T.C. DÜZCE ÜNİVERSİTESİ

İŞLETME FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

2023-2024 AKADEMİK YILI

YBS302 BİLGİSAYAR AĞLARI ÖDEVİ

Makine Öğrenmesi yöntemleri aracılığıyla oltalama (phishing) web sitelerinin tespiti ve ilgili algoritmaların karşılaştırılması: USOM örneği

Hazırlayanlar:

Cankat Çakmak
Kübra Yılmaz
Elif Şahin

Özet:

Günümüzün gelişen siber teknolojisi, oltalama saldırılarını göz ardı edilemeyecek bir seviyeye getirmiştir. Her geçen gün hızla artan siber suçların önde gelen yöntemlerinden biri olan oltalama saldırıları, birçok şirket ve son kullanıcı için tehdit oluşturmaktadır. Bu saldırılarda, saldırganlar sahte siteler oluşturarak çeşitli sosyal mühendislik yöntemleriyle kurbanlara ait kredi kartı, hassas belgeler, parolalar gibi hassas bilgileri kolayca elde edebilirler. Bu çalışmanın amacı, son yıllarda adını sıkça duyduğumuz ve hızla gelişmekte olan yapay zekâ teknolojilerini kullanarak makine öğrenmesi yöntemi ile oltalama sitelerinin tespit edilmesini sağlamaktır. Oltalama sitelerinin sınıflandırılmasında, scikit-learn ve Weka adlı iki farklı makine öğrenmesi yazılımı kullanılmıştır. Scikit-learn ile 96 bin (48,009 bin oltalama ve 48,009 bin meşru) ve Weka ile 3 bin (1500 oltalama ve 1500 meşru) olmak üzere 6 farklı özniteliğe sahip iki farklı veri seti test edilmiştir. Çalışmanın sonucunda, scikit-learn ile Rastgele Orman (RO) algoritmasında %99,1 doğruluk oranı elde edilirken Weka ile Destek Vektör Makinesi (DVM) algoritmasında %99,06 doğruluk oranı elde edilmiştir. Bu çalışmanın sonucu, veri miktarı arttıkça SVM algoritmasının doğruluk oranı azalırken Rastgele Orman algoritmasının doğruluk oranın arttığını göstermektedir.

Anahtar kelimeler: Makine öğrenmesi, oltalama, yapay zekâ, scikit-learn, Weka, oltama tespiti, python, rastgele orman, desktek vektör makinesi, SVM, karar ağacı, USOM, Ulusal Siber Olaylara Müdahale Merkezi, phishing

1- Giriş

1940'ların ortasından bu yana gelişen bilgisayar teknolojisi ile önceleri Amerika Birleşik Devletleri'nin iletişim aracı olan ve ABD Savunma Bakanlığı tarafından yürütülen ARPANET projesi, günümüzde yerini "uluslararası çalışma ağı" olarak da tanımlanan Internet'e ve onun en büyük bilgi kaynağı olan web teknolojilerine bıraktı (Soysal, 2006: 484-486). Teknolojide yaşanan bu devrim ile web sitelerinin yayılımı, birçok insanın bilgiye erişimini kolaylaştırmayı sağlarken, çeşitli suçlular için yeni bir alan açılmasına da neden olmuştur. Günden güne artan teknoloji kullanımı, beraberinde tehdit aktörlerinin veya siber suçluların yeni yöntemler keşfetmesini de sağlamıştır. Bu yöntemlerin başında gelen oltalama saldırıları, "web uygulamaları üzerinde kullanıcıların kişisel bilgilerini, banka kartı veya kredi kartı bilgilerini, sosyal medya bilgilerini, şifrelerini almaya yönelik hazırlanmış eski ve etkili elektronik dolandırıcılık yöntemlerinden biridir." (Toğaçar, 2021: 1603). Temel olarak, oltalama saldırılarında gerçekte var olan ve genellikle kullanıcılarının kişisel bilgilerini taşıyan bir web sitesinin bir kopyasının oluşturulup içine kullanıcıların bilgilerini ele geçiren çeşitli zararlı yazılım kodları eklenir ve çeşitli sosyal mühendislik yöntemleri aracılığıyla hedef kullanıcıların bu web adresine giriş yapılması beklenmektedir. Son dönemde oltalama saldırılarının en büyük örneklerinden bir tanesi 2016 yılında bir grup saldırgan tarafından gerçekleştirildi. Bu saldırganlar, New York Federal Bankası alt yapısını ve SWIFT ağını kullanan Bangladeş Bankasını hedef alan bir oltalama saldırısı düzenlediler ve başarıya ulaştılar. Sistemin kontrolünü ele geçiren saldırganlar, yaklaşık olarak 1 milyar ABD dolarını Sri Lanka ve Filipinler'de bulunan hesaplarına gönderen otuz beş farklı talimat yayınladılar. Bu talimatların büyük bir kısmı SWIFT ağının onayından geçmese bile beş tanesi başarıya ulaştı ve saldırganlar 20 milyonu Sri Lanka'ya ve 81 milyonu Filipinler'e olmak üzere toplamda 101 milyon ABD doları çalmayı başardılar. Eğer SWIFT ağı kalan otuz işlemi engelleyemeseydi, Bangladeş bankası toplamda 850 milyon ABD doları kaybedecekti. Yetkililer, talimatlar gönderildikten bir süre sonra Sri Lanka'nın transfer işlemlerindeki uyumsuzluk nedeniyle durumu erkenden fark etti ve transfer edilen tüm paranın kurtarılmasını sağladılar. Ancak, Filipinler'e transfer edilen paranın yalnızca 18 milyonu kurtartılabildi (Bukth & Huda, 2017 akt. Hossain vd., 2020: 378). Geçmişten günümüze oltalama saldırıları ile ilgili birçok araştırma yapılmıştır. Elde edilen nicel bulgulardan bazıları şu şekildedir: “McAfee tarafından yapılan 2016 raporunda 2015 yılı boyunca 100 milyona yakın yeni kötücül URL tespit edilmiştir. Ayrıca 2015 yılının dördüncü çeyreğinde 1.5 milyona yakın yeni kimlik avı URL’i gözlemlenmiştir." (Koşan vd., 2017: 276). "Verizon adlı firmanın 2019 yılı veri araştırmaları raporuna göre veri ihlallerinin yaklaşık %33,33 oranında oltama saldırılarından kaynaklandığını belirtmektedir. Amerika Federal Soruşturma Bürosu'nun İnternet Şikâyet Birimi'nin istatistik bilgilerine göre oltalama saldırıları ile işlenen suçların 2016-2019 yılı arasında vermiş olduğu maddi kayıp dünya genelinde 26 milyar $ üzerinde olduğunu belirtti.“ (Abdelhamid, 2020 akt. Toğaçar, 2021: 1604). Buna ek olarak, Anti-Phishing Working Group (APWG), 45 farklı ülkeden aldığı istatistksel sonuçları

kullanarak oltalama saldırıları aracılığıyla en çok etki altında bulunan ülkelerde Çin'den sonra %42.88'lik bir oran ile Türkiye'yi 2nci en çok etkilenen ülke olarak belirtmiştir (Buber vd,, 2017: 609). Bu çalışmanın amacı, gittikçe popülerleşen oltalama saldırılarına karşı, bu saldırıların barındırdığı spesifik veriler dikkate alınarak tahminleme ve tespit yapabilecek bir makine öğrenmesi modeli ortaya koymaktır. Bu model, üç farklı makine öğrenmesi algoritması olan Karar Ağacı (KA), Rastgele Orman (RO) ve Destek Vektör Makinesi (DVM) yöntemlerinin karşılaştırması esas alınarak oluşturulmuştur. İlgili yöntemlerin detayları 4. Araştırma Yöntemleri ve Makine Öğrenmesi Algoritmaları bölümünde detaylandırılmıştır. Buna ek olarak bu çalışma altı bölümden oluşmaktadır, ikinci bölümde literatür araştırmalarının incelenmesine, üçüncü bölümde veri seti bilgilendirmesine ve veri ön işleme adımlarına, dördüncü bölümde kullanılan makine öğrenmesi algoritmalarına, beşinci bölümde yazılım kütüphanesinin seçilmesine ve ilgili algoritmaların modellenmesine, altıncı bölümde algoritmaların karşılaştırılmasına ve sonuç kısımlarına yer verilmiştir.

2- Literatür Taraması

Miyamoto vd. 2008 yılında gerçekleştirdikleri çalışmalarında AdaBoost, Bagging, Destek Vektör Makineleri, Sınıflandırma ve Regresyon Ağaçları, Lojistik Regresyon, Rastgele Ormanlar, Sinir Ağları, Naif Bayes ve Bayesian Additive Regresyon Ağaçları dahil olmak üzere 9 makine öğrenmesi yöntemini kullanmışlardır. Bu yöntemler ile yapılan analizlerin performans ölçütleri; f1 ölçüsü, hata oranı ve ROC Eğrisi Altında Alan (AUC) olarak belirlenmiştir. Veri seti, eşit sayıda oltalama ve meşru site içeren URL'ler bir araya getirilerek oluşturulmuştur. Toplamda 3.000 adet veri içeren bu veri seti üzerinde yapılan analizler sonucunda tüm algoritmalar karşılaştırılmış ve Adaboost algoritması; f1 ölçüsü 0.8581, hata oranı 0.1415, AUC 0.9342 değerlerini elde ederek en iyi performansı sergileyen algoritma olarak gözlemlenmiştir (Miyamoto vd., 2008: 7).

James vd. 2013 yılında yapmış oldukları çalışmalarında naive bayes, J48 karar ağacı, k-en yakın komşu, destek vektör makinesi gibi çeşitli sınıflandırma algoritmalarını kullanmışlardır. Bu algoritmaların karşılaştırılması ve değerlendirilmesi için WEKA ve MATLAB platformları kullanılmıştır. Bu çalışmada kullanılan veri seti toplamda, 17.000 oltalama URL'si ve 20.000 güvenli URL içermektedir. Bu veri setinin %40'ı eğitim, %60'ı ise test için kullanılmıştır ve algoritmalar karışıklık matrisi, başarı oranı, hata oranı gibi ölçütlerle karşılaştırılmıştır. Çalışmanın sonucunda; WEKA üzerinden yapılan analizlerde en iyi başarı oranının 93.78 ile karar ağacı algoritmasından elde edildiği görülmüştür (James vd., 2013: 308). MATLAB üzerinden yapılan analizler sonucunda da yine 91.08 başarı oranı ile en iyi performans karar ağacı algoritması ile gözlemlenmiştir (James vd., 2013: 308).

Bu makale, Akinyelu & Adewumi tarafından 2014 yılında aldatıcı uygulamaların inceliklerini ve bu saldırıları önlemek için gereken koruma önlemlerini araştırmıştır. Makale, bilgisayar korsanları ile sosyal mühendisler arasındaki ayrımı belirleyerek başlamakta ve güvenlik sistemlerini ihlal etme girişimlerinde kullanılan farklı yaklaşımları açıklamaktadır. Ayrıca, sosyal mühendislik mağdurlarının sergilediği ortak özelliklerin dikkatlice incelenmesi, kötü niyetli aktörlerin istismar ettiği güvenlik açıklarının ortaya çıkarılmasına katkı sağlamaktadır. Makalenin sonuçlarına göre, random forest sınıflandırıcısı kullanılarak gerçekleştirilen phishing tespit yöntemi, %99.7 doğruluk oranı elde etmiştir (Akinyelu & Adewumi , 2014: 4). Bu yüksek doğruluk, eğitim aşamasındaki bilgi kazancı hesaplamaları sayesinde belirlenen en iyi sekiz özellikle oluşturulan karar ağaçlarıyla başarıyla

sınıflandırma yapılmasından kaynaklanmaktadır. Yapılan 10 katlı çapraz doğrulama testlerinde, en büyük veri setinde elde edilen %0.06 yanlış pozitif oranı ve %2.50 yanlış negatif oranı, yöntemin gerçek dünya veri setlerinde etkili bir şekilde çalışabileceğini göstermektedir. Bu çalışma, mevcut phishing tespit tekniklerinin ötesinde bir içerik tabanlı yaklaşım sunarak, siber güvenlik alanında önemli bir adım olarak değerlendirilebilir. Gelecekteki araştırmalar, doğa ilhamlı tekniklerin entegrasyonuyla phishing özelliklerini dinamik olarak belirleyerek sınıflandırıcıyı daha da iyileştirmeyi hedeflemektedir. Bu, siber güvenlik önlemlerinin phishing saldırılarındaki değişen modellere daha etkili bir şekilde adapte olmasına katkı sağlayabilir. Manipülasyon taktiklerinden oltalama saldırılarına kadar çeşitli stratejilere odaklanarak makale, bireyleri ve kurumları bu tür tehditlere karşı güçlendirmek için acil koruma önlemlerinin gerekliliğini vurgulamaktadır. Sonuç olarak, makale, dijital ortamda daha güvenli bir gelecek için sürekli çaba ve bilinçlendirme gerekliliğini vurgulamaktadır (Akinyelu & Adewumi, 2014 : 4-5).

Moghimi ve Varjani, 2016 yılında yeni bir kural tanımına dayanan bir oltalama tespit modeli ortaya koymuşlardır. Çalışmada açıklanan bu yeni model, oltalama adreslerini iki yeni özellik setini önermektedir (Moghimi & Varjani, 2016: 233). Önerilen ilk özellik setinde, sayfa içerik ile URL arasındaki ilişkiyi belirleyen dize eşleme algoritmaları kullanılmıştır. Bu adreslerin sınıflandırılmasında Destek Vektör Makinesi (SVM) algoritması kullanılmıştır (Moghimi & Varjani, 2016: 236). Moghimi ve Varjanı, kural tabanlı bu modelin sonuçlarında %99,14 doğru tespit oranı ve 0,086 hata yapıyla bankacılık oltalama adreslerini doğru olarak sınıflandırabilmiştir (Moghimi & Varjani, 2016: 241).

Koşan vd. 2017 yılında yaptıkları çalışmalarında, Weka yazılımını kullanarak, Phistank web sitesinden aldıkları bir veri seti üzerinden 11055 örnek ile oltalama web sitelerinde kullanılmak üzere C4.5, ID3, RİPPER, Naif Bayes, k-En Yakın Komşu, Rastgele Ormanlar algoritmalarının karşılaştırmalı analizini gerçekleştirmişlerdir (Koşan vd., 2017: 277-279). Onlar, çalışmalarında ilgili algoritmaların yalnızca başarı oranlarını değil aynı zamanda kendilerinden önceki literatür çalışmalarına ek olarak ilgili algoritmaların performans ölçümlerini de çalışmalarına eklemişlerdir. Yapılan deneylerin sonucunda Rastgele Orman yönteminin %97.3 ve ID3 algoritmasının %96.5 oranında başarılı sonuç verdiği elde edilmiştir. Buna ek olarak, ilgili makine öğrenmesi modelinin oluşturulması ve tahminleme süreçlerinde en başarılı sonucu ~0 ile Naif Bayes yöntemi elde etmiştir. Ancak Naif Bayes yönteminin başarılı oranının %93.0 ile diğer tüm algoritmaların gerisinde kaldığı görülmektedir (Koşan vd., 2017: 280-281).

Buber vd. 2017 yılında yaptıkları çalışmalarında, ilgili diğer Makine Öğrenmesi yöntemlerinden farklı olarak Rastgele Orman (RO), Sıralı Minimum Optimizasyon (SMO), Naif Bayes (NB) algoritmalarını Doğal Dil İşleme Özellikleri, Kelime Vektörleri ve Hibrid adını verdikleri üç farklı test yöntemi ile test etmişlerdir. Amaçları oltalama tespit sistemi olmasa bile hedeflenen tipteki URL adreslerini tespit etmeye çalışmaktır. Test amaçlı kullanılacak veri seti Phistank web sitesinden alınan 37.175 adet zararlı oltama adresi ve 36.400 adet yasal adres olmak üzere toplam 73575 adet URL barındırmaktadır (Buber vd., 2017: 4-5). Çalışmadan elde edilen sonuçlara göre en başarılı yöntem %89,9 kesinlik ile Hibrid yönteminin Rastgele Orman algoritması olmuştur. Buna ek olarak, çalışmanın sonucunda en düşük başarılı oranına %59 Doğal Dil İşleme yönteminin Naif Bayes (NB)

algoritması üzerindeki testiyle ulaşılmıştır. İlgili algoritmalar üzerindeki diğer test yöntemlerinin bu aralıklarda olduğu saptanmıştır (Buber vd., 2017: 6).

