıf ağırlıkları daha dengeli veri seti lazım olabilir. Ayrıca Karar Ağacı ve Rastgele Orman Yöntemleri özellik artışı ile performans artışı gösterebilir.

Tablo 9. Çalışmamızda yer alan tüm algoritmaların ölçü sonuçlarının karşılaştırılması

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ML Yöntemi | Accurracy | Recall | Precision | F1 score |
| KNN | %94,33 | %95,39 | %94,69 | %95,04 |
| Karar Ağacı | %95,72 | %96,72 | %95,80 | %96,26 |
| SVM Lineer | %92,40 | %94,02 | %92,74 | %93,38 |
| SVM RBF | %95,24 | %96,66 | %95,05 | %95,85 |
| **RF** | **%96,71** | **%98,04** | **%96,26** | **%97,14** |
| Naive Bayes | %59,00 | %28,16 | %99,44 | %43,89 |
| Lineer Regresyon | %91,86 | %94,18 | %91,75 | %92,95 |

**KAYNAKÇA**

1. <https://www.paynet.com.tr/blog/phishing-oltalama-nedir>
2. <https://corporate.comcast.com/press/releases/comcast-business-2024-cybersecurity-threat-report-artificial-intelligence-new-era>
3. <https://blog.cloudflare.com/tr-tr/2023-phishing-report/>
4. <https://keepnetlabs.com/blog/twilio-suffers-phishing-based-data-breach>
5. Rami M. Mohammad, Fadi Thabtah, Lee McCluskey “Phishing Websites”, 2013
6. Abdul Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, S. Ramana Kumar Joga “Phishing Detection System Through Hybrid Machine Learning Based on URL”, 2023, IEEE Access, Volume 11
7. M. Korkmaz, O. K. Şahingöz, B. Diri “Detection of Phishing Websites by Using Machine Learning-Based URL Analysis”, 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT
8. Ankit Kumar Jain, Ninmoy Debnath, Arvind Kumar Jain “APuML: An Efficient Approach to Detect Mobile Phishing Webpages Using Machine Learning”, 2022, Wireless Personal Communications, Volume 125, pages 3227–3248, (2022)
9. <https://github.com/BayramEnesAtay/BMT218_Proje_PhishingWebistes>