OLTALAMA SALDIRILARININ TESPİTİ İÇİN ÖZELLİK SEÇİM YAKLAŞIMI

2023
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Ahmet Selim KÜÇÜKKARA

Tez Danışmanı
Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU

OLTALAMA SALDIRILARININ TESPİTİ IÇIN ÖZELLİK SEÇİM YAKLAŞIMI

Ahmet Selim KÜÇÜKKARA

Tez Danışmanı
Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

KARABÜK
Temmuz 2023

Ahmet Selim KÜÇÜKKARA tarafından hazırlanan "OLTALAMA SALDIRILARININ TESPITI IÇIN ÖZELLIK SEÇIM YAKLAŞIMI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU .....
Tez Danışmanı, Bilgisayar Donanımı Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 28/07/2023

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Doç. Dr. Baha ŞEN (AYBÜ)	.....
Üye : Doç. Dr. İlhami Muharrem ORAK (KBÜ)	.....
Üye : Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU (KBÜ)	.....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Müslüm KUZU .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Ahmet Selim KÜÇÜKKARA

ÖZET

Yüksek Lisans Tezi

OLTALAMA SALDIRILARININ TESPITI IÇIN ÖZELLIK SEÇIM YAKLAŞIMI

Ahmet Selim KÜÇÜKKARA

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU
Temmuz 2023, 74 sayfa

Bu çalışmada özellik sayısını azaltma ve özellik seçimine yönelik yeni bir yaklaşım önerilmiştir. Yaklaşım ile iki popüler özellik önem yöntemini ve GRIS ortalama yöntemini birleştirerek yüksek algılama doğruluğunu korumak hedeflenmiştir. 2 farklı oltalama benchmark veri seti üzerinde 12 farklı algoritma kullanılarak sonuçlar toplanmıştır. Hem veri seti hem de algoritma bazında sonuçlar karşılaştırılmıştır. Önerilen özellik seçme yaklaşımının algoritmalar için eğitim ve test süresini iyileştirdiği görülmüştür. Mendeley 2018 veri seti için özelliklerin sadece %27,08'i ile LightGBM algoritmasında doğruluğun %98,37'ye ulaşabildiği görülmüştür. Mendeley 2020 veri seti için ise özelliklerin sadece %17,12'si ile Random Forest algoritmasında doğruluğun %97,12'ye ulaşabildiği görülmüştür. Aynı zamanda bellek kullanımında Mendeley 2018 veri setinde %72,95, Mendeley 2020 veri setinde ise %82,88 düşüş gözlemlenmiştir. Böylece önerilen yaklaşımla elde edilen

verimli özellikler kullanılarak çok daha az bellek kullanımı ile daha kısa sürede yüksek doğruluğun korunabileceği ortaya konulmuştur.

Anahtar Sözcükler : Oltalama saldırısı, Özellik seçimi, Makine öğrenmesi, Algoritma.

Bilim Kodu : 92432

ABSTRACT

Master Thesis

FEATURE SELECTION APPROACH FOR PHISHING DETECTION

Ahmet Selim KÜÇÜKKARA

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Oğuzhan MENEMENCİOĞLU
July 2023, 74 pages

This study introduces a novel approach to feature selection and reduction, aiming to uphold a high level of detection accuracy through the amalgamation of two prevalent feature importance methods and the GRIS averaging technique. The results were obtained by applying 12 distinct algorithms across two diverse phishing benchmark datasets, subsequently comparing both dataset and algorithm-based outcomes. The proposed feature selection approach exhibited the capacity to notably diminish the training and testing duration for the algorithms. In the case of the Mendeley 2018 dataset, it was observed that the LightGBM algorithm achieved an accuracy of 98.37%, utilizing just 27.08% of the features. Similarly, for the Mendeley 2020 dataset, the Random Forest algorithm attained an accuracy of 97.12% with a mere 17.12% of the features. Concurrently, a substantial reduction of 72.95% in memory usage was observed in the Mendeley 2018 dataset, along with an 82.88% reduction in the Mendeley 2020 dataset. These findings collectively demonstrate that by utilizing the efficient features derived from the proposed approach, it is possible to

maintain high accuracy levels while significantly reducing memory usage and expediting processing time.

Key Word **:** Phishing, Feature selection, Machine learning, Algorithm.
Science Code : 92432

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Dr. Öğr. Üyesi Oğuzhan MENEMENCİOĞLU'na sonsuz teşekkürlerimi sunarım.

Akademik hayatımın bu başlangıç aşamalarında bana yol gösteren ve yardımcı olan Zonguldak Bülent Ecevit Üniversitesi'ndeki çalışma arkadaşlarım ve hocalarıma çok teşekkür ederim.

Son olarak tüm hayatım boyunca benim yanımda olan, aldığım kararları her zaman destekleyen, sadece bu çalışma sürecinde değil tüm hayatım boyunca bana moral veren babam Hamza KÜÇÜKKARA'ya, annem Ayşe KÜÇÜKKARA'ya ve ağabeyim Mehmet Esad KÜÇÜKKARA'ya sonsuz şükranlarımı sunar ve teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

Sayfa

KISALTMALAR DİZİNİ

KISALTMALAR

ADA : Adaboost

APWG : Anti-Phishing Working Group (Anti-Oltama Çalışma Grubu)

CART : Classification and Regression Tree (Sınıflandırma ve Regresyon Ağacı)

DCT : Decision Tree (Karar Ağacı)

GBDT : Gradient-Boosted Decision Trees (Gradyan-Artırılmış Karar Ağaçları)

GNB : Gaussian Naive Bayes

GRIS : Golden Ratio in Statistics (İstatistikte Altın Oran)

KNN : K-Nearest Neighbors (K-En Yakın Komşular)

LDA : Linear Discriminant Analysis (Lineer Ayrım Analizi)

LGB : Light Gradient Boosting Machines

LR : Logistic Regression (Lojistik Regresyon)

MDI : Mean Decrease in Impurity (Karışıklıktaki Ortalama Azalma)

NB : Naive Bayes

RF : Random Forest (Rastgele Orman)

SGD : Stochastic Gradient Descent (Stokastik Gradyan Artırımı)

SVM : Support Vector Machine (Destek Vektör Makinesi)

XGB : Extreme Gradient Boosting

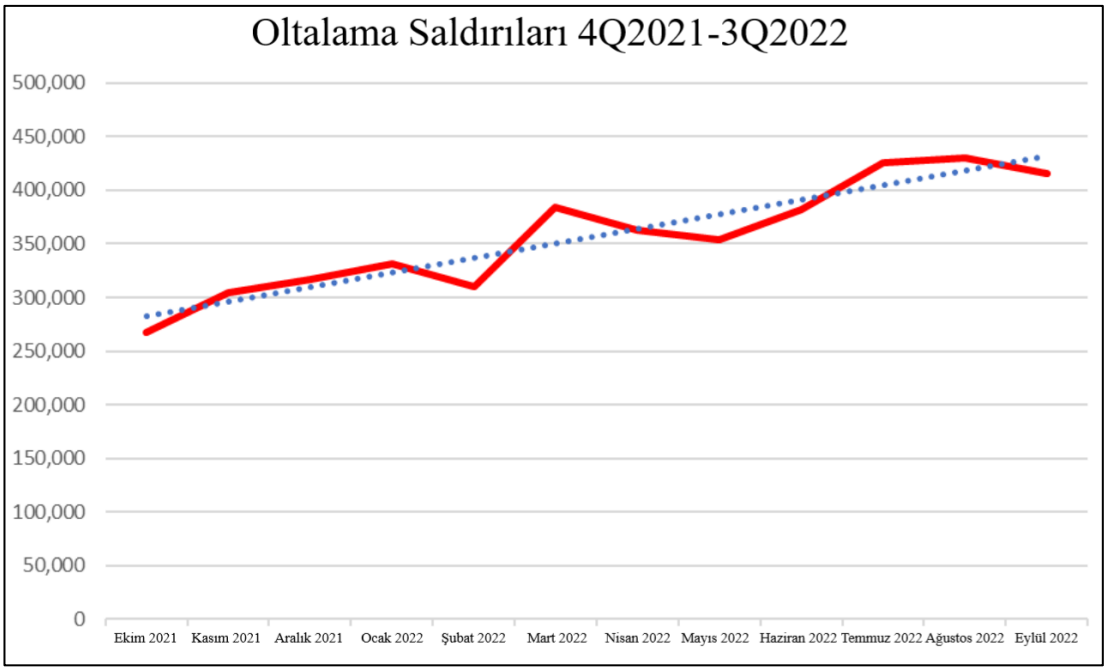
BÖLÜM 1

GİRİŞ

1.1. ARAŞTIRMANIN KONUSU

Son yıllarda, internetin hızlı yayılması ve dijitalleşmenin artması ile kullanıcılar ve organizasyonlar için siber güvenlik tehditleri giderek artmaktadır [1]. Özellikle oltalama saldırıları, siber güvenlik açısından ciddi bir tehdit oluşturmaktadır. Oltalama saldırıları, kötü niyetli aktörlerin kullanıcıların hassas bilgilerini ele geçirmek için sahte veya hileli e-postalar, web sayfaları veya mesajlar aracılığıyla kullanıcıları aldatma yöntemini kullanır. Bu tür saldırılar, kullanıcıların güvenilir bir markaya bürünmüş fakat sahte olan bir kaynağa girdiklerinde, kimlik bilgilerini, finansal verilerini veya diğer hassas bilgilerini ifşa etmelerine neden olabilir.

Dünyayı etkisi altına alan COVID-19 pandemisi nedeniyle insanların yaşam tarzlarında büyük değişimler yaşandı. Dünyanın dört bir yanındaki insanlar çevrimiçi alışverişe, öğrenmeye ve uzaktan çalışmaya giderek daha fazla uyum sağladılar. Neticede internetin yaygın kullanımı, oltalama saldırılarını daha da yaygın hale getirmiştir. Verizon'un 2022 raporuna göre, kötü amaçlı saldırıların %82'si insan faktörü içermektedir. Sosyal mühendisliğin bir yolu olan oltalama saldırıları da her yıl daha popüler hale gelmektedir. Yine Verizon'un 2020 raporuna göre, oltalama saldırılarının oranı %25 iken, 2021'de %11 artarak %36'ya yükseldiği görülmüştür [1]. APWG (Anti-Phishing Working Group) raporuna göre 2022'nin üçüncü çeyreğinde, toplam 1.270.883 oltalama saldırısı gözlemlenmiş ve bu APWG'nin şu zamana dek gözlemlediği en kötü oltalama çeyreği olarak kayıtlara geçmiştir. Şekil 1.1'de benzersiz oltalama saldırılarının aylara göre dağılımı görülmektedir [2].

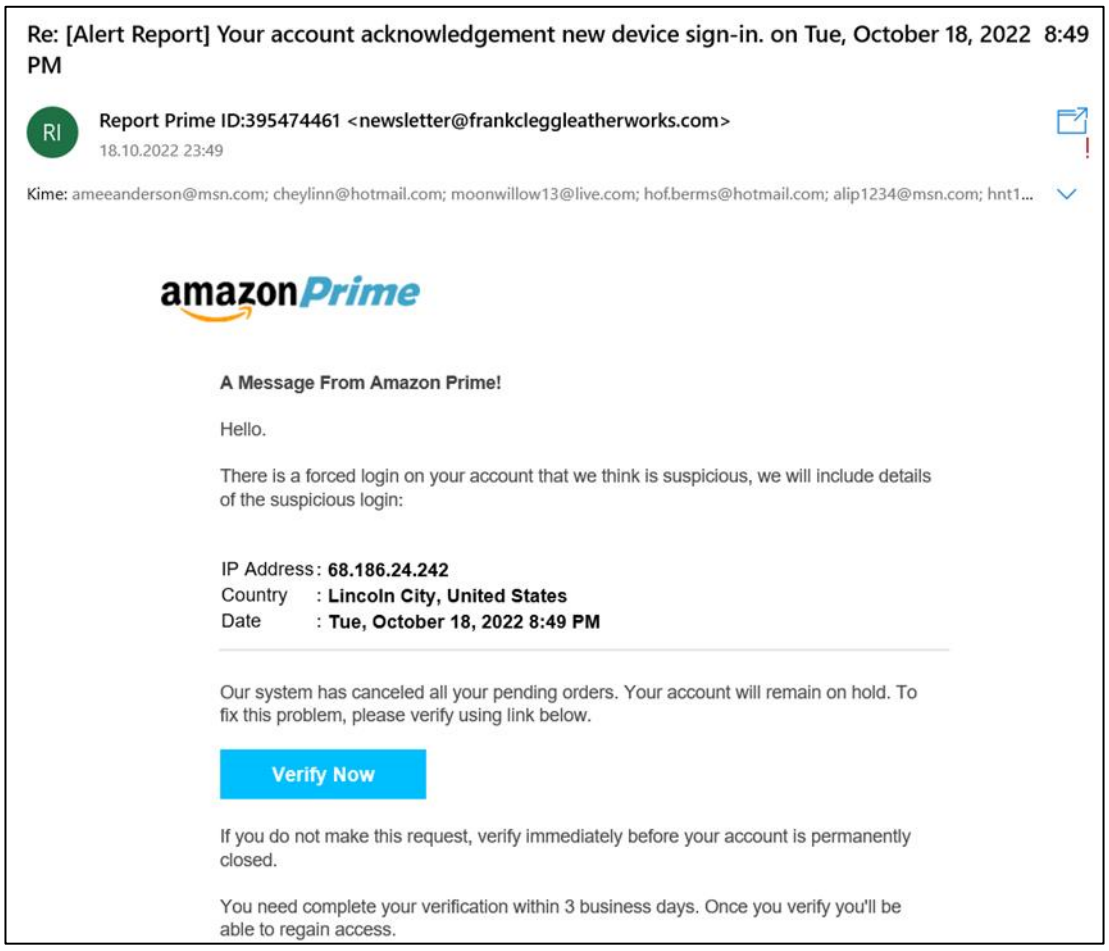


Şekil 1.1. 2021'in 4. çeyreği ile 2022'nin 3. çeyreği arasındaki benzersiz oltalama saldırılarının ay bazında dağılımı [2].

1.2. ARAŞTIRMA MOTİVASYONU

Finansal kurumlar, yazılım hizmetleri ve sosyal medya sektörleri oltalama saldırılarına en fazla hedef olan sektörlerdir [2]. Rockstar Games ve Uber gibi büyük şirketler de oltalama saldırılarının kurbanı olmuştur [3]. Oltalama, sürekli evrimleşen ve devam eden bir tehlikedir. Rastgele birçok alıcıya oltalama e-postaları gönderilir ve yalnızca çok küçük bir yüzdesinin tepki vermesi beklenir. Şekil 1.2'de bir oltalama e-postası örneği görülmektedir. Bilinen ve güvenilir bir markadan gelen bir e-posta, kurbanın hesabına yabancı bir yerden giriş yapıldığını ve hesabın kalıcı olarak kapanmasından kaçınmak için kurbanın doğrulamada bulunması gerektiğini belirtmektedir. Burada "tehdit" unsuru olarak "kalıcı kapanma" kullanılmıştır ve bu "tehdit" unsurları oltalama e-postalarında önemli bir rol oynar. Kurban "doğrula" butonuna tıkladığında, hesabına ilişkin kişisel verileri girmesi istenen ve orijinalinin kopyası olarak tasarlanmış sahte bir web sitesi açılır. Eğer kurban bu web sitesinin sahte olduğunu fark etmez ve bilgilerini verir ise saldırgan amacına ulaşmış olur.

Çoğu oltalama tespit sisteminin iki önemli gereksinimi vardır. İlki, gerçek zamanlı bir ortamda hızlı erişim, ikincisi ise yüksek tespit oranıdır. Oltalama web sitelerini tespit etmek için makine öğrenmesi kullanılarak bu gereksinimler yerine getirilebilir.



Şekil 1.2. Oltalama e-postası.

1.3. AMAÇ VE KATKI

Bu çalışmanın amacı, bir oltalama veri setinde önerilen yaklaşım kullanılarak, verimli özelliklerin seçilmesi ve farklı algoritmalarla elde edilen verimli özellikler ve bütün özellikler kullanıldığında elde edilen sonuçların karşılaştırılmasıdır. Ana hedef, verimli özellikler seçilirken yüksek doğruluk oranının korunulmasıdır. Çalışmanın ana katkısı ise, yeni bir yaklaşımla özellik seçimi yapmaktır. Özellik seçim yaklaşımı olarak, iki popüler özellik önem hesaplama yöntemi ve GRIS ortalama hesabı kullanılarak yüksek tahmin doğruluğu elde etmek için verimli

özellikler seçilmiştir. Özellik sayısı azalması beklendiği için bellek kullanımının azalması ve model eğitimi ve testi için gereken sürenin azalması öngörülmüştür.

1.4. TEZİN BÖLÜMLERİ

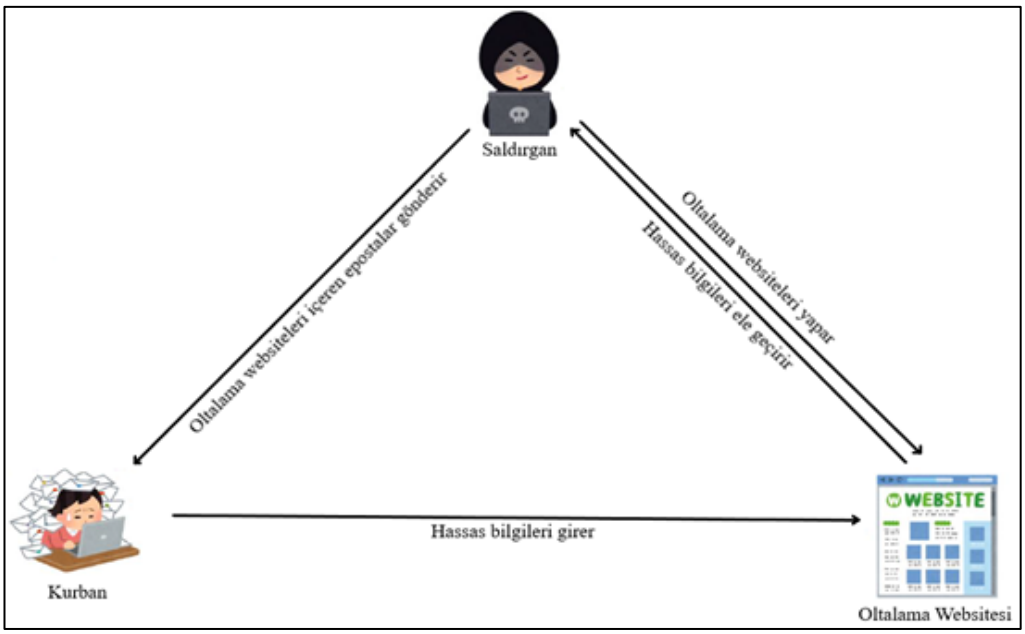
Tez 5 bölümden oluşmaktadır. Bu ilk bölümde çalışma ile ilgili kısa bir bilgi ve çalışmanın motivasyonu, amacı ve katkıları verilmiştir. İkinci bölümde, çalışma ile ilgili arka plan bilgileri: oltalama mekanizması, oltalama tespit yöntemleri ve literatür taraması yer almaktadır. Üçüncü bölümde bu çalışmada kullanılacak veri setleri, makine öğrenmesi modelleri, özellik önem yaklaşımları, GRIS hesabı, önerilen yaklaşım ve performans metrikleri yer almaktadır. Dördüncü bölümde makine öğrenmesi algoritmaları kullanılarak elde edilen sonuçlar yer almaktadır. Beşinci ve son bölümde ise elde edilen sonuçlar çalışmanın amacı doğrultusunda yorumlanarak tartışılmış ve gelecekte yapılabilecek çalışmalardan bahsedilmiştir.

BÖLÜM 2

ARKAPLAN

2.1. OLTALAMA MEKANİZMASI

"Oltalama" kelimesi internet üzerinde ilk kez 1996 yılında dikkatsiz bir America Online (AOL) kullanıcısını kandırıp şifresini ele geçirip hesabını çalan bir grup bilgisayar korsanı tarafından kullanılmıştır [4]. Oltalama saldırısının motivasyonu, kurbanları gizli bilgiler vermeleri için manipüle etmektir [5]. Oltalama saldırısını gerçekleştirmek için bilgisayar korsanlarının meşru web sitelerini taklit etmesi ve ardından bu kötü amaçlı web sitelerini e-postalar, anlık mesajlar, sosyal ağ siteleri, çok oyunculu oyunlar ve diğer yollarla kurbanlara göndermesi gerekir. Mızrak oltalama saldırısı, balina avcılığı, siteler arası komut dosyası çalıştırma (XSS), QRishing, drive-by download, kötü amaçlı tarama uzantıları vb. gibi çeşitli oltalama saldırıları mevcuttur [6-10].



Şekil 2.1. E-posta ile oltalama mekanizması [11].

2.2. TESPİT YÖNTEMLERİ

Oltalama web sitelerini tespit etmek için birçok yöntem vardır ve oltalama saldırılarını önlemek için çeşitli yöntemler önerilmiştir. Yöntemler genellikle iki bölüme ayrılır; yazılımla tespit yöntemleri ve kullanıcıların gelecekteki saldırılara karşı hazırlıklı olmaları için eğitilmesi [12].

Şekil 2.2. Oltalama tespit yöntemlerine genel bakış [12].

2.2.1. Kullanıcı Eğitimi

Oltalama tespitinde kullanıcı farkındalığı oldukça önemlidir. Bilgisayar kullanıcıları herhangi bir bağlantıya tıklayıp bilgi girerlerse hiç şüphesiz kandırılacak ana hedef olacaklardır. Meslek öğrencilerinin oltalama saldırılarına karşı farkındalığını artırmaya yönelik bir çalışmada yazarlar, öğrenciler arasında farkındalık düzeyini yükseltmek için bir model geliştirmişler ve sonuç olarak modelin öğrencilerin özellikle oltalama saldırılarına karşı bilgi güvenliği farkındalığı becerilerini geliştirmelerine yardımcı olabileceğini görmüşlerdir [13]. Kullanıcıların farkındalığını artırmanın yanı sıra, kullanıcıların korunması için ek oltalama tespit destek sistemleri gereklidir. Ayrıca, insanları URL'lere erişme, bunları ayrıştırma ve kontrol etme konusunda eğitmeyi amaçlayan, güvenilir ve güvenilir olmayan web sitelerini ayırt etmelerini sağlayan NoPhish adında bir mobil oyun bulunmaktadır [14].

2.2.2. Liste Tabanlı Tespit Yaklaşımı

Liste tabanlı oltalama tespit yaklaşımı, oltalama ve meşru web sitelerini sınıflandırmak için iki liste kullanır: beyaz ve kara liste. Jain vd., bireysel kullanıcılar tarafından erişilen meşru sitelerin otomatik olarak güncellenen beyaz listesini kullanarak oltalama saldırılarına karşı koruma sağlamak için yeni bir yaklaşım önerdiler. Deneysel sonuçları, önerilen yaklaşımın %86,02 gerçek pozitif oranı ve %1,48'den daha düşük yanlış negatif oranına sahip olduğu için oltalama saldırılarına karşı koruma sağlayabildiğini göstermektedir [15]. Kara listeler, internet kullanıcılarını oltalama saldırılarına karşı korumada hayati bir rol oynar. Kara listelerin etkinliği, diğer özelliklerinin yanı sıra boyutlarına, kapsamlarına, güncellenme hızlarına, sıklıklarına ve doğruluklarına bağlıdır [16]. 3 temel oltalama kara listesi vardır: Google Safe Browsing (GSB), OpenPhish (OP) ve PhishTank (PT) [17-19]. Liste tabanlı çözümler hızlı erişim süresine sahiptir, ancak düşük algılama oranı dezavantajına sahiptirler.

2.2.3. Görsel Benzerliğe Dayalı Tespit Yaklaşımı

Kullanıcıları doğru web sitesinde gezindiklerine inandırmak için, oltalama web siteleri görünüşte kimliğine büründüğü meşru web sitelerine benzemektedirler. Görsel benzerliğe dayalı oltalama tespit teknikleri, karar vermek için metin içeriği, metin formatı, HTML etiketleri, Basamaklanmış Stil Katmanları (CSS), resim ve benzeri özelliklerden yararlanır [20]. Benzer renkli alt türler izlenerek oltalama web siteleri kapsamlı bir şekilde tespit edilebilir. Sahar vd. bir benzerlik ölçüsüyle web siteleri için profilleri öğrenebilen, yeni görsel görünümlere sahip sayfalara genellenebilen, bir benzerlik tabanlı oltalama tespit çerçevesi olan VisualPhishNet adında bir yaklaşım önermişlerdir [21]. Benzerliğe dayalı algılama yöntemleri özellikle görünmeyen oltalama web sitelerine karşı yeterli koruma sağlamaz.