Aksu vd. 2017 yılında bu makalede kullanıcıları sahte ve gerçek URL'ler arasındaki farkı anlamada zorlananlara yardımcı olmak amacıyla oluşturulmuş birçok kurallı phishing tespit sistemi bulunduğunu açıklamaktadır. Çalışma, PhishTank'ten alınan phishing URL'lerini ve gezinme geçmişinden alınan non-phishing URL'lerini içeren bir veri seti oluşturarak başlamaktadır. Özellik seçimi, URL uzunluğu, nokta sayısı, IP adresi kullanımı, SSL bağlantısı, "@" sembolü varlığı ve çizgi sembolü görünümü gibi temel karakteristikleri belirlemeyi içerir. SVM, bu özelliklere dayalı olarak ikili sınıflandırma (phishing veya non-phishing) için kullanılmaktadır. Sistemin tasarımı, veri seti oluşturma, özellik seçimi, özellik çıkarma, SVM eğitimi ve test etme adımlarını içerir (Aksu vd., 2017: 140). Sonuçlar, karışıklık matrisinde sunulduğu gibi, gerçek phishing durumlarının %88 oranında tespit edildiğini ve non-phishing durumlarının %97 doğrulukla tanımlandığını göstermektedir. Doğruluk, hassasiyet ve F1 skoru gibi performans metrikleri, sistemimizin etkinliğini gösterir ve sırasıyla %95, %88, %91.66 ve %89.79 oranlarını sergiler (Aksu vd., 2017: 141).

Bu çalışma, Aydın tarafından 2018 yılında kalabalık şehirlerde itfaiye istasyonlarının doğru yer seçimini, yangınlara hızlı müdahale etmeyi ve can-mal kaybını en aza indirmeyi amaçlamaktadır. Mevcut itfaiye istasyonlarından yola çıkarak makine öğrenmesi algoritmaları kullanılarak bölgelere göre itfaiye istasyonu ihtiyacının sınıflandırılması yapılmıştır. İzmir Büyükşehir Belediyesi'nin 808 bölgesi üzerinden yapılan çalışmada, 2015-2017 yangın kayıtları analiz edilerek en başarılı sınıflandırma algoritmasının %93.84 doğruluk oranıyla Random Forest olduğu belirlenmiştir (Aydın, 2017: 173). Veri seti, itfaiye araçlarının ulaşım süreleri, nüfus yoğunluğu, bölgeye giden ortalama araç sayıları gibi faktörleri içermekte ve bu veriler kullanılarak istasyon ihtiyacının tahmini yapılmaktadır. En iyi performansı gösteren algoritmanın belirlenmesi için doğruluk, ortalama mutlak hata (MAE), kök hata kareler ortalaması (RMSE) ve Kappa değerleri kullanılmıştır. Sonuçlar, mekânsal verilerin makine öğrenmesi yöntemleriyle itfaiye istasyonu ihtiyacını sınıflandırmada etkili olduğunu göstermiş ve gelecekteki çalışmalarda trafik yoğunluğu, yol genişliği gibi ek verilerin eklenerek daha kesin sonuçlara ulaşılabileceği öne sürülmüştür (Aydın, 2017: 171).

Bu makale, Çelik & Altunaydın tarafından 2018 yılında yapay zeka alanının bir alt dalı olarak 1950'lerde gelişen ve geliştirilen makine öğreniminin tarihini, kullanılan yöntemleri ve uygulama alanlarını ele almaktadır. İlk makine öğrenimi adımları 1950'lerde atılmış, ancak bu alanda önemli araştırma ve geliştirmeler olmamıştır. Ancak, 1990'lı yıllarda makine öğrenimi üzerine yapılan çalışmalar tekrar başlamış, geliştirilmiş ve günümüze gelinmiştir. Bu gelişmenin arkasındaki neden, hızla artan veriyi analiz etme ve işleme zorluğudur. Makine öğrenimi, esasen artan veri sayesinde yeni veri için en iyi modeli bulma prensibine dayanmaktadır. Bu nedenle, makine öğrenimi çalışmalarının verinin artmasıyla paralel olarak devam edeceği öngörülmektedir. Makine öğreniminin tanımı, öğrenmenin makineler tarafından gerçekleştirildiği ve bu sürecin, veri ve deneyim kullanarak insan beyni gibi öğrenme yeteneği kazandırma süreci olduğunu belirtmektedir. Makine öğreniminin ana amacı, kendini eğitebilen modeller oluşturmak, karmaşık desenleri algılamak ve önceki verileri kullanarak yeni problemlere çözüm bulmaktır. (Çelik & Altunaydın, 2018: 26) Sonuç olarak, makine öğrenimi, yapay zeka alanındaki önemli bir alt alan olarak gelişmiş ve

evrimleşmiştir. Tarihsel olarak 1950'lerde başlayan bu süreç, özellikle 1990'lardan itibaren hız kazanmış ve günümüze kadar gelmiştir. Veri analizi ve işleme zorluğu, makine öğrenimi çalışmalarının ivme kazanmasının arkasındaki ana sebeplerden biridir. Makine öğrenimi, denetimli ve denetimsiz öğrenme gibi çeşitli yöntemlere dayanmaktadır. Denetimli öğrenme, mevcut verilerin kullanılarak belirli sonuçlara ulaşma amacını taşırken, denetimsiz öğrenme, veri setindeki ilişkileri kullanarak öğrenme gerçekleştirir. Bu yöntemler, sınıflandırma, kümeleme, regresyon ve ilişkilendirme gibi farklı görevleri yerine getirir. Bu bağlamda, bilgi teknolojisi ve makine öğrenimi güçlü bir şekilde ele alınmalı ve bu teknolojik gelişmelere uyum sağlanmalıdır. Makine öğrenimi, hem iş dünyasında hem de bireylerin günlük yaşamlarında önemli bir rol oynamaya devam edecektir (Çelik & Altunaydın, 2018: 34).

Ravi vd. 2018 yılında oltalama e-mailleri ile kimlik hırsızlığı ve zararlı bağlantılara yönlendirme yapan avcıların tespiti için yaptıkları çalışmalarında rastgele orman, adaboost, naive bayes, karar ağacı, destek vektör makinesi gibi makine öğrenmesi algoritmalarını kullanmışlardır. Çalışma için toplamda 18.778 örnek içeren bir veri seti kullanılmış ve bu verilerin 10.283 adeti eğitim, 8.495 adeti ise test süreçleri için kullanılmıştır. Verilerin bölünmesi için Python dili ile scikit-learn kütüphanesi kullanılmıştır. Araştırmanın sonuçlarına göre; test miktarı arttıkça sonuçlarda değişimler görülse de destek vektör makinesi %98.7 doğruluk oranıyla en iyi performansı sergileyen algoritma olarak gözlemlenmiştir (Ravi vd., 2018: 4).

Kalaycı, 2018 yılında yapmış olduğu çalışmasında içerisinde 1353 örnek bulunan, bu örneklerin her biri için 9 özellik ve ait olduğu sınıf bilgisinin yer aldığı, Abdelhamid vd. tarafından ilişkili sınıflandırma veri madenciliği yöntemiyle elde edilmiş olan website phishing hazır veri kümesi kullanılmıştır. Bu veri kümesinin %70'i eğitim, %30'u ise test işlemleri için kullanılmıştır. Çalışmada veri kümesi üzerinde, sınıflandırma algoritmaları olan; AdaBoost, çok katmanlı algılayıcı, destek vektör makinesi, karar ağacı, en yakın k komşu, naive bayes ve rastgele orman algoritmaları kullanılmıştır. Algoritmaları ölçmek için hata matrisi kullanılmış, bu matrisin ölçütleri; doğruluk, kesinlik, geri çağırım, f ölçütü ve ROC AUC olarak belirlenmiştir. Elde edilen verilere göre tüm karşılaştırma ölçütlerinde en iyi sonucun rastgele orman algoritması tarafından alındığı görülmüştür ancak Tahir Kalaycı bu ölçütlerin yanında algoritmaları çalışma sürelerine göre de karşılaştırmıştır ve çalışma süreleri açısından yapılan karşılaştırmalarda en hızlı eğitimi gerçekleştiren algoritma naive bayes, en hızlı sınamayı gerçekleştiren algoritma ise karar ağacı olmuştur (Kalaycı, 2018: 876). Rastgele orman, karar ağaçlarının hızlı eğitim ve kestirim özelliklerinden yararlanarak ve bunun yanı sıra karar ağaçlarının aşırı uyum problemlerine uygun bir çözüm bularak en iyi sonuçları üretmeyi başarmış, %89 oranı ile diğer algoritmaları gerisinde bırakmıştır (Kalaycı, 2018: 875).

Mahajan & Siddavatam 2018 yılında yaptıkları çalışmada karar ağacı, rastgele orman ve destek vektör algoritmalarını; doğruluk oranı, yanlış pozitif oranı, yanlış negatif oranı gibi ölçütler ile karşılaştırmışlardır. Bu çalışma için ALEXA ve Phish Tank platformları üzerinden toplanan; 17,058'si güvenli, 19,653'ü oltalama olmak üzere 36,711 adet URL içeren bir veri seti kullanılmıştır. Elde edilen bulgulara göre, rastgele orman algoritması

%97.14 doğruluk oranı ve 3.14 yanlış negatif oranı ile en verimli sonuçları elde eden algoritma ilan edilmiştir. (Mahajan & Siddavatam, 2018: 47)

Kulkarni ve Brown, 2019'da yaptıkları çalışmalarında, 1,353 gerçek web adresini içeren bir veri seti üzerinden Karar Ağacı, Naif Bayes, Destek Vektör Makinesi (SVM) ve sinir ağları algoritmalarını kullanarak makine öğrenmesi yöntemleriyle oltalama adreslerini tespit etmeye çalıştılar. Çalışmalarının sonucunda sınıflandırıcı algoritmaların %90'dan fazla başarı oranıyla oltalama adreslerini tespit ettiğini ortaya koymuştur (Kulkarni & Brown, 2019: 12-13).

Şahin ve Chouseinoglou, 2019 yılında yaptıkları çalışmalarında, Alexa Top Sites aracılığıyla hazırlanmış olan ve 63 farklı parametre içeren 45.543 adet web sayfasıyla (en fazla kelime sayısı içeren ilk 10 sınıf hedeflenmiş) yaptıkları araştırmalarında bulguları İkili Sınıflandırma Algoritmaları ve Çok Sınıflı Sınıflandırma Algoritmaları olarak iki farklı alt başlık altında ayırmıştır. İkili Sınıflandırma kategorisinde kullanılan yöntemler Tam Bağlantılı Yapay Sinir Ağları, Lojistik Regresyon ve Bernoulli Naif Bayes olmak üzere üç farklı algoritma seçilmiştir. İlgili üç algoritma kendi içinde başarım, kesinlik, duyarlılık ve F1 Skor olmak üzere performans ölçümleri üzerinden değerlendirilmiştir. Çok Sınıflı Sınıflandırma kategorisinde Çok Terimli Naif Bayes, Rastgele Orman ve DVM algoritmaları belirlenmiş ve belirlenen algoritmalar TF-IDF, BOW ve Word2Vec isimli kelime vektörleştirme yöntemleri aracılığıyla test edilmiştir (Şahin & Chouseinoglu, 2019: 34-36). Elde edilen testler sonucunda İkili Sınıflandırma yaklaşımının belirlenen Lojistik Regresyon algoritmasının %98 oranında en başarılı sonucu verdiği tespit edilmiştir (Şahin & Chouseinoglou, 2019: 40-41).

Korkmaz & Büyükgöze, 2019 yılında yapmış oldukları çalışmada sınıflandırma algoritmaları olan rastgele orman, destek vektör makinesi, J48, K-en yakın komşu ve naive bayes algoritmalarını kullanmıştır. Çalışmada veri analizleri, R programlama dili kullanılarak bir web sitesine ait seçilmiş 10 özellik üzerinde çapraz endüstri standart süreç modeli ile gerçekleştirilmiştir. Veri seti, Machine Learning Repository (UCI)'ye bağışlanan verilerden olup, Phish Tank arşivi ve Yahoo'dan PHP'de geliştirilen bir script dosyası ile toplanmıştır. Bu veri seti 1353 özelliğe, 10 niteliğe sahiptir ve verilerin 548'i meşru site, 702'si kimlik avı, 103 URL ise şüphelidir. Yapılan çalışmaların sonucunda, algoritmaların karşılaştırılması karşılık matrisi ile açıklanmış ve hedef sınıf olarak "Meşru" sınıfı seçilmiştir. Bu doğrultuda elde edilen verilere göre; en yüksek doğruluk oranı rastgele orman algoritması tarafından, en düşük doğruluk oranı ise naive bayes algoritması tarafından kazanılmıştır. Kappa istatistik değerlerine göre aralık değeri baz alındığında da yine en iyi sonucun rastgele orman ve J48 algoritmaları tarafından elde edildiği görülmüştür. Genel olarak belirleyicilik, duyarlılık, f-ölçütü, negatif etiketli ve pozitif etiketli sınıflara ait verilerin tahmin edilme oranları gibi ölçütlerin de sonuçlarında en yüksek oranlar rastgele orman algoritmasına aittir. Çalışmanın sonucunda diğer algoritmalara göre çok daha fazla kök ayrımlarını ayırt edebildiği görülen rastgele orman algoritması %95 oranında doğruluğa sahip olarak en iyi sonucu yansıtmıştır (Korkmaz & Büyükgöze, 2019: 833).

Hossain vd. 2020 yılında yayınladıkları araştırmalarında, Mendeley'in çevrimiçi deposundan elde ettikleri 48 farklı özniteliğe sahip, 5000 oltalama ve 5000 meşru web sitesi olmak üzere 10000 farklı web adresini KNN, Karar Ağaçları, Rastgele Orman, DVM ve Lojistik Regresyon algoritmalarını kullanarak test ettiler (Hossain vd., 2020: 381). Onlar, çalışmalarının sonucunda test edilen algoritmalar arasından en başarılı sonucun F1 puanı

ölçütü (kesinlik ve duyarlılık değerlerinin harmonik ortalaması) ile Rastgele Orman yöntemi olduğu sonucuna ulaştılar. Çalışmanın sonucunda Rastgele Orman yönteminin %99 başarı oranı ile sonuç verdiğini ve F1 puanı gibi birçok performans ölçütü açısından en başarılı yöntem olduğunu ifade ettiler (Hossain vd., 2020: 387).

Toğaçar, 2021 yılında yaptığı araştırmasında, Python programlama diline ait Sklean (scikit-learn) kütüphanesi üzerinde 30 farklı parametreye sahip bir veri seti ile deneysel çalışmalar yapmış ve Destek Vektör Makineleri, k-En Yakın Komşu, Karar Ağacı ve Rastgele Ormanlar algoritmalarının testlerinde çapraz doğrulama değerini (k=5) olarak almıştır (Toğaçar, 2021: 1605-1606). Toğaçar'ın deneysel çalışmaları sonucunda, doğruluk oranı ve performans kriterleri açısından en başarılı algoritma Rastgele Ormanlar olarak saptanmıştır. Rastgele Ormanlar yöntemi ile elde edilen başarı oranı %96.53 ile ilgili yöntemin diğer metrik başarında özgünlük başarısı %94.86, duyarlılık başarısı %95.95, geri çağırma başarısı %97.88 ve F1-Skor başarısı %96.90 olarak saptanmıştır (Toğaçar, 2021: 1613).

Balogun vd. 2021 yılında oltalama adreslerinin tespiti için Rotasyon Ormanı temelli Lojistik Model Ağaçları (RF-LMT) yöntemini benimsemişlerdir. Temel alınan bu algoritma modelini incelemek için dengeli ve dengesiz olarak farklı değerlere ve dağılımlara sahip olan üç farklı veri seti kullanmışlardır (Balogun vd., 2021: 158-159). Yapılan deneysel çalışmalar sonucunda, LMT'nin belirlenen sınıflandırıcılara kıyasla ortalama bir performans çıktısı alınmıştır. Ancak, RF-LMT yönteminin oltalama sitelerini tespit etme aşamasında başarılı bir orana sahip olduğu görülmüştür. RF-LMT'nin en yüksek doğruluk oranı %98,24 olarak belirlenmiştir. Buna ek olarak, F ölçütü 0,982 olarak elde edilmiştir (Balogun vd., 2021: 163-164).

Bu araştırma makalesi, Sevgen & Tanrıvermiş tarafından 25 Haziran 2020 yılında, son yıllarda birçok alanda geleneksel yöntemlerle yapılan çalışmaların, Makine Öğrenimi olarak adlandırılan yöntemlerle değiştirildiği bir dönemde gerçekleştirilen bir çalışmayı ele almaktadır. Özellikle, veri boyutu büyüdükçe, Makine Öğrenimi yöntemlerini kullanarak hız ve performans gibi etkilerin arttığı, Büyük Veri adı verilen yeni bir araştırma dalı ortaya çıkmıştır. Bu makale, emlak biliminde değerleme işleminin bireysel olarak veya bir arada gerçekleştirilmesine odaklanmaktadır. Birçok makine öğrenimi tekniği kullanılarak kitle değerleme çalışmaları yapılmıştır. Yapay sinir ağları algoritması kullanılarak kitle değerleme çalışması gerçekleştirildi. Bu çalışmadan sonra, YSA algoritması kullanılarak gerçekleştirilen kitle değerleme çalışmalarının sayısı artmıştır. Birçok çalışmada, araştırmacılar YSA'nın emlak değerlemede iyi sonuçlar verdiğini gözlemlemişlerdir Bununla birlikte, bazı araştırmacılar YSA'nın klasik istatistik yöntemlerine (hedonik, regresyon) göre çok fazla katkıda bulunmadığını karşılaştırmışlardır. Araştırmacı, YSA makine öğrenimi yöntemini kullanarak kitle değerleme çalışması gerçekleştirdi. ABD'nin Kentucky eyaletine bağlı Louisville'de 16.366 veri örneği ve bu örneklerin 18 farklı değişkeni toplandı. Sonuçlarına göre, farklı değişkenlerle farklı senaryolar ürettiler ve heterojen olmayan değişkenlerin dağılımının homojen dağılıma göre daha iyi sonuçlar verdiğini gözlemlediler. İstanbul ilinin Çekmeköy ilçesinde 100 daire ve bunların alan, oda sayısı, bina yaşı, asansör varlığı, konum vb. gibi 12 değişkeni kullanarak YSA yöntemiyle kitle değerleme çalışması gerçekleştirildi ve kitlesel değerleme konusunda güvenilir sonuçlar elde edildi. YSA'nın kitlesel değerlemede başarılı olmasının ardından, araştırmacılar kitlesel değerleme çalışmaları için diğer makine öğrenimi algoritmalarını denemeye başlamışlardır. Örneğin, Amerika Birleşik Devletleri (ABD) emlak satış sitelerinden topladığı verilerde Rastgele Orman, Destek Vektör

Makineleri (SVM), YSA ve çoklu regresyon yöntemlerini kullanarak gerçekleştirilen bir emlak değerleme çalışmasında, RO algoritmasının SVM ve YSA'ya göre daha iyi sonuçlar verdiğini gözlemledi. 100 mülkiyet ve 12 değişken (alan, yaş, ısıtma türü, kanalizasyon sistemi ve evin malzemesi gibi) kullanılarak gerçekleştirilen bir kitlesel değerleme çalışmasında, lineer regresyon modeli Destek Vektör Makineleri (SVM) ile karşılaştırıldı ve lineer regresyon sonuçlarının SVM sonuçlarından daha iyi olduğu bulundu. Bulanık mantık tekniğini kullanarak nominal değerleme sonucunda arazi değer haritaları üretti. Sonuçlarına göre, yazarlar, bulanık mantık tekniğinin kitlesel değerleme çalışmaları için uygun olduğunu gözlemledi. 2006-2017 tarihleri arasında 16,601 örneği içeren bir çalışmada, dairelerin 26 parametrelik ve RO yaklaşımıyla gerçekleştirilen bir çalışmada, RO yönteminin hedonik modellere iyi bir tamamlayıcı olabileceği bulundu. Araştırmacılar, 21 çalışmanın 16'sında YSA tekniğinin, Hedonik Fiyatlandırma Modeli tarafından ya aşıldığını ya da iyi bir alternatif olduğunu belirttiler, yalnızca birkaç çalışma YSA modelinin öngörüsel doğruluğunu gözlemledi. Özellikle RF algoritmalarını diğer makine öğrenimi teknikleriyle özellikle karşılaştıran birçok çalışma bulunmaktadır. Örneğin, bir kitle değerleme için Brezilya'da RF ve Tekrarlayan Sinir Ağları (RNN) karşılaştırıldı ve RF'nin sayısal özelliklerle daha iyi çalıştığı gözlemlendi. İsveç'te bir karşılaştırma çalışması için 57,974 emlak verisi ve bunların 44 özgü özellikleri kullanıldı. Üç farklı makine öğrenimi tekniği; RO, SVM ve YSA, kitle değerleme çalışması için karşılaştırıldı. Araştırmacılar, RO algoritmalarının daha iyi sonuçlar verdiğini gözlemledi. 29,680 emlak verisi ve 55 değişkeni kullanarak Pune şehrinde RF yöntemlerini kullanarak bir çalışma gerçekleştirildi ve RF'nin kitlesel değerleme teknikleri için güvenilir sonuçlar verdiğini kanıtlamaya çalıştı. (Sevgen & Tanrıvermiş, 2020: 302). Literatürden görüldüğü gibi, RO algoritması kitlesel değerleme çalışmaları için en iyi performansı vermektedir. Bu nedenle, bu çalışmada kitlesel değerleme RF Regresyon algoritması ile gerçekleştirilmeyi amaçlamaktadır. Ayrıca, RF ve diğer makine öğrenimi algoritmalarıyla gerçekleştirilen birçok kitlesel değerleme çalışmasında, veriler genellikle resmi kurumlardan elde edilmekte ve genellikle daha fazla emlak verisi ve değişken içermektedir. Yenimahalle ilçesinden alınan daire verileri ve dairelerin 13 bağımlı değişkeni kullanılarak popüler bir makine öğrenimi algoritması olan RF regresyonu ile kitlesel değerleme çalışması gerçekleştirildi ve sonuçlar ArcGIS sürüm 10.1 ile görselleştirildi. RF regresyonu tarafından gerçekleştirilen kitlesel değerleme sonuçları, sonuçları etkileyen değişkenlerin önemine göre değerlendirildi (Sevgen & Tanrıvermiş, 2020: 306).

Veranyurt vd. 2020 yılında yapmış olduğu çalışmasında, sağlık yönetiminde hastalıkların doğru teşhisinde makine öğrenmesi tekniklerinin etkinliğini karşılaştırmayı amaçlamaktadır. Vanderbilt Üniversitesi tarafından sağlanan ve 390 hastaya ait 15 değişkenden oluşan bir veri seti üzerinde Random Forest (RF), K-Nearest Neighbour (KNN) ve AdaBoost algoritmalarının sınıflandırma performansları incelenmiştir. (Veranyurt vd., 2020: 279) Çalışmanın bulgularına göre, RF ve KNN algoritmalarıyla yapılan sınıflandırma başarısı %92,30, AdaBoost algoritması ile yapılan sınıflandırma başarısı ise %90,59 olarak belirlenmiştir. Bu sonuçlar, makine öğrenmesi algoritmalarının hastalıkların doğru sınıflandırılmasında yüksek başarı elde etme potansiyeline sahip olduğunu göstermektedir. Diyabet hastalığının erken teşhis edilebilmesi amacıyla seçilen algoritmaların, %25'in altında yanlış teşhis oranlarıyla etkin bir şekilde çalıştığı belirlenmiştir. Sağlık yönetimi ve hizmetlerinde yapay zekâ ve makine öğrenmesi tekniklerinin kullanımının giderek arttığı ve hastalıkların doğru teşhis edilmesinde bu tekniklerin önemli bir rol oynayabileceği vurgulanmıştır (Veranyurt vd., 2020: 282-283).

Bu çalışmasında, Özen vd. 2021 yılında, COVID-19 pandemisi döneminde Amerika Birleşik Devletleri'nde doğrulanmış vaka sayılarını tahminlemek için çeşitli makine öğrenmesi modellerini kullanmıştır. Prophet, Polinom Regresyon, ARIMA, Doğrusal Regresyon ve Random Forest modelleri Python ve R programlama dilleriyle implemente edilmiş ve performansları ortalama mutlak yüzde hatası (MAPE), ortalama karekök sapması (RMSE) ve ortalama mutlak hata (MAE) kullanılarak değerlendirilmiştir (Özen vd., 2021: 135). Çalışmanın sonuçlarına göre, Polinom Regresyon algoritması %1.86 MAPE oranıyla en iyi tahminleri verirken, ARIMA, Prophet, Random Forest ve Doğrusal Regresyon sırasıyla takip etmiştir. Pandemi sürecindeki belirsizlikleri göz önüne alarak, bu çalışma Amerika Birleşik Devletleri'nin kaynak planlaması, aşı dağıtımı, hastane personel ve malzeme planlaması gibi senaryoları olası en düşük hatayla tahminlemek için Polinom Regresyon modelini öner Segoe UI mektedir. Gelecekte, farklı ülkelerde benzer çalışmalar yapılarak ülke bazlı stratejiler geliştirilebilir ve pandemiyle ilgili belirsizliklerin azaltılmasına katkı sağlanabilir. Makine öğrenmesi modellerinin bu tür kriz durumlarında kullanımı, hızlı ve etkili kararlar alınmasına yardımcı olabilir (Özen vd., 2021: 138).

Bu çalışma, kardiyovasküler hastalıkları tespit etmek amacıyla makine öğrenmesi tekniklerinin uygulandığı bir örnek veri seti üzerinde Coşar & Deniz tarafından 2021 yılında gerçekleştirilmiştir. Kalp hastalıkları için erken teşhisin önemine vurgu yaparak, makine öğrenmesi ve yapay zekâ gibi yeni tekniklerin kullanımının tıp alanında yapılan çalışmalara katkı sağladığı belirtilmiştir. Veri seti analizi sonucunda kalp rahatsızlığını gösteren belirtiler tespit edilmiş, ardından üç farklı makine öğrenmesi yöntemi kullanılarak bir model oluşturulmuştur (Coşar & Deniz, 2021: 1112). Sonuçlar, Random Forest algoritmasının %88'lik doğruluk oranıyla en başarılı olduğunu göstermiştir. Lojistik Regresyon %85 ve kNN algoritması ise %70 doğruluk oranlarına sahiptir. Göğüs ağrısı tipinin hastalığı doğrulamada önemli bir etken olduğu, erkeklerin kadınlara göre daha sık kalp hastalığına yakalandığı tespit edilmiştir (Coşar & Deniz, 2021: 1114-1115). Çalışmanın genel sonuçları, kalp rahatsızlıklarının temel verilerle başarılı bir şekilde tespit edilebileceğini göstermektedir. Ayrıca, makine öğrenme algoritmalarının kullanımının, hastalıkların doğru teşhisi ve tedavisi konusunda sağlık çalışanlarına yardımcı olabileceği düşünülmektedir.

Arslan, 2021 yılında yapmış olduğu çalışmasında Doc2Vec modeli ve makine öğrenmesi yaklaşımlarını kullanmıştır. ISCX2016URL veri setinden toplamış olan URL'ler 5 farklı sınıfa ayrılmış ve toplam 165372 adet farklı tür URL adresi üzerinden analizler yapılmıştır. Çalışmada kullanılan veri seti ilk aşamada ikili sınıflandırma tekniği ile kötücül ve iyicil olarak sınıflandırılmış, bu sınıflandırma işlemi için lojistik regresyon, K-en yakın komşu, rastgele orman, destek vektör makinesi, karar ağacı, lineer discriminant analizi, naïve bayes, ektra ağaç, gradyan artırma ve adaboost algoritmaları kullanılmıştır. Analizler için gerekli özellikler DBOW modeli ile çıkarılmıştır ve sonucunda en yüksek doğruluk değeri %87.8 oranı ile lojistik regresyon algoritması tarafından elde edilmiştir. Kesinlik, duyarlılık ve f-skor değerleri ise sırasıyla %88.8, %89.8 ve %87.7 olarak saptanmıştır (Arslan, 2021: 798). Bir diğer Doc2Vec modeli olan DM modeli ile çıkarılan özellikler ile yapılan analizde ise rastgele orman algoritması %98.8, k-en yakın komşu algoritması %98.9, extra tree algoritması %98.9, ve SVC algoritması da %98.8 doğruluk değerlerini elde etmiştir. İkinci aşamada ise aynı veri seti üzerinde çoklu sınıflandırma tekniği uygulanmıştır ve kötücül URL adresleri spam, malware, phishing, defacement olarak farklı sınıflara ayrılmıştır. Ayrıştırılan

bu sınıflar üzerinde DBOW ve DM modellerinin birleşiminden oluşan CONCAT modeli ile ikinci bir test yapılmış, test sonucunda en yüksek performansın %88,3 oranı ile SVC algoritması ile elde edildiği görülmüştür. Recep Sinan Arslan, çoklu sınıflandırma sonuçlarının ikili sınıflandırma sonuçlarından bu şekilde düşük olması nedenini 4 kümeye ait örneklerin ciddi dengesizliği olarak açıklıyor. Bu çalışmada iki aşamadan oluşan analizlerin sonucunda en verimli sonucun %98,9 oranı ile lojistik regresyon ve k-en yakın komşu sınıflandırıcıları ile edilmiştir (Arslan, 2021: 800).

Dinler & Şahin 2021 yılında yapmış oldukları çalışmalarında kimlik hırsızı web sitelerinin tespiti için rastgele orman, destek vektör makinesi, k-en yakın komşu, çok katmanlı algılayıcı ve derin öğrenme yöntemlerini kullanarak WEKA ortamında analizlerini gerçekleştirmişlerdir. Kullanılan algoritmalar ile sınanacak olan veri seti 1353 örnek içermektedir ve bu özelliklerin her biri 9 altsınıfa sahiptir. Bu çalışmada algoritmaların performansları doğruluk, kesinlik, duyarlılık ve f-ölçütü oranları ile analiz edilmiştir. Analizlerin sonucuna göre rastgele orman ve derin öğrenme yöntemleri sırasıyla %89.9 ve %90 oranları ile en iyi performansı sağlayan yöntemler olmuşken k-en yakın komşu algoritmasının, çapraz doğrulama kullanıldığında daha iyi bir performans sergilediği görülmüştür (Dinler & Şahin, 2021: 39).

Baktır & Atay 2022 yılında gerçekleştirdiği çalışmasında; Kaggle, UCI, Machine Learning Repository, Google Dataset Search Engine gibi platformlardan çeşitli veri setleri elde ederek birleştirmiş ve en sonunda 24.152 adet veri üzerinden analizlerini gerçekleştirmiştir. Çalışmada kullanılan algoritmalar; naive bayes, lojistik regresyon, karar ağacı ve k-en yakın komşu algoritmalarıdır. Dağınık veri setini standart ve kullanılabilir hale getirmek için karakter işlemleri, etkisiz kelimeler, köklendirme ve kelime frekansını bulma gibi veri ön işlemleri adımları uygulanmıştır. Python programlama dili ile Jupyter Notebook ortamında geliştirilen algoritmalar ile deneyler gerçekleştirilmiş ve ölçüt metrikleri doğruluk, kesinlik, duyarlılık, F1-skor olarak belirlenmiştir. Kullanılan veri setleri üzerinde bu ölçütlerin ve algoritmaların başarım oranları ayrı ayrı incelenmiş ve sunulmuştur (Baktır & Atay, 2022: 360). Çalışmanın sonucunda elde edilen verilere göre, tüm veri setleri üzerinde ve ölçütler bazında genel olarak en iyi sonuçları yansıtan algoritmanın lojistik regresyon algoritması olduğuna saptanmıştır. En iyi başarımı kazandıran veri seti ise CS440/ECE448 olarak açıklanmıştır ve bu veri seti üzerinde k-en yakın komşu algoritmasının kesinlik, lojistik regresyon algoritmasının da duyarlılık ölçütlerinde %100 oranında performans sergiledikleri gözlenmiştir (Baktır & Atay, 2022: 362).

Shaik vd. 2022 tarihli çalışmalarında rastgele orman, karar ağacı, hafif GBM, lojistik regresyon ve destek vektör makinesi algoritmalarını kullanarak kötü niyetli web sayfalarının tespit edilmesini amaçlamışlardır. Analizler için 1500'ü güvenli, 1500'ü oltalama olan toplamda 3000 adet URL içeren ve Phish Tank üzerinden elde edilen veri seti kullanılmıştır. Bu veri setinin %80'i eğitim, %20'si ise test için kullanılmıştır. Bu çalışma sonunda, hafif GBM algoritmasının %89.5 eğitim doğruluğu ve %86 test doğruluğu oranları ile en iyi performansa sahip algoritma olduğu gözlemlenmiştir (Shaik vd. 2022: 4).

Bu makale, Wei & Sekiya tarafından 2022 yılında oltalama web sitelerini tespit etmek için kullanılan çeşitli makine öğrenimi yöntemlerinin performansını değerlendirmektedir. Makale, çeşitli makine öğrenimi ve derin öğrenme yöntemlerini karşılaştırmak amacıyla gerçekleştirilen deneylerin sonuçlarını sunmaktadır. Deney sonuçlarına göre, ensemble

makine öğrenimi algoritmalarının, tespit doğruluğu ve hesaplama tüketimi açısından diğer yöntemlere göre daha etkili olduğu belirlenmiştir. Özellikle, ensemble yöntemlerinin, veri setindeki özellik sayısı keskin bir şekilde azaldığında bile etkileyici bir yetenek sergilediği ifade edilmektedir. Hem makine öğrenimi tabanlı hem de derin öğrenme tabanlı phishing tespit tekniklerini ele almaktadır. Makine öğrenimi tabanlı yöntemler arasında, destek vektör makinesi, sınıflandırma ve regresyon ağaçları, rastgele orman, AdaBoost gibi algoritmalar bulunmaktadır. Derin öğrenme tabanlı yöntemler arasında ise, çok katmanlı algılayıcılar, uzun kısa dönemli bellek ve evrişimli sinir ağları gibi popüler algoritmalar yer almaktadır (Wei & Sekiya vd., 2022: 2-3). Sonuçlar, ensemble makine öğrenimi yöntemlerinin seçilen özelliklerle gerçek zamanlı uygulamalarda ve mevcut özellikleriyle phishing web sitelerini ayırt etme konusunda avantajlı ve son derece etkili olduğunu doğrulamaktadır. Makale, gelecekteki çalışmalar için, sonuçlarını çeşitli veri kümeleri üzerinde ve daha fazla özellik ve örneklemle doğrulamayı ve sıfır-gün saldırıları tespit ederken verimliliği artırmayı planlamaktadır (Wei & Sekiya vd., 2022: 3-4).