2.2.4. Sezgisel Yöntemler ve Makine Öğrenmesi Temelli Tespit Yaklaşımı

Sezgisel yöntemler, benzer sorunları çözmek için önceki deneyimlerden türetilen stratejilerdir. Makine öğrenmesi, insan müdahalesini en aza indirgeyerek veri

kümelerinden öğrenme yeteneğine sahip olup, kalıpları tanımlayabilir ve tahminlerde bulunabilir. Oltalama tespitinde kullanılan makine öğrenmesi algoritmalarının çoğu, denetimli makine öğrenmesi olarak sınıflandırılır. Oltalama tespitine dayalı makine öğrenmesi, URL'ler, bağlantı bilgileri, sayfa içeriği, dijital sertifika, web sitesi trafiği ve diğer kaynaklar gibi özellikleri çıkarır [22-24]. Oltalama tespitinin doğruluğu, özellik kümesine, eğitim verilerine ve makine öğrenmesi algoritmalarına bağlıdır.

M. Korkmaz vd. web sayfasının URL'sini analiz ederek, URL'lerden 58 özellik belirlemiş ve 8 farklı algoritmayı karşılaştırmışlardır. Aynı zamanda, deneysel sonuçları elde etmek için üç veri kümesi kullanmışlardır. Sonuç olarak, RF %94,59, %90,50 ve %91,26 ile en yüksek doğruluk oranlarını göstererek diğer algoritmalara kıyasla daha iyi bir performans göstermiştir. Ayrıca makine öğrenmesi algoritmalarının farklı veri setlerinde de verimli olduğunu göstermiştir [25].

Gururaj vd. DCT, KNN, Lineer SVC, RF, tek sınıf SVM gibi birçok teknik kullanmışlar ve bunlardan RF algoritmasının yaklaşık %96,87 ile en yüksek doğruluğa sahip olduğunu gözlemlemişlerdir [26].

Vahid vd. PhisTank sitesinden 11000 örnek web sitesi kullanmış ve 30 özellik çıkarmışlardır. LR, DCT, SVM, Ada Boost, RF, Sinir Ağları, KNN, Gradient Boost ve XGBoost dahil olmak üzere 12 algoritmayı değerlendirmişlerdir. Elde ettikleri sonuçlara göre, topluluk tabanlı makine öğrenmesi algoritmalarının birkaç zayıf öğreniciyi daha güçlü bir hale getirebileceğini gösteren RF ve XGBoost algoritmalarının hem hesaplama süresi hem de doğruluk açısından bir araya getirmede çok iyi bir performans elde ettiğini gözlemlemişlerdir [12].

Ammar vd. makine öğrenmesi tekniklerini kullanarak oltalama web sitesi tespiti için en son teknikleri içeren bir araştırma sunmaktadırlar. Bu araştırmada, makine öğrenmesi tekniklerine dayalı olarak web sitelerinin oltalama sorununa yönelik çözümleri tanımlamışlardır. İncelenen yaklaşımların çoğunun geleneksel makine öğrenmesi tekniklerine odaklandığını gözlemlemişlerdir. RF, SVM, NB ve Ada Boost, literatürde incelenen güçlü makine öğrenmesi teknikleridir. Bu çalışmada tanımlanan makine öğrenmesi tekniklerine yönelik zorluklar, aşırı öğrenme

(overfitting), düşük doğruluk ve yeterli eğitim verisinin bulunmaması durumunda makine öğrenmesi tekniklerinin etkisizliğini içermektedir [27].

Yi ve Sekiya, oltalama ve meşru web siteleri arasındaki bariz farkları ve ilişkileri araştırmak için 2020'de yayınlanan oltalama web siteleri veri setinin 111 özelliğini analiz etmişlerdir. Yaygın olarak kullanılan 11 makine öğrenmesi algoritmasını uygulayarak ve performanslarını değerlendirerek, oltalama tespit algoritması olarak RF algoritmasını seçmişlerdir. Özellik önem yöntemlerine dayanarak, yüksek algılama doğruluğunu korurken özellik sayısını azaltmak için bir çerçeve önermişlerdir. Önerilen özellik seçim çerçevesinin birleştirilmesiyle, yalnızca 14 özellik kullanılarak, oltalama tespit doğruluğunun %97,0'a ulaşabileceğini gözlemlemişlerdir [11].

BÖLÜM 3

METODOLOJİ

3.1. VERİ SETLERİ

Dünya çapında kabul görmüş bir veri seti varsa, araştırmacılar modellerinin performanslarını kıyaslayarak karşılaştırabilirler. Ancak oltalama web siteleri hızlı güncellendikleri ve kısa ömürlü oldukları için oltalama web sitelerinin özelliklerini kaydetmek için standart bir veri seti yoktur.

Algoritmaların ve yöntemlerin performansını objektif bir şekilde değerlendirmek ve karşılaştırmak için kullanılan standardize edilmiş veri kümesine "benchmark veri seti" denir. Benchmark veri setleri, genellikle belirli bir problemin çözümüne yönelik test verilerini içerir ve bu veriler, farklı algoritmaların veya yöntemlerin nasıl performans gösterdiğini nesnel bir şekilde değerlendirmeye yardımcı olur. Bir benchmark veri setinin başarıyla kullanılabilmesi için, veri setindeki gerçek sonuçların, tahmin edilen sonuçlarla karşılaştırılabilmesi gerekmektedir. Bu gerçek ve kesin sonuçlara "ground truth" denir. Ground truth, veri setinin hazırlandığı sırada uzmanlar veya daha önce doğrulanmış kaynaklar tarafından belirlenir. Algoritmaların performansını değerlendirmek için, bu algoritmaların ürettiği sonuçlar ile ground truth arasındaki farklar hesaplanır [28].

Bu çalışmada 2 farklı benchmark veri seti kullanılacaktır. 2 veri setinin farklı sayıda örnek sayısı, özellik sayısı, meşru-oltalama web sitesi dağılımına sahip olması önerilen yaklaşımın daha iyi test edilmesini sağlayacaktır.

3.1.1. Mendeley 2018 Veri Seti

Kullanılacak ilk veri seti 2018 yılında Choon Lin tarafından yayınlanan, Ocak-Mayıs 2015 ve Mayıs-Haziran 2017 arasında indirilen 5.000 oltalama web sitesi ve 5.000 meşru web sitesinden çıkarılan 48 özelliği içeren veri setidir. Oltalama web siteleri PhishTank ve OpenPhish'ten, meşru web siteleri Alexa ve Common Crawl ile elde edilmiştir [29]. Çizelge 3.1, Mendeley 2018 veri setinde bulunan özellikleri göstermektedir.

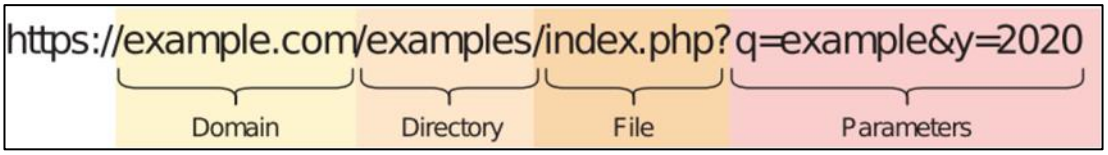
Çizelge 3.1. Mendeley 2018 veri setinin özellikleri [29].

No	**Özellik Adı**	**No**	**Özellik Adı**
Ö1	NumDots	Ö25	NumSensitiveWords
Ö2	SubdomainLevel	Ö26	EmbeddedBrandName
Ö3	PathLevel	Ö27	PctExtHyperLinks
Ö4	UrlLength	Ö28	PctExtResourceUrls
Ö5	NumDash	Ö29	ExtFavicon
Ö6	NumDashInHostname	Ö30	InsecureForms
Ö7	AtSymbol	Ö31	RelativeFormAction
Ö8	TildeSymbol	Ö32	ExtFormAction
Ö9	NumUnderscore	Ö33	AbnormalFormAction
Ö10	NumPercent	Ö34	PctNullSelfRedirectHyperlinks
Ö11	NumQueryComponents	Ö35	FrequentDomainNameMismatch
Ö12	NumAmpersand	Ö36	FakeLinkInStatusBar
Ö13	NumHash	Ö37	RightClickDisabled
Ö14	NumNumericChars	Ö38	PopUpWindow
Ö15	NoHttps	Ö39	SubmitInfoToEmail
Ö16	RandomString	Ö40	IframeOrFrame
Ö17	IpAddress	Ö41	MissingTitle
Ö18	DomainInSubdomains	Ö42	ImagesOnlyInForm
Ö19	DomainInPaths	Ö43	SubdomainLevelRT
Ö20	HttpsInHostname	Ö44	UrlLengthRT
Ö21	HostnameLength	Ö45	PctExtResourceUrlsRT
Ö22	PathLength	Ö46	AbnormalExtFormActionR
Ö23	QueryLength	Ö47	ExtMetaScriptLinkRT
Ö24	DoubleSlashInPath	Ö48	PctExtNullSelfRedirectHyperlinksRT

3.1.2. Mendeley 2020 Veri Seti

Kullanılacak ikinci veri seti 2020 yılında Vrbančič tarafından yayınlanan "Phishing Websites Dataset"dir. Mendeley 2020: URL özelliklerine, URL çözümleme metriklerine ve dış hizmetlere dayalı olarak oltalama web sitelerini tespit etme görevi için çeşitli sınıflandırma yöntemleri oluşturmak ve değerlendirmek amacıyla toplanmış ve hazırlanmıştır. Veri setinin özellikleri altı gruba ayrılabilir:

1. Tüm URL özelliklerine dayalı
2. Alan adı özelliklerine dayalı
3. URL dizini özelliklerine dayalı
4. URL dosyası özelliklerine dayalı
5. URL parametre özelliklerine dayalı
6. URL çözümleme verilerine ve harici metriklere dayalı



Şekil 3.1. Özniteliklerin ayrılması [30].

İlk grup, URL dizgisinin tamamındaki özniteliklerin değerlerine dayanırken, ilk grubu izleyen dört grubun değerleri, Şekil 3.1'de gösterildiği gibi belirli alt dizgilere dayanmaktadır. Son grup, URL çözümleme metriklerinin yanı sıra Google arama dizini gibi harici hizmetlere dayanmaktadır.

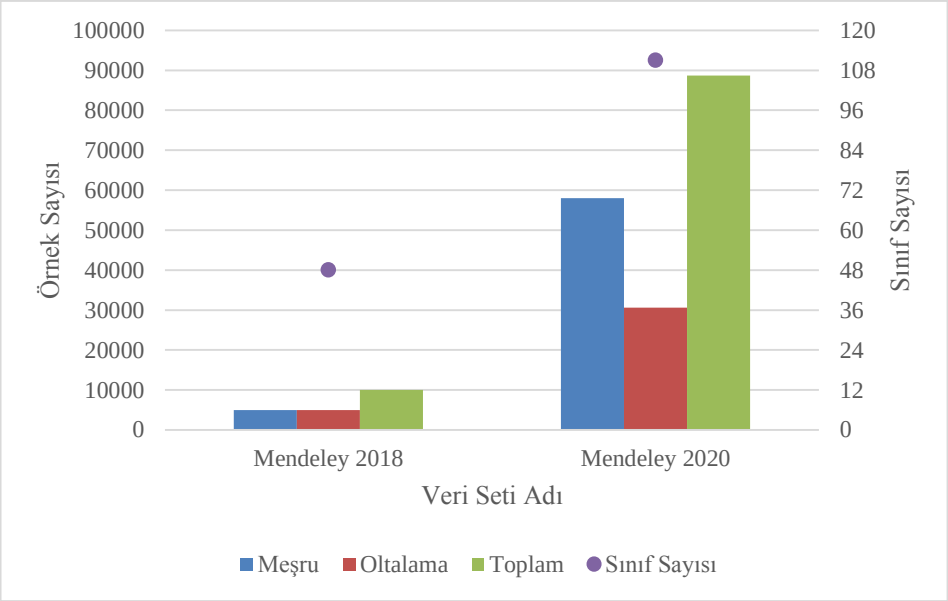
Veri setinde 58.000 adet meşru web sitesi, 30.647 adet oltalama web sitesi olmak üzere toplam 88.647 adet örnek bulunmaktadır. Belirli bir örneğin meşru (değeri 0) veya oltalama (değeri 1) web sitesi olduğunu gösteren hedef oltalama özelliği ile beraber toplam 111 özellik içerir [30].

Çizelge 3.2. Mendeley 2020 veri setinin özellikleri [30].

Grup	İndeks	Açıklama	Veri Tipi
1	1-17	Tüm URL'deki her bir ".-_/?=@&!~,+*#"$%" işareti sayısı	Numerik
2	18-34	Alan adındaki her bir ".-_/?=@&!~,+*#"$%" işareti sayısı	Numerik
3	35-51	Alan adının directorysindeki her bir ".-_/?=@&!~,+*#"$%" işareti sayısı	Numerik
4	52-68	Alan adının dosyasındaki her bir ".-_/?=@&!~,+*#"$%" işareti sayısı	Numerik
5	69-85	Alan adının parametrelerindeki her bir ".-_/?=@&!~,+*#"$%" işareti sayısı	Numerik
6	86-96	Sesli harf sayısı, parametre sayısı, time_response, asn_ip, time_domain_activation, time_domain_expiration, çözümlenen Ips sayısı, çözümlenen NS sayısı, MX sunucusu sayısı, yaşam süresi, yönlendirme sayısı	Numerik
7	97-102	Üst düzey etki alanı karakter uzunluğu, tüm URL'deki karakter sayısı, etki alanı karakter sayısı, dizin karakter sayısı, dosya karakter sayısı, parametre karakter sayısı	Numerik
8	103-111	E-posta var mı, IP adresi biçiminde URL alanı mı, etki alanında "sunucu" veya "istemci" mi, parametrelerde TLD var mı, alan SPF'ye sahip mi, URL geçerli TLD/SSL sertifikasına sahip mi, URL Google'da dizine eklenmiş mi, Google'da dizine eklenmiş alan adı, URL kısaltılmış mı?	Mantıksal

3.1.3. Veri Setlerinin Karşılaştırılması

Çalışmada kullanılacak olan iki veri seti Şekil 3.2'de karşılaştırılmıştır. Mendeley 2018; 5.000 meşru, 5.000 oltalama web sitesi örneğine ve 48 özelliğe sahipken Mendeley 2020; 58.000 meşru 30.647 oltalama web sitesi örneğine ve 111 özelliğe sahiptir. Önerilen yaklaşımda özellikler eleneceği için fazla sayıda özelliğe sahip bir veri setindeki performansı ile az sayıda özelliğe sahip veri setindeki performansı karşılaştırılabilir hale gelmiştir. Mendeley 2020 daha fazla meşru web sitesi örneğine sahip olduğu için gerçek dünya problemine daha yakın bir veri setidir.



Şekil 3.2. Veri setlerinin karşılaştırılması.

3.2. MAKİNE ÖĞRENMESİ MODELLERİ

Makine öğrenmesi, veri kümelerinden öğrenerek kalıpları tanımlayabilir ve minimum insan müdahalesiyle tahminlerde bulunabilir. Makine öğrenmesi modelleri, Yahoo, Gmail ve Outlook gibi önde gelen internet servis sağlayıcıları tarafından filtrelemek ve sınıflandırmak için yaygın olarak kullanılmaktadır [31].

3.2.1. Random Forest (RF)

RF, denetimli makine öğrenimi algoritmasıdır ve sınıflandırma ve regresyon görevlerini gerçekleştirebilir. RF, birçok ağaçtan oluşan topluluk tabanlı bir tekniktir ve tahmin doğruluğunu artırmasıyla bilinir. Algoritma, öncelikle her tahmin ağacını rastgele değerlerle oluşturur ve tüm tahmin ağaçlarından alınan oylamaya göre sonunda kararlar verir. RF, karar ağaçlarında aşırı öğrenmeyi azaltabilir, ancak birçok ağaç inşa ettiği için hesaplama gücü ve kaynaklara gereksinim duyar [32,33].

3.2.2. Logistic Regression (LR)

LR, sınıflandırma problemleri için kullanılan bir denetimli öğrenme algoritmasıdır. Yöntemin temelinde kullanılan lojistik sigmoid fonksiyonu ile çıktısını dönüştürerek, sürekli sayısal değerler yerine olasılık değerleri döndürür. Bu olasılık değeri daha sonra iki ya da daha fazla ayrık sınıfa eşlenerek sınıflandırma yapar [33,34].

3.2.3. Linear Discriminant Analysis (LDA)

LDA, boyut azaltma için yaygın olarak kullanılan bir lineer dönüşüm tekniğidir ve aynı zamanda bir sınıflandırıcı olarak da kullanılabilir. LDA, yeni bir girdi kümesinin her sınıfa ait olma olasılığını hesaplayarak tahminlerde bulunur. En yüksek olasılığa sahip sınıf, çıkış sınıfı olarak belirlenir ve tahmin yapılır. Model, olasılıkları tahmin etmek için Bayes Teoremi'ni kullanır [33,35].

3.2.4. Classification and Regression Tree (CART)

CART, sınıflandırma ve regresyon öğrenme görevleri için kullanılan karar ağacı algoritmalarını tanımlamak için kullanılan bir terimdir. CART algoritması, torbalı (bagged) karar ağaçları, rastgele orman ve güçlendirilmiş (boosted) karar ağaçları gibi önemli algoritmalara temel sağlar. CART modeli, uygun bir ağaç oluşturulana kadar girdi değişkenleri ve bölme noktalarını seçerek ağaç yapısını oluşturur. Her düğüm için belirli bölme noktalarını belirlemek için açgözlü bir yaklaşım kullanılarak, maliyet fonksiyonunu en aza indirmek için girdi özellikleri seçilir. Ağaçların inşası, önceden tanımlanmış bir kriter kullanılarak durdurulur [33,36].

3.2.5. Naïve Bayes (NB)

NB, tahminciler arasındaki bağımsızlık varsayımı ile Bayes Teoremi'ne dayanan bir sınıflandırma tekniğidir. Koşullu olasılığa dayanan basit bir olasılıksal sınıflandırıcıdır. NB, her sınıf için sonsal olasılığı hesaplamak için NB denklemi kullanılarak kolayca oluşturabilir. En yüksek sonsal olasılığa sahip sınıf tahminin sonucudur. NB sınıflandırıcısının sınırlaması, bağımsız tahminciler varsayımıdır,

çünkü gerçek hayatta tamamen bağımsız bir veri kümesi elde etmek neredeyse imkânsızdır [33,37].

3.2.6. K-Nearest Neighbors (KNN)

KNN, yeni bir durum ile mevcut durumlar arasındaki benzerliğe dayalı olarak yeni durumu mevcut kategorilere en benzer kategoriye yerleştirme varsayımına dayalı, klasik bir denetimli makine öğrenme algoritmasıdır. Mesafe ölçümü ile sınıflandırma kararlarını vermek için K-en yakın komşu kullanılır. KNN, sınıflandırma ve regresyon tahmin problemleri için kullanılabilir [33,38].

3.2.7. Support Vector Machine (SVM)

SVM, yaygın olarak kullanılan denetimli makine öğrenme yöntemlerinden biridir. SVM'nin temel amacı, n-boyutlu uzayı sınıflara bölebilecek en iyi çizgiyi veya karar sınırını oluşturmaktır, böylece gelecekte yeni veri noktalarını doğru kategoriye yerleştirebilir. Bu en iyi karar sınırına hiper düzlem denir ve veri kümesini farklı sınıflara ayırabilir. SVM'nin dezavantajı, veri kümesinde gürültü olduğunda iyi performans göstermemesidir. SVM, sınıflandırma ve regresyon problemlerini çözmek için kullanışlıdır [33,39]. Büyük veri setlerinde SVM çok uzun süre harcadığı için[40] büyük veri setlerine daha uygun olan Lineer SVM kullanılacaktır.

3.2.8. Stochastic Gradient Descent (SGD)

SGD, sürekli optimizasyon problemlerini çözmek için kullanılan bir iteratif optimizasyon algoritmasıdır. SGD, büyük veri kümesi üzerinde eğitim yaparken geleneksel Gradyan Artırım (GA) algoritmasının zorluklarından kaçınmak için geliştirilmiştir. GA, tüm veri kümesini kullanarak her güncelleme adımında gradyanı hesaplar ve model parametrelerini güncellerken büyük hesaplama yüküne neden olabilirken, SGD, her güncelleme adımında yalnızca bir örnekten gradyanı hesaplar ve parametreleri hızlı bir şekilde güncelleyerek daha hızlı bir eğitim süreci sağlar. SGD, büyük veri kümesi üzerinde hızlı ve verimli bir şekilde çalışan bir optimizasyon algoritmasıdır. Ancak, varyasyonlar, yerel minimumlara sıkışma riski

gibi dezavantajları da dikkate alınmalı ve kullanılacak veri kümesi ve problem alanına göre dikkatlice seçilmelidir [41,42].

3.2.9. Gradient-Boosted Decision Trees (GBDT)

GBDT makine öğreniminde Gradient Boosting Machine (GBM) temelli, tahmin sonuçlarını öğrenme sürecinde ardışık adımlarla optimize etmek için kullanılan bir tekniktir. Her iterasyonda, karar ağaçları, kayıp fonksiyonunu en aza indirmeyi hedefleyerek katsayıların ve çarpımların değerlerini ayarlar. Gradyan, sürecin her adımında yapılan artımlı ayarı temsil eder ve hedefi basitleştirerek ve yeterince optimal bir çözüme ulaşmak için gereken iterasyon sayısını azaltarak öğrenme sürecini iyileştirir [43].

3.2.10. AdaBoost

Adaptive Boosting (AdaBoost veya kısaca Ada) Yoav Freund ve Robert Schapire tarafından formüle edilen bir istatistiksel makine öğrenimi sınıflandırma algoritmasıdır. Zayıf öğrenicilerin (karar ağaçları gibi) tahminlerini birleştiren bir topluluk tabanlı makine öğrenimi algoritmasıdır. AdaBoost, zayıf öğrenicileri art arda ekleyerek ve daha önceki sınıflandırıcılar tarafından yanlış sınıflandırılan örnekleri lehine ayarlayarak uyarlanabilir bir şekilde çalışır. Bazı problemlerde diğer öğrenme algoritmalarına göre aşırı öğrenmeyi daha az etkileyebilir [44,45].

3.2.11. LightGBM

Light Gradient Boosting Machine (LightGBM) aynı zamanda sıralama, sınıflandırma ve birçok diğer makine öğrenimi görevi için kullanılan hızlı, dağıtık, yüksek performanslı bir gradyan arttırılmış karar ağacı çerçevesidir. LightGBM, XGBoost'un seyrek optimizasyon, paralel eğitim, çoklu kayıp fonksiyonları, düzenleme, torbalama ve erken durdurma gibi birçok avantajını içerir. LightGBM ve XGBoost arasındaki önemli fark ise, ağaçların inşa sürecidir. LightGBM, ağaçları sıra sıra büyütmez, aksine ağaçları yaprak bazında büyütür ve kayıp fonksiyonunda en büyük azalmayı sağlayacağına inandığı yaprağı seçer. Ayrıca, LightGBM, yaygın

olarak kullanılan sıralı tabanlı karar ağacı öğrenme algoritmasını kullanmaz, bunun yerine yüksek derecede optimize edilmiş bir histogram tabanlı karar ağacı öğrenme algoritması uygular. Böylece verimlilik ve bellek tüketimi açısından büyük avantajlar sağlar [46].

3.2.12. XGBoost

Extreme Gradient Boosting (XGBoost veya kısaca XGB), hız ve performans için optimize edilmiş bir gradyan arttırılmış karar ağacı algoritmasıdır. Paralelleştirilmiş uygulama kullanarak ardışık ağaçlar inşa eder ve 'max_depth' parametresini belirtilen şekilde kullanarak ağaçları geriye doğru budar, bu da hesaplama performansını önemli ölçüde artırır. XGBoost, regresyon, sınıflandırma, sıralama ve kullanıcı tanımlı tahmin problemlerini çözmek için kullanılabilir [47].

3.3. ÖZELLİK ÖNEM YAKLAŞIMLARI

Açıklanabilir yapay zekâ, makine öğrenimi modellerinin kullanıcılarının veya geliştiricilerinin modellerin neden bu şekilde davrandıklarını anlamalarına yardımcı olan, gelişmekte olan bir araştırma yönüdür. En popüler açıklama tekniği özellik önemidir [48]. Özelliğin önemi, hangi girdi özelliklerinin önemli olduğunu ve bunların sonuçları tahmin etmede ne kadar yararlı olduğunu açıklar. Özellik önemini ölçmek için birçok farklı yöntem vardır. Çalışmada kullanılan iki özellik önem yöntemi ilerleyen başlıklarda açıklanmıştır.

3.3.1. Mean Decrease in Impurity (MDI)

MDI, Karışıklıktaki Ortalama Azalmayı (Mean Decrease in Impurity) ifade eder, özellik önemleri, her ağaçtaki kirlilik azalmasının ortalama ve standart sapması olarak hesaplanır. Bu yöntem, ağaç tabanlı bir modeli ve topluluklarını açıklamanın en basit ve popüler yoludur. Ancak, yüksek kardinalite özelliklerinden de kolayca etkilenebilir [49].