Bu makalede Rathee & Mann 2022 yılında odak noktası olarak, özellikle phishing e-postalarını algılamak için hem makine öğrenimi hem de derin öğrenme yaklaşımlarına dikkat çekmiştir. Ayrıca, son birkaç on yılda önerilen çeşitli DL ve ML modellerinin karşılaştırmalı analizi ve değerlendirmesini sunmaktadır. Yazarlar, tüm e-postaların karakteristik özelliklerini içeren bir çeşitli özellikleri kullanarak geniş bir veri kümesi topladılar. Kullanılan özellikler arasında metin içeriği, gönderici bilgileri, gönderilen URL'ler ve zararlı dosya türleri gibi bir dizi şey bulunmaktadır. Elde edilen veri setini bir tür dil modeline beslemek için birçok konvolüsyonel sinir ağı (CNN) mimarisi kullandılar. Özellikle, YSA, CNN, LSTM ve BERT mimarilerini değerlendirdiler. CNN mimarisi, tüm mimariler arasında en iyi performansı gösterdi ve %95,34 doğruluk oranına sahipti (Rathee & Mann, 2022: 4). Makine Öğrenimi Temelli Phishing E-posta Algılama: Fette vd. tarafından önerilen PILFERS yöntemi, e-postaları değerlendirmek için 10 özelliğe dayanıyor ve URL'leri ve JavaScript'in varlığını kontrol ederek e-postaları phished olarak işaretlemeye odaklanıyor. Abu-Nimeh vd. farklı sınıflandırıcıların performansını inceledi ve rastgele ormanların en iyi sonucu verdiğini buldu, ancak yüksek yanlış pozitif oranı olduğunu gözlemledi. Rawal vd., RF ve SVM kullanarak phishing e-postalarını tespit etmek için özellik çıkarma tabanlı bir sistem önerdi ve en yüksek doğruluğu elde etti (Rathee & Mann, 2022: 3-4). Derin Öğrenme (DL) Temelli Phishing E-posta Algılama: Jameel vd., beslemeli ileri yayılımlı sinir ağı kullanarak phishing e-postalarını sınıflandırmak için bir yöntem önerdi ve yüksek bir doğruluk elde etti. Bir diğer paradigma, karakter seviyesinde bir evrişimli sinir ağı (CNN) kullanarak e-postaların özelliklerini çıkarmayı amaçlar ve yüksek doğruluk sağlar, ancak kısa URL'ler veya gizli kelimeler içeren URL'ler gibi durumlarda yanlış sonuçlar verebilir (Rathee & Mann, 2022: 4-5). Makale, bu ve daha fazla yaklaşımı ayrıntılı olarak ele alıyor ve her birinin avantajlarını ve dezavantajlarını tartışıyor. Ayrıca, farklı yöntemlerin karşılaştırmalı analizini ve farklı araştırmacıların önerdiği yöntemlerin doğruluklarını içeren bir tablo sunuyor. Phishing e-postalarını algılamak için hem geleneksel ML yöntemlerini hem de DL yöntemlerini inceleyerek geniş bir bakış sunmuştur.

İlgün ve Samet, 2023 yılında yaptıkları çalışmalarında, veri setine ölçekleme, kategorik veri kodlama, hibrit öznitelik seçimi gibi ön işleme süreçleri uygulayarak bu veri üzerinden makine öğrenmesi destekli bir yapay zekâ saldırı tespit sistemi ortaya koydular.

Çalışmalarında K-En Yakın Komşu, Çok Katmanlı Algılayıcı, Rastgele Orman, XGBOOST, LightGBM makine öğrenmesi algoritmalarını kullandılar ve bu algoritmalar üzerinden farklı tiplerde saldırı tespit modelleri ortaya koydular (İlgun & Samet, 2023: 683). Çalışmalarının sonucunda, eğitilen modellere hiper-parametre optimizasyonu yaparak ilgili veri seti üzerinde başarı değerlendirmesi yapmışlardır. İlgün ve Samet, elde ettikleri sonuçlarında eğitim yapılan veri setinde 0,373 saniyede %96,1 saldırı tespit oranını ve test verisinde 0,005 saniyede %100 saldırı tespit oranını elde ettiler (İlgun & Samet, 2023: 691).

Kulkarni, 2023 yılında oltalama sitelerini tespit etmek için yapay sinir ağlarını (Neural Networks) kullanmıştır. CNN olarak kısaltılan yapay sinir ağı modelleri, görüntüleri işleme ve bu görüntülerden özellikler çıkarmak yaygın olarak kullanılmaktadır. Ayrıca CNN modelleri, bu özelliklerle çeşitli sınıflandırmalar yapabilmektedir. İlgili çalışmada oltalama, legal ve meşru olmak üzere üç farklı kategöriye ayrılmış 1,353 gerçek web sitesi veri seti olarak kullanılmıştır. Yapılan testler sonucunda CNN modelinin sınıflandırma başarısı %86,5 olarak tespit edilmiştir (Kulkarni, 2023: 15-18).

3- Veri Seti ve Veri Ön İşleme

3.1 Veri Setinin Tanıtılması

Oltalama saldırılarının tespitinde kullanılacak veri seti, Ulusal Siber Olaylara Müdahale Merkezi'nin API modeli aracılığıyla toplanmıştır (https://www.usom.gov.tr/api/). USOM'un belirtilen API modeli kullanılarak ve USOM tarafından tanımlanmış API yapılandırma kuralları dikkate alınarak '/api/adress/index' uzantısı üzerinden url, type, datasource, connectiontype, desc, criticalitylevel, date olmak üzere altı farklı öznitelik belirlenmiş ve oltalama verileri bu şekilde elde edilmiştir. İlgili özniteliklerin karşılıkları Tablo 1.0'da belirtilmiştir. Veri seti, USOM'un 'Zararlı Bağlantılar Listesi' aracılığıyla (https://www.usom.gov.tr/url-list.txt) kanıtlamış veri setleri üzerinden (hedeflenen altı öznitelik gözetilerek) yaklaşık 120 bin farklı oltalama web adresini içermektedir. Veri setine ait altı farklı parametrenin işlevi Tablo 1.1'de açıklanmıştır.

Meşru siteleri içeren veri seti, Alto Üniversite tarafından yayınlamış "PhishStorm - phishing / legitimate URL dataset" isimli veri setinden alınmıştır. 96,018 adet oltalama ve meşru site içeren bu veri seti, "label" etiketi üzerinden sıfır (0) meşru adresler ve bir (1) oltalama adresleri olmak üzere ayrıştırılmıştır. Bu sayede, veri setinde bulunan toplam 48,009 meşru web sitesinden faydalanılmıştır.

Çalışmamızda kullanılan her iki veri seti, 'desc' özniteliği üzerinden oltalamalar için 'Bankacılık-Oltalama', 'Oltalama', 'Zararlı Yazılım Yayan URL', meşru siteler için de 'Mesru' etiketleri kullanılarak birleştirilmiştir. Ayrıca, veri setleri 'label' özniteliği üzerinden oltalama adresler 1 ve meşru adresler 0 olmak üzere tanımlanmıştır.

Öznitelik Adı	Altsınıf Bilgisi	Açıklama
URL	-	Her biri birbirinden bağımsız oltalama ve zararlı yazılım barındıran/yayan URL adreslerini içerir.

TYPE	Domain, IP ve URL olmak üzere 3 altsınıfa sahiptir.	Verilerin türünü gösteren kategorik bir özniteliktir.
DATASOURCE	USOM, SOME ve İhbar olmak üzere 3 altsınıfa sahiptir.	Verilerin hangi kaynaklar tarafından tespit edildiği bilgilerini içerir.
CONNECTION TYPE	Oltalama, APT C&C(Advanced Persistent Threat Command and Control), Zararlı dosya indirme, Mobil C&C, BOTNET, C&C, Diğer, Mining zararlısı ve Exploit kit olmak üzere 8 altsınıfa sahiptir.	Saldırganların ağ veya cihazlara hangi bağlantı türü ile eriştiğini gösteren özniteliktir.
DESC	Bankacılık-oltalama ve oltalama olmak üzere 2 altsınıfa sahiptir.	Verilerle ilgili yapılan açıklama bilgilerini içerir.
CRITICALITY LEVEL	Minimum 1 ve maksimum 10 değerleri arasında tanımlanır.	Saldırıların kritiklik seviyelerini içeren özniteliktir.
DATE	-	Saldırı tespit işlemlerinin gerçekleştiği tarih bilgilerini içerir.

(Tablo 1.1 USOM özniteliklerinin tanıtımı)

Tablo 1.1'de belirtildiği gibi veri setimiz toplamda 7 öznitelikten ve yaklaşık 20 altsınıftan oluşmaktadır. Bu veri seti içerisinde yer alan URL özniteliği, saldırganların URL'ler üzerindeki manipülatif hareketlerini ve bu sayede gerçekleştirdikleri sızma çeşitlerini barındırır ve en yüksek ayırt ediciliğe sahip özniteliktir. Type özniteliği, verilerin internet hizmetleri üzerindeki türlerini gösteren özniteliktir. Datasource özniteliği, verilerin hangi kaynaklardan elde edildiği bilgilerini içerir. Bu özniteliğin altsınıflarından olan SOME (Siber Olaylara Müdahale Ekipleri), USOM bünyesi altında oluşturulan bir ekiptir ve bu iki ekip tarafından tespit edilen saldırıların yanı sıra ihbar hattı üzerinden elde edilen saldırılar da bu veri seti içerisinde USOM tarafından birleştirilmiştir. Connection type özniteliği içerisinde yer alan altsınıflar ise, saldırganların eriştikleri ağ ve cihazlar içerisinde tutunma stratejilerini ve kullanılan farklı türlerdeki merkezi komuta ve kontrol mekanizmalarını gösterir. Criticality level özniteliğinde, gerçekleştirilen saldırılar kritiklik seviyelerine göre sınıflandırılmıştır ve bu saldırı türlerinden, Oltalamalar genellikle 4 kritiklik değerini alırken zararlı yazılım barındıran URL/IP'ler ise 7 değerini almaktadır. Saldırı tespit işlemlerinin gerçekleştiği tarih bilgilerini içeren Date özniteliği ise veri ön işleme adımında, benzersizlik oranı %0 olması ve analiz esnasında hiçbir ayırt ediciliğe sahip olmayan bu öznitelik üzerinde çalışılması verimli sonuçlar çıkarmayacağı nedenleriyle veri setinden ayrıştırılmıştır.

3.2 Veri Ön İşleme

Veri ön işleme, veri analitiği veya makine öğrenimi projelerinde kritik bir adımdır ve veri setlerini hazırlamanın temelini oluşturur (Gurusamy vd., 2014: 1). Bu süreç, veriyi daha anlamlı ve etkili bir şekilde kullanmak için veri setlerinin temizlenmesini, dönüştürülmesini

ve hazırlanmasını içerir (Han vd., 2006: 112). Veri ön işleme adımları, veri setinin düzenli, anlaşılır ve modelleme için hazır hale getirilmesini sağlar. Bu adımlar sırasında, veri setindeki hatalar düzeltilir, eksik veya boş değerler ele alınır ve veriler makine öğrenimi algoritmalarının daha iyi çalışabileceği bir formata dönüştürülür (Kotsiantis vd., 2006: 116).

Aykırı değerlerin tespit edilip kaldırılması, modelin yanlış sonuçlar üretmesini önler ve daha güvenilir bir analiz yapılmasını sağlar. (Hawkins, 1980: 25). Kategorik değişkenlerin dönüştürülmesi ve ölçeklendirilmesi, farklı özellikler arasında karşılaştırma yapılmasını ve modelin daha tutarlı sonuçlar üretmesini sağlar (Jain vd., 1997: 155-156).

Veri ön işleme neden önemlidir?

Model Performansını Artırma: Temizlenmiş, dönüştürülmüş ve özellikleri seçilmiş bir veri seti, makine öğrenimi modellerinin daha iyi performans göstermesine yardımcı olabilir (Garreta vd., 2013: 46-47).

Overfitting'i Önleme: Gereksiz veya fazla özellikler, modelin veriye aşırı uyum sağlamasına neden olabilir. Ön işleme ile bu tür durumlar önlenebilir (James vd., 2013: 65).

Doğruluk ve İşlenebilirlik: Temizlenmiş ve düzenlenmiş bir veri seti, analiz ve yorumlama süreçlerini daha doğru ve işlenebilir hale getirebilir (Provost vd., 2013: 54).

Veri Eksikliğini Ele Alma: Veri ön işleme, eksik verileri ele alarak model performansını artırır ve yanlış sonuçlara yol açma riskini azaltır (Little vd., 2019: 13-14).

1. Eksik Değerlerin Kontrol Edilmesi:

Eksik değerlerin kontrolü, veri setindeki eksiklikleri belirlemek için önemlidir ve veri bütünlüğünü sağlar. Bu adım literatürde genellikle 'eksik veri analizi' olarak adlandırılır ve eksik verilerin etkilerini inceleyen bir araştırma alanıdır. Eksik değerlerin kontrol edilmesinde yaygın olarak kullanılan yöntemler arasında, her bir değişken için eksik değerlerin sayısının ve yüzdesinin hesaplanması, eksik değerlerin nedenlerinin incelenmesi (örneğin, rasgele mi yoksa sistematik mi olduğu) ve eksik değerlerin yapısının ve dağılımının görselleştirilmesi bulunur. Bu adım, eksik değerlerin analiz ve modelleme süreçlerinde potansiyel etkilerini belirlemek ve eksik değerlerin uygun şekilde ele alınması için stratejiler geliştirmek açısından kritik öneme sahiptir (Little vd., 2019: 29).

2. Eksik Değerlerin Doldurulması:

Eksik verilerin doldurulması, veri setinin tamamlanmasını sağlar ve eksik bilgilerin analizdeki etkisini azaltır. Bu adım, eksik veri problemini ele alan ve doldurma tekniklerini tartışan birçok makalede ele alınmıştır. (Azur vd., 2011: 41). Eksik değerlerin doldurulması işlemi, veri setinin doğasına ve analiz amacına bağlı olarak değişebilir. Bu nedenle, eksik değerlerin doldurulması sürecinde dikkatli bir şekilde yöntem seçilmeli ve eksik değerlerin doldurulmasının analiz sonuçlarına etkisi göz önünde bulundurulmalıdır.

3. Kategorik Değişkenlerin Dönüştürülmesi:

Kategorik değişkenlerin dönüştürülmesi, makine öğrenimi algoritmalarının kategorik verilerle çalışmasını sağlar. Bu adım, sınıflandırma ve kümeleme problemlerinde yaygın olarak kullanılan bir veri ön işleme tekniğidir. Bu adımda kullanılan birkaç yöntem vardır; One-Hot Encoding, Label Encoding , Ordinal Encoding. Kategorik değişkenlerin

dönüştürülmesi, veri setinin analiz ve modelleme sürecinde daha iyi performans elde etmek için önemlidir. Ancak, dönüşüm yöntemi seçilirken veri setinin özellikleri ve analiz amacı dikkate alınmalıdır (James vd., 2015: 42).

4. Korelasyon Matrisinin Hesaplanması:

Korelasyon matrisi, değişkenler arasındaki ilişkiyi ölçmek için kullanılır ve birçok istatistiksel analiz için önemlidir. Bu adım, değişkenler arasındaki ilişkiyi belirlemek ve gereksiz değişkenleri eleme veya birleştirme kararları almak için kullanılır. Bu matris, her bir değişken çifti arasındaki ilişkiyi belirler ve genellikle Pearson korelasyon katsayısı kullanılarak hesaplanır. Korelasyon matrisi, veri setinin yapısını anlamak, değişkenler arasındaki güçlü ilişkileri belirlemek ve gereksiz veya fazla ilişkili değişkenleri eleme veya birleştirme kararları almak için önemlidir. Bu adım, veri setinin daha iyi anlaşılmasına ve daha etkili analizlere olanak sağlar (Rencher vd., 1987: 77).

5. Min-max Ölçekleme:

Min-max ölçekleme, veri değerlerini belirli bir aralığa ölçeklemek için kullanılır ve verileri karşılaştırılabilir hale getirir. Bu adım, ölçek farklılıklarının model performansını etkileyebileceği durumlarda sıklıkla kullanılır. Böylece, tüm değişkenler aynı ölçek aralığında olur ve farklı ölçek aralıklarına sahip değişkenler arasındaki karşılaştırmalar daha tutarlı hale gelir. Min-max ölçekleme, özellikle makine öğrenimi algoritmalarıyla çalışırken, değişkenler arasındaki farklı ölçeklerin model performansını olumsuz etkilemesini önlemek için yaygın olarak kullanılır. Bu adım, veri setinin homojenleştirilmesine ve daha güvenilir sonuçların elde edilmesine yardımcı olur (Han vd., 2000: 78).