3.3.2. Permütasyon Tabanlı Özellik Önem Yöntemi

Permütasyon tabanlı özellik önem yöntemi, tek bir özellik değeri rastgele karıştırıldığında model puanındaki düşüş olarak tanımlanır. Bu teknik, bir özelliğe izin verirseniz veya karıştırırsanız performanstaki farkı ölçer. Ana fikir, bir özelliğin karıştırılması durumunda model performansının düşmesi, o özelliğin önemli olduğunu ortaya koyar. Permütasyona dayalı önemler, ortalama kirlilik azalmasıyla hesaplanan varsayılan özellik öneminin dezavantajlarının üstesinden gelmek için kullanılabilir. Yüksek kardinalite özelliklerine karşı bir önyargıları yoktur.

Ağaçlar için MDI güçlü bir şekilde önyargılıdır ve ikili özellikler veya az sayıda olası kategoriye sahip kategorik değişkenler gibi düşük kardinalite özelliklerine göre yüksek kardinalite özelliklerini (tipik olarak sayısal özellikler) tercih eder.

Permütasyon tabanlı özellik önem yöntemi böyle bir önyargı sergilemez. Ek olarak, permütasyon tabanlı özellik önem yöntemi sonuçları, model tahminlerinde hesaplanan performans metriği olabilir ve herhangi bir model sınıfını analiz etmek için kullanılabilir [50].

3.4. İSTATİSTİKTE ALTIN ORAN (GRIS)

Gunver vd. ortalama ve sapma hesabı için yeni bir yaklaşım sunmuşlardır. Önerilen yeni yaklaşıma göre, ortancaya yakın olan bileşenlerin ortalamaya daha yüksek, daha uzak olanların ise daha az katkı yapması gerekir. Bunu sağlamak için, ortancada φ ve uçlarda 1/φ alan doğrusal bir "GRIS ortalama katsayı maskesi" formüle edilmiştir. 'O', GRIS ortalamasının sembolü olarak önerilmektedir.

Amaç, standart bir katsayı dizisine bağlı olarak her bir elemanın ortalamaya katkısını değerlendirmektir. Veri yığınındaki her elemanı kendi katsayısıyla eşleştirirken, önce veri yığını artan düzende sıralanır. Dolayısıyla, X_1 en küçük veri yığınını, X_n ise en büyük veri yığınını temsil eder. Her elemanın katsayılarını hesaplamak için veri yığını artan düzende kullanılır.

Çizelge 3.3. GRIS ortalamasının hesaplanması [51].

İndeks(i)	**Veriler(Küçükten büyüğe doğru)**	**Ağırlıklandırma Katsayıları M_c**	**Verinin Medyandan Ağırlıklandırılmış Farkı**
1	X_1	DİNAMİK ORTALAMA KATSAYI MASKESİ	$M_{C1}*(X_1\text{-Med})$
2	X_2		$M_{C2}*(X_2\text{-Med})$
3	X_3		$M_{C3}*(X_3\text{-Med})$
…	…		…
n-2	X_{n-2}		$M_{C(n-2}*(X_{(n-2)}\text{-Med})$
n-1	X_{n--1}		$M_{C(n-1)}*(X_{(n-1)}\text{-Med})$
n	X_n		$M_{Cn}*(X_n\text{-Med})$

Çizelge 3.3'te GRIS ortalama hesabında her bir veri için hesaplanması gereken değerler görülmektedir. Ağırlıklandırma katsayıları Eşitlik 3.1'deki formül ile hesaplanır.

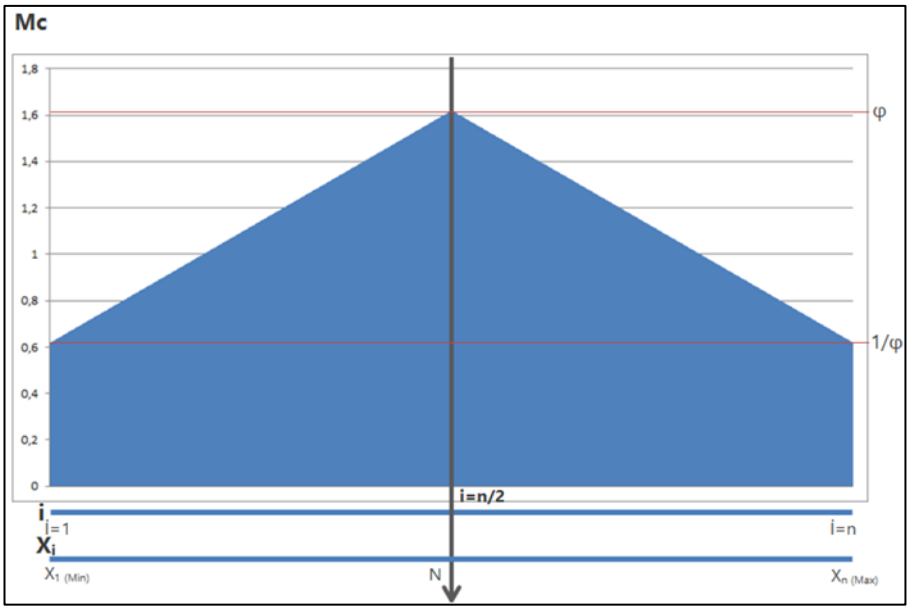
$$M_{C_i} = \begin{cases} if\ x_i < med\ \Rightarrow \dfrac{1}{\varphi} + 2 * (i-1)/(n-1) \\ else\ \Rightarrow\ 1 + \varphi - 2 * (i-1)/(n-1) \end{cases} \tag{3.1}$$

Ortalama katsayı maskesi dinamiktir, çünkü her eleman için katsayı n'ye (örnek boyutu) bağlı olarak değişir. Şekil 3.3'te de görülebileceği üzere medyana göre simetriktir. Sonraki adım, her bir elemanın medyana olan ağırlıklı mesafesini hesaplamaktır ve Eşitlik 3.2'daki formül ile hesaplanır.

$$verinin\ medyandan\ ağırlıklandırılmış\ farkı = M_{C_i} * (X_i - Med) \tag{3.2}$$

GRIS ortalamasının hesaplanması için gereken son adım, Eşitlik 3.3'teki formül ile medyandan sapmayı hesaplamaktır.

$$medyandan\ sapma = \frac{\sum M_{C_i} * (X_i - Med)}{\sum M_{C_i}} \tag{3.3}$$



Şekil 3.3. Dinamik ortalama katsayı maskesi [51].

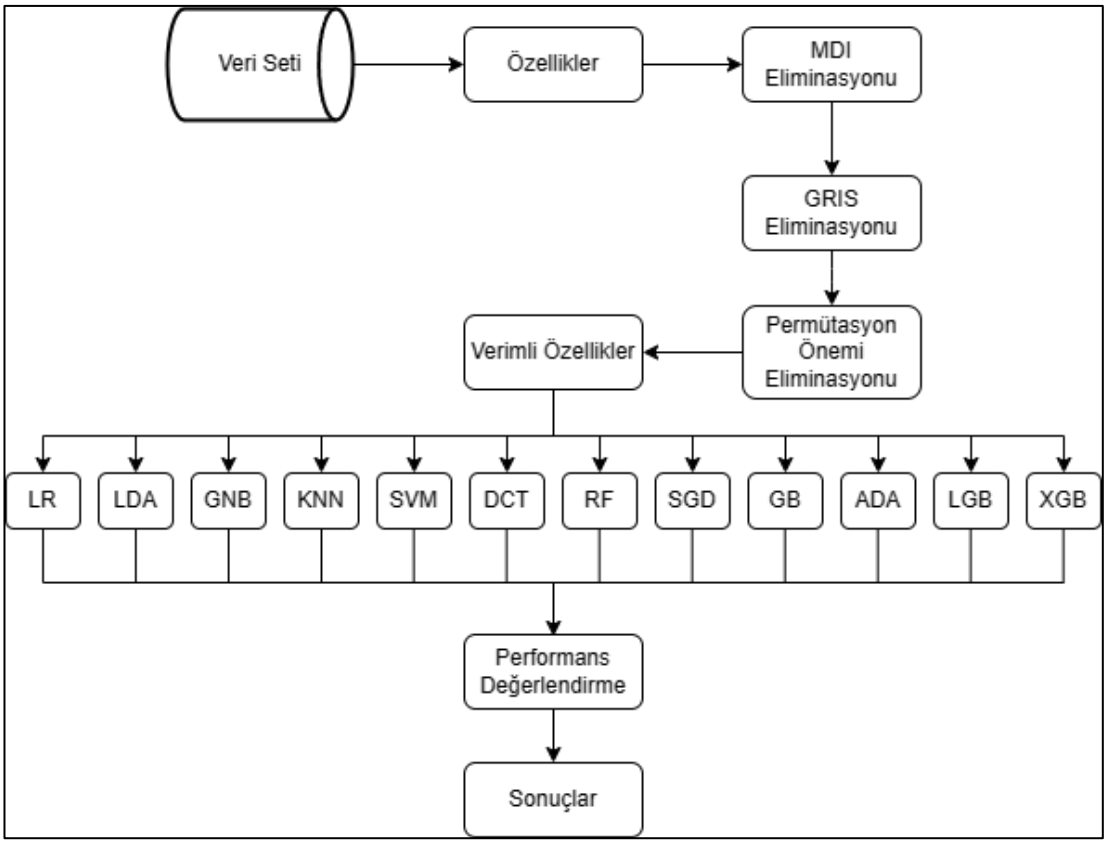
Eşitlik 3.4'teki hesaplama olan medyan ve medyandan sapmanın toplamı GRIS ortalamasını verir.

$$GRiS\ Ortalaması(\boldsymbol{O}) = medyan + medyandan\ sapma \tag{3.4}$$

GRIS ortalamasının (O) bağlantı noktası, uzaktaki elemanların katkısı azaldıkça, O'nun her zaman medyana yakın ve aritmetik ortalamadan daha yakın olmasını sağlar [51].

3.5. ÖNERİLEN YAKLAŞIM

Önerilen yaklaşım iki özellik önem yönteminin GRIS ortalaması ile birleştirilmesini içermektedir. Önerilen özellik seçimi yaklaşımının akış şeması Şekil 3.4'te gösterilmektedir.



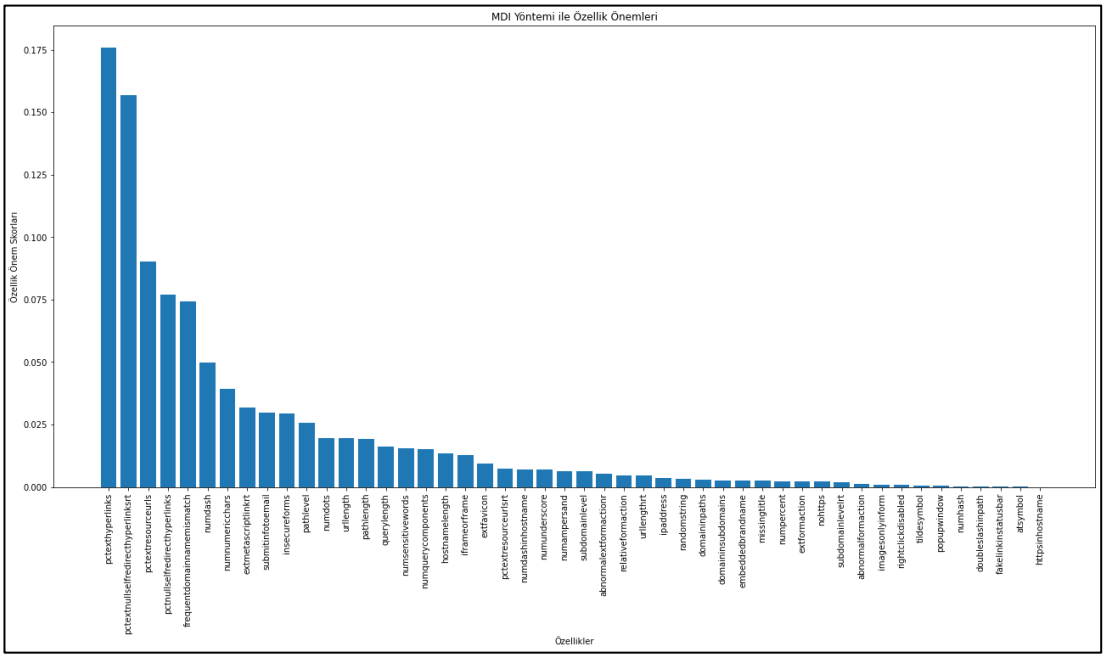
Şekil 3.4. Önerilen yaklaşımın akış şeması.

Önerilen yaklaşım üç aşamadan oluşmaktadır. İlk aşama, MDI eliminasyonudur. Bu aşamada özelliklerin özellik önem skorları MDI yöntemine göre hesaplanır ve özellik önem puanı sıfır olan özellikler elenir. İkinci aşama, GRIS eliminasyonudur. Bu aşamada bir önceki aşamada hesaplanan özellik önem skorları kullanılarak GRIS ortalaması hesaplanmaktadır. Özellik önem skoru bu ortalamanın altında olan özellikler elenir. Son aşama, Permütasyon Önemi eliminasyonudur. Bu aşamada kalan özelliklerin özellik önem skorları permütasyon önem yöntemine göre hesaplanır ve özellik önem puanı sıfır olan özellikler elenir. Böylece sona verimli özellikler kalmış olur. Ardından bu verimli özellikler makine öğrenmesi algoritmalarında (3.2'de bahsedilen) kullanılır ve performans değerlendirme aşamasından geçilerek sonuçlar elde edilir.

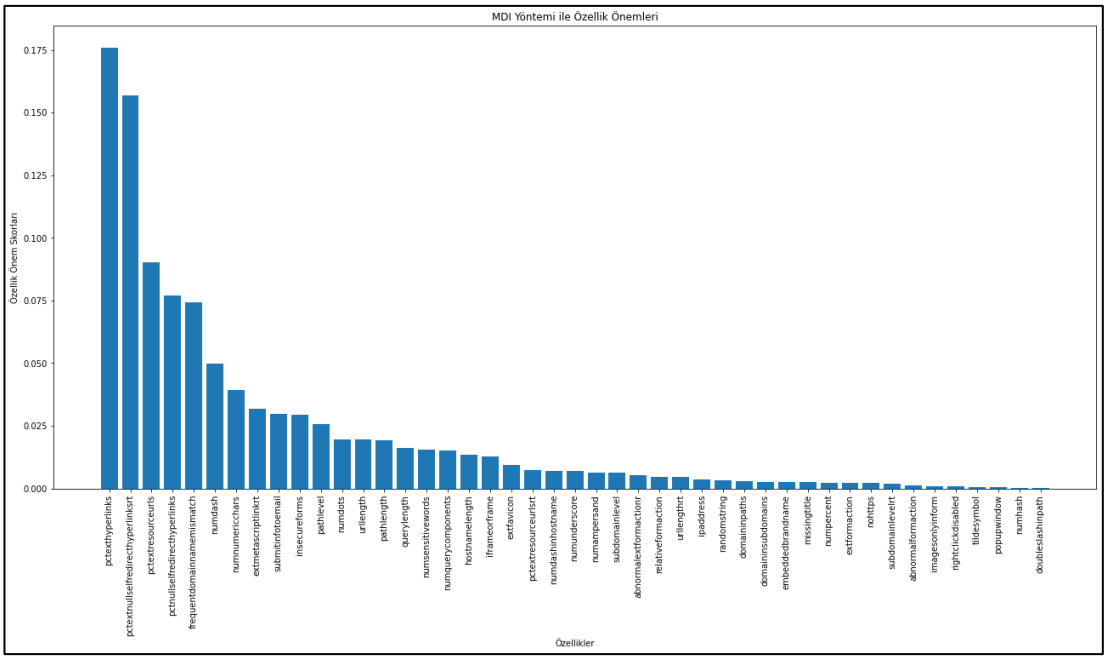
3.6. ÖNERİLEN YAKLAŞIMLA ÖZELLİKLERİN ELİMİNASYONU

Bu başlıkta önerilen yaklaşımdaki her bir eliminasyonda veri setlerinde kalan özellikler gösterilmiştir. Bütün özelliklerin bulunduğu durum ve her bir eliminasyondan sonraki durum özellik önem skorları ile verilmiştir.

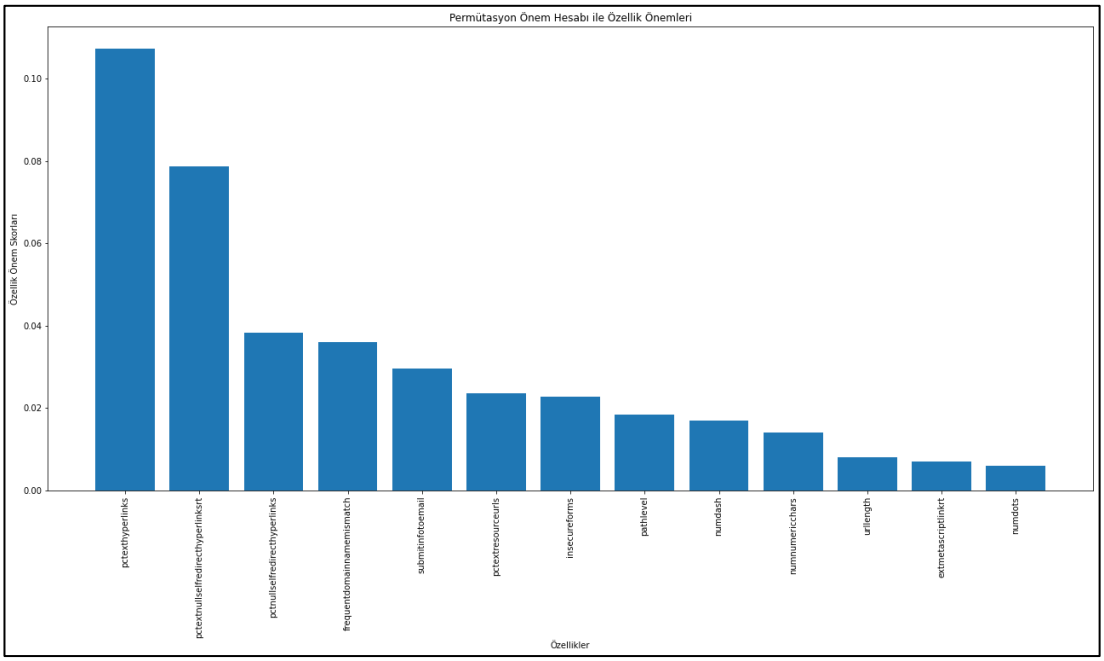
Şekil 3.5'te Mendeley 2018, Şekil 3.9'da Mendeley 2020 veri setinde bulunan bütün özelliklerin MDI yöntemi ile özellik önem skorları hesaplanmış ve büyükten küçüğe doğru sıralanmıştır. Şekil 3.6 ve Şekil 3.10'da önerilen yaklaşımdaki MDI eliminasyonundan sonra kalan özelliklerin MDI yöntemi ile özellik önem skorlarına göre büyükten küçüğe doğru sıralanmıştır. Şekil 3.7 ve Şekil 3.11'de önerilen yaklaşımdaki GRIS eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorları hesaplanmış ve büyükten küçüğe doğru sıralanmıştır. Şekil 3.8 ve Şekil 3.12'de önerilen yaklaşımdaki permütasyon önemi eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorlarına göre büyükten küçüğe doğru sıralanmıştır.



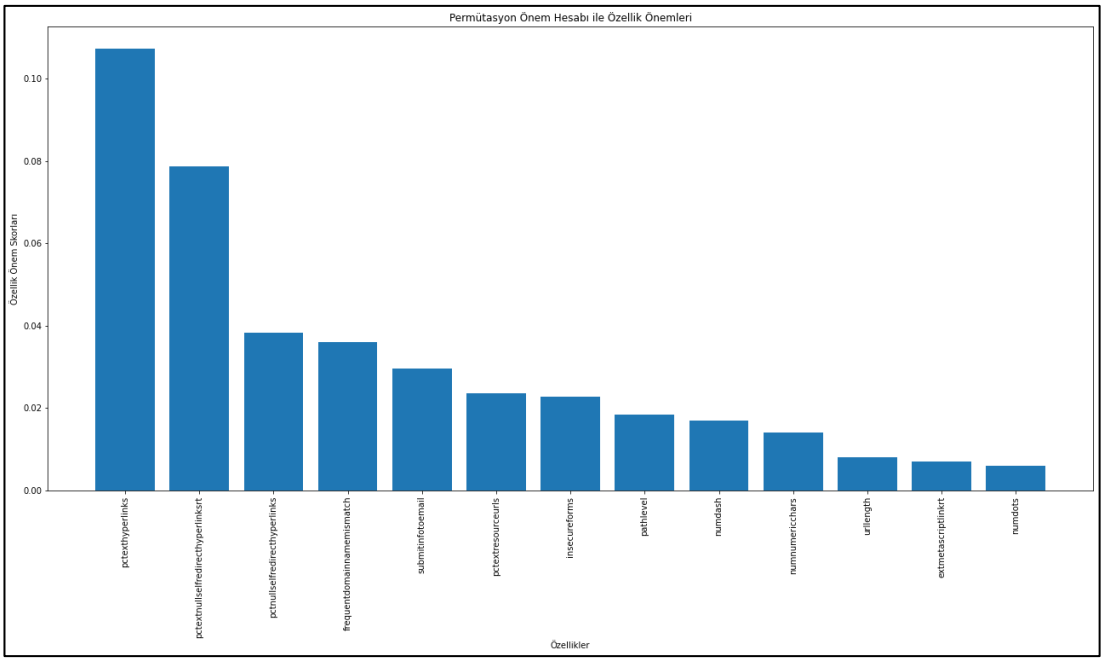
Şekil 3.5. Mendeley 2018 veri setinin bütün özelliklerinin MDI yöntemi ile özellik önem skorları.



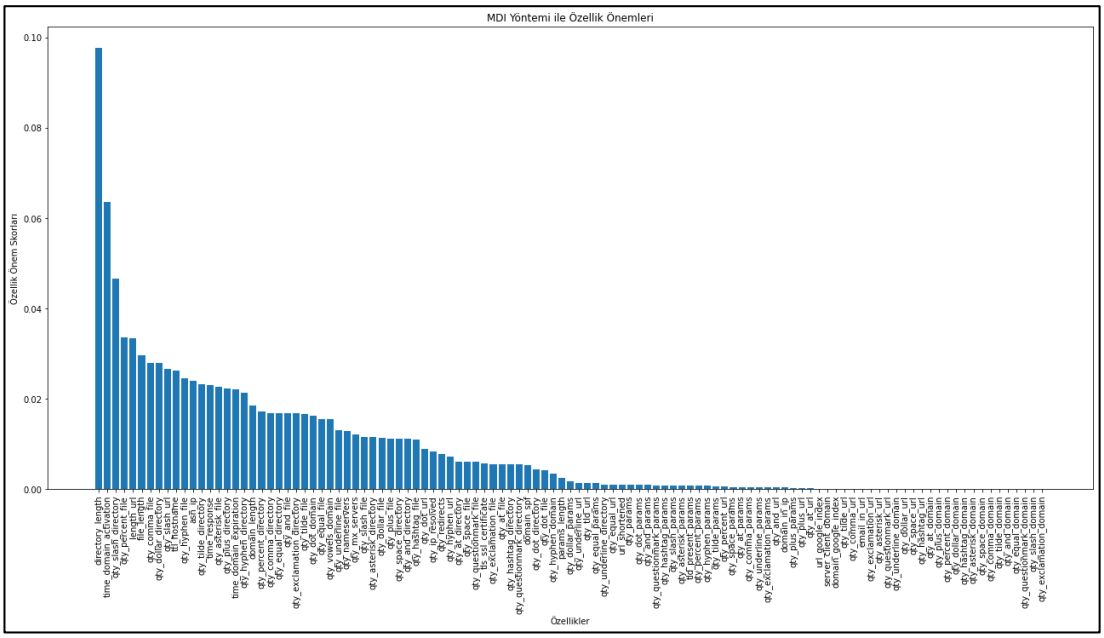
Şekil 3.6. Mendeley 2018 veri setinin MDI eliminasyonundan sonra kalan özelliklerin MDI yöntemi ile özellik önem skorları.



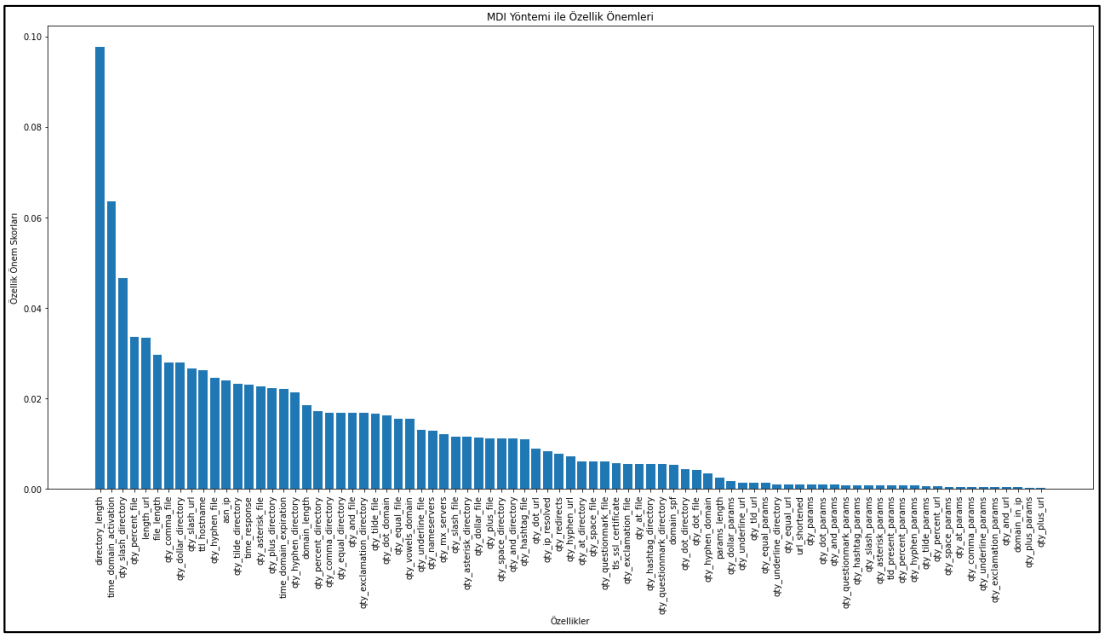
Şekil 3.7. Mendeley 2018 veri setinin GRIS eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorları.



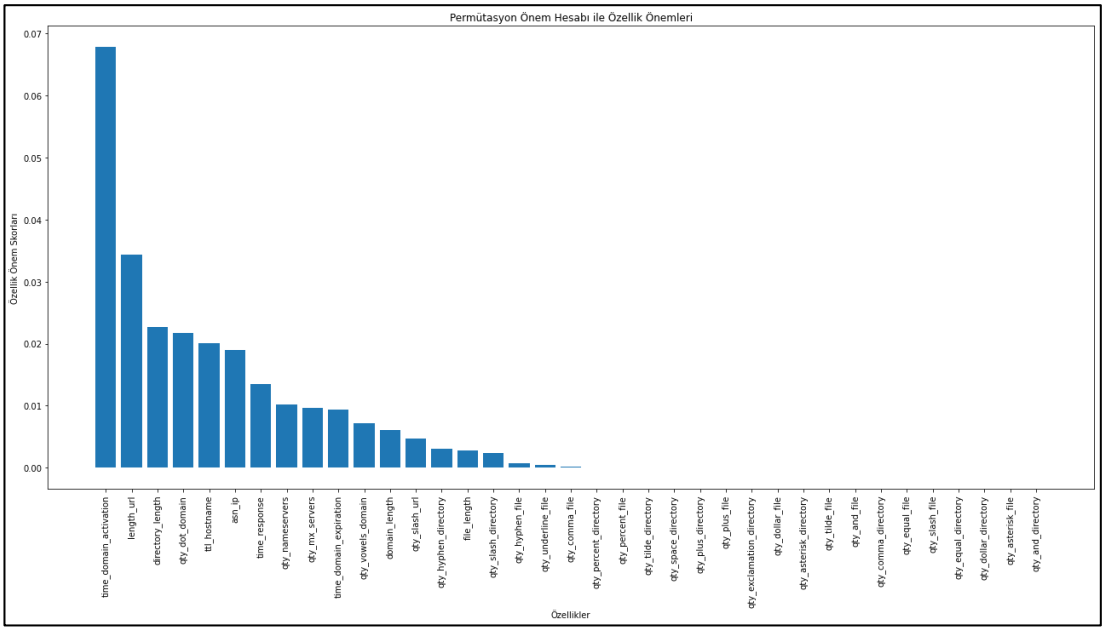
Şekil 3.8. Mendeley 2018 veri setinin permütasyon önemi eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorları.



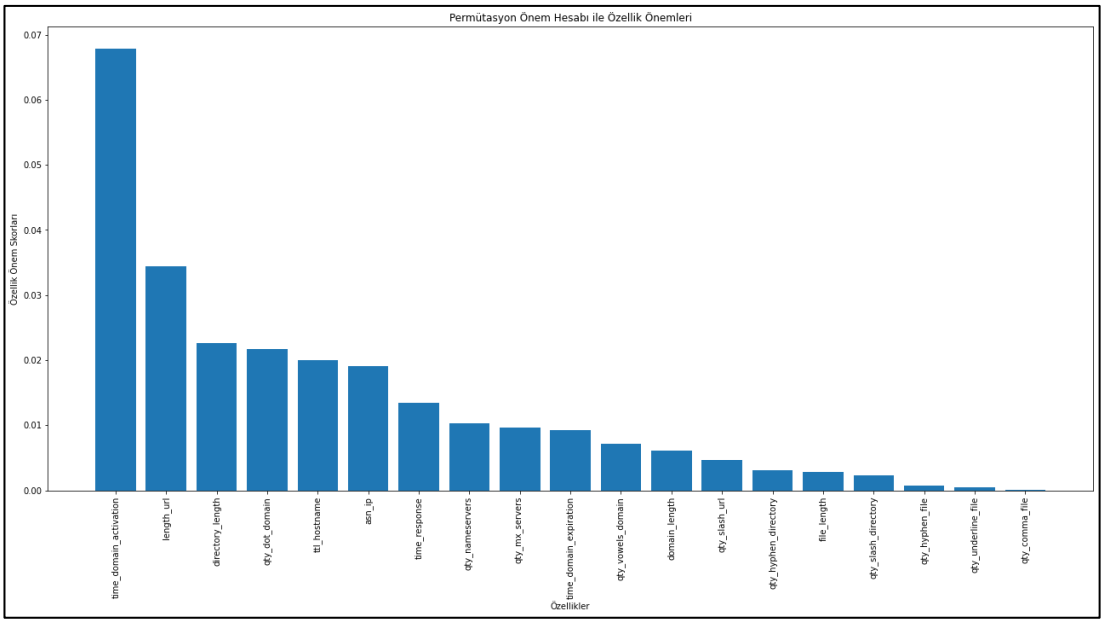
Şekil 3.9. Mendeley 2020 veri setinin bütün özelliklerinin MDI yöntemi ile özellik önem skorları.



Şekil 3.10. Mendeley 2020 veri setinin MDI eliminasyonundan sonra kalan özelliklerin MDI yöntemi ile özellik önem skorları.



Şekil 3.11. Mendeley 2020 veri setinin GRIS eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorları.



Şekil 3.12. Mendeley 2020 veri setinin permütasyon önemi eliminasyonundan sonra kalan özelliklerin permütasyon önem yöntemi ile özellik önem skorları.

3.7. PERFORMANS METRİKLERİ

Makine öğrenmesinin temellerinden biri de bir makine öğrenmesi modelinin performansını değerlendirmektir. Bu, parametrelerin iyileştirilmesine ve test edilecek en iyi ve en uygun modelin seçilmesine yardımcı olur. Ayrıca, uygun bir değerlendirme parametresi seçmek, modelin başarısını veya başarısızlığını verimli bir şekilde görselleştirmeye yardımcı olur. Bu nedenle geliştirilen modelleri ifade etmek için uygun performans değerlendirme parametrelerinin bulunması bir zorunluluktur.

Sınıflandırma eğitimi sırasında, birçok üretken sınıflandırıcı, en iyi çözümü belirlemek için doğruluğu bir ölçü olarak kullanır. Bununla birlikte, doğrulukta benzersizlik eksikliği, ayırt edilebilirlik, bilgilendiricilik ve çoğunluk sınıfından gelen verilere yönelik bir önyargı dahil olmak üzere birkaç kusur vardır [52]. Bu çalışmada doğruluk, kesinlik, duyarlılık ve f1 skoru gibi tahmin doğruluğunu esas alan metrikler ve eğitim süresi, test süresi ve bellek kullanımı gibi sistem kullanımını esas alan metrikler kullanılmıştır.

"TN" terimi, Gerçek Negatif anlamına gelir ve doğru sınıflandırılmış negatif örneklerin sayısını temsil eder. "TP" terimi, Gerçek Pozitif anlamına gelir ve doğru şekilde sınıflandırılmış pozitif örneklerin sayısını belirtir. "FN" terimi, negatif olarak sınıflandırılan gerçek pozitif örneklerin sayısını ifade ederken, "FP", pozitif olarak sınıflandırılan gerçek negatif örneklerin sayısını ifade eder [53].

3.7.1. Doğruluk

Doğruluğun en büyük avantajı, daha az karmaşıklıkla hesaplama kolaylığı sağlaması ve aynı zamanda insanlar tarafından anlaşılmasını kolaylaştırırken çok sınıflı ve çok etiketli sınıflandırma problemlerine uygun olmasıdır. Eşitlik 3.5 doğruluğun nasıl hesaplanabileceğini göstermektedir.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.5}$$

3.7.2. Kesinlik

Kesinlik, pozitif bir sınıfın toplam tahminlerinden uygun şekilde tahmin edilen pozitif modellerin bir ölçüsü olarak tanımlanabilir. Kesinliği hesaplamak için Eşitlik 3.6 kullanılabilir.

$$Kesinlik(k) = \frac{TP}{TP + FP} \tag{3.6}$$

3.7.3. Duyarlılık

Duyarlılık, tipik olarak, doğru şekilde sınıflandırılan pozitif modellerin fraksiyonunu hesaplamak için kullanılır. Eşitlik 3.7, duyarlılığı belirlemek için kullanılabilir.

$$Duyarlılık(d) = \frac{TP}{TP + FN} \tag{3.7}$$

3.7.4. F1 Skoru

F1 Skoru, duyarlılık ve kesinlik arasındaki harmonik ortalamayı gösteren bir ölçümdür. Eşitlik 3.8, F1 skorunun formülünü göstermektedir.

$$F1 - Skoru = \frac{2 * k * d}{k + d} \tag{3.8}$$

Mevcut çok sayıda performans değerlendirme parametresi olmasına rağmen, önerilen yaklaşımın etkinliğini belirlemek için bu çalışmada sadece yukarıda bahsedilen parametreler kullanılmıştır. Modellerin performansı bu değerlendirme parametrelerine dayalı olarak tahmin edilebilir.

BÖLÜM 4

SONUÇLAR

Bu çalışmada 2 farklı veri setinin özelliklerinden, önerilen yaklaşım ile elde edilen verimli özellikler ve veri setinde verilen bütün özellikler 12 farklı algoritma ile test edilmiştir. Bütün özelliklerin kullanıldığı senaryonun sonuçları ile verimli özelliklerin kullanıldığı senaryonun sonuçları, 12 farklı algoritmanın sonuçları ve yöntemin etkinliği veri seti ve algoritmalar bazında karşılaştırılmıştır.

Testler Google Colab üzerinden Python dili kullanılarak yapılmıştır. Testlerin gerçekleştirildiği Google Colab sunucusu; Intel(R) Xeon(R) CPU 2.30GHz, 12 GB RAM ve Tesla T4 GPU sistem özelliklerine sahiptir. Algoritmalar için bazı Python kütüphaneleri kullanılmıştır. Her bir algoritma için çağırılan kütüphane, kütüphanenin kullanılan versiyonu, eklenen parametreler Çizelge Ek A.1'de verilmiştir. Eklenen parametreler kısmı boş olan algoritmalar için o versiyondaki standart parametrelerin kullanıldığı anlamına gelmektedir.

Bu bölümde veri setine göre algoritmalardan elde edilen sonuçlar karşılaştırılmıştır. Farklı performans metriklerinde hangi algoritmanın daha iyi olduğunun tespit edilmesi amaçlanmıştır. Çizelge 4.1'de Mendeley 2018 veri seti, Çizelge 4.2'de Mendeley 2020 veri seti için 12 algoritmanın 100 bağımsız teste göre ortalama sonuçları verilmiştir. "BTN "sütunları bütün özellikler kullanıldığında alınan sonuçları gösterirken "VER" sütunları önerilen yaklaşımla elde edilen verimli özelikler kullanıldığında alınan sonuçları göstermektedir. O sütundaki en iyi sonuç kalın fontla belirtilmiştir.

Her algoritma için ayrıntılı tablolar Çizelge Ek B.1 ile Çizelge Ek B.24 arasındaki çizelgelerde verilmiştir. Tabloların içeriğine bakıldığında 4 sütun bulunmaktadır. İlk iki sütunda bütün özellikler ile verimli özellikler kullanılarak elde edilen sonuçlar yer alırken, üçüncü sütunda numerik fark ve dördüncü sütunda da yüzdesel fark yer almaktadır. Numerik fark 2. sütundaki verinin 1. sütundaki veriden farkını vermekte, yüzdesel fark ise 1. sütundaki veriden 2. sütundaki veriye olan düşüş yüzdesini vermektedir. Böylece yüzdesel fark sütunundaki verilere bakılarak kayıp veya kazancın yüzde olarak ne kadar azalıp veya arttığı gözlemlenebilir. Bölüm 3.7'de bahsedilen performans metriklerinin en iyi, ortalama ve en kötü sonuçları verilmişken, eğitim süresi, test süresi ve bellek kullanımının sadece ortalama sonuçları verilmiştir. Bunun sebebi eğitim ve test süresinde en iyi ve en kötü sonuçların yanıltıcı olabilme ihtimalinden kaynaklanmaktadır. Son 3 satırdaki verilerdeki düşüşler negatif işaretli ise bu kazanç anlamına gelmektedir. Bu satırlar süre ve bellek kullanımını gösterdikleri için buradaki düşüş performans metriklerindeki düşüşün aksine olumludur. Buradaki farkın ayırt edilebilmesi için son üç satır çift çizgi ile ayrılmıştır.

Çizelge 4.1. Mendeley 2018 veri setinde algoritmaların sonuçlarının karşılaştırılması.

Veri Seti	Mendeley 2018							
Metrik	Doğruluk		Kesinlik		Duyarlılık		F1-Skoru	
Özellikler	BTN	VER	BTN	VER	BTN	VER	BTN	VER
LR	0,9322	0,9073	0,9250	0,8990	0,9408	0,9177	0,9328	0,9082
LDA	0,9360	0,9127	0,9249	0,9070	0,9490	0,9197	0,9368	0,9133
GNB	0,8443	0,8266	0,9307	0,9167	0,7441	0,7187	0,8269	0,8056
KNN	0,8688	0,8901	0,8465	0,8814	0,9010	0,9016	0,8729	0,8914
SVM	0,9362	0,8986	0,9206	0,88302	0,9600	0,9460	0,9532	0,9322
DCT	0,9653	0,9602	0,9620	0,9569	0,9690	0,9638	0,9655	0,9603
RF	0,9823	0,9759	0,9825	0,9742	0,9821	0,9777	0,9823	0,9759
SGD	0,9380	0,9080	0,9353	0,8987	0,9412	0,9203	0,9382	0,9091
GB	0,9770	0,9705	0,9753	0,9689	0,9789	0,9724	0,9771	0,9706
ADA	0,9711	0,9650	0,9694	0,9606	0,9729	0,9699	0,9712	0,9652
LGB	**0,9850**	**0,9771**	**0,9844**	**0,9758**	**0,9856**	**0,9786**	**0,9850**	**0,9772**
XGB	0,9764	0,9700	0,9738	0,9682	0,9791	0,9720	0,9765	0,9701

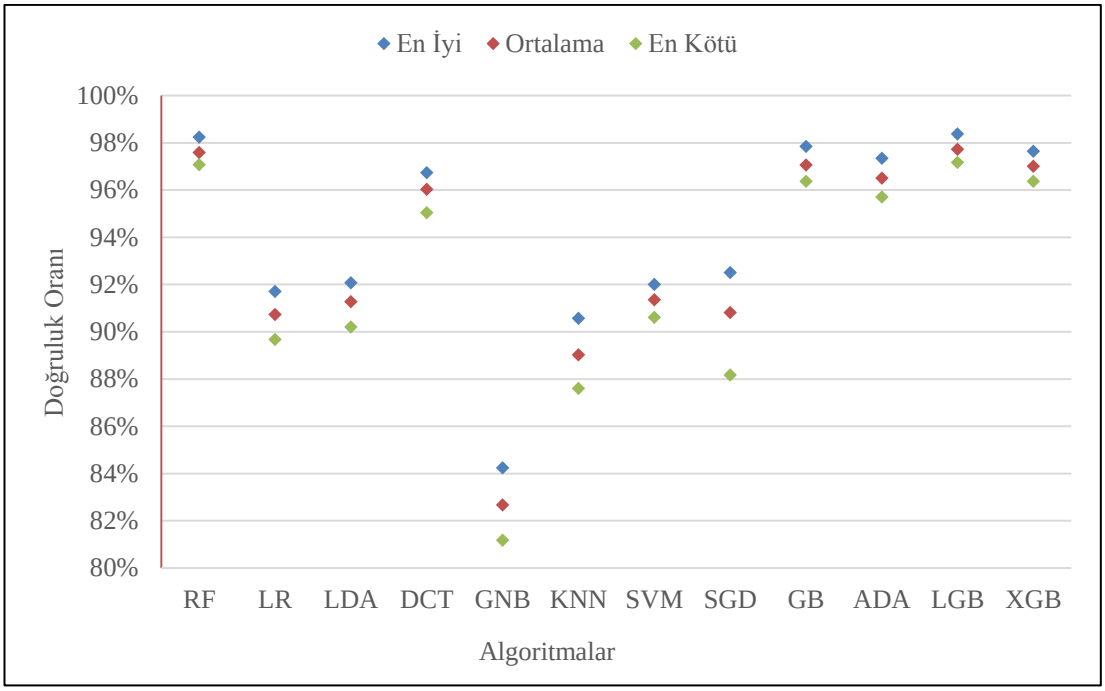
Çizelge 4.2. Mendeley 2020 veri setinde algoritmaların sonuçlarının karşılaştırılması.

Veri Seti	Mendeley 2020							
Metrik	Doğruluk		Kesinlik		Duyarlılık		F1-Skoru	
Özellikler	BTN	VER	BTN	VER	BTN	VER	BTN	VER
LR	0,8972	0,8846	0,8815	0,8918	0,8125	0,7583	0,8445	0,8196
LDA	0,9152	0,9020	0,8276	0,8025	0,9532	0,9503	0,8859	0,8702
GNB	0,8428	0,8684	0,8831	0,8825	0,6285	0,7146	0,7343	0,7897
KNN	0,8796	0,8787	0,8311	0,8279	0,8180	0,8195	0,8245	0,8237
SVM	0,8909	0,8605	0,8832	0,8541	0,9254	0,922	0,9181	0,9165
DCT	0,9528	0,9513	0,9320	0,9301	0,9314	0,9290	0,9317	0,9295
RF	**0,9700**	**0,9689**	**0,9547**	**0,9518**	**0,9587**	**0,9584**	**0,9567**	**0,9551**
SGD	0,9285	0,9195	0,8905	0,8638	0,9058	0,9114	0,8975	0,8867
GB	0,9538	0,9510	0,9316	0,9272	0,9350	0,9312	0,9333	0,9292
ADA	0,9360	0,9337	0,9084	0,9052	0,9064	0,9027	0,9074	0,9040
LGB	0,9661	0,9642	0,9502	0,9465	0,9519	0,9501	0,9510	0,9483
XGB	0,9530	0,9505	0,9273	0,9236	0,9375	0,9341	0,9324	0,9288

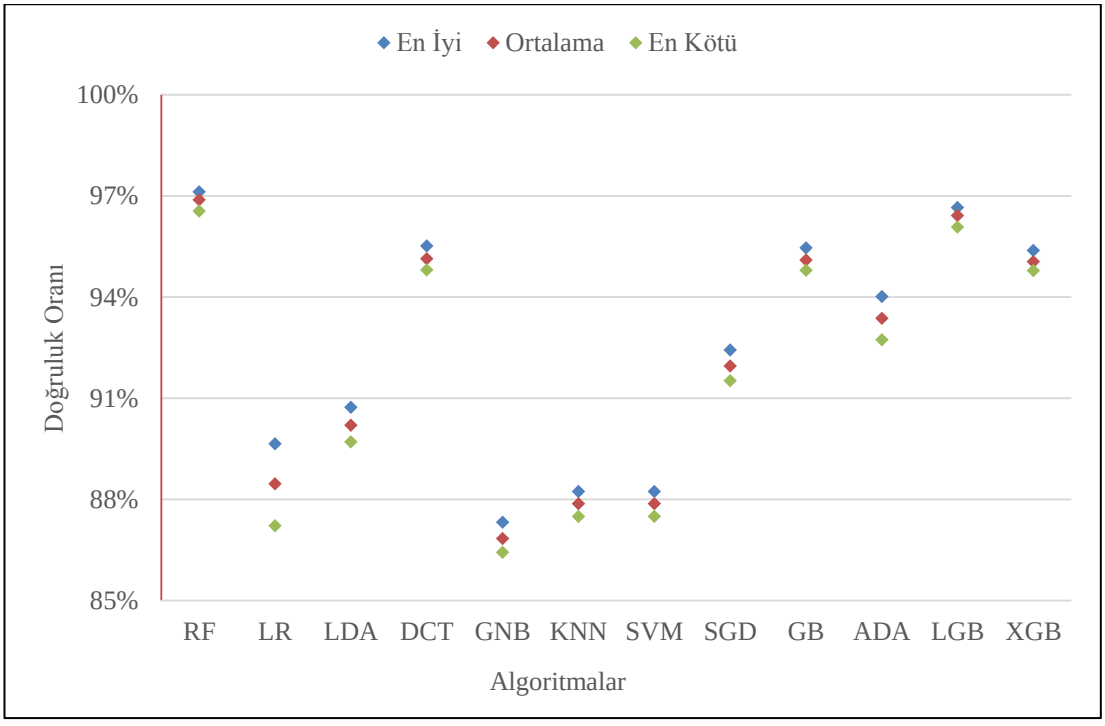
Her bir algoritma için verimli özellikler kullanılarak elde edilen sonuçlardan en iyi, ortalama ve en kötü doğruluk değerleri Mendeley 2018 veri seti için Şekil 4.1'de, Mendeley 2020 veri seti için Şekil 4.2'de verilmiştir.

Bütün özellikler kullanılarak elde edilen sonuçlardan en iyi ortalama ve en kötü doğruluk değerleri Mendeley 2018 veri seti için Şekil Ek B.1'de, Mendeley 2020 veri seti için Şekil Ek B.2'de verilmiştir.

Kesinlik, duyarlılık ve f1-skoru için hazırlanan karşılaştırma grafikleri Şekil Ek B.3 ile Şekil Ek B.14 arasındaki şekillerde verilmiştir.



Şekil 4.1. Mendeley 2018 veri setinde verimli özellikler kullanılarak algoritmaların doğruluk oranlarının karşılaştırılması.



Şekil 4.2. Mendeley 2020 veri setinde verimli özellikler kullanılarak algoritmaların doğruluk oranlarının karşılaştırılması.

Her bir algoritma için verimli özellikler kullanılarak elde edilen sonuçlardan ortalama eğitim süresi, test süresi ve toplam süre Mendeley 2018 veri seti için Çizelge 4.3'te, Mendeley 2020 veri seti için Çizelge 4.4'te verilmiştir. Tablolardaki değerler saniye cinsinden yazılmış olup toplam süreye göre küçükten büyüğe doğru sıralanmıştır.

Bütün özellikler kullanılarak elde edilen sonuçlardan ortalama eğitim süresi, test süresi ve toplam süre Mendeley 2018 veri seti için Çizelge Ek B.25'te, Mendeley 2020 veri seti için Çizelge Ek B.26'da verilmiştir.

Ranking yönteminde seçilen metriğe göre algoritmaların sonuçları sıralanır ve en iyi sonuca sahip olan en yüksek puanı en kötü sonuca sahip olan en düşük puanı alır. Bu çalışmada 12 algoritma kullanıldığı için en iyi sonuca sahip olan 12 puan, en kötü sonuca sahip olan 1 puan almıştır. Doğruluk, kesinlik, duyarlılık, f1-skoru ve zaman metriklerinde bu puanlamalar yapılmış ve her metriğin ağırlığı eşittir. Mendeley 2018 veri seti için ranking tablosu Çizelge 4.5'te, Mendeley 2020 veri seti için ranking tablosu Çizelge 4.6'da verilmiştir. İki veri setindeki sonuçlar Çizelge 4.7'de toplanmış ve en iyiden en kötüye doğru sıralanmıştır.

Çizelge 4.3. Mendeley 2018 veri setinde verimli özellikler kullanılarak algoritmaların eğitim ve test sürelerinin karşılaştırılması.

Algoritma	Eğitim Süresi (sn)	Test Süresi (sn)	Toplam (sn)
GNB	0,00326	0,00165	0,00491
LDA	0,02298	0,00254	0,02552
DCT	0,02613	0,00182	0,02795
SGD	0,02873	0,00259	0,03132
LR	0,10469	0,00202	0,10671
KNN	0,01371	0,11951	0,13323
LGB	0,14476	0,01207	0,15683
ADA	0,26408	0,02621	0,29030
XGB	0,31900	0,00736	0,32636
RF	0,43101	0,10343	0,53444
SVM	0,55199	0,00751	0,55950
GB	0,82245	0,00681	0,82926

Çizelge 4.4. Mendeley 2020 veri setinde verimli özellikler kullanılarak algoritmaların eğitim ve test sürelerinin karşılaştırılması.