6. Standartlaştırma:

Standartlaştırma, veri ön işleme adımlarından biridir ve veri setindeki değerleri bir standart dağılımı takip edecek şekilde dönüştürmek için kullanılır. Bu adımda, her bir değişkenin değerleri ortalama değerinden çıkarılır ve standart sapmasıyla bölünerek standart normal dağılımı takip edecek şekilde ölçeklenir. Standartlaştırma, veri setinin ortalamasını ve standart sapmasını kullanarak verileri bir standart normal dağılıma dönüştürür. Bu adım, veri setinin dağılımını düzeltir ve bazı makine öğrenimi algoritmalarının daha iyi performans göstermesini sağlar. (Hastie vd., 2009: 67).

7. Aykırı Değerlerin Tespiti ve Kaldırılması:

Aykırı değerlerin tespiti ve kaldırılması, veri ön işleme adımlarından biridir ve veri setindeki istatistiksel olarak anormal veya aşırı değerlerin belirlenmesi ve ele alınması sürecidir. Bu adım, genellikle veri setindeki aykırı değerlerin analize veya modele zarar verebileceği durumlarda kullanılır. Aykırı değerler, analiz sonuçlarını yanıltabilir ve modelin performansını olumsuz yönde etkileyebilir. Bu adım, aykırı değerlerin tanımlanması, etkilerinin incelenmesi ve gerektiğinde veri setinden çıkarılmasıyla ilgili birçok araştırmada ele alınmıştır. Aykırı değerlerin tespitinde kullanılan yaygın yöntemler arasında, ortalama ve standart sapma gibi istatistiksel ölçütler kullanarak belirlenen eşik değerlerin ötesindeki değerlerin tanımlanması, kutu grafiği gibi görselleştirme yöntemlerinin kullanılması ve makine öğrenimi tekniklerinin uygulanması yer alır. Ardından, belirlenen aykırı değerler veri setinden kaldırılabilir veya bunlarla ilgili farklı işlemler uygulanabilir, örneğin ortalama değerlerle değiştirme veya özellik mühendisliği teknikleri kullanma gibi (Aggarwal, 2016:

67). Bu işlem, veri setinin doğruluğunu artırır, modelin daha güvenilir sonuçlar üretmesine olanak tanır ve modelin genelleme yeteneğini iyileştirir.

8. Boyut Azaltma (PCA):

Boyut Azaltma Analizi (PCA), bir veri setindeki değişkenliği daha az sayıda bileşenle temsil etmek için kullanılır. PCA, veri boyutunu azaltarak veri setinin karmaşıklığını azaltır ve temel değişkenlerin yapısal özelliklerini ortaya çıkarır. Bu sayede, veri setinin anlaşılması ve yorumlanması kolaylaşırken, modelleme süreçlerinde daha az işlemci gücü ve bellek kullanımı gerektirir. PCA, veri setindeki değişkenliği daha az sayıda bileşenle temsil etmek için kullanılır. PCA'nın matematiksel detayları ve uygulamaları birçok istatistik ve makine öğrenimi kitabında kapsamlı bir şekilde ele alınmıştır (Jolliffe, 2011: 92).

4- Algoritmaların Tanıtılması ve Performans Ölçütlerinin Belirlenmesi

4.1 Destek Vektör Makinesi

Destek vektör makinesi, *Cortes ve Vladimir N. Vapnik* tarafından ortaya atılan ve 1995'ten itibaren birçok sınıflandırma ve regresyon probleminde başarılı sonuçlar veren gözetimli öğrenme yöntemlerinden biridir. Destek Vektör Makinesi yönteminin en önemli amacı, gelen yeni veri noktalarını etkin ve verimli bir şekilde en doğru kategoriye koymaktır. Bu yöntem, birbirinden ayrıştırılması istenen veri noktalarını bir düzlem üzerine yerleştirerek, noktalar arasındaki uzaklığın maksimum seviyede olmasını amaçlayan bir vektör çizer ve iki farklı sınıfa ayırır. Bu vektöre hiper düzlem (hyperplane) adı verilir. Daha karmaşık bir tabirle; "Destek vektör makinesi çok boyutlu bir uzayda herhangi bir girdi noktasından en uzakta yer alan en uygun lineer ayırıcı hiperdüzlemi bularak sınıflandırma gerçekleştiren bir yöntemdir." (Kalaycı, 2018: 873). Başka bir kaynakta ise "Destek vektör makinesinin temel fikri, veriyi giriş uzayından daha yüksek boyutlu bir özellik uzayına haritalamak ve iki sınıf arasında en yakın noktalar arasındaki marjı maksimize ederek optimal ayırıcı hiperdüzlemi bulmaktır." şeklinde tanımlanmıştır (Cortes & Vapnik, 1995 akt. Miyamoto vd., 2009: 3). Hiper düzlemle ayrıştırılan sınıflar arasında kalan boşluğa "Margin" adı verilir ve margin ne kadar geniş ise, sınıflandırma işlemi o kadar başarılı kabul edilir. Bir margin boşluğu ne kadar iyi belirlenirse, elde edilen fayda da aynı oranda maksimize edilir. Margin noktasında karar sınırına yani hiper düzleme en uzak nokta maksimum faydayı sağlayan alandır ve bu noktadan itibaren sınıflandırma işlemi yapılır. Karar sınırına en yakın noktaya ise "Destek" adı verilir. Destek noktaları belirlendikten sonra yeni gelen veriler bu noktalarla karşılaştırılır ve veriler farklılıklarına göre sınıflandırılır. Bu noktanın konumu değişirse, karar noktasının konumu da değişeceği için sınıflandırma işlemlerinde destek noktalarını doğru bir şekilde belirlemek çok önemlidir (Mahajan & Siddavatam, 2018: 46; Dinler & Şahin, 2021: 37).

4.2 Karar Ağaçları

Karar ağacı algoritması; 1985 yılında J.R. Quinlan tarafından ortaya atılmış ve temel yapısı özyinelemeye dayanan, denetimli öğrenme yöntemlerinden biri olan regresyon ve sınıflandırma algoritmasıdır. Bu algoritma, böl-yönet yaklaşımını benimseyerek öncelikle veri kümelerinde yer alan özellikler arasından en iyi ayırıcıyı seçer ve bunu ağacın kökü olarak kabul eder. Sonradan gelen veri girişlerini bu kök düğüm ile karşılaştırarak en doğru sınıfa yerleştirmeye çalışır. Karşılaştırma işlemini yaptığı düğümlere karar düğümü denir ve bu düğümlerin her biri farklı bir ayırıcı özelliği içerir buna test fonksiyonu da denebilir. Gelen verilere bu test fonksiyonları uygulanır ve sonucuna göre veri için en uygun dal seçilir,

bu işlem bir bitiş düğümüne ulaşana kadar özyinelemeli şekilde devam eder. Bu doğrultuda, karar ağaçlarının bitiş düğümüne gelindiğinde belirli bir sınıfın etiketini yani özelliğini içeren yaprağa ulaşılır bu sayede de veri girdileri en az iterasyon ile verimli bir şekilde sınıflandırılmış olur. Karar ağaçlarında bu dallara ayırma işlemleri için en çok kullanılan yöntemler gini indeksi ve bilgi kazancı (information gain) metotlarıdır. Bu metotlar, herkes tarafından anlaşılır olması ve hesaplama maliyetlerinin düşük olması gibi nedenlerden dolayı tercih edilse de aşırı uyum problemi yaratabileceği için bir budama veya topluluk öğrenmesi yöntemlerine ihtiyaç duyabilir (Baktır & Atay, 2022: 356; Kalaycı, 2018: 873; Mahajan & Siddavatam, 2018: 46; James vd., 2013: 307; Quinlan: 1985: 88).

4.3 Rastgele Orman

Rastgele orman algoritması ilk olarak 1996 yılında *Leo Breiman ve Adele Cutler* tarafından önerilmiş ve 2001 yılında yayınladıkları makalede, "Rastgele ormanlar, her biri ormandaki tüm ağaçların bağımsız ve aynı dağılıma sahip bir rastgele vektörün değerlerine bağlı olan ağaç tahmincilerin bir kombinasyonudur." (Breiman, 2001: 1) şeklinde tanıtılmıştır. Bu algoritma; genel olarak birden fazla modelin bir araya getirilerek daha güçlü bir model ortaya koyulmasını amaçlayan, sınıflandırma ve regresyon amacıyla kullanılan topluluk öğrenme algoritmasıdır. Bu amacını; veri kümelerindeki belirli özelliklere göre sınıflandırılan karar ağaçlarını rastgele vektörler kullanımı ile oluşturarak ve oluşturulan karar ağaçlarının en yaygın sınıfını çoğunluk oylamasına göre seçerek yerine getirir. Bu rastgele seçim işlemine "Bagging" yöntemi adı verilir. Daha sonrasında ise "Rastgele alt uzay" yöntemi oluşturulmuştur ve bu yöntem, "KA' larında aşırı uyma durumunu önleyebilmek adına, örneklemlere ait özelliklerin arasında rastgele bir alt uzay seçilir. KA'nın oluşturulması esnasında ayırma noktalarında kullanılacak özellik ise rastgele seçilmiş olan bu alt uzaydan seçilir." (Ho, 1998 akt. Bayraktar, 2019: 42) şeklinde tanımlanır. Rastgele orman algoritması, bu iki yöntemin bir araya getirilmesi ile oluşturulur. Kullanılacak olan özellikleri bagging ve rastgele alt uzay yöntemleri aracılığıyla seçer sonrasında aynı karar ağacı algoritmasında olduğu gibi gini indeksi ve bilgi kazancı (information gain) metotları ile en iyi ayırıcıyı seçerek bireysel karar ağaçlarını oluşturur. Bu işlemler istenilen miktardaki ağaç sayısına ulaşana kadar devam eder ve her bir ağaç hedef değeri tahmin eder, bu tahminlere "Oy" adı verilir. Oluşturulan bu karar ağaçlarına test veri setinde bulunan örnekler uygulanır ve tüm karar ağaçları için sınıflandırma tahmininde bulunulur. En sonunda bu tahminlerden çoğunluk oylamasına göre en çok oy alan sınıf seçilir (Mahajan & Siddavatam, 2018: 46, Shaik vd., 2022: 3, Dinler & Şahin, 2021: 37).

4.4 Performans Ölçütlerinin Belirlenmesi

Çalışmamızda belirlenen sınıflandırma algoritmaları çapraz doğrulama değeri (k=5) beş üzerinden hesaplanacaktır. Çalışma sürecinde belirlenen algoritmalar test sonunda elde edilen çıktılarla başta Doğruluk (Accuracy), Kesinlik (Precision) ve Karmaşıklık Matrisi (Confusion Matrix) olmak üzere, Doğru Pozitif Oranı (TP), Yanlış Pozitif Oranı (FP), F-Ölçütü (F-Measure), Duyarlılık (Recall), ROC Alanı (ROC Arena), Matthew'un Korelasyon Katsayısı (MCC), peformans ölçümleri üzerinden karşılaştırılacaktır.

Çapraz Doğrulama (Cross-Validation): Çapraz Doğrulama ya da Basit Çapraz Doğrulama, sınıflandırılacak veri bu değerlendirme modeline verildiğinde öncelikle eğitim ve test verisi olarak iki parçaya bölünecektir. Daha sonra veri, eğitim verisi kullanılarak eğitilecek ve k (k-fold) kadar eşit veri kümesine bölünecek ve her veri kümesi test verisinde kullanılmak üzere

sınıflandırıcı modele verilecektir (Selvan & Muthuraman, 2016: 21). Çalışmamızda bu “k” değeri beş olarak belirlenmiştir.

Doğruluk (Accuracy): Sonuç olarak doğruluk ölçütü bu formül ile ifade edilebilir: Doğruluk= (DP + DN) \ (DP + DN + YP + YN) (İlgün & Samet, 2023: 684).

Karmaşıklık Matrisi (Confusion Matrix): Yapılan deneyler sonucunda test edilen algoritmanın başarısını ölçmeye yarayan bir ölçüt yapısıdır. Karmaşıklık Matrisi, deneyler sonucunda sahip olduğu bilgilere dayanarak, gerçek ve tahmin edilen verilerin her birinin sayısal değerlerinin bilinmesini ve bu sayede test edilen algoritmanın başarısının ölçülebilmesini sağlar (Koşan vd., 2017: 279).

Doğru Pozitif (TP): Deney sonucunun başarıya ulaşmış olması durumudur, örneğin bizim çalışmamız için gerçek bir oltalama (phishing) web adresinin tahminleme yapan modelin amacına uygun olarak başarıyla oltalama (phishing) kategorisinde tahminlenmiş olması bir Doğru Pozitif (TP) örneğidir (Şahin & Chouseinoglou, 2019: 37).

Yanlış Pozitif (FP): Deney sonucunda yanlış tahminleme yapmış bir modelin gösterim türüdür. Örneğin ikili sınıflandırma üzerinde "1 olarak tahmin edilen sınıfın gerçekte 0 olması durumu" olarak özetlenebilir (Şahin & Chouseinoglou, 2019: 37).

F-ölçütü (F1-skoru): F ölçütü olarak adlandırdığımız değerlendirme modeli, ayrıca F1-skoru, F1-ölçütü isimleriyle de bilinir. Bu model temel olarak iki farklı değerlendirme ölçütü olan kesinlik (precision) ve duyarlılık (recall) değerlendirme modellerinin harmonik ortalamalarının tanımlanmış bir halidir. F ölçütü, çoğunlukla hassasiyet ve duyarlılık ölçütlerine karşı farklı sonuçlar elde edilen endeksler aracılığıyla hesaplanabilir (Dalianis, 2018: 47).

F ölçütü, aşağıdaki formülle hesaplanabilir:

F ölçütü: F1 = F = 2 ∗ P ∗ R / P + R

Kesinlik (Precision): Kesinlik, mevcut veride bulunmuş Doğru Pozitif (DP) olarak tanımlanan verilerin, doğru bulunan (Doğru Pozitif ve Yanlış Pozitif) tüm veriler üzerindeki oranını ölçer. Bu oran kesinlik (precision) olarak tanımlanır ve aşağıdaki formülle özetlenebilir: (Dalianis, 2018: 47)

Kesinlik: Doğru Pozitif / (Doğru Pozitif + Yanlış Pozitif)

Duyarlılık (Recall): Duyarlılık, bir veride bulunan ve Doğru Pozitif (DP) olarak tanımlanmış sonuçların tüm Pozitif (+P) oranında kapsanma derecesini gösterir. Diğer bir değişle, bu kapsama derecesi gerçek anlamda sahip olunan doğru sonuçların (DP) Yanlış Negatif (YN) sonuçları da dahil edilerek tüm pozitifler için oranlanmış halidir (Powers, 2011: 38). Aşağıdaki formülle özetlenebilir:

Duyarlılık: Doğru Pozitif / (Doğru Pozitif + Yanlış Negatif)

Matthew'un Korelasyon Katsayısı (MCC): Hedeflenen modelin kalitesini belirlemek amacıyla hesaplanan istatistiksel bir orandır. Bu oran, yüksek yüzdelik oranlara sahip olduğunda verinin kalitesine hizmet edecektir. MCC, hedeflenen tahmin modelinin karışıklık matrisi üzerinde dört farklı sınıf aracılığıyla (doğru pozitif, yanlış pozitif, gerçek negatif ve yanlış negatif) hesaplanabilir (Balogun vd., 2021: 161).

ROC Alanı (ROC Arena): Performans testleri sonucunda elde edilen Doğru Pozitif (DP) ve Yanlış Pozitif (YP) sonuçlarının grafiksel eğrileri üzerinden hesaplanmaktadır. Bu değerlendirme ölçütünde, 0 ve 1 değeri olmak üzere iki farklı ölçüt kategorisi bulunur ve değerlendirme sonuçları 1 kategorisine yaklaştıkça testlerin başarı oranı artar (Koşan vd., 2017: 279).