Algoritma	Eğitim Süresi (sn)	Test Süresi (sn)	Toplam (sn)
GNB	0,03113	0,01046	0,04159
LDA	0,19758	0,00587	0,20345
SGD	0,37621	0,00827	0,38448
DCT	0,64045	0,00825	0,64871
LGB	1,25205	0,12048	1,37252
LR	3,15097	0,00570	3,15666
ADA	3,86544	0,26717	4,13261
XGB	4,47800	0,08840	4,56640
RF	7,15149	0,41423	7,56571
KNN	0,01209	10,37996	10,39205
GB	14,92956	0,05330	14,98286
SVM	15,45482	0,00758	15,46240

Çizelge 4.5. Mendeley 2018 veri setinde verimli özellikler kullanıldığındaki sonuçlar için ranking tablosu.

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	Zaman	Toplam
LGB	12	12	12	12	6	54
RF	11	11	11	11	3	47
GB	10	10	10	10	1	41
XGB	9	9	9	9	4	40
DCT	7	7	7	7	10	38
ADA	8	8	8	8	5	37
LDA	6	5	4	5	11	31
SGD	5	3	5	4	9	26
LR	4	4	3	3	8	22
GNB	1	6	1	1	12	21
SVM	3	2	6	6	2	19
KNN	2	1	2	2	7	14

Çizelge 4.6. Mendeley 2020 veri setinde verimli özellikler kullanıldığındaki sonuçlar için ranking tablosu.

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	Zaman	Toplam
RF	12	12	12	12	4	52
LGB	11	11	10	11	8	51
DCT	10	10	7	10	9	46
XGB	8	8	9	8	5	38
GB	9	9	8	9	2	37
LDA	5	1	11	4	11	32
ADA	7	7	4	6	6	30
SGD	6	4	5	5	10	30
LR	4	6	2	2	7	21
GNB	2	5	1	1	12	21
SVM	1	3	6	7	1	18
KNN	3	2	3	3	3	14

Çizelge 4.7. İki ranking tablosunun toplamı.

Algoritma	Mendeley 2018	Mendeley 2020	Toplam
LGB	54	51	105
RF	47	52	99
DCT	38	46	84
GB	41	37	78
XGB	40	38	78
ADA	37	30	67
LDA	31	32	63
SGD	26	30	56
LR	22	21	43
GNB	21	21	42
SVM	19	18	37
KNN	14	14	28

BÖLÜM 5

TARTIŞMA

Bu çalışmada, 2 popüler özellik önem yöntemi ve GRIS ortalama hesabı ile yeni bir yaklaşım önerilmiştir. 2 farklı oltalama veri seti kullanılarak önerilen yaklaşımın etkinliği test edilmiştir. Bölüm 4'te bahsedilen test sonuçları, oltalama tespitinde genel olarak topluluk tabanlı makine öğrenmesi algoritmalarının (RF, SGD, ADA, GB, LGB, XGB) geleneksel makine öğrenmesi algoritmalarından (DCT, LR, LDA, GNB, SVM, KNN) daha iyi performans gösterdiğini göstermiştir. Seçilen verimli özellikler, yüksek algılama doğruluğunu büyük oranda korurken eğitim ve test süresi ile bellek kullanımını azalttığı görülmüştür. Mendeley 2018 veri seti için bellek kullanımı %72,95 düşerken, Mendeley 2020 veri seti için ise %82,88 düştüğü gözlemlenmiştir.

Mendeley 2018 veri setinde LGB algoritması ile önerilen yaklaşım kullanılarak, orijinal özellik sayısının yalnızca %27,08'i kullanılarak doğruluğun %98,37'ye ulaşabildiği görülmüştür. Çizelge 4.1 incelendiğinde performans metrikleri göz önüne alındığında Mendeley 2018 veri seti için en iyi 3 algoritma sırasıyla LGB, RF ve GB olduğu görülmüştür. Zaman göz önüne alındığında en iyi 3 algoritma sırasıyla GNB, LDA ve DCT olduğu görülmüştür. Toplama (Çizelge 4.5) bakıldığında en iyi 3 algoritma sırasıyla LGB, RF ve GB olduğu görülmüştür.

Mendeley 2020 veri setinde RF algoritması ile önerilen yaklaşım kullanılarak, orijinal özellik sayısının yalnızca %17,12'si kullanılarak doğruluğun %97,12'ye ulaşabildiği görülmüştür. Çizelge 4.2 incelendiğinde performans metrikleri göz önüne alındığında Mendeley 2020 veri seti için en iyi 3 algoritma sırasıyla RF, LGB ve DCT olduğu görülmüştür. Zaman göz önüne alındığında en iyi 3 algoritma sırasıyla GNB, LDA ve SGD olduğu görülmüştür. Toplama (Çizelge 4.6) bakıldığında en iyi 3 algoritma sırasıyla RF, LGB ve DCT olduğu görülmüştür.

Çizelge 4.7 incelendiğinde en iyi 3 algoritmanın sırasıyla LGB, RF ve DCT olduğu görülmüştür. Her iki veri setinde de yakın sonuçlar alan LGB ve RF algoritmalarında sonucu belirleyen zaman faktörü olmuştur. LGB algoritması, RF algoritmasından daha hızlı çalıştığı için, sıralama tablolarında puan farkını açmış ve öne geçmiştir. RF'nin daha basit bir versiyonu olan DCT algoritmasının genel toplam tablosunda üçüncülüğe gelmesindeki en önemli pay, iki veri setinde de zaman faktöründe öne çıkmış olmasıdır. Geleneksel makine öğrenmesi algoritmalarının zaman puanı ortalaması 7,75 iken topluluk tabanlı makine öğrenmesi algoritmalarının zaman puanı ortalaması 5,25'tir. Geleneksel makine öğrenmesi algoritmalarının performans metriklerinin puan ortalaması 4,25 iken topluluk tabanlı makine öğrenmesi algoritmalarının performans metriklerinin puan ortalaması 8,75'tir. Bu da topluluk tabanlı makine öğrenmesi algoritmalarının geleneksel makine öğrenmesi algoritmalarından zaman olarak geride olsa da performans metrikleri açısından oldukça önde olduğunu ortaya koymaktadır. Sıralama tablolarındaki ağırlıklar eşit verilmiştir. Bu ağırlıklar değiştirilerek önem verilen metriğe göre en iyi algoritma seçilebilir.

SVM ve KNN uzun sürelerde çalışmaları nedeniyle kullanılması tercih edilmemesi gereken algoritmalar olarak tespit edilmiştir. GNB, her iki veri setinde de en hızlı algoritma olmasına karşın performansı diğer algoritmaların gerisinde kalmıştır. Duyarlılık konusunda sıkıntılı olduğu görülmüştür. LDA ve LR; (LDA 2020 veri setinde kesinlik metriğinde, LR 2020 veri setinde duyarlılık metriğinde) geride kalmış olsa da diğer metrikler göz önüne alındığında istikrarlı algoritmalardır. DCT, geleneksel algoritmalar içerisinde genel olarak en iyi performansa sahip algoritmadır. Topluluk tabanlı algoritmalar verimli özellikler kullanıldığında beklendiği gibi doğruluk oranını korumuş ve eğitim ve test sürelerini kısaltmışlardır.

Yapılan çalışmada doğruluk metriği modelin oltalama ve meşru web sitelerinin ne kadarını doğru olarak sınıflandırdığını, kesinlik metriği model tarafından oltalama olarak tespit edilenlerin ne kadarının gerçekten oltalama web sitesi olduğunu, duyarlılık metriği modelin var olan oltalama web sitelerinin ne kadarını tespit ettiğini ve f1-skoru ise modelin oltalama tespitinde ne kadar etkili olduğunu bize verir.

Çizelge 5.1. Önerilen yaklaşımın Yi ve Sekiya'nın çalışması ile karşılaştırılması.

Özellik Sayısı	Test Süresi (sn)		Doğruluk		Duyarlılık		Bellek Kullanımı (MB)
	Colab	Yi, Sekiya	Colab	Yi, Sekiya	Colab	Yi, Sekiya	
10	0,3887	0,2884	0,9696	0,9693	0,9641	0,9614	7,4
11	0,3941	0,3017	0,9704	0,9696	0,9649	0,9615	8,1
14	0,4156	0,2787	0,9708	0,9703	0,9644	0,9622	10,1
19	0,4142	-	0,9712	-	0,9629	-	12,85
111	0,5398	-	0,9725	-	0,9640	-	75,07

Yi ve Sekiya'nın çalışmasında [11] Mendeley 2020 veri seti için en iyi sonucu 14 özellikle, en iyi ikinci sonucun 10 özellikle ve en iyi dördüncü sonucun 11 özellikle elde edildiğini öne sürmüşlerdir. Çizelge 5.1'deki metrikleri kullanarak formülünü paylaşmadıkları "anti-phishing" skoru ile karşılaştırmışlardır. En iyi üçüncü sonucu 26 özellik ile elde edildiğini belirttikleri halde bu 26 özelliğin hangi özellikler olduğunu paylaşmadıkları için, çizelgeye en iyi üçüncü yerine dördüncü dahil edilmiştir. Çizelge 5.1'de Yi ve Sekiya'nın çalışmasında açıkça belirtilen özellikler kullanılarak kendi test ortamımızda Random Forest algoritması ile 100 bağımsız test gerçekleştirerek aynı tablo oluşturulmuştur. Tabloda Yi ve Sekiya'nın hazırlamış olduğu tablodakilerle benzer olacak şekilde özellik sayılarına göre en iyi doğruluk ve en iyi duyarlılık oranı, ortalama test süresi ve bellek kullanımı verilmiştir. Yi ve Sekiya'nın almış oldukları orijinal sonuçlar koyu arka planla verilmiştir. Yi ve Sekiya çalışmalarında 19 ve 111 özelliği test etmedikleri için ilgili hücreler boş bırakılmıştır. Aynı özellik sayıları ile elde edilen sonuçlar karşılaştırıldığında Yi ve Sekiya'nın testlerinin daha hızlı gerçekleştiği fakat performans metrikleri açısından geride kaldığı görülmektedir. Hem Yi ve Sekiya'nın aldığı sonuçlara bakıldığında hem de bizim test ortamımızda aldığımız sonuçlara bakıldığında özellik sayısı arttıkça doğruluk artmış, test süresi ise düşmüştür. Yi ve Sekiya en iyi seçimin 14 özellik olduğunu çalışmalarında vurgulamışlardır. Bizim yaklaşımımızla seçtiğimiz 19 özellikle aldığımız sonuçları karşılaştırdığımızda doğruluk olarak bizim seçimimizin daha önde olduğu görülmüştür. Yi ve Sekiya'nın sunmuş olduğu 14 özellikle elde edilen sonuçlarda duyarlılığın bizim yaklaşımımızdan daha iyi olduğu görülmüştür. Fakat doğruluk olarak bizim seçtiğimiz özelliklerin önde olmasına

bakılarak 19 özellik ile daha iyi kesinlik ve f1-skoruna ulaşıldığı çıkarımına varılabilir. Bu da 19 özellik ile elde edilen sonuçlarda oltalama web sitesi olarak tespit ettiklerinin gerçekten oltalama web sitesi olma oranının daha yüksek olduğu (kesinlik) ve oltama tespit yetkinliğinin daha yüksek olduğu (f1-skoru) fakat gerçek oltama web sitelerini oltalama web sitesi olarak bulma oranının daha düşük olduğu (duyarlılık) anlamına gelmektedir.

Önerilen yaklaşımın nihai hedefi verimli özellikleri seçmektir. Mendeley 2018 veri setinde 48 özellikten 13'ü, Mendeley 2020 veri setinde 111 özellikten 19'u verimli özellik olarak seçilmiştir. Fazla sayıda özellik barındıran veri setinde daha az doğruluk düşüşü gözlemlenirken az sayıda özellik barındıran veri setinde daha fazla doğruluk düşüşü gözlemlenmiştir. Buradan yaklaşımın fazla sayıda özellik barındıran veri setlerinde daha etkin çalıştığı çıkarımı yapılmıştır.

Gelecek çalışmalarda önerilen yaklaşımdaki özellik önem skoru hesaplama yöntemleri yerine farklı özellik önem skoru hesaplama yöntemleri, GRIS ortalama hesabı yerine farklı bir ortalama hesap yöntemi denenerek yaklaşımın performansı iyileştirilmeye çalışılabilir. Önerilen yaklaşım, oltalama saldırıları tespit eden bir sisteme entegre edilerek (e-mail vs.) kısa sürede saldırıların tespit edilmesinde kullanılabilir.

KAYNAKLAR

1. İnternet: Verizon Business, "2022 Data Breach Investigations Report", ***https://www.verizon.com/business/resources/reports/dbir/*** (2022).
2. İnternet: APWG, "APWG Phishing Activity Trends Reports", ***https://apwg.org/trendsreports/*** (2023).
3. İnternet: Dymoke E., "GTA 6 leaks and Uber hacked through social engineering", ***https://www.hoxhunt.com/blog/gta-6-leaks-and-uber-hacked-through-social-engineering*** (2022).
4. Gupta, B.B., Tewari, A., Jain, A. K., and Agrawal, D. P., "Fighting against phishing attacks: state of the art and future challenges", ***Neural Comput. Appl.***, 28(12): 3629-3654 (2017).
5. Shankar, A., Shetty, R., and K, B. N., "A Review on Phishing Attacks", ***International Journal of Applied Engineering Research***, 14(9): 2171-2175 (2019).
6. Caputo, D. D., Pfleeger, S. L., Freeman, J. D., and Johnson M. E., "Going Spear Phishing: Exploring Embedded Training and Awareness", ***IEEE Security & Privacy***, 12(1): 28-38 (2014).
7. Alabdan, R., "Phishing Attacks Survey: Types, Vectors, and Technical Approaches" ***Future Internet***, 12(10): 168 (2020).
8. Shar, L. K., and Tan, H. B. K., "Defending against Cross-Site Scripting Attacks", ***Computer***, 45(3): 55-62 (2012).
9. Vidas T., Owusu E., Wang S., Zeng C., Cranor L.F., and Christin N., "QRishing: The Susceptibility of Smartphone Users to QR Code Phishing Attacks", ***Financial Cryptography and Data Security***, 7862: 52-69 (2013).
10. Cova, M., Kruegel, C., and Vigna, G. "Detection and analysis of drive-bydownload attacks and malicious JavaScript code" ***19th International Conference on World Wide Web***, 281-290 (2010).
11. Wei, Y., and Sekiya, Y., "Feature selection approach for phishing detection based on machine learning", ***Networks and Systems***, 61-70 (2022).

12. Shahrivari, V., Darabi, M. M., and Izadi, M., "Phishing Detection Using Machine Learning Techniques", ***Cryptography and Security***, (2020).
13. bin Othman Mustafa, M. S., Nomani Kabir, M., Ernawan, F., and Jing, W. "An Enhanced Model for Increasing Awareness of Vocational Students Against Phishing Attacks," 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 10-14 (2019).
14. Canova, G., Volkamer, M., Bergmann, C., and Borza, R. "NoPhish: An AntiPhishing Education App", Security and Trust Management, 8743:188-192 (2014).
15. Jain, A.K., and Gupta, B.B., "A novel approach to protect against phishing attacks at client side using auto-updated white-list", ***EURASIP J.***, 2016 (9) (2016).
16. Bell, S., and Komisarczuk, P., "An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank", ***ACSW '20: Australasian Computer Science Week Multiconference***, 3:1-11, (2020).
17. İnternet: Google, "Google Safe Browsing", ***https://safebrowsing.google.com/*** (2005).
18. İnternet: OpenPhish, "OpenPhish", ***https://openphish.com/*** (2014).
19. İnternet: PhishTank, "PhishTank", ***https://phishtank.com/*** (2006).
20. Jain, A.K., and Gupta, B.B., "Phishing Detection: Analysis of Visual Similarity Based Approaches", ***Security and Communication Networks***, 2017: 1-20 (2017).
21. Abdelnabi, S., Krombholz, K., and Fritz, M., "VisualPhishNet: ZeroDay Phishing Website Detection by Visual Similarity", ***CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security***, 1681-1698 (2020).
22. Sahingoz, O. K., Buber, E., Demir, O., and Diri, B., "Machine learning based phishing detection from URLs", ***Expert Systems with Applications***, 117: 345-357 (2019).
23. Jain, A.K., and Gupta, B.B., "A machine learning based approach for phishing detection using hyperlinks information", ***Journal of Ambient Intell Human Computing***, 10(5): 2015-2028 (2019).

24. Peng, T., Harris, I., and Sawa, Y., "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning", ***2018 IEEE 12th International Conference on Semantic Computing (ICSC)***, 300-301 (2018).
25. Korkmaz, M., Sahingoz, O. K., and Diri, B., "Detection of phishing websites by using machine learning-based URL analysis", ***2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*** (2020).
26. Lokesh, G.H., and BoreGowda, G., "Phishing website detection based on effective machine learning approach", ***Journal of Cyber Security Technology***, 5(1), 1–14 (2021).
27. Odeh, A., Keshta, I., and Abdelfattah, A., "Machine Learning Techniques for detection of website phishing: A review for promises and challenges", ***2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*** (2021).
28. Cardoso, J. R., Pereira, L. M., Iversen, M. D., and Ramos, A. L.; "What is gold standard and what is ground truth?", ***Dental press journal of orthodontics***, 19(5): 27–30 (2014).
29. Tan, C. L., "Phishing Dataset for Machine Learning: Feature Evaluation", ***Mendeley*** (2018).
30. Vrbančič, G., "Phishing Websites Dataset" ***Mendeley***, (2020).
31. Gangavarapu, T., Jaidhar, C.D., and Chanduka, B., "Applicability of machine learning in spam and phishing email filtering: review and approaches." ***Artif Intell Rev***, 53:5019–5081 (2020).
32. Breiman, L., "Random Forests", ***Machine Learning***, 45 (1):5–32 (2001).
33. Singh, A., Narina T., and Aakanksha S., "A Review of Supervised Machine Learning Algorithms." ***3rd International Conference on Computing for Sustainable Global Development (INDIACom)***, 1310–15 (2016).
34. Kleinbaum, D. G., and Mitchel K., "Introduction to Logistic Regression", Statistics for Biology and Health, ***Springer***, New York, 1-39 (2010).
35. Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems", ***Annals of Eugenics***, 7 (2):179–88 (1936).
36. Kingsford, C., and Salzberg, S. L., "What Are Decision Trees?", ***Nature Biotechnology***, vol. 26(9): 1011–1013, (2008).

37. Bishop, C. M., “Pattern Recognition and Machine Learning”, ***Springer-Verlag***, New York, (2006).
38. Peterson, L. E., “K-Nearest Neighbor”, ***Scholarpedia Journal***, 4(2): 1883, (2009).
39. Noble, W. S. “What Is a Support Vector Machine?”, ***Nature Biotechnology***, 24(12): 1565–1567, (2006).
40. Yunxiang, L., and Jiongjun, D., "Parameter Optimization of the SVM for Big Data", ***2015 8th International Symposium on Computational Intelligence and Design (ISCID)***, 341-344, (2015).
41. Bottou, L., “Large-scale machine learning with stochastic gradient descent”, ***COMPSTAT'2010***, 177-186 (2010).
42. Bottou, L., “Stochastic Gradient Descent Tricks.” Lecture Notes in Computer Science, ***Springer Berlin Heidelberg***, 421-436 (2012).
43. Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., and Hsieh, C. “Gradient Boosted Decision Trees for High Dimensional Sparse Output”, ***34th International Conference on Machine Learning***, 70:3182–90. (2017).
44. Schapire, R. E., “Explaining AdaBoost.” Empirical Inference, ***Springer Berlin Heidelberg***, 37-52 (2013).
45. Freund, Y. and Robert E. S., “A Short Introduction to Boosting”, ***Ucsd.edu***, (1999).
46. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., and Liu, T. Y., “Lightgbm: A highly efficient gradient boosting decision tree”, ***Advances in Neural Information Processing Systems***, 3149-3157 (2017).
47. Chen, T., and Guestrin, C., “Xgboost: A scalable tree boosting system”, ***22nd ACM SIGKDD***, 785-794 (2016).
48. Saarela, M., and Jauhiainen, S., “Comparison of feature importance measures as explanations for classification models”, ***SN Applied Sciences***, 3 (2) (2021).
49. İnternet: Scikit-Learn, “Permutation importance vs random forest feature importance (MDI)”, ***https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html*** (2022).
50. İnternet: Scikit-Learn, “Permutation feature importance”, ***https://scikit-learn.org/stable/modules/permutation_importance.html*** (2022).

51. Gunver, M. G., Senocak, M. S., and Vehid, S., “To determine skewness, mean and deviation with a new approach on continuous data”, ***PONTE International Scientific Researchs Journal***, 74(2) (2018).
52. Hossin, M., and Sulaiman, M. N. “A review on evaluation metrics for data classification evaluations”, ***International Journal of Data Mining & Knowledge Management Process***, 5 (2), 01–11 (2015).
53. Kulkarni, A., Chong, D., and Batarseh, F. A. “Foundations of data imbalance and solutions for a data democracy”, ***Data Democracy***, 83–106 (2020).

EK AÇIKLAMALAR A.

TESTLERİN GERÇEKLEŞTİRİLDİĞİ ORTAM

Çizelge Ek A.1. Algoritmaların çağırıldığı kütüphaneler ve eklenen parametreler.

Algoritma	**Çağırılan Kütüphane**	**Kütüphanenin Kullanılan Versiyonu**	**Eklenen Parametreler**
Random Forest (RF)	scikit-learn	1.2.1	max_features='auto', n_jobs=-1
Logistic Regression (LR)	scikit-learn	1.2.1	n_jobs=-1
Linear Discriminant Analysis (LDA)	scikit-learn	1.2.1	
Decision Tree (DCT)	scikit-learn	1.2.1	
GaussianNB (GNB)	scikit-learn	1.2.1	
K-Nearest Neighbors (KNN)	scikit-learn	1.2.1	n_neighbors=3, n_jobs=-1
Support Vector Machine (SVM)	scikit-learn	1.2.1	tol=1e-5
Stochastic Gradient Descent (SGD)	scikit-learn	1.2.1	n_jobs=-1
GradientBoosting (GB)	scikit-learn	1.2.1	
AdaBoost (ADA)	scikit-learn	1.2.1	
LightGBM (LGB)	lightgbm	2.2.3	
XGBoost (XGB)	xgboost	1.7.4	

EK AÇIKLAMALAR B.

DETAYLI SONUÇ

Çizelge Ek B.1. RF algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9867	0,9823	-0,0043	-0,44%
	Ortalama	0,9823	0,9759	-0,0064	-0,66%
	En Kötü	0,9777	0,9707	-0,0070	-0,72%
Kesinlik	En İyi	0,9899	0,9813	-0,0086	-0,87%
	Ortalama	0,9825	0,9742	-0,0084	-0,85%
	En Kötü	0,9750	0,9627	-0,0123	-1,26%
Duyarlılık	En İyi	0,9893	0,9867	-0,0027	-0,27%
	Ortalama	0,9821	0,9777	-0,0044	-0,45%
	En Kötü	0,9753	0,9700	-0,0053	-0,55%
F1-Skoru	En İyi	0,9866	0,9824	-0,0042	-0,43%
	Ortalama	0,9823	0,9759	-0,0064	-0,65%
	En Kötü	0,9776	0,9706	-0,0070	-0,71%
Eğitim Süresi	Ortalama	0,4653 s	0,4310 s	-0,0343	-7,36%
Test Süresi	Ortalama	0,1050 s	0,1034 s	-0,0016	-1,53%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.2. RF algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9725	0,9712	-0,0013	-0,13%
	Ortalama	0,9700	0,9689	-0,0012	-0,12%
	En Kötü	0,9676	0,9655	-0,0020	-0,21%
Kesinlik	En İyi	0,9592	0,9562	-0,0030	-0,31%
	Ortalama	0,9547	0,9518	-0,0029	-0,30%
	En Kötü	0,9506	0,9462	-0,0044	-0,47%
Duyarlılık	En İyi	0,9640	0,9629	-0,0011	-0,11%
	Ortalama	0,9587	0,9584	-0,0003	-0,03%
	En Kötü	0,9530	0,9531	0,0001	0,01%
F1-Skoru	En İyi	0,9603	0,9585	-0,0018	-0,19%
	Ortalama	0,9567	0,9551	-0,0016	-0,17%
	En Kötü	0,9532	0,9503	-0,0028	-0,29%
Eğitim Süresi	Ortalama	7,4759 s	7,1515 s	-0,3244	-4,34%
Test Süresi	Ortalama	0,5398 s	0,4142 s	-0,1256	-23,26%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.3. LR algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9437	0,9170	-0,0267	-2,83%
	Ortalama	0,9322	0,9073	-0,0250	-2,68%
	En Kötü	0,9193	0,8967	-0,0227	-2,47%
Kesinlik	En İyi	0,9417	0,9141	-0,0276	-2,93%
	Ortalama	0,9250	0,8990	-0,0260	-2,81%
	En Kötü	0,9042	0,8829	-0,0214	-2,36%
Duyarlılık	En İyi	0,9567	0,9327	-0,0240	-2,51%
	Ortalama	0,9408	0,9177	-0,0232	-2,46%
	En Kötü	0,9207	0,8940	-0,0267	-2,90%
F1-Skoru	En İyi	0,9438	0,9174	-0,0265	-2,81%
	Ortalama	0,9328	0,9082	-0,0246	-2,64%
	En Kötü	0,9199	0,8967	-0,0232	-2,52%
Eğitim Süresi	Ortalama	0,6966 s	0,1047 s	-0,5919	-84,97%
Test Süresi	Ortalama	0,0027 s	0,0020 s	-0,0007	-25,16%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.4. LR algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9168	0,8965	-0,0203	-2,21%
	Ortalama	0,8972	0,8846	-0,0126	-1,40%
	En Kötü	0,8746	0,8722	-0,0024	-0,28%
Kesinlik	En İyi	0,8950	0,9008	0,0058	0,65%
	Ortalama	0,8815	0,8918	0,0103	1,16%
	En Kötü	0,8520	0,8819	0,0298	3,50%
Duyarlılık	En İyi	0,9052	0,7962	-0,1090	-12,04%
	Ortalama	0,8125	0,7583	-0,0542	-6,67%
	En Kötü	0,7340	0,7161	-0,0178	-2,43%
F1-Skoru	En İyi	0,8807	0,8418	-0,0389	-4,42%
	Ortalama	0,8445	0,8196	-0,0249	-2,95%
	En Kötü	0,8018	0,7948	-0,0070	-0,88%
Eğitim Süresi	Ortalama	13,1210 s	3,1510 s	-9,9701	-75,99%
Test Süresi	Ortalama	0,0149 s	0,0057 s	-0,0092	-61,84%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.5. LDA algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9443	0,9207	-0,0237	-2,51%
	Ortalama	0,9360	0,9127	-0,0233	-2,49%
	En Kötü	0,9247	0,9020	-0,0227	-2,45%
Kesinlik	En İyi	0,9396	0,9219	-0,0177	-1,89%
	Ortalama	0,9249	0,9070	-0,0179	-1,93%
	En Kötü	0,9077	0,8920	-0,0157	-1,73%
Duyarlılık	En İyi	0,9653	0,9347	-0,0307	-3,18%
	Ortalama	0,9490	0,9197	-0,0293	-3,09%
	En Kötü	0,9333	0,8973	-0,0360	-3,86%
F1-Skoru	En İyi	0,9451	0,9213	-0,0238	-2,52%
	Ortalama	0,9368	0,9133	-0,0235	-2,51%
	En Kötü	0,9253	0,9028	-0,0225	-2,44%
Eğitim Süresi	Ortalama	0,0564 s	0,0230 s	-0,0335	-59,28%
Test Süresi	Ortalama	0,0025 s	0,0025 s	0,0000	0,00%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.6. LDA algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9188	0,9073	-0,0115	-1,25%
	Ortalama	0,9152	0,9020	-0,0132	-1,44%
	En Kötü	0,9085	0,8970	-0,0115	-1,26%
Kesinlik	En İyi	0,8350	0,8126	-0,0223	-2,67%
	Ortalama	0,8276	0,8025	-0,0250	-3,02%
	En Kötü	0,8159	0,7927	-0,0232	-2,84%
Duyarlılık	En İyi	0,9588	0,9549	-0,0039	-0,41%
	Ortalama	0,9532	0,9503	-0,0029	-0,30%
	En Kötü	0,9487	0,9455	-0,0032	-0,33%
F1-Skoru	En İyi	0,8903	0,8764	-0,0139	-1,56%
	Ortalama	0,8859	0,8702	-0,0157	-1,78%
	En Kötü	0,8776	0,8645	-0,0131	-1,49%
Eğitim Süresi	Ortalama	2,0480 s	0,1976 s	-1,8504	-90,35%
Test Süresi	Ortalama	0,0156 s	0,0059 s	-0,0097	-62,35%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.7. DCT algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9733	0,9673	-0,0060	-0,62%
	Ortalama	0,9653	0,9602	-0,0052	-0,53%
	En Kötü	0,9580	0,9503	-0,0077	-0,80%
Kesinlik	En İyi	0,9778	0,9691	-0,0087	-0,89%
	Ortalama	0,9620	0,9569	-0,0051	-0,53%
	En Kötü	0,9507	0,9383	-0,0124	-1,31%
Duyarlılık	En İyi	0,9800	0,9767	-0,0033	-0,34%
	Ortalama	0,9690	0,9638	-0,0052	-0,54%
	En Kötü	0,9580	0,9507	-0,0073	-0,77%
F1-Skoru	En İyi	0,9735	0,9673	-0,0062	-0,64%
	Ortalama	0,9655	0,9603	-0,0051	-0,53%
	En Kötü	0,9583	0,9508	-0,0075	-0,78%
Eğitim Süresi	Ortalama	0,0541 s	0,0261 s	-0,0280	-51,71%
Test Süresi	Ortalama	0,0024 s	0,0018 s	-0,0005	-22,86%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.8. DCT algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9558	0,9551	-0,0007	-0,07%
	Ortalama	0,9528	0,9513	-0,0015	-0,15%
	En Kötü	0,9497	0,9480	-0,0017	-0,17%
Kesinlik	En İyi	0,9384	0,9371	-0,0014	-0,15%
	Ortalama	0,9320	0,9301	-0,0018	-0,20%
	En Kötü	0,9262	0,9204	-0,0057	-0,62%
Duyarlılık	En İyi	0,9385	0,9359	-0,0026	-0,28%
	Ortalama	0,9314	0,9290	-0,0024	-0,26%
	En Kötü	0,9247	0,9217	-0,0030	-0,33%
F1-Skoru	En İyi	0,9361	0,9350	-0,0011	-0,12%
	Ortalama	0,9317	0,9295	-0,0021	-0,23%
	En Kötü	0,9271	0,9252	-0,0019	-0,21%
Eğitim Süresi	Ortalama	1,1792 s	0,6405 s	-0,5387	-45,69%
Test Süresi	Ortalama	0,0210 s	0,0083 s	-0,0127	-60,67%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.9. GNB algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,8583	0,8423	-0,0160	-1,86%
	Ortalama	0,8443	0,8266	-0,0176	-2,09%
	En Kötü	0,8263	0,8117	-0,0147	-1,77%
Kesinlik	En İyi	0,9478	0,9393	-0,0086	-0,90%
	Ortalama	0,9307	0,9167	-0,0140	-1,51%
	En Kötü	0,9135	0,9024	-0,0111	-1,21%
Duyarlılık	En İyi	0,7700	0,7453	-0,0247	-3,20%
	Ortalama	0,7441	0,7187	-0,0254	-3,41%
	En Kötü	0,7167	0,6973	-0,0193	-2,70%
F1-Skoru	En İyi	0,8446	0,8228	-0,0218	-2,58%
	Ortalama	0,8269	0,8056	-0,0213	-2,57%
	En Kötü	0,8049	0,7874	-0,0176	-2,18%
Eğitim Süresi	Ortalama	0,0046 s	0,0033 s	-0,0013	-29,14%
Test Süresi	Ortalama	0,0024 s	0,0017 s	-0,0007	-30,97%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.10. GNB algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,8579	0,8732	0,0153	1,79%
	Ortalama	0,8428	0,8684	0,0256	3,04%
	En Kötü	0,8359	0,8643	0,0284	3,39%
Kesinlik	En İyi	0,8915	0,8917	0,0001	0,02%
	Ortalama	0,8831	0,8825	-0,0006	-0,07%
	En Kötü	0,8757	0,8747	-0,0010	-0,12%
Duyarlılık	En İyi	0,6723	0,7275	0,0553	8,22%
	Ortalama	0,6285	0,7146	0,0860	13,69%
	En Kötü	0,6066	0,7005	0,0939	15,47%
F1-Skoru	En İyi	0,7658	0,7983	0,0325	4,24%
	Ortalama	0,7343	0,7897	0,0554	7,54%
	En Kötü	0,7188	0,7814	0,0625	8,70%
Eğitim Süresi	Ortalama	0,1409 s	0,0311 s	-0,1098	-77,91%
Test Süresi	Ortalama	0,0456 s	0,0105 s	-0,0351	-77,05%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.11. KNN algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,8820	0,9057	0,0237	2,68%
	Ortalama	0,8688	0,8901	0,0214	2,46%
	En Kötü	0,8563	0,8760	0,0197	2,30%
Kesinlik	En İyi	0,8603	0,8975	0,0372	4,32%
	Ortalama	0,8465	0,8814	0,0349	4,12%
	En Kötü	0,8293	0,8683	0,0390	4,70%
Duyarlılık	En İyi	0,9207	0,9220	0,0013	0,14%
	Ortalama	0,9010	0,9016	0,0007	0,07%
	En Kötü	0,8840	0,8860	0,0020	0,23%
F1-Skoru	En İyi	0,8860	0,9066	0,0206	2,33%
	Ortalama	0,8729	0,8914	0,0185	2,12%
	En Kötü	0,8608	0,8772	0,0165	1,91%
Eğitim Süresi	Ortalama	0,0049 s	0,0137 s	0,0089	182,27%
Test Süresi	Ortalama	0,2432 s	0,1195 s	-0,1237	-50,86%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.12. KNN algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,8827	0,8824	-0,0003	-0,04%
	Ortalama	0,8796	0,8787	-0,0009	-0,10%
	En Kötü	0,8767	0,8750	-0,0018	-0,20%
Kesinlik	En İyi	0,8378	0,8363	-0,0014	-0,17%
	Ortalama	0,8311	0,8279	-0,0031	-0,37%
	En Kötü	0,8236	0,8192	-0,0044	-0,53%
Duyarlılık	En İyi	0,8254	0,8249	-0,0005	-0,07%
	Ortalama	0,8180	0,8195	0,0015	0,19%
	En Kötü	0,8084	0,8119	0,0036	0,44%
F1-Skoru	En İyi	0,8287	0,8282	-0,0005	-0,06%
	Ortalama	0,8245	0,8237	-0,0008	-0,09%
	En Kötü	0,8193	0,8181	-0,0012	-0,15%
Eğitim Süresi	Ortalama	0,0589 s	0,0121 s	-0,0468	-79,48%
Test Süresi	Ortalama	21,0959 s	10,3800 s	-10,7160	-50,80%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.13. SVM algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9513	0,9200	-0,0313	-3,29%
	Ortalama	0,9441	0,9135	-0,0306	-3,24%
	En Kötü	0,9343	0,9060	-0,0283	-3,03%
Kesinlik	En İyi	0,9460	0,9084	-0,0375	-3,97%
	Ortalama	0,9362	0,8986	-0,0376	-4,02%
	En Kötü	0,9206	0,8830	-0,0375	-4,08%
Duyarlılık	En İyi	0,9600	0,9460	-0,0140	-1,46%
	Ortalama	0,9532	0,9322	-0,0210	-2,20%
	En Kötü	0,9433	0,9227	-0,0207	-2,19%
F1-Skoru	En İyi	0,9516	0,9214	-0,0302	-3,17%
	Ortalama	0,9446	0,9151	-0,0295	-3,13%
	En Kötü	0,9349	0,9082	-0,0267	-2,86%
Eğitim Süresi	Ortalama	0,6919 s	0,5520 s	-0,1399	-20,22%
Test Süresi	Ortalama	0,0061 s	0,0075 s	0,0014	23,82%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.14. SVM algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9353	0,9218	-0,0135	-1,45%
	Ortalama	0,9328	0,9198	-0,0130	-1,40%
	En Kötü	0,9303	0,9174	-0,0129	-1,39%
Kesinlik	En İyi	0,8963	0,8689	-0,0274	-3,06%
	Ortalama	0,8909	0,8605	-0,0304	-3,41%
	En Kötü	0,8832	0,8541	-0,0291	-3,30%
Duyarlılık	En İyi	0,9254	0,9220	-0,0034	-0,36%
	Ortalama	0,9181	0,9165	-0,0015	-0,17%
	En Kötü	0,9109	0,9109	0,0000	0,00%
F1-Skoru	En İyi	0,9081	0,8907	-0,0174	-1,91%
	Ortalama	0,9043	0,8876	-0,0166	-1,84%
	En Kötü	0,9004	0,8849	-0,0155	-1,73%
Eğitim Süresi	Ortalama	46,7562 s	15,4548 s	-31,3014	-66,95%
Test Süresi	Ortalama	0,0328 s	0,0076 s	-0,0252	-76,90%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.15. SGD algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9473	0,9250	-0,0223	-2,36%
	Ortalama	0,9380	0,9080	-0,0300	-3,19%
	En Kötü	0,9247	0,8817	-0,0430	-4,65%
Kesinlik	En İyi	0,9521	0,9323	-0,0198	-2,08%
	Ortalama	0,9353	0,8987	-0,0366	-3,91%
	En Kötü	0,9128	0,8594	-0,0534	-5,85%
Duyarlılık	En İyi	0,9620	0,9533	-0,0087	-0,90%
	Ortalama	0,9412	0,9203	-0,0209	-2,22%
	En Kötü	0,9140	0,8240	-0,0900	-9,85%
F1-Skoru	En İyi	0,9477	0,9258	-0,0219	-2,31%
	Ortalama	0,9382	0,9091	-0,0291	-3,10%
	En Kötü	0,9254	0,8744	-0,0509	-5,50%
Eğitim Süresi	Ortalama	0,0527 s	0,0287 s	-0,0240	-45,49%
Test Süresi	Ortalama	0,0036 s	0,0026 s	-0,0010	-27,94%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.16. SGD algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9340	0,9243	-0,0097	-1,04%
	Ortalama	0,9285	0,9195	-0,0090	-0,97%
	En Kötü	0,9167	0,9152	-0,0015	-0,16%
Kesinlik	En İyi	0,9341	0,9040	-0,0301	-3,22%
	Ortalama	0,8905	0,8638	-0,0268	-3,00%
	En Kötü	0,8303	0,8350	0,0047	0,57%
Duyarlılık	En İyi	0,9540	0,9405	-0,0135	-1,41%
	Ortalama	0,9058	0,9114	0,0055	0,61%
	En Kötü	0,8199	0,8656	0,0457	5,57%
F1-Skoru	En İyi	0,9055	0,8925	-0,0130	-1,43%
	Ortalama	0,8975	0,8867	-0,0108	-1,20%
	En Kötü	0,8733	0,8773	0,0041	0,47%
Eğitim Süresi	Ortalama	1,5553 s	0,3762 s	-1,1791	-75,81%
Test Süresi	Ortalama	0,0297 s	0,0083 s	-0,0214	-72,11%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.17. GB algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9837	0,9783	-0,0053	-0,54%
	Ortalama	0,9770	0,9705	-0,0065	-0,66%
	En Kötü	0,9717	0,9637	-0,0080	-0,82%
Kesinlik	En İyi	0,9832	0,9779	-0,0053	-0,54%
	Ortalama	0,9753	0,9689	-0,0064	-0,66%
	En Kötü	0,9691	0,9580	-0,0111	-1,14%
Duyarlılık	En İyi	0,9867	0,9807	-0,0060	-0,61%
	Ortalama	0,9789	0,9724	-0,0065	-0,67%
	En Kötü	0,9693	0,9600	-0,0093	-0,96%
F1-Skoru	En İyi	0,9837	0,9784	-0,0053	-0,54%
	Ortalama	0,9771	0,9706	-0,0065	-0,66%
	En Kötü	0,9716	0,9635	-0,0081	-0,83%
Eğitim Süresi	Ortalama	1,4104 s	0,8224 s	-0,5880	-41,69%
Test Süresi	Ortalama	0,0079 s	0,0068 s	-0,0011	-14,18%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.18. GB algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9562	0,9546	-0,0016	-0,17%
	Ortalama	0,9538	0,9510	-0,0028	-0,30%
	En Kötü	0,9513	0,9480	-0,0033	-0,35%
Kesinlik	En İyi	0,9370	0,9333	-0,0037	-0,39%
	Ortalama	0,9316	0,9272	-0,0043	-0,46%
	En Kötü	0,9237	0,9195	-0,0042	-0,46%
Duyarlılık	En İyi	0,9420	0,9379	-0,0041	-0,44%
	Ortalama	0,9350	0,9312	-0,0038	-0,41%
	En Kötü	0,9284	0,9245	-0,0039	-0,42%
F1-Skoru	En İyi	0,9368	0,9345	-0,0024	-0,25%
	Ortalama	0,9333	0,9292	-0,0041	-0,44%
	En Kötü	0,9297	0,9250	-0,0047	-0,51%
Eğitim Süresi	Ortalama	26,6795 s	14,9296 s	-11,7499	-44,04%
Test Süresi	Ortalama	0,1004 s	0,0533 s	-0,0471	-46,89%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.19. ADA algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9793	0,9733	-0,0060	-0,61%
	Ortalama	0,9711	0,9650	-0,0061	-0,63%
	En Kötü	0,9640	0,9570	-0,0070	-0,73%
Kesinlik	En İyi	0,9778	0,9731	-0,0047	-0,48%
	Ortalama	0,9694	0,9606	-0,0089	-0,91%
	En Kötü	0,9587	0,9425	-0,0161	-1,68%
Duyarlılık	En İyi	0,9820	0,9813	-0,0007	-0,07%
	Ortalama	0,9729	0,9699	-0,0030	-0,31%
	En Kötü	0,9600	0,9560	-0,0040	-0,42%
F1-Skoru	En İyi	0,9794	0,9734	-0,0060	-0,62%
	Ortalama	0,9712	0,9652	-0,0060	-0,61%
	En Kötü	0,9639	0,9570	-0,0069	-0,72%
Eğitim Süresi	Ortalama	0,3754 s	0,2641 s	-0,1113	-29,65%
Test Süresi	Ortalama	0,0304 s	0,0262 s	-0,0042	-13,85%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.20. ADA algoritmasının Mendeley 2020'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9432	0,9401	-0,0031	-0,33%
	Ortalama	0,9360	0,9337	-0,0023	-0,25%
	En Kötü	0,9318	0,9273	-0,0045	-0,48%
Kesinlik	En İyi	0,9240	0,9202	-0,0038	-0,41%
	Ortalama	0,9084	0,9052	-0,0031	-0,35%
	En Kötü	0,9008	0,8930	-0,0078	-0,87%
Duyarlılık	En İyi	0,9148	0,9106	-0,0042	-0,46%
	Ortalama	0,9064	0,9027	-0,0037	-0,41%
	En Kötü	0,8984	0,8933	-0,0051	-0,57%
F1-Skoru	En İyi	0,9173	0,9130	-0,0043	-0,47%
	Ortalama	0,9074	0,9040	-0,0034	-0,38%
	En Kötü	0,9012	0,8951	-0,0061	-0,68%
Eğitim Süresi	Ortalama	7,2469 s	3,8654 s	-3,3815	-46,66%
Test Süresi	Ortalama	0,5471 s	0,2672 s	-0,2800	-51,17%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.21. LGB algoritmasının Mendeley 2018'deki sonuçları.

Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9893	0,9837	-0,0057	-0,57%
	Ortalama	0,9850	0,9771	-0,0079	-0,80%
	En Kötü	0,9780	0,9717	-0,0063	-0,65%
Kesinlik	En İyi	0,9919	0,9833	-0,0086	-0,87%
	Ortalama	0,9844	0,9758	-0,0086	-0,88%
	En Kötü	0,9768	0,9670	-0,0097	-1,00%
Duyarlılık	En İyi	0,9927	0,9873	-0,0053	-0,54%
	Ortalama	0,9856	0,9786	-0,0070	-0,71%
	En Kötü	0,9767	0,9660	-0,0107	-1,09%
F1-Skoru	En İyi	0,9893	0,9837	-0,0057	-0,57%
	Ortalama	0,9850	0,9772	-0,0078	-0,80%
	En Kötü	0,9780	0,9716	-0,0064	-0,65%
Eğitim Süresi	Ortalama	0,2386 s	0,1448 s	-0,0938	-39,32%
Test Süresi	Ortalama	0,0131 s	0,0121 s	-0,0010	-7,83%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.22. LGB algoritmasının Mendeley 2020'deki sonuçları.

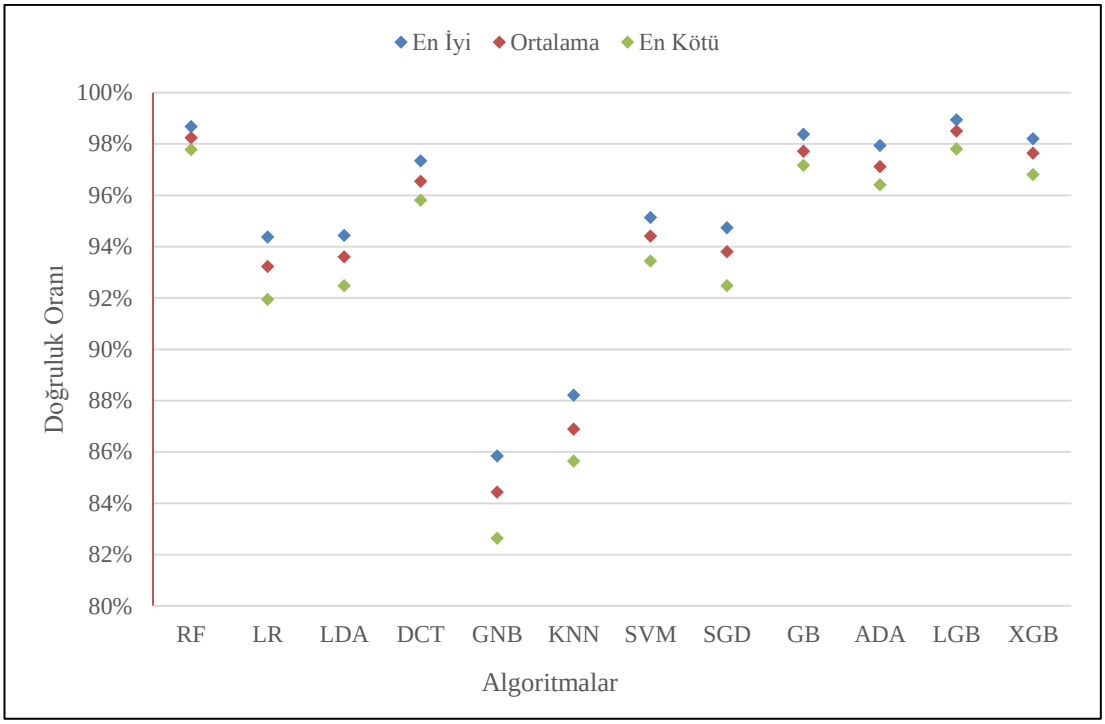
Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9685	0,9665	-0,0020	-0,21%
	Ortalama	0,9661	0,9642	-0,0019	-0,20%
	En Kötü	0,9635	0,9608	-0,0027	-0,28%
Kesinlik	En İyi	0,9550	0,9540	-0,0011	-0,11%
	Ortalama	0,9502	0,9465	-0,0036	-0,38%
	En Kötü	0,9442	0,9415	-0,0026	-0,28%
Duyarlılık	En İyi	0,9562	0,9570	0,0009	0,09%
	Ortalama	0,9519	0,9501	-0,0018	-0,19%
	En Kötü	0,9467	0,9434	-0,0033	-0,34%
F1-Skoru	En İyi	0,9546	0,9517	-0,0028	-0,30%
	Ortalama	0,9510	0,9483	-0,0027	-0,29%
	En Kötü	0,9473	0,9434	-0,0040	-0,42%
Eğitim Süresi	Ortalama	3,3578 s	1,2520 s	-2,1058	-62,71%
Test Süresi	Ortalama	0,1596 s	0,1205 s	-0,0392	-24,53%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%

Çizelge Ek B.23. XGB algoritmasının Mendeley 2018'deki sonuçları.