5. Deneysel Sonuçlar

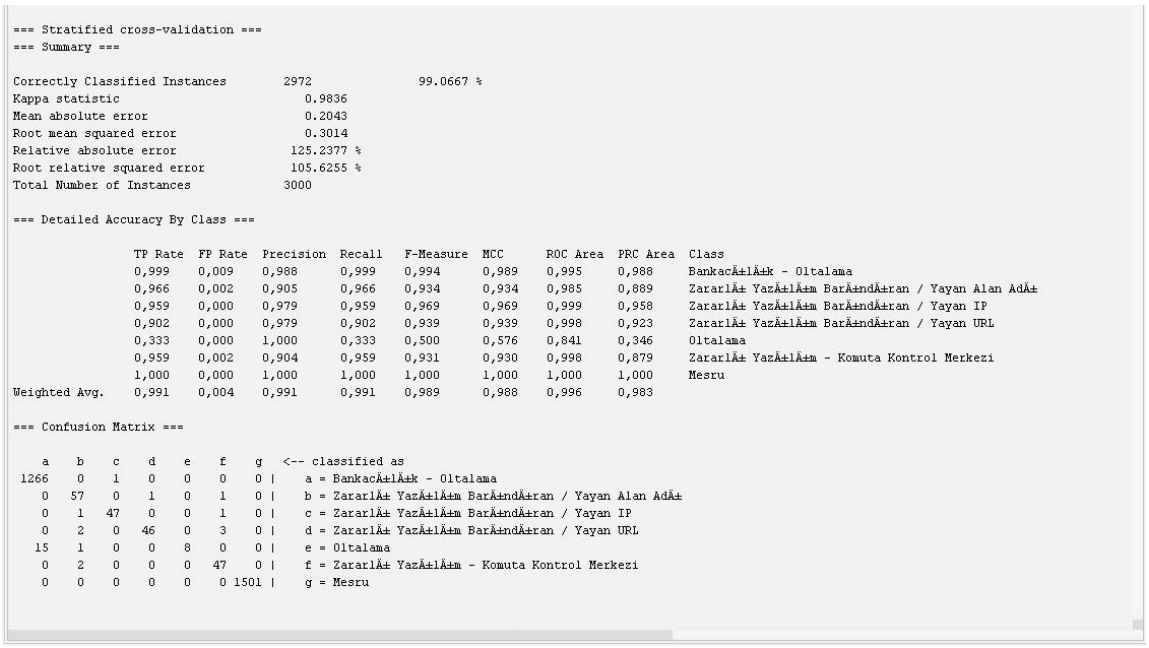
5.1 Weka ile Modelin Test Edilmesi

Bu çalışmada önceki bölümlerde açıklanmış olan çeşitli sınıflandırma algoritmalarını kullanarak bir web sayfasının kimlik hırsızı olup olmadığının tespit edilmesi amaçlanmıştır. Aynı zamanda bu çalışma makine öğrenimi ile kimlik hırsızı web sitelerinin tespiti için en doğru ve tutarlı sonucu sunan sınıflandırma algoritmasının belirlenmesine de yardımcı olmaktadır. Ele alınan algoritmalar, 1500 adeti meşru, 1500 adeti ise oltalama olmak üzere toplamda 3000 URL içeren veri seti üzerinde WEKA ortamında test edilmiştir. Veri setinin %80'i eğitim, %20'si ise test amaçlı kullanılmıştır, aynı zamanda veri setinin bu kadar dar olmasının sebebi ise Weka uygulamasında yüksek miktarda verilerle çalışmanın oldukça zor olması ve güvenli sonuçlar elde edilememesidir. Wekanın bu kadar kısıtlı veri ile çalışmasından dolayı sonuçların yeterli olmayacağından ötürü aynı algoritmalarla daha büyük veri setleriyle çalışabilen Scikit-Learn kütüphanesi ile aynı testler yapılmıştır ve sonuçlar bir sonraki maddede açıklanmıştır.

Weka ortamında gerçekleştirdiğimiz deneyler toplamda 30 kez tekrarlanmıştır ve yapılan deneylerin ortalaması alınarak sunulmuştur. Böylelikle uygulama içerisindeki yüksek standart sapmaların önüne geçilerek sonuçlar güçlendirilmiş ve daha tutarlı, güvenilir sonuçlar elde edilmiştir (Baktır & Atay, 2022: 353). Deney aşamasında çapraz doğrulama değeri 5 olarak kabul edilmiştir. Çapraz doğrulama modeli, veri setini belirtilen miktarda kümelere böler ve her bir küme elemanının en az bir kere sınama elemanı olmasını sağlayarak sonuçların daha doğru ve güvenilir olmasını sağlar, yanlılığı azaltır (Burman, 1989 akt. Kalaycı, 2018: 874; Miyamoto vd., 2008: 5). Aynı zamanda çapraz doğrulama eğitim verisinde yer alan elemanların performanslarını tutar, sonrasında ise test aşamasına gelen elemanların performanslarını daha rahat değerlendirme imkânı sağlar ve böylelikle eğitim aşamasında kullanılan verilerin gerçek hayata nasıl entegre olduklarını daha kolay gözlemleyebiliriz.

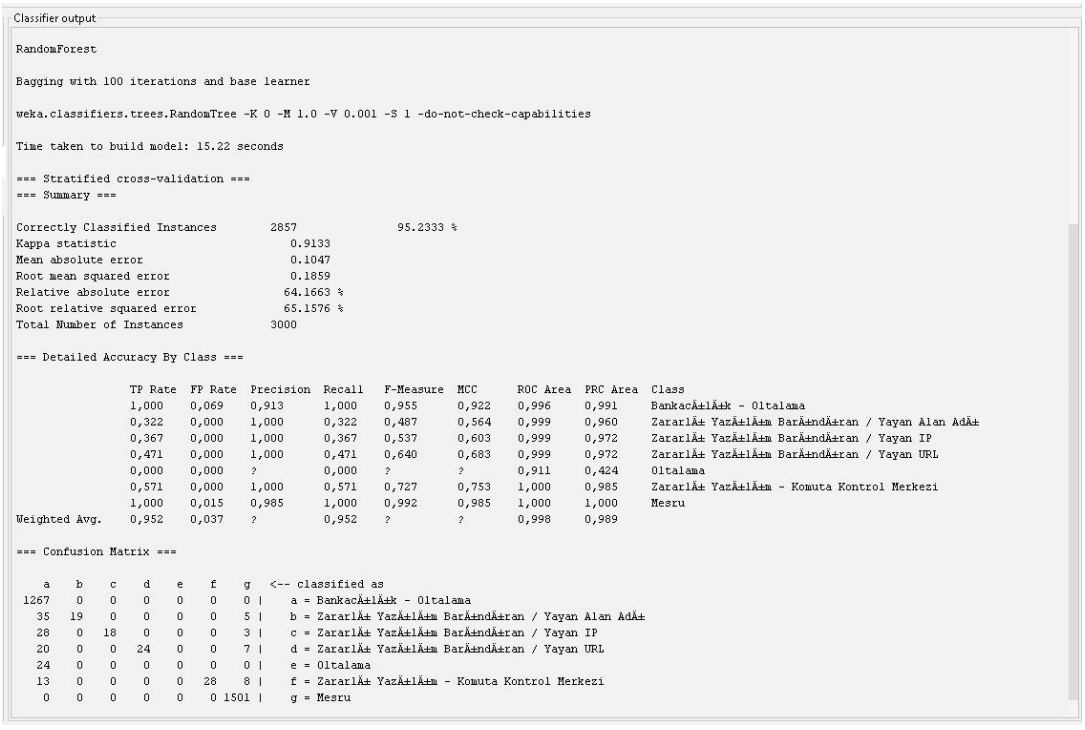
Elde edilen bulgulara göre; bu çalışmada kullanılan algoritmalardan en yüksek performansı sergileyen algoritma %99.06 oranı ile Destek Vektör Makinesi algoritması olmuştur. En düşük faydayı sağlayan algoritma ise %95.23 oranı ile Rastgele Orman algoritması olmuştur. Karar ağacı algoritması ise %98.1 oranı ile iyi bir performans sergilemiştir. Yapılan deneylerin sonuçlarında performans ölçütlerine baktığımızda, Destek Vektör Makinesi (Görsel 1.1) %0.991 oranında kesinlik ve yine %0.991 oranında duyarlılık değerlerine sahip olarak yine en başarılı algoritma olduğu gözlemlenmiştir. Bu iki ölçütün yakın değerlere sahip olması modelin aşarı öğrenme probleminin önüne geçtiğini gösterir. Ancak Rastgele Orman (Görsel 1.2) algoritmasının sonuçlarına baktığımızda %0.913 oranında kesinlik ve %1.00 oranında duyarlılık değerlerine sahip olduğunu gözlemliyoruz. Bu değerlerden anlaşıldığı üzere Rastgele Orman algoritması aşırı öğrenme sorununa maruz kalarak yanıltıcı sonuçlar doğurmuştur. Karar Ağaçlarında

(Görsel 1.3) ise bu değerler, %0.981 oranında kesinlik ve %0.999 oranında duyarlılık şeklindedir. Genel sonuçlara bakıldığında Destek Vektör Makinesi algoritmasının daha verimli olduğu görülse de Karar Ağacı algoritmasının da yüksek duyarlılık oranı ile pozitif durumların tahmin edilmesinde daha başarılı olduğu gözlemlenmiştir.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2972               99.0667 %
Kappa statistic                          0.9836
Mean absolute error                      0.2043
Root mean squared error                  0.3014
Relative absolute error                125.2377 %
Root relative squared error            105.6255 %
Total Number of Instances             3000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,999    0,009    0,988      0,999    0,994      0,989    0,995     0,988     BankacÃ±lÃ±k - Oltalama
               0,966    0,002    0,905      0,966    0,934      0,934    0,985     0,889     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan Alan AdÃ±
               0,959    0,000    0,979      0,959    0,969      0,969    0,999     0,958     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan IP
               0,902    0,000    0,979      0,902    0,939      0,939    0,998     0,923     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan URL
               0,333    0,000    1,000      0,333    0,500      0,576    0,841     0,346     Oltalama
               0,959    0,002    0,904      0,959    0,931      0,930    0,998     0,879     ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     Mesru
Weighted Avg.  0,991    0,004    0,991      0,991    0,989      0,988    0,996     0,983

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
 1266    0    1    0    0    0    0 |    a = BankacÃ±lÃ±k - Oltalama
    0   57    0    1    0    1    0 |    b = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan Alan AdÃ±
    0    1   47    0    0    1    0 |    c = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan IP
    0    2    0   46    0    3    0 |    d = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan URL
   15    1    0    0    8    0    0 |    e = Oltalama
    0    2    0    0    0   47    0 |    f = ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
    0    0    0    0    0    0 1501 |    g = Mesru
```

(Görsel 1.1 Destek Vektör Makinesi algoritması deneysel sonuçları)



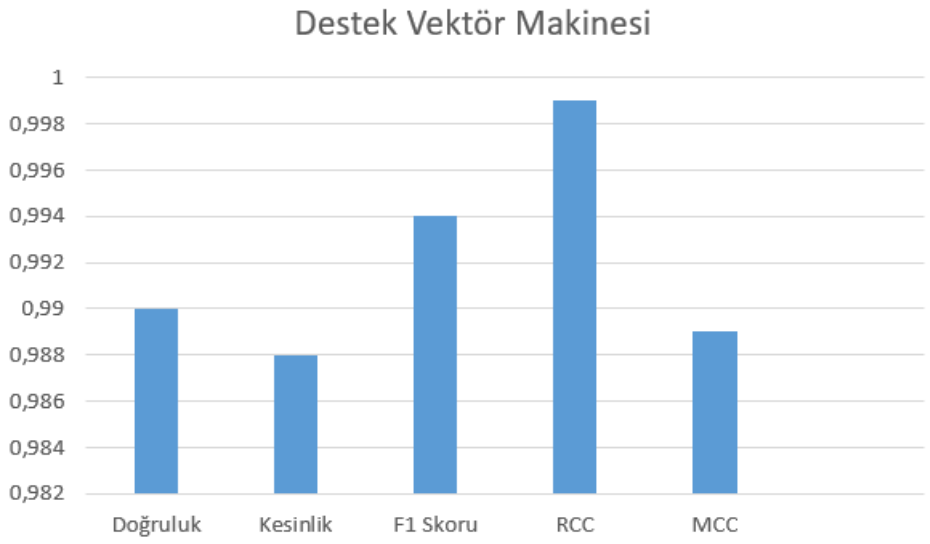
```
Classifier output

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 15.22 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2857               95.2333 %
Kappa statistic                          0.9133
Mean absolute error                      0.1047
Root mean squared error                  0.1859
Relative absolute error                 64.1663 %
Root relative squared error             65.1576 %
Total Number of Instances             3000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1,000    0,069    0,913      1,000    0,955      0,922    0,996     0,991     BankacÃ±lÃ±k - Oltalama
               0,322    0,000    1,000      0,322    0,487      0,564    0,999     0,960     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan Alan AdÃ±
               0,367    0,000    1,000      0,367    0,537      0,603    0,999     0,972     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan IP
               0,471    0,000    1,000      0,471    0,640      0,683    0,999     0,972     ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan URL
               0,000    0,000    ?          0,000    ?          ?        0,911     0,424     Oltalama
               0,571    0,000    1,000      0,571    0,727      0,753    1,000     0,985     ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
               1,000    0,015    0,985      1,000    0,992      0,985    1,000     1,000     Mesru
Weighted Avg.  0,952    0,037    ?          0,952    ?          ?        0,998     0,989

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
 1267    0    0    0    0    0    0 |    a = BankacÃ±lÃ±k - Oltalama
   35   19    0    0    0    0    5 |    b = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan Alan AdÃ±
   28    0   18    0    0    0    3 |    c = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan IP
   20    0    0   24    0    0    7 |    d = ZararlÃ± YazÃ±lÃ±m BarÃ±ndÃ±ran / Yayan URL
   24    0    0    0    0    0    0 |    e = Oltalama
   13    0    0    0    0   28    8 |    f = ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
    0    0    0    0    0    0 1501 |    g = Mesru
```

(Görsel 1.2 Rastgele Orman algoritması deneysel sonuçları)

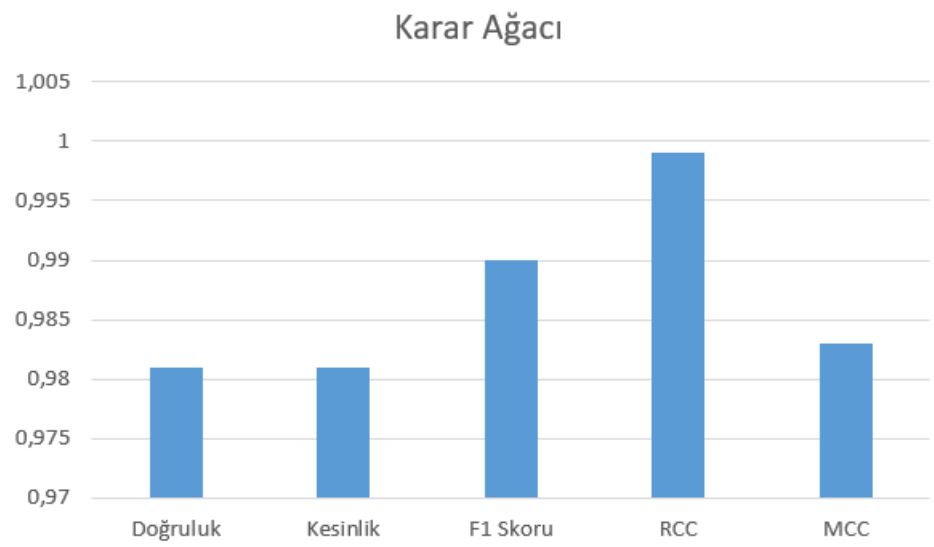
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2943               98.1    %
Kappa statistic                          0.9665
Mean absolute error                      0.009
Root mean squared error                  0.0687
Relative absolute error                  5.5249 %
Root relative squared error             24.0716 %
Total Number of Instances             3000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,999    0,014    0,981      0,999    0,990      0,983    0,992     0,979     BankacÃ±lÃ±k - Oltalama
               0,983    0,002    0,921      0,983    0,951      0,950    0,999     0,926     ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan Alan AdÃ±
               0,939    0,001    0,958      0,939    0,948      0,948    0,989     0,894     ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan IP
               0,490    0,000    1,000      0,490    0,658      0,697    0,985     0,768     ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan URL
               0,000    0,000    ?          0,000    ?          ?        0,754     0,016     Oltalama
               0,959    0,009    0,644      0,959    0,770      0,782    0,975     0,613     ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     Mesru
Weighted Avg.  0,981    0,006    ?          0,981    ?          ?        0,994     0,970

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
 1266    0    1    0    0    0    0 |    a = BankacÃ±lÃ±k - Oltalama
    0   58    0    0    0    1    0 |    b = ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan Alan AdÃ±
    0    2   46    0    0    1    0 |    c = ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan IP
    0    2    0   25    0   24    0 |    d = ZararlÃ± YazÃ±lÃ±m BarÃ¤ndÃ¤ran / Yayan URL
   23    0    1    0    0    0    0 |    e = Oltalama
    1    1    0    0    0   47    0 |    f = ZararlÃ± YazÃ±lÃ±m - Komuta Kontrol Merkezi
    0    0    0    0    0    0 1501 |    g = Mesru
```

(Görsel 1.3 Karar Ağacı algoritması deneysel sonuçları)

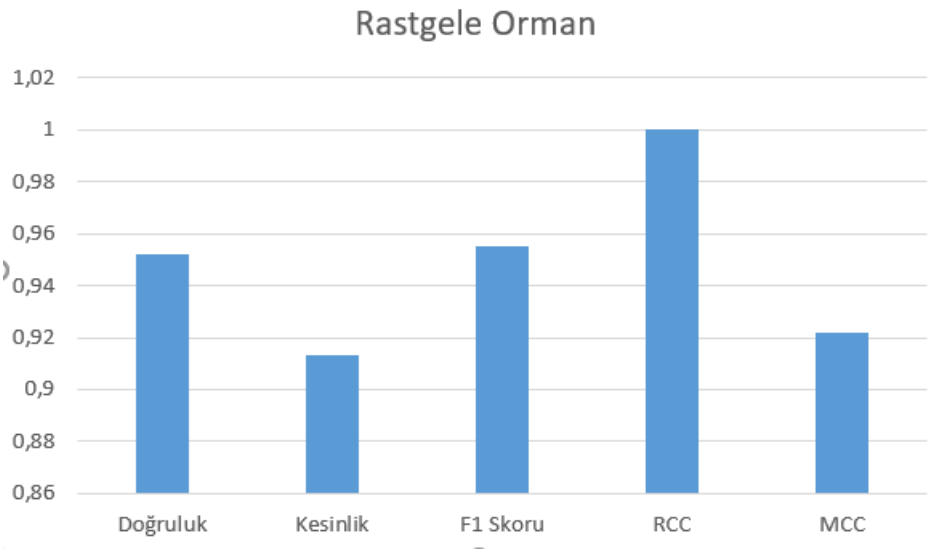
Weka ortamında test edilen algoritmaların doğruluk oranlarının yanında diğer ölçüm kriterleri üzerindeki oranları ilgili görsellerle açıklanmıştır. Elde edilen sonuçlar grafikler ile desteklenerek daha açıklayıcı bir hale getirilmiştir. Gözlemlenen değerlere göre; Destek Vektör Makinesi algoritması (Tablo 1.2), %99.06 doğruluk oranı ve 0,009 hata oranı ile en iyi sonucu elde eden algoritma olmuşken RCC ölçüm kriterine göre diğer algoritmaların gerisinde kalmıştır. Karar Ağacı algoritması (Tablo 1.3) ise %98.1 doğruluk oranına sahipken 0,999 RCC ve 0,981 kesinlik oranlarını elde ederek daha iyi sonuç elde etmiştir. Rastgele Orman (Tablo 1.4) algoritması, genel sonuçlarına baktığımızda diğer algoritmalardan daha düşük sonuçlara sahip olsa da RCC kriterine göre 1.000 oranı ile en iyi sonucu elde ettiği gözlenmiştir.