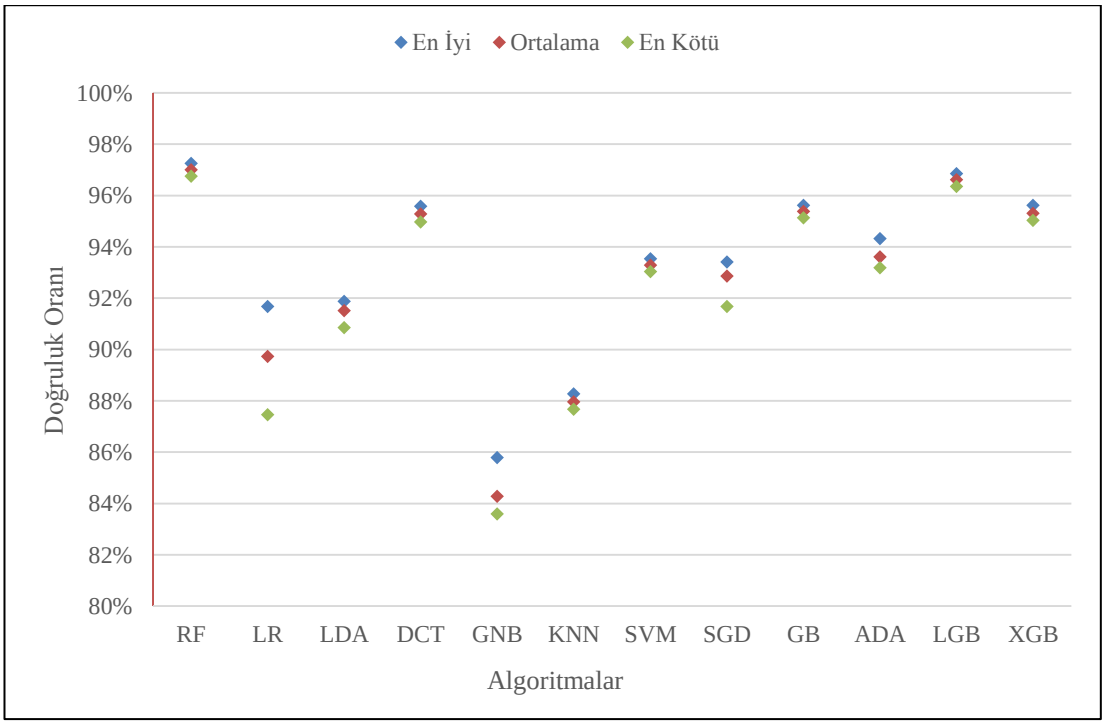
Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9820	0,9763	-0,0057	-0,58%
	Ortalama	0,9764	0,9700	-0,0064	-0,66%
	En Kötü	0,9680	0,9637	-0,0043	-0,45%
Kesinlik	En İyi	0,9839	0,9784	-0,0055	-0,56%
	Ortalama	0,9738	0,9682	-0,0057	-0,58%
	En Kötü	0,9668	0,9592	-0,0075	-0,78%
Duyarlılık	En İyi	0,9867	0,9820	-0,0047	-0,47%
	Ortalama	0,9791	0,9720	-0,0071	-0,73%
	En Kötü	0,9693	0,9627	-0,0067	-0,69%
F1-Skoru	En İyi	0,9820	0,9763	-0,0057	-0,58%
	Ortalama	0,9765	0,9701	-0,0064	-0,66%
	En Kötü	0,9680	0,9637	-0,0044	-0,45%
Eğitim Süresi	Ortalama	0,6746 s	0,3190 s	-0,3556	-52,71%
Test Süresi	Ortalama	0,0095 s	0,0074 s	-0,0021	-22,37%
Bellek Kullanımı	Ortalama	3,66 MB	0,99 MB	-2,6700	-72,95%

Çizelge Ek B.24. XGB algoritmasının Mendeley 2020'deki sonuçları.

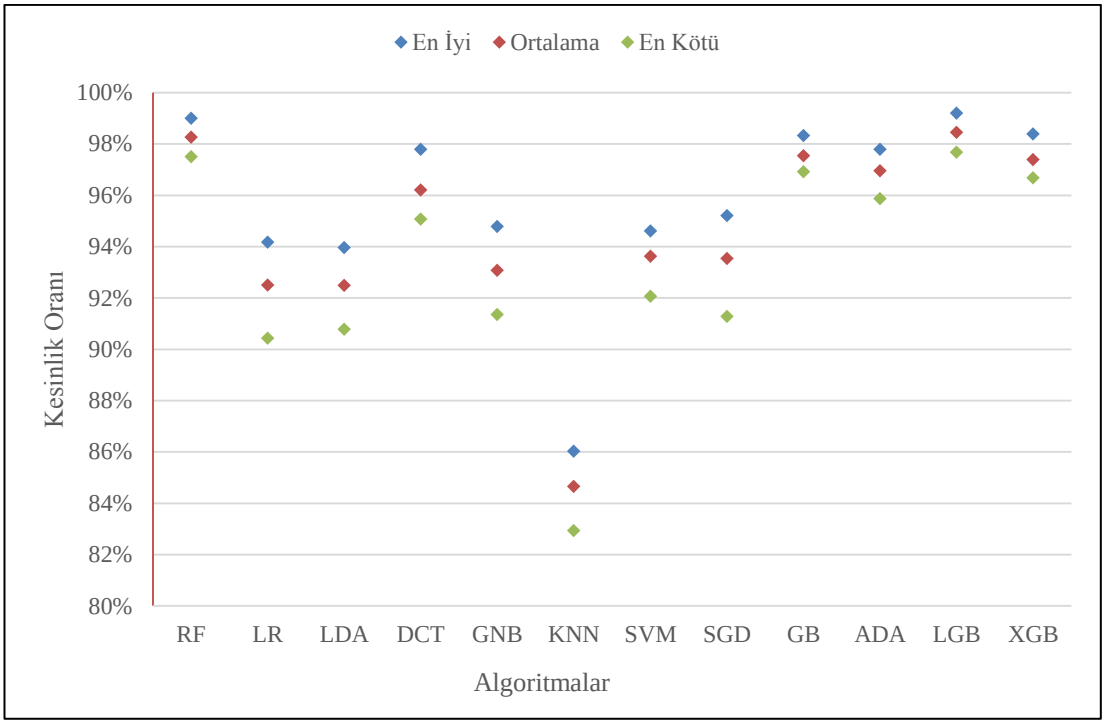
Metrikler		Bütün Özellikler Kullanıldığında	Verimli Özellikler Kullanıldığında	Numerik Fark	Yüzdesel Fark
Doğruluk	En İyi	0,9561	0,9538	-0,0023	-0,24%
	Ortalama	0,9530	0,9505	-0,0025	-0,26%
	En Kötü	0,9503	0,9478	-0,0025	-0,26%
Kesinlik	En İyi	0,9338	0,9296	-0,0042	-0,45%
	Ortalama	0,9273	0,9236	-0,0037	-0,40%
	En Kötü	0,9190	0,9175	-0,0015	-0,17%
Duyarlılık	En İyi	0,9461	0,9405	-0,0055	-0,59%
	Ortalama	0,9375	0,9341	-0,0034	-0,36%
	En Kötü	0,9323	0,9269	-0,0054	-0,58%
F1-Skoru	En İyi	0,9369	0,9335	-0,0034	-0,36%
	Ortalama	0,9324	0,9288	-0,0035	-0,38%
	En Kötü	0,9288	0,9249	-0,0038	-0,41%
Eğitim Süresi	Ortalama	15,8798 s	4,4780 s	-11,4018	-71,80%
Test Süresi	Ortalama	0,1271 s	0,0884 s	-0,0387	-30,47%
Bellek Kullanımı	Ortalama	75,07 MB	12,85 MB	-62,2200	-82,88%



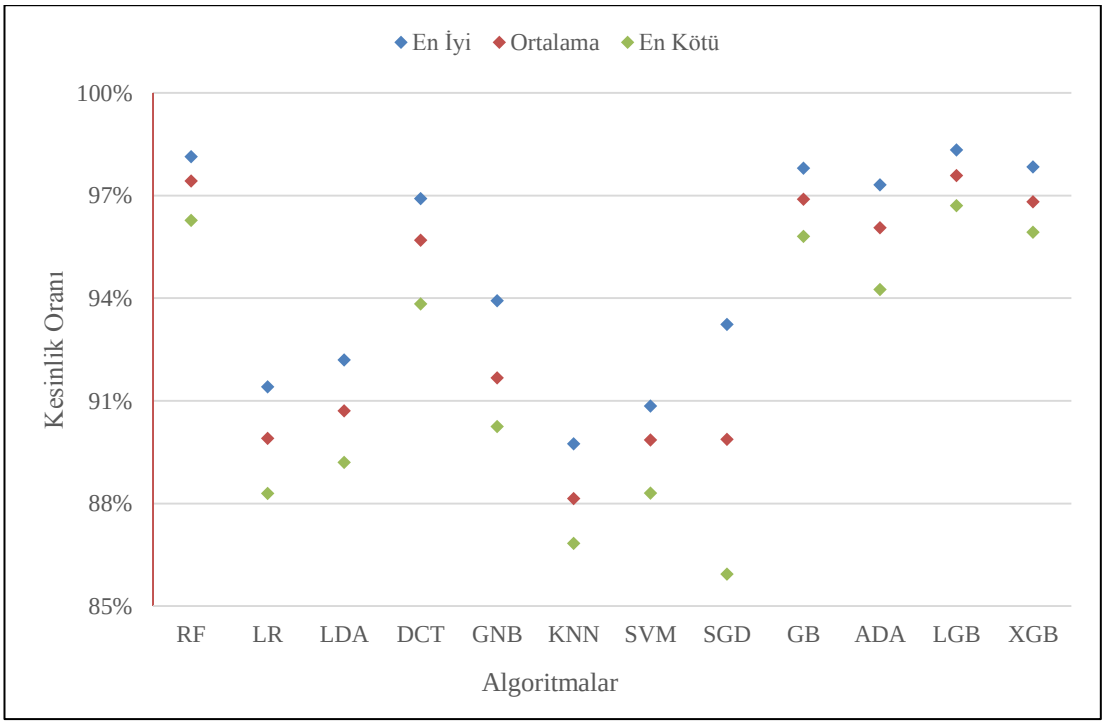
Şekil Ek B.1. Mendeley 2018 veri setinde bütün özellikler kullanılarak algoritmaların doğruluk oranlarının karşılaştırılması.



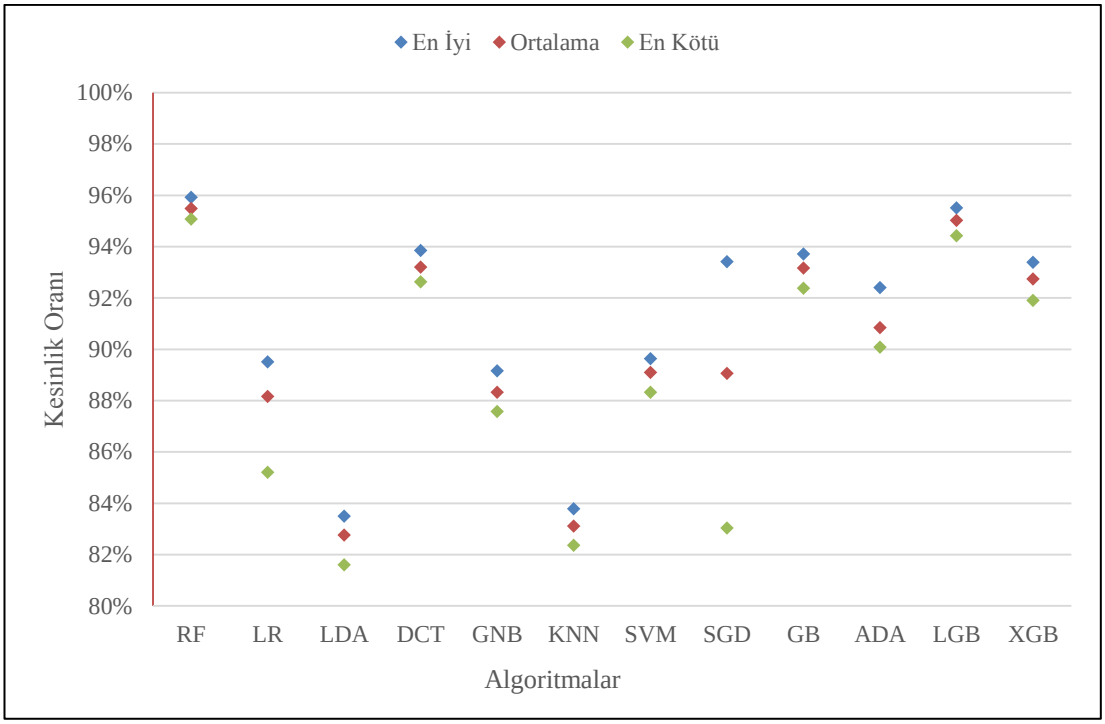
Şekil Ek B.2. Mendeley 2020 veri setinde bütün özellikler kullanılarak algoritmaların doğruluk oranlarının karşılaştırılması.



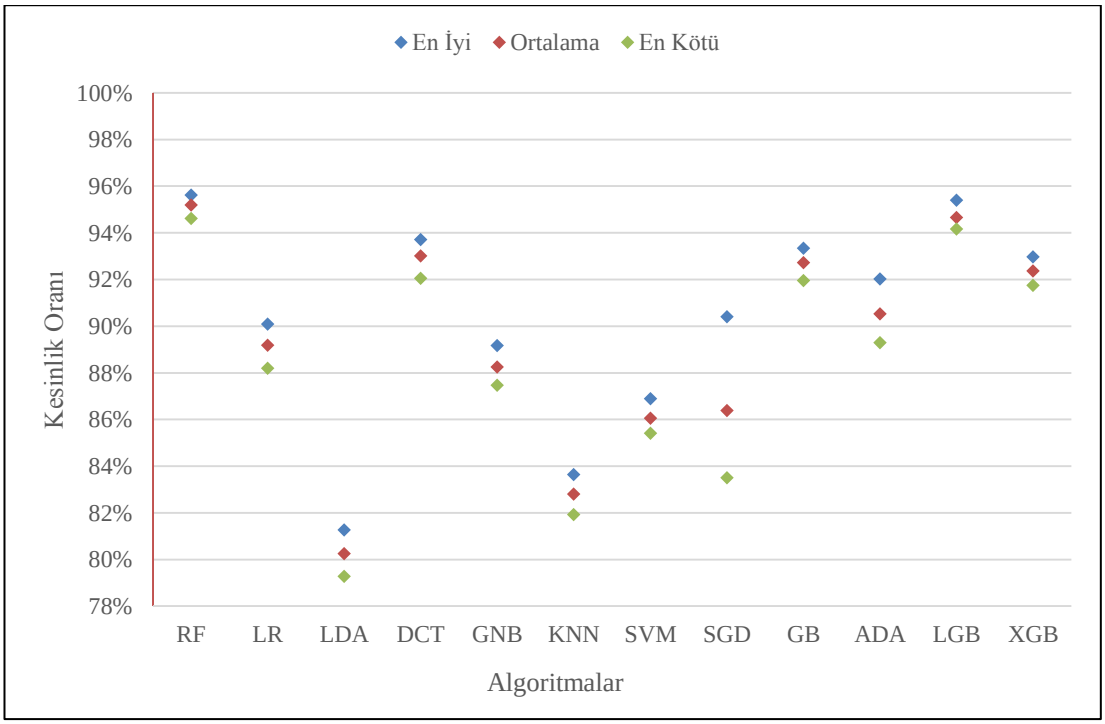
Şekil Ek B.3. Mendeley 2018 veri setinde bütün özellikler kullanılarak algoritmaların kesinlik oranlarının karşılaştırılması.



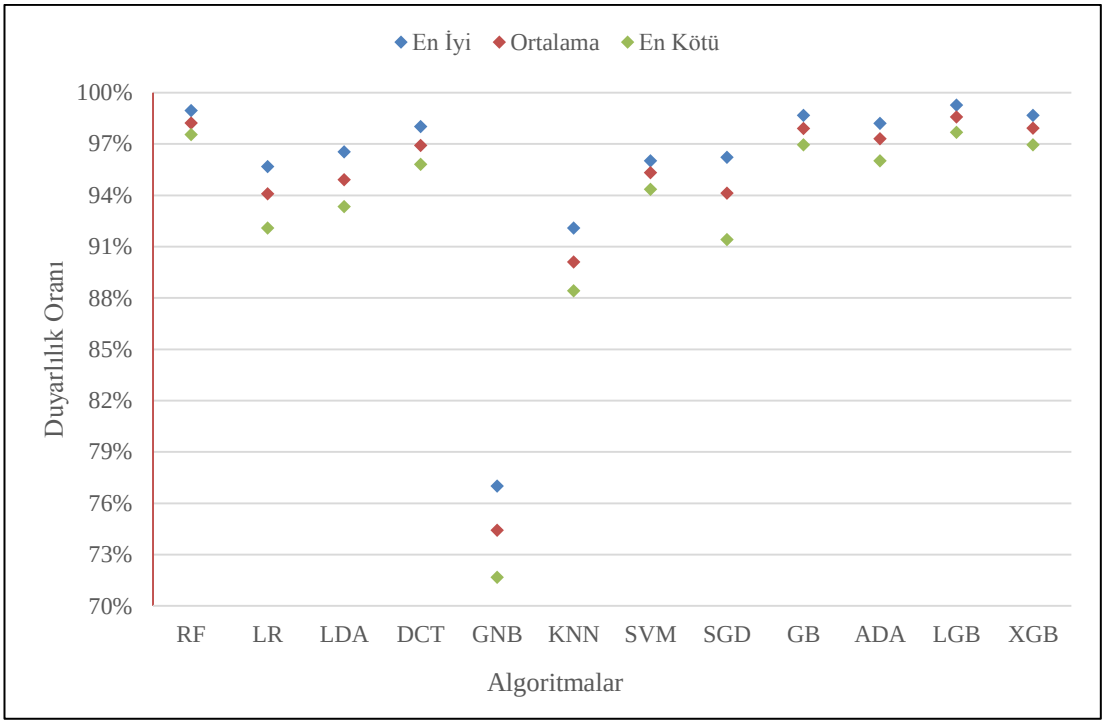
Şekil Ek B.4. Mendeley 2018 veri setinde verimli özellikler kullanılarak algoritmaların kesinlik oranlarının karşılaştırılması.



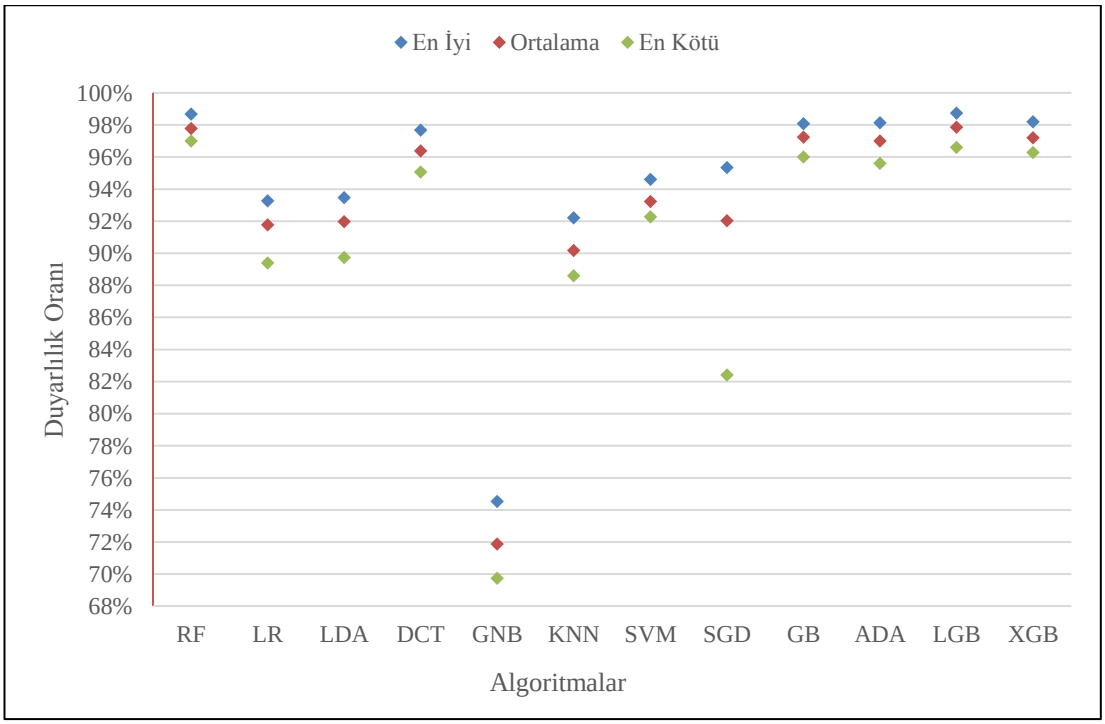
Şekil Ek B.5. Mendeley 2020 veri setinde bütün özellikler kullanılarak algoritmaların kesinlik oranlarının karşılaştırılması.



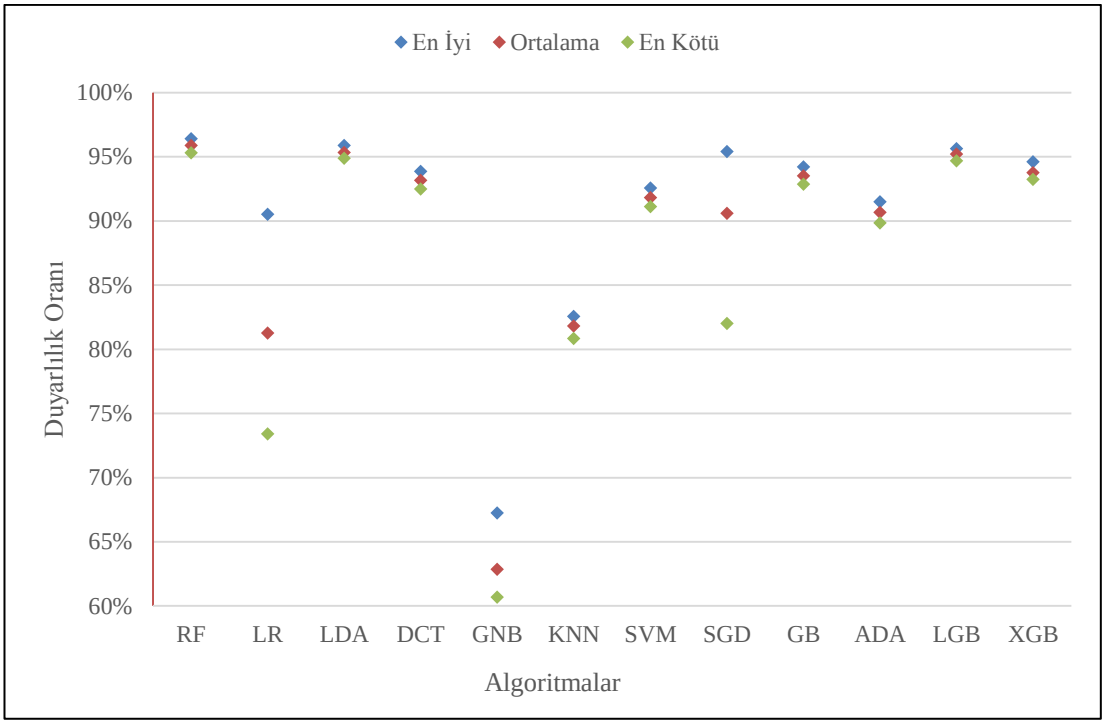
Şekil Ek B.6. Mendeley 2020 veri setinde verimli özellikler kullanılarak algoritmaların kesinlik oranlarının karşılaştırılması.



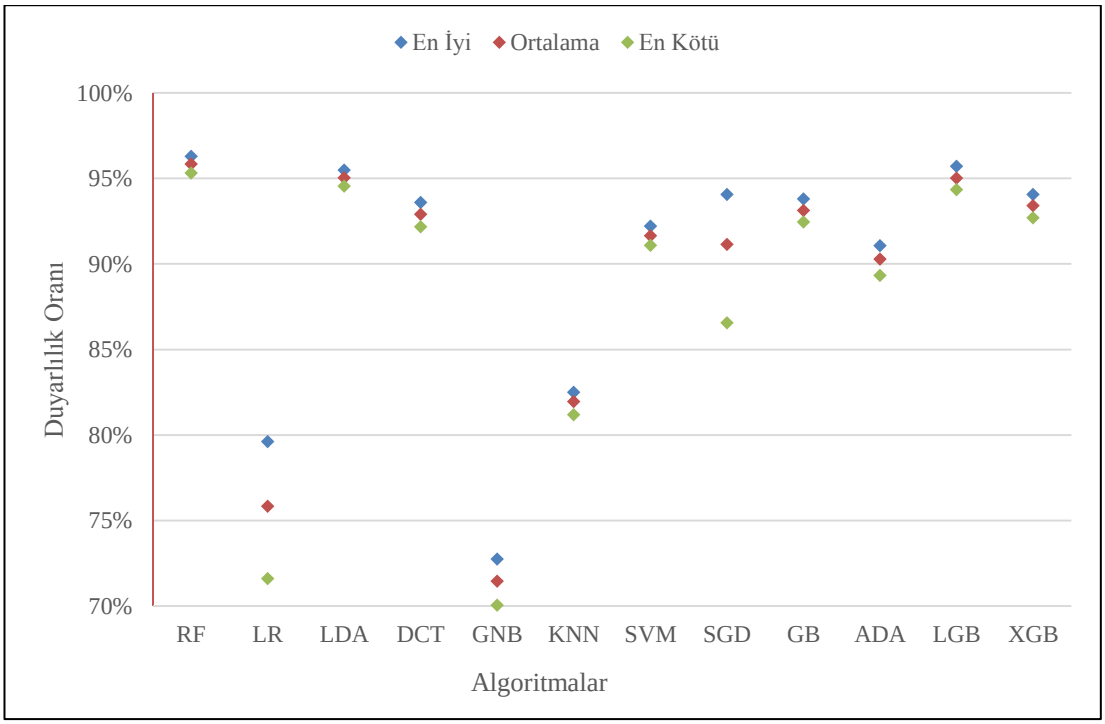
Şekil Ek B.7. Mendeley 2018 veri setinde bütün özellikler kullanılarak algoritmaların duyarlılık oranlarının karşılaştırılması.



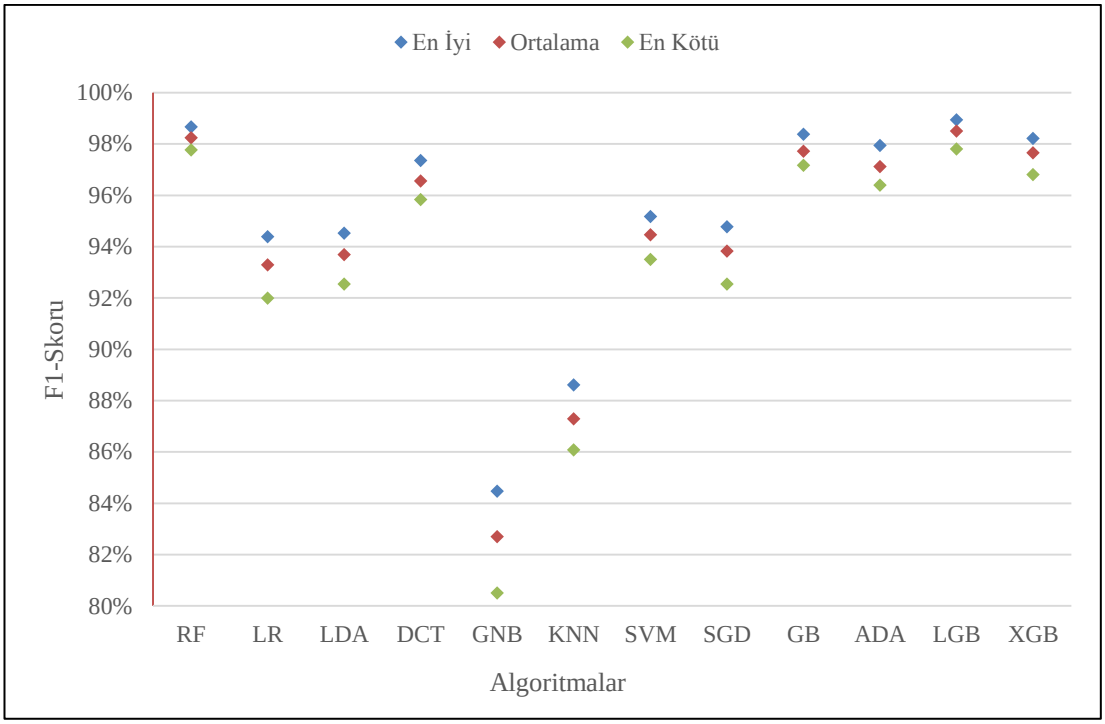
Şekil Ek B.8. Mendeley 2018 veri setinde verimli özellikler kullanılarak algoritmaların duyarlılık oranlarının karşılaştırılması.



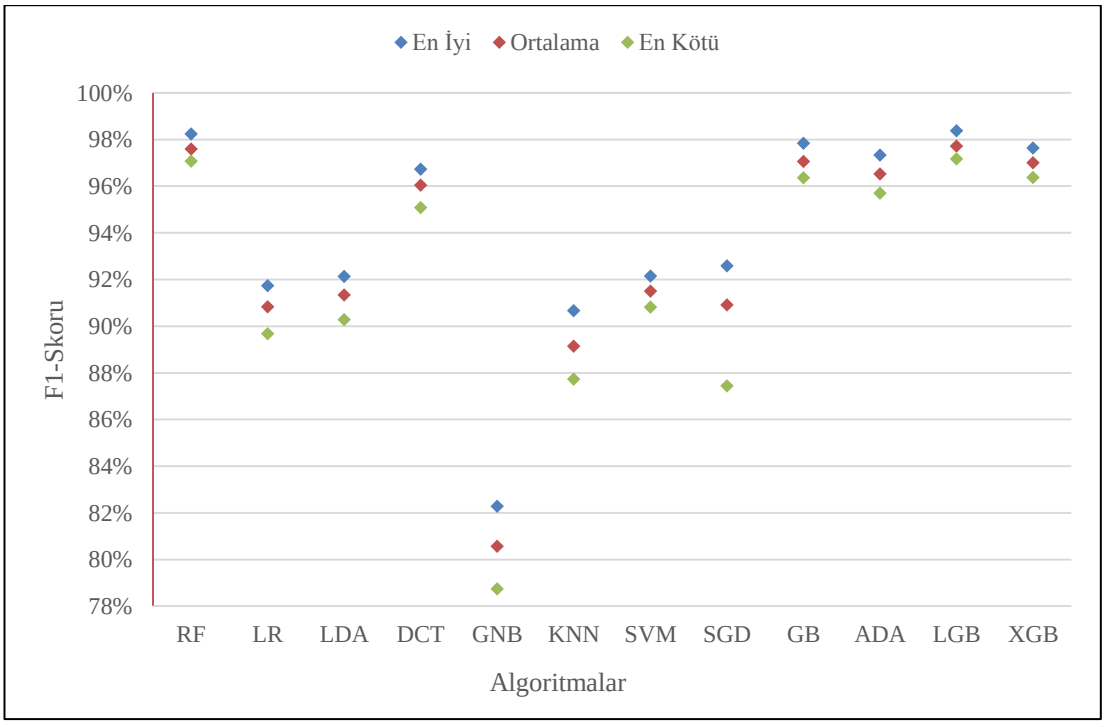
Şekil Ek B.9. Mendeley 2020 veri setinde bütün özellikler kullanılarak algoritmaların duyarlılık oranlarının karşılaştırılması.



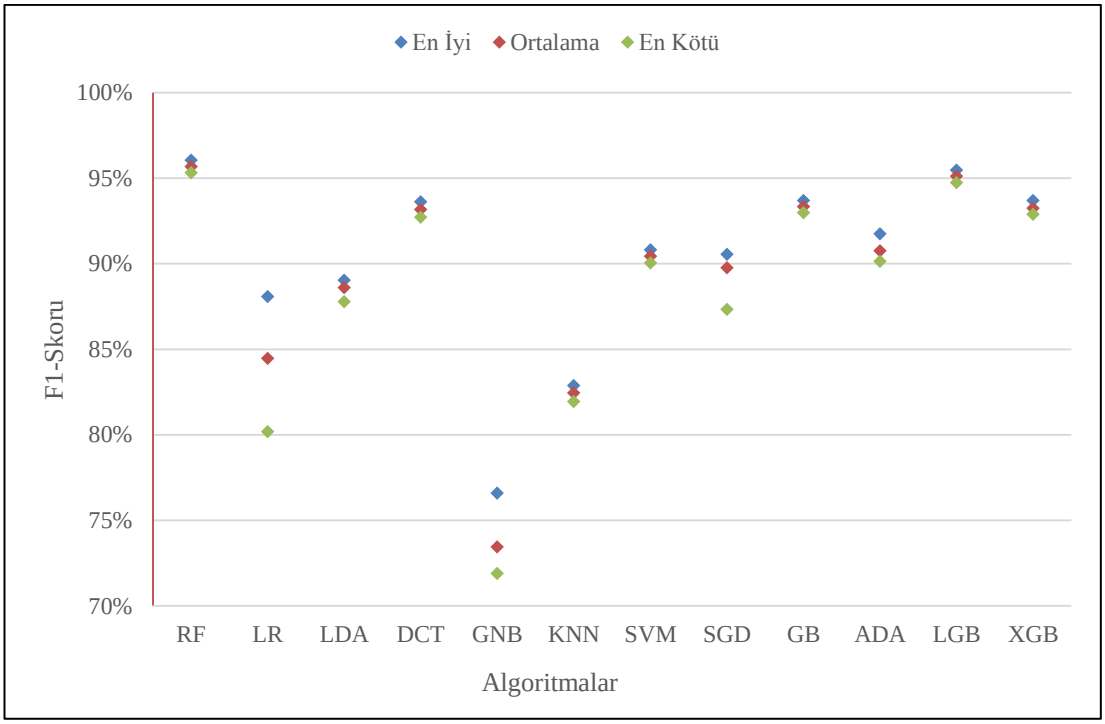
Şekil Ek B.10. Mendeley 2020 veri setinde verimli özellikler kullanılarak algoritmaların duyarlılık oranlarının karşılaştırılması.



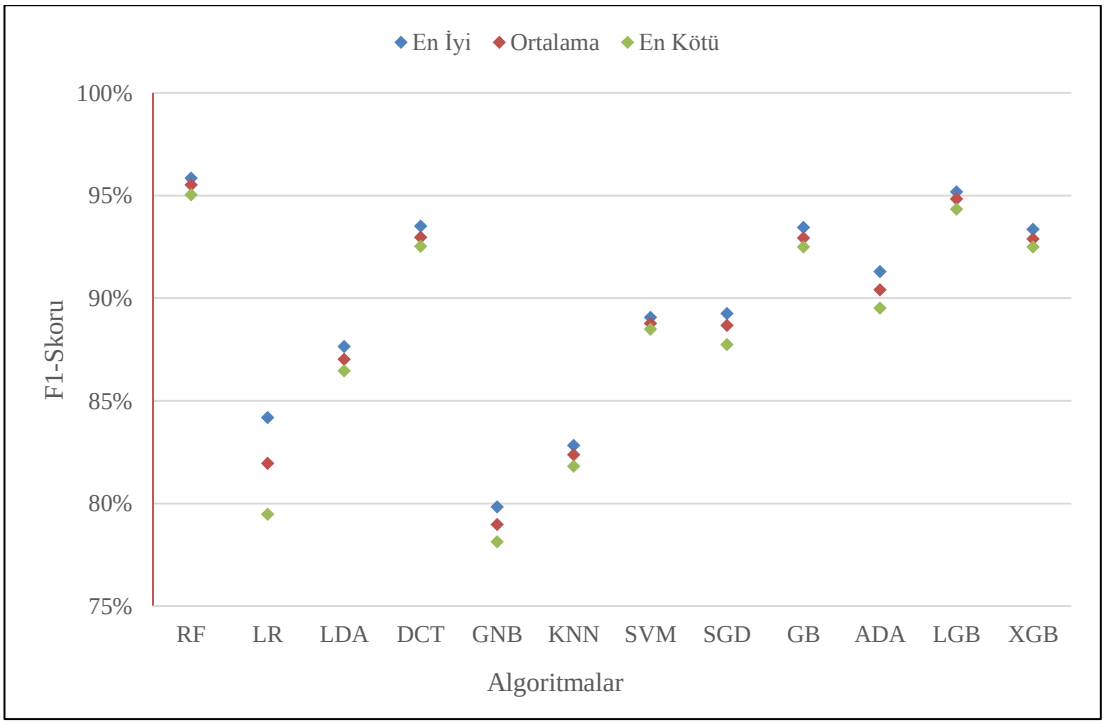
Şekil Ek B.11. Mendeley 2018 veri setinde bütün özellikler kullanılarak algoritmaların f1-skorlarının karşılaştırılması.



Şekil Ek B.12. Mendeley 2018 veri setinde verimli özellikler kullanılarak algoritmaların f1-skorlarının karşılaştırılması.



Şekil Ek B.13. Mendeley 2020 veri setinde bütün özellikler kullanılarak algoritmaların f1-skorlarının karşılaştırılması.



Şekil Ek B.14. Mendeley 2020 veri setinde verimli özellikler kullanılarak algoritmaların f1-skorlarının karşılaştırılması.

Çizelge Ek B.25. Mendeley 2018 veri setinde bütün özellikler kullanılarak algoritmaların eğitim ve test sürelerinin karşılaştırılması.

Algoritma	Eğitim Süresi	Test Süresi	Toplam
GNB	0,00460	0,00239	0,00699
SGD	0,05271	0,00359	0,05630
DCT	0,05410	0,00236	0,05646
LDA	0,05644	0,00252	0,05896
KNN	0,00486	0,24320	0,24806
LGB	0,23855	0,01309	0,25165
ADA	0,37536	0,03043	0,40579
RF	0,46527	0,10504	0,57030
XGB	0,67463	0,00948	0,68411
SVM	0,69190	0,00607	0,69797
LR	0,69663	0,00269	0,69932
GB	1,41041	0,00794	1,41834

Çizelge Ek B.26. Mendeley 2020 veri setinde bütün özellikler kullanılarak algoritmaların eğitim ve test sürelerinin karşılaştırılması.

Algoritma	Eğitim Süresi	Test Süresi	Toplam
GNB	0,14094	0,04558	0,18652
DCT	1,17918	0,02098	1,20016
SGD	1,55534	0,02965	1,58499
LDA	2,04798	0,01559	2,06358
LGB	3,35780	0,15963	3,51743
ADA	7,24690	0,54714	7,79403
RF	7,47591	0,53981	8,01572
LR	13,12102	0,01493	13,13596
XGB	15,87984	0,12714	16,00698
KNN	0,05892	21,09594	21,15486
GB	26,67947	0,10035	26,77983
SVM	46,75620	0,03281	46,78901

Çizelge Ek B.27. RF algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,439%	-0,131%	0,308%
	Ortalama	-0,656%	-0,119%	0,537%
	En Kötü	-0,716%	-0,210%	0,506%
Kesinlik	En İyi	-0,870%	-0,312%	0,558%
	Ortalama	-0,851%	-0,302%	0,548%
	En Kötü	-1,258%	-0,467%	0,791%
Duyarlılık	En İyi	-0,270%	-0,113%	0,157%
	Ortalama	-0,450%	-0,032%	0,418%
	En Kötü	-0,547%	0,011%	0,558%
F1-Skoru	En İyi	-0,427%	-0,190%	0,237%
	Ortalama	-0,651%	-0,167%	0,483%
	En Kötü	-0,713%	-0,295%	0,418%
Eğitim Süresi	Ortalama	-7,363%	-4,340%	3,023%
Test Süresi	Ortalama	-1,526%	-23,264%	-21,738%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.28. LR algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-2,826%	-2,211%	0,615%
	Ortalama	-2,681%	-1,405%	1,277%
	En Kötü	-2,466%	-0,275%	2,190%
Kesinlik	En İyi	-2,934%	0,646%	3,580%
	Ortalama	-2,812%	1,163%	3,975%
	En Kötü	-2,362%	3,501%	5,863%
Duyarlılık	En İyi	-2,509%	-12,040%	-9,532%
	Ortalama	-2,463%	-6,672%	-4,208%
	En Kötü	-2,896%	-2,430%	0,466%
F1-Skoru	En İyi	-2,805%	-4,419%	-1,613%
	Ortalama	-2,639%	-2,952%	-0,313%
	En Kötü	-2,520%	-0,876%	1,644%
Eğitim Süresi	Ortalama	-84,972%	-75,985%	8,987%
Test Süresi	Ortalama	-25,159%	-61,839%	-36,680%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.29. LDA algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-2,506%	-1,252%	1,254%
	Ortalama	-2,487%	-1,439%	1,048%
	En Kötü	-2,451%	-1,262%	1,189%
Kesinlik	En İyi	-1,887%	-2,672%	-0,785%
	Ortalama	-1,931%	-3,023%	-1,093%
	En Kötü	-1,728%	-2,844%	-1,116%
Duyarlılık	En İyi	-3,177%	-0,408%	2,768%
	Ortalama	-3,092%	-0,302%	2,790%
	En Kötü	-3,857%	-0,332%	3,525%
F1-Skoru	En İyi	-2,521%	-1,562%	0,959%
	Ortalama	-2,508%	-1,777%	0,731%
	En Kötü	-2,436%	-1,493%	0,942%
Eğitim Süresi	Ortalama	-59,284%	-90,352%	-31,069%
Test Süresi	Ortalama	0,634%	-62,353%	-62,987%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.30. DCT algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,616%	-0,071%	0,546%
	Ortalama	-0,534%	-0,154%	0,380%
	En Kötü	-0,800%	-0,174%	0,626%
Kesinlik	En İyi	-0,893%	-0,146%	0,746%
	Ortalama	-0,527%	-0,198%	0,329%
	En Kötü	-1,305%	-0,620%	0,685%
Duyarlılık	En İyi	-0,340%	-0,278%	0,062%
	Ortalama	-0,537%	-0,261%	0,276%
	En Kötü	-0,765%	-0,329%	0,436%
F1-Skoru	En İyi	-0,637%	-0,120%	0,516%
	Ortalama	-0,532%	-0,230%	0,302%
	En Kötü	-0,782%	-0,208%	0,574%
Eğitim Süresi	Ortalama	-51,710%	-45,687%	6,023%
Test Süresi	Ortalama	-22,860%	-60,669%	-37,809%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.31. GNB algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-1,864%	1,788%	3,652%
	Ortalama	-2,090%	3,039%	5,129%
	En Kötü	-1,775%	3,392%	5,166%
Kesinlik	En İyi	-0,903%	0,016%	0,919%
	Ortalama	-1,506%	-0,068%	1,437%
	En Kötü	-1,212%	-0,115%	1,097%
Duyarlılık	En İyi	-3,203%	8,219%	11,422%
	Ortalama	-3,415%	13,688%	17,102%
	En Kötü	-2,698%	15,474%	18,172%
F1-Skoru	En İyi	-2,584%	4,241%	6,825%
	Ortalama	-2,575%	7,541%	10,116%
	En Kötü	-2,185%	8,700%	10,885%
Eğitim Süresi	Ortalama	-29,145%	-77,914%	-48,769%
Test Süresi	Ortalama	-30,968%	-77,050%	-46,082%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.32. KNN algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	2,683%	-0,038%	-2,722%
	Ortalama	2,458%	-0,098%	-2,556%
	En Kötü	2,297%	-0,202%	-2,498%
Kesinlik	En İyi	4,324%	-0,171%	-4,495%
	Ortalama	4,120%	-0,374%	-4,495%
	En Kötü	4,698%	-0,528%	-5,226%
Duyarlılık	En İyi	0,145%	-0,066%	-0,211%
	Ortalama	0,074%	0,186%	0,112%
	En Kötü	0,226%	0,444%	0,218%
F1-Skoru	En İyi	2,325%	-0,057%	-2,383%
	Ortalama	2,121%	-0,093%	-2,213%
	En Kötü	1,913%	-0,150%	-2,064%
Eğitim Süresi	Ortalama	315,068%	-63,853%	-378,920%
Test Süresi	Ortalama	-60,600%	-8,926%	51,674%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.33. SVM algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-3,294%	-1,447%	1,846%
	Ortalama	-3,244%	-1,397%	1,847%
	En Kötü	-3,032%	-1,386%	1,646%
Kesinlik	En İyi	-3,968%	-3,060%	0,908%
	Ortalama	-4,016%	-3,410%	0,606%
	En Kötü	-4,078%	-3,296%	0,783%
Duyarlılık	En İyi	-1,458%	-0,364%	1,094%
	Ortalama	-2,202%	-0,169%	2,033%
	En Kötü	-2,191%	0,000%	2,191%
F1-Skoru	En İyi	-3,174%	-1,914%	1,260%
	Ortalama	-3,127%	-1,840%	1,287%
	En Kötü	-2,858%	-1,726%	1,133%
Eğitim Süresi	Ortalama	-20,222%	-66,946%	-46,724%
Test Süresi	Ortalama	23,819%	-76,896%	-100,716%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.34. SGD algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-2,357%	-1,039%	1,319%
	Ortalama	-3,195%	-0,971%	2,224%
	En Kötü	-4,650%	-0,164%	4,486%
Kesinlik	En İyi	-2,080%	-3,222%	-1,142%
	Ortalama	-3,912%	-3,004%	0,908%
	En Kötü	-5,849%	0,566%	6,416%
Duyarlılık	En İyi	-0,901%	-1,414%	-0,513%
	Ortalama	-2,217%	0,611%	2,828%
	En Kötü	-9,847%	5,572%	15,419%
F1-Skoru	En İyi	-2,311%	-1,434%	0,877%
	Ortalama	-3,097%	-1,199%	1,899%
	En Kötü	-5,505%	0,468%	5,973%
Eğitim Süresi	Ortalama	-45,487%	-75,811%	-30,325%
Test Süresi	Ortalama	-27,937%	-72,115%	-44,177%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.35. GB algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,542%	-0,165%	0,377%
	Ortalama	-0,664%	-0,296%	0,368%
	En Kötü	-0,823%	-0,348%	0,475%
Kesinlik	En İyi	-0,541%	-0,392%	0,149%
	Ortalama	-0,660%	-0,464%	0,196%
	En Kötü	-1,141%	-0,457%	0,684%
Duyarlılık	En İyi	-0,608%	-0,439%	0,169%
	Ortalama	-0,665%	-0,406%	0,260%
	En Kötü	-0,963%	-0,422%	0,541%
F1-Skoru	En İyi	-0,540%	-0,253%	0,287%
	Ortalama	-0,663%	-0,435%	0,228%
	En Kötü	-0,832%	-0,507%	0,325%
Eğitim Süresi	Ortalama	-41,687%	-44,041%	-2,354%
Test Süresi	Ortalama	-14,182%	-46,892%	-32,710%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.36. ADA algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,613%	-0,327%	0,286%
	Ortalama	-0,626%	-0,250%	0,376%
	En Kötü	-0,726%	-0,480%	0,246%
Kesinlik	En İyi	-0,480%	-0,412%	0,067%
	Ortalama	-0,914%	-0,346%	0,568%
	En Kötü	-1,681%	-0,866%	0,815%
Duyarlılık	En İyi	-0,068%	-0,464%	-0,396%
	Ortalama	-0,309%	-0,407%	-0,098%
	En Kötü	-0,417%	-0,569%	-0,152%
F1-Skoru	En İyi	-0,616%	-0,472%	0,144%
	Ortalama	-0,614%	-0,377%	0,237%
	En Kötü	-0,716%	-0,676%	0,039%
Eğitim Süresi	Ortalama	-29,646%	-46,661%	-17,015%
Test Süresi	Ortalama	-13,847%	-51,170%	-37,323%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.37. LGB algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,573%	-0,206%	0,367%
	Ortalama	-0,798%	-0,200%	0,598%
	En Kötü	-0,648%	-0,281%	0,367%
Kesinlik	En İyi	-0,869%	-0,111%	0,759%
	Ortalama	-0,877%	-0,384%	0,493%
	En Kötü	-0,997%	-0,277%	0,719%
Duyarlılık	En İyi	-0,537%	0,091%	0,628%
	Ortalama	-0,713%	-0,193%	0,520%
	En Kötü	-1,092%	-0,345%	0,747%
F1-Skoru	En İyi	-0,573%	-0,296%	0,277%
	Ortalama	-0,795%	-0,288%	0,507%
	En Kötü	-0,650%	-0,421%	0,230%
Eğitim Süresi	Ortalama	-39,316%	-62,712%	-23,396%
Test Süresi	Ortalama	-7,826%	-24,527%	-16,702%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

Çizelge Ek B.38. XGB algoritması ile elde edilen yüzdesel farkların karşılaştırılması.

Metrikler		Mendeley 2018	Mendeley 2020	Fark
Doğruluk	En İyi	-0,577%	-0,244%	0,333%
	Ortalama	-0,656%	-0,259%	0,397%
	En Kötü	-0,448%	-0,261%	0,187%
Kesinlik	En İyi	-0,562%	-0,447%	0,115%
	Ortalama	-0,583%	-0,398%	0,185%
	En Kötü	-0,780%	-0,168%	0,613%
Duyarlılık	En İyi	-0,473%	-0,586%	-0,113%
	Ortalama	-0,729%	-0,361%	0,368%
	En Kötü	-0,688%	-0,583%	0,104%
F1-Skoru	En İyi	-0,580%	-0,358%	0,222%
	Ortalama	-0,656%	-0,379%	0,277%
	En Kötü	-0,451%	-0,411%	0,040%
Eğitim Süresi	Ortalama	-52,715%	-71,801%	-19,086%
Test Süresi	Ortalama	-22,375%	-30,473%	-8,098%
Bellek Kullanımı	Ortalama	-72,951%	-82,883%	-9,932%

ÖZGEÇMİŞ

Ahmet Selim KÜÇÜKKARA, 2016 yılında Karabük Anadolu Öğretmen Lisesinden mezun oldu. 2020 yılında Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği bölümünden şeref öğrencisi olarak, Anadolu Üniversitesi Kamu Yönetimi bölümünden onur öğrencisi olarak mezun oldu. 2020 yılında Karabük Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisansa başladı. 2021 yılında Zonguldak Bülent Ecevit Üniversitesi Bilgisayar Yazılımı Anabilim Dalı'nda araştırma görevlisi olarak göreve başladı ve halen aynı yerde çalışmaya devam etmektedir.