Tablo 1.2



Tablo 1.3



Tablo 1.4

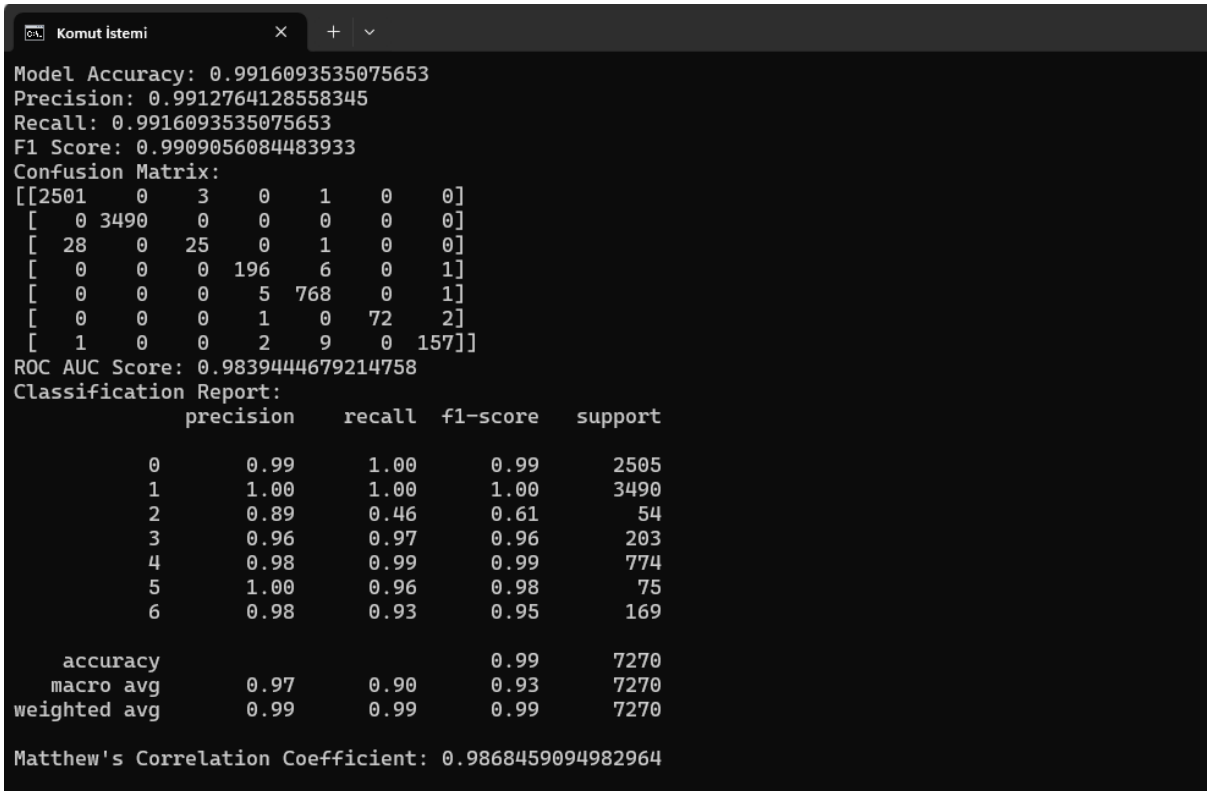
5.2 Scikit-Learn ile Modelin Test Edilmesi

Weka programı yalnızca sınırlı büyüklükteki veri setleriyle çalışabildiği için modelin ek olarak bir python makine öğrenimi kütüphanesi olan scikit-learn üzerinde de test edilme gereksinimi doğmuştur. Bu bölümün amacı, Weka programına oranla daha büyük bir veri seti kullanarak kesinlik ve doğruluk gibi performans ölçütlerinin daha doğru sonuçlar vermesini sağlamaktır. Çalışmanın bu bölümünde, 48,009 adet oltalama adresi (USOM) ve 48,009 meşru web adresi (Alto Üniversitesi meşru adresleri) olmak üzere toplamda 96,018 adet web sitesi içeren veri seti ön işleme adımlarındaki aykırı değerlerin, hatalı

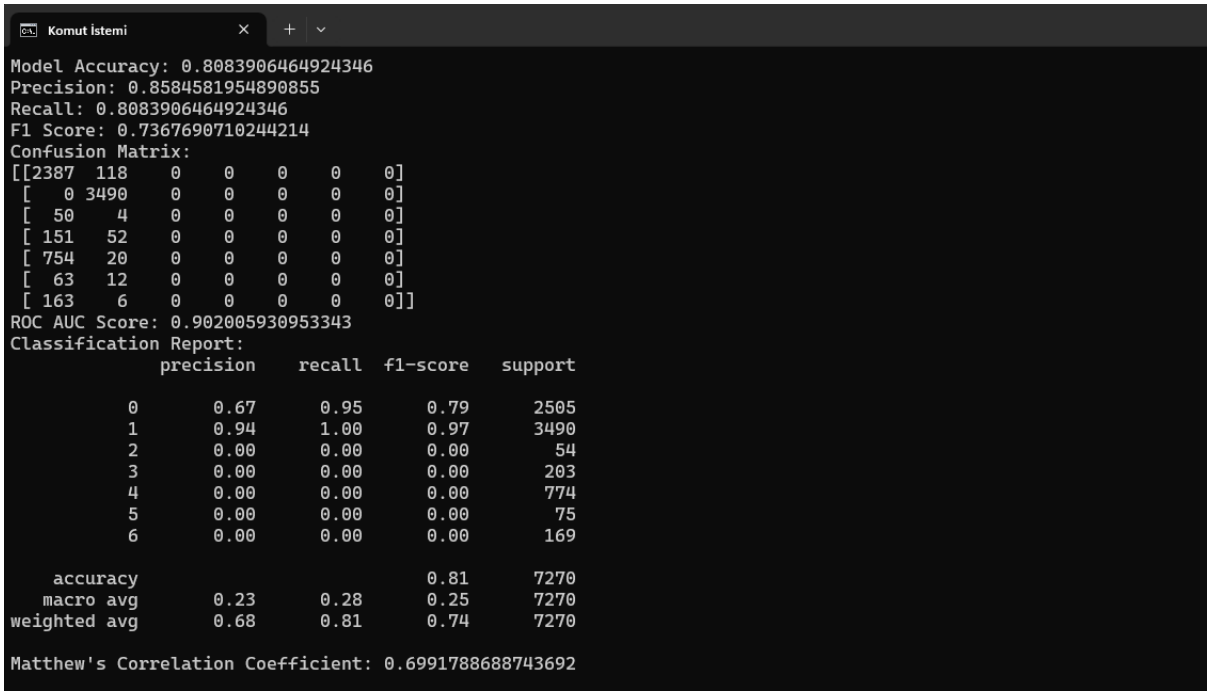
girdilerin, uzantı bazlı alanların çıkarılması sonucu ön işlenmiş 36.347 adet site olarak düzenlenmiştir.

Veri setinin %80'lik kısmı 5.1 Weka ile Modelin Test Edilmesi bölümünde de belirtildiği gibi eğitim verisi olarak, %20'lik kısmı test verisi olarak kullanılmıştır. Ek olarak, 4.4 Performans Ölçütlerinin Belirlenmesi bölümünde de bahsedildiği gibi, yapılan deneysel testler boyunca bu bölümde belirtilen performans ölçütleri kullanılacaktır.

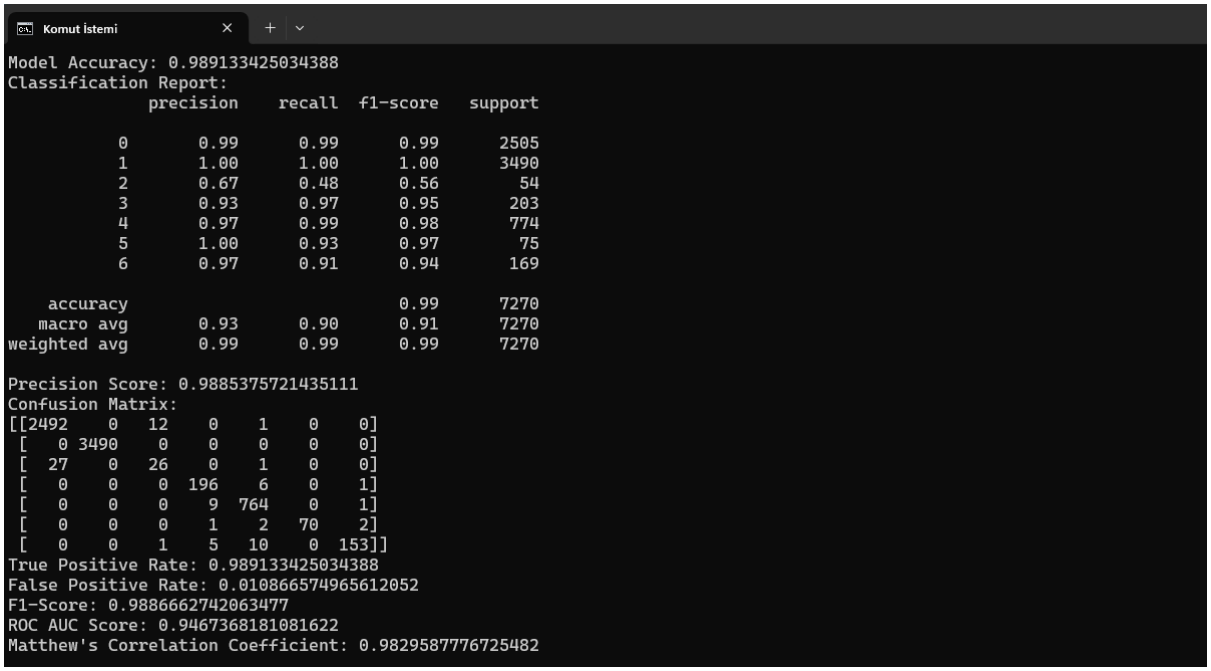
Aşağıdaki üç görsel sırasıyla Görsel 2.1 Rastgele Orman algoritması, Görsel 2.2 Destek Vektör Makinesi algoritması, Görsel 2.3 Karar Ağacı algoritmasını kullanan üç farklı python sckit-learn betiğini ifade etmektedir. İlgili python betiklerinin kodları ödev dosyasında paylaşılmıştır.



Görsel 2.1 Rastgele Orman algoritması ile modelin deneysel sonuçları

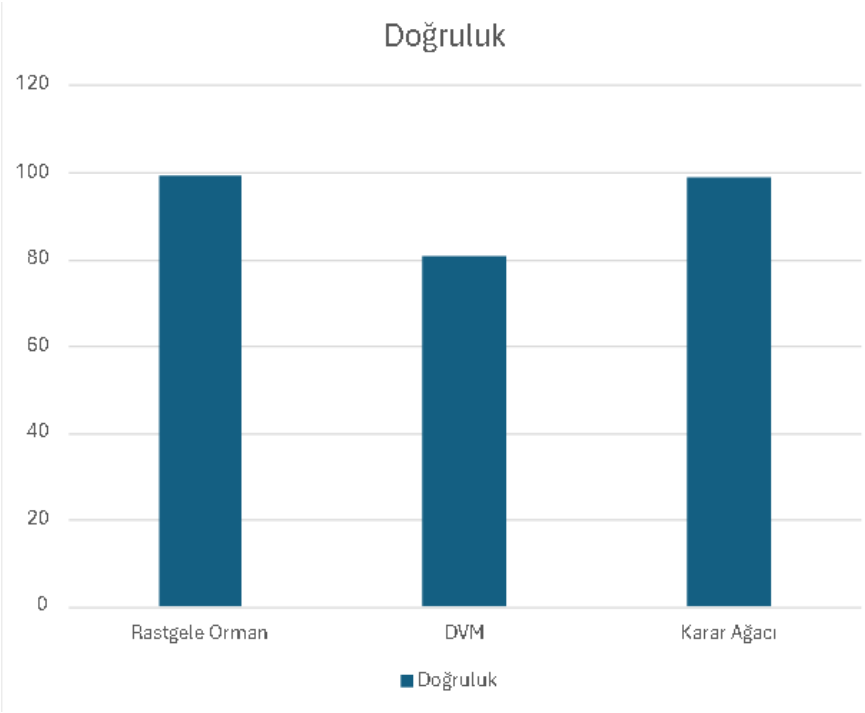


Görsel 2.2 DVM algoritması ile modelin deneysel sonuçları

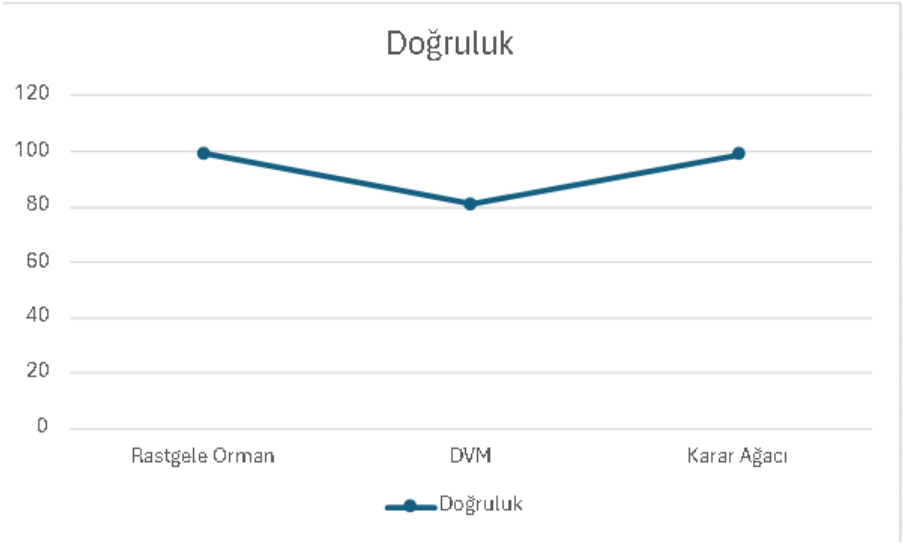


Görsel 2.3 Karar Ağacı algoritması ile modelin deneysel sonuçları

Yukarıda belirtildiği üzere, her üç algoritma da ön işlenmiş veri seti üzerinden test edilmiştir. Yapılan testler sonucunda, en yüksek doğruluk oranı %99.1 ile Rastgele Orman algoritması olmuştur. Ardından %98.8 ile Karar Ağacı ve %80.8 ile Destek Vektör Makinesi algoritması yer almaktadır. Yapılan testlerin doğruluk üzerinden grafiksel açıklaması Tablo 1.2 ve Tablo 1.3'de gösterilmiştir.

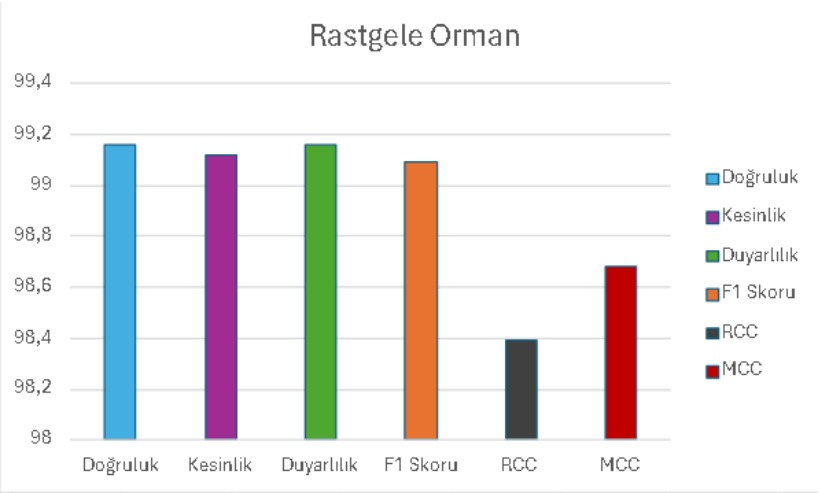


Tablo 1.5

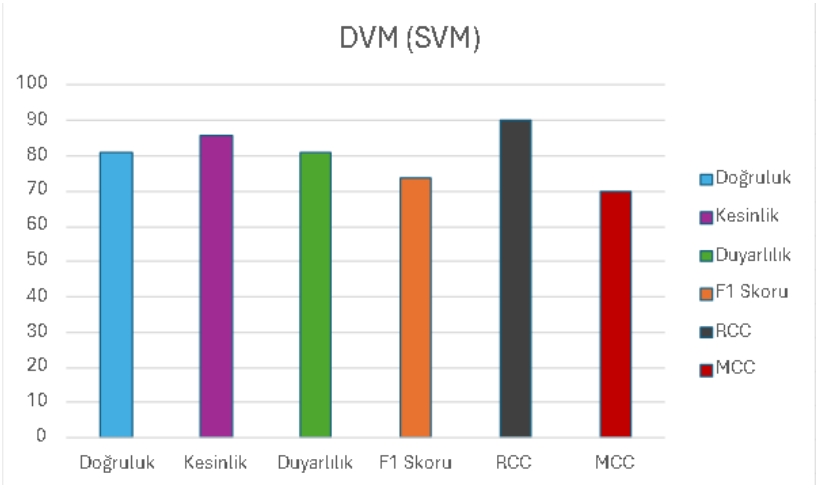


Tablo 1.6

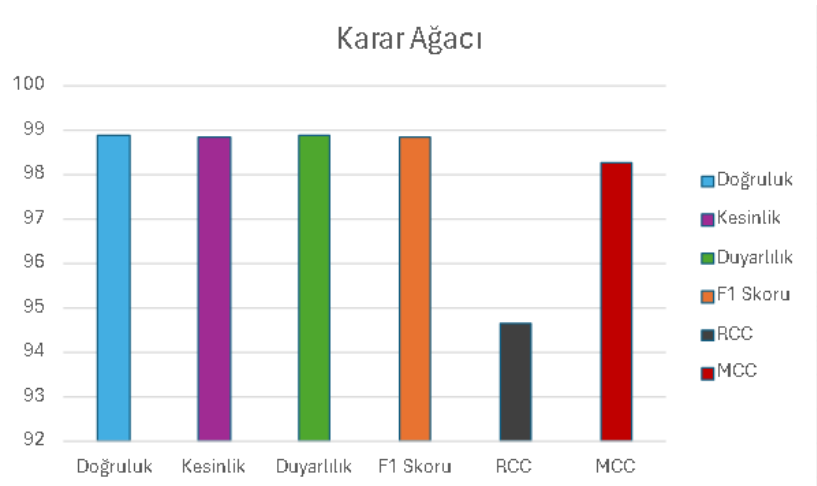
Belirtilen tabloları daha fazla detaylandırmak, performans ölçütleriyle beraber geçerliliğini ve güvenilirliğini arttırmak amacıyla aşağıda belirtilen Tablo 1.6 ve Tablo 1.7 oluşturulmuştur. Bu tablolar temel olarak, her üç algoritmanın deneysel sonuçlarından elde edilen performans çıktıları arasındaki uyumun kolay bir şekilde anlaşılmasını sağlamaktadır.



Tablo 1.7



Tablo 1.8



Tablo 1.9

Görseller ve tablolardan anlaşılacağı üzere, her üç algoritmanın performans ölçütlerinde belirtilen presicion (kesinlik) ve recall (duyarlılık) oranları birbirine yakın değerler almıştır. Bu durum, kesin olmamakla beraber her üç algoritma için yapılan deneylerde

overfitting (aşırı uyumluluk) durumunun yaşanmamış olmasını destekleyebilir. Bu çalışmanın sonucu, veri miktarı arttıkça SVM algoritmasının doğruluk oranı azalırken Rastgele Orman algoritmasının doğruluk oranın arttığını göstermektedir.

6- Sonuç

Günümüzde yoğun kullanım alanına sahip olan web sayfalarının güvenirliği her geçen gün azalmaktadır. Bilgi teknolojilerinin gelişimi ile siber saldırı çeşitlerinin sayısı da aynı oranda artmaktadır. Bu saldırı çeşitlerinden birisi olan oltalama saldırıları ise tahmin edilmesi ve ayırt ediciliği düşük olan, kullanıcılar tarafından bilinçsizce erişimi sonrasında büyük zararlara yol açabilecek bir saldırı türüdür. Günlük hayatta her alanda ortaya çıkan bu saldırılar; kullanıcılara masum ve ilgi çekici içerik gibi görünerek kimlik hırsızlığını hedefleyen, kullanıcıların asıl erişmeyi hedeflediği sayfaların kopyalarını üreterek kendi ağına çekip sonrasında kullanıcı bilgilerinin girilmesini sağlar. Bu sayede kullanıcı kimliğinin siber suçluların eline geçmesi birçok zarara ve mahremiyet ihlaline yol açar. Makine öğrenimi algoritmaları ise bu kimlik hırsızı web adreslerinin tespitinde sıklıkla kullanılan çözüm modellerinden birisidir. Makine öğrenimi algoritmaları sayesinde kimlik hırsızı web adresleri belirli özniteliklere göre ayırt edilerek sınıflandırılır. Bu sınıflandırılma sonucunda ise erişilen bir web adresinin oltalama adresi olup olmadığı önceden anlaşılarak bireyler ve kurumlar için oluşabilecek zararlar en aza indirilmiş olur.

Bu çalışmada, makine öğrenmesi algoritmaları olan Rastgele Orman (RO), Destek Vektör Makinesi (DVM) ve Karar Ağacı (KA) kullanılarak oltalama web adreslerinin doğru bir şekilde sınıflandırılması amaçlanmıştır. Çalışmanın deneysel analiz bölümünde, 6 farklı özniteliğe sahip Ulusal Siber Olaylara Müdahale Merkezi (USOM) tarafından yayınlanan oltalama web adresleri ile Aalto Üniversitesine ait meşru web adresleri bulunan iki farklı veri seti kullanılmıştır. Bu veri setleri, Weka ve Scikit-learn ortamlarında ilgili algoritmalar üzerinden teste tabii tutulmuştur. Weka için 1500’ü meşru, 1500’ü oltalama olmak üzere toplamda 3000 adet web adresinin %80'i eğitim, %20'si test amaçlı kullanılmıştır. Elde edilen bulgulara göre en başarılı algoritmanın %99,06 oranıyla Destek Vektör Makinesi olduğu tespit edilmiştir. Scikit-Learn yazılımı için ise 48,009 adet oltama ve 48,009 adet meşru web adresi ile toplamda 96 bin URL içeren geniş bir veri seti oluşturulmuştur ve %80'i eğitim, %20'si test amaçlı kullanılmıştır. Scikit-learn için ise en başarılı algoritmanın %99,1 oranında Rastgele Orman (RO) algoritması olduğu belirlenmiştir.

Bu çalışmadan çıkarılacak sonuçlara göre, kullandığımız veri setinin miktarı artıkça Destek Vektör Makinesi (DVM) algoritmasının başarı oranı düşerken Rastgele Orman (RO) algoritmasının başarı oranının artığı gözlemlenmiştir. Bunun sebebi, Destek Vektör Makinesi (DVM) algoritması veri setimiz üzerinde çalışırken küçük miktardaki verileri daha başarılı bir şekilde sınıflandırırken daha büyük verilerde iyi bir performans gösterememesidir. Rastgele Orman (RO) ise küçük miktardaki veriler üzerinde çalışırken çok fazla budama işlemi gerçekleştirerek eğitim aşamasında yanlı bir performans göstermiştir. Bu demek oluyor ki veri setimizin miktarı artıkça, Rastgele Orman (RO) algoritması daha verimli bir şekilde budama imkânı bulmuştur, bu da daha iyi bir performans sergileyebilmesine olanak sağlamıştır. Aynı zamanda Rastgele Orman (RO) algoritması, kendi içerisinde oluşan birden fazla karar ağacını belirli kriterlere göre budayarak daha anlamlı ve verimli bir hale getirerek

deney verilerindeki aşırı uyum problemini de ortadan kaldırarak oltalama adresi tespiti konusunda en başarılı sonuçları verebilmektedir (Bayraktar, 2019: 42).

Kaynakça

1) Abdelhamid, M. (2020). The role of health concerns in phishing susceptibility: Survey design study. *Journal of medical Internet research*, *22*(5), e18394.
2) Aggarwal, C. C. (2016). Outlier analysis second edition 67.
3) Ahammad, S. H. Vd. (2022). Phishing URL detection using machine learning methods. *Advances in Engineering Software*, 173, 103288.
4) Akinyelu, A. A., & Adewumi, A. O. (2014). Phishing Detection Using Machine Learning: A Comprehensive Analysis, 87-102.
5) Arslan, R. S. (2021). Kötücül Web Sayfalarının Tespitinde Doc2Vec Modeli ve Makine Öğrenmesi Yaklaşımı. *Avrupa Bilim ve Teknoloji Dergisi*, Sayı: 27, 792 - 801.
6) AYDIN, C. (2018). Makine öğrenmesi algoritmaları kullanılarak itfaiye istasyonu ihtiyacının sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, (14),139-141
7) Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 41.
8) Baktır, N. & Atay, Y. (2022). Makine Öğrenmesi Yaklaşımlarının Spam-Mail Sınıflandırma Probleminde Karşılaştırmalı Analizi. *Bilişim Teknolojileri Dergisi*, Cilt: 15, Sayı: 3, 349 - 364.
9) Balogun, A. Vd. (2021). Rotation forest-based logistic model tree for website phishing detection. In *Computational Science and Its Applications–ICCSA 2021: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part IX 21* (pp. 154-169). Springer International Publishing.
10) Bayraktar, B. (2019). Rastgele Orman ve Aşırı Öğrenme Makineleri Teknikleri ile Oltalama Saldırılarının Tespiti. Yüksek Lisans Tezi. İstanbul Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, İstanbul, Toplam Sayfa Sayısı: 102.
11) Breiman, L. (2001). Random Forests. *Kluwer Academic Publishers*, 45, 5–32.
12) Buber, E. Vd. (2017, October). Detecting phishing attacks from URL by using NLP techniques. In *2017 International conference on computer science and Engineering (UBMK)* (pp. 337-342). IEEE.
13) Bukth, T., & Huda, S. S. (2017). The soft threat: *The story of the Bangladesh bank reserve heist*. SAGE Publications: SAGE Business Cases Originals.
14) Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, *76*(3), 503-514.
15) Chouseinoglou, O., & Şahin, İ. (2019). Metin madenciliği, makine ve derin öğrenme algoritmaları ile web sayfalarının sınıflandırılması. *Yönetim Bilişim Sistemleri Dergisi*, *5*(2), 29-43.
16) Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

17) Coşar, M., & Deniz, E. (2021). Makine öğrenimi algoritmaları kullanarak kalp hastalıklarının tespit edilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (28),1112-1115.
18) Çelik, Ö. (2018). A research on machine learning methods and its applications. *Journal of Educational Technology and Online Learning, 1(3),* 26-38.
19) Dalianis, H. (2018). *Clinical text mining: Secondary use of electronic patient records*. Springer Nature, 181.
20) Dinler, Ö. B. & Şahin, C. B. (2021). Prediction of Phishing Web Sites with Deep Learning Using WEKA Environment. *Avrupa Bilim ve Teknoloji Dergisi*, Sayı: 24, 35-41.
21) Dogukan, A. K. S. U., Abdulwakil, A., & Aydin, M. A. (2017). Detecting phishing websites using support vector machine algorithm. *PressAcademia Procedia*, 5(1), 2-5.
22) Garreta, R., & Moncecchi, G. (2013). *Learning scikit-learn: machine learning in python* (Vol. 2013). Birmingham: Packt Publishing, 46-47.
23) Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques.* Morgan kaufmann, 78.
24) Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer, 67.
25) Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). London: Chapman and Hall, 25.
26) Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8), 832-844.
27) Hossain, S. Vd. (2020). Machine learning-based phishing attack detection. *International Journal of Advanced Computer Science and Applications*, *11*(9).
28) Ilgun, E. G., & Samet, R. (2024). Increasing the performance of intrusion detection models developed using machine learning method with preprocessing applied to the dataset. *JOURNAL OF THE FACULTY OF ENGINEERING AND ARCHITECTURE OF GAZI UNIVERSITY*, *39*(2), 679-692.
29) Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, 19(2), 155-156.
30) James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer,42.
31) James, J. Vd. (2013, December). Detection of phishing URLs using machine learning techniques. *In 2013 international conference on control communication and computing (ICCC)* (pp. 304-309). IEEE.
32) Jolliffe, I. T. (2011). Principal Component Analysis. Springer, 92.
33) Kalaycı, T. E. (2018). Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 24[5], 870-878.
34) Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2014). *Preprocessing techniques for text mining. International Journal of Computer Science & Communication Networks*, 5(1), 1.
35) Korkmaz, A. & Büyükgöze, S. (2019). Sahte web sitelerinin sınıflandırma algoritmaları ile tespit edilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, Sayı: 16, 826-833.

36) Koşan, M. Vd. (2018). Kimlik avı web sitelerinin tespitinde makine öğrenmesi algoritmalarının karşılaştırmalı analizi. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *24*(2), 276-282.
37) Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). *Data preprocessing for supervised leaning. International journal of computer science*, 1(2), 116.
38) Kulkarni, A. & Brown, L. (2019). Phishing Websites Detection using Machine Learning. *International Journal of Advanced Computer Science and Applications*, *Vol. 10*, No. 7.
39) Kulkarni, A. (2023). Convolution Neural Networks for Phishing Detection. *International Journal of Advanced Computer Science and Applications*, *Vol. 14*, No. 4.
40) Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons, 29.
41) Mahajan, R. & Siddavatam, I. (2018). Phishing website detection using Machine Learning algorithms. *International Journal of Computer Applications*, 181(23), 45-47.
42) Miyamoto, D. Vd. (2009). An evaluation of machine learning-based methods for detection of phishing sites. In *Advances in Neuro-Information Processing: 15th International Conference, ICONIP 2008, Auckland, New Zealand, November 25-28, 2008, Revised Selected Papers, Part I 15* (pp. 539-546). Springer Berlin Heidelberg.
43) Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert systems with applications*, *53*, 231-242.
44) Özen, N. S., Saraç, S., & Koyuncu, M. (2021). COVID-19 vakalarının makine öğrenmesi algoritmaları ile tahmini: Amerika Birleşik Devletleri örneği. *Avrupa Bilim ve Teknoloji Dergisi*, (22), 135-138
45) Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
46) Provost, F., & Fawcett, T. (2013). Data Science for Business: *What you need to know about data mining and data-analytic thinking*. " O'Reilly Media, Inc.",54.
47) Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1, 81-106.
48) Rathee, D., & Mann, S. (2022). E-Posta Phishing Saldırılarının Algılanması – Makine Öğrenimi ve Derin Öğrenme Kullanımı, 1-5.
49) Ravi, V. Vd. (2020). A Machine Learning approach towards Phishing Email Detection. *CEUR-WS.org*, Vol. 2124, Paper 7.
50) Rencher, A. C., & Christensen, W. F. (1987). Methods of Multivariate Analysis, 77.
51) SEVGEN, S. C., & ALİEFENDİOĞLU, Y. (2020). Mass apprasial with a machine learning algorithm: random forest regression. *Bilişim Teknolojileri Dergisi*, 13(3), 301-309
52) Selvan, K., & Vanitha, M. (2016). A Machine Learning Approach for Detection of Phished Websites Using Neural Networks. *International Journal of Recent Technology and Engineering (IJRTE)*, *4*(6), 19-23.
53) Soysal, T. (2006). İnternet Alan ve Adları Sistemi ve Tahkim Kuruluşlarının UDRP Kurallarında Göre Verdikleri Kararlara Eleştirel Bir Yaklaşım-1. *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *1*(21), 481-507.

54) Toğaçar, M. (2021). Web Sitelerinde Gerçekleştirilen Oltalama Saldırılarının Yapay Zekâ Yaklaşımı ile Tespiti. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *10*(4), 1603-1614.
55) Veranyurt, Ü., DEVECİ, A., ESEN, M. F., & VERANYURT, O. (2020). MAKİNE ÖĞRENMESİ TEKNİKLERİYLE HASTALIK SINIFLANDIRMASI: RANDOM FOREST, K-NEAREST NEIGHBOUR VE ADABOOST ALGORİTMALARI UYGULAMASI. *Uluslararası Sağlık Yönetimi ve Stratejileri Araştırma Dergisi,* 6(2), 276-284
56) Wei, Y., & Sekiya, Y. (2022). Sufficiency of ensemble machine learning methods for phishing websites detection. *IEEE Access*, 1-7.