Black Sea Journal of Engineering and Science
doi: 10.34248/bsengineering.1792907

Open Access Journal
e-ISSN: 2619 – 8991

Araştırma Makalesi (Research Article)
Cilt 9 - Sayı 1: 41-51 / Ocak 2026
(Volume 9 - Issue 1: 41-51 / January 2026)

NASA METRİCS DATA PROGRAM VERİ SETİ ÜZERİNDE YENİDEN ÖRNEKLEME YÖNTEMLERİNİN YAZILIMDA HATA TAHMİNİ BAŞARIMINA ETKİSİ

Emre Can YILMAZ[1]*, Recai OKTAŞ[2]

[1]*Ondokuz Mayis University, Institute of Graduate Studies, Computational Sciences Department, 55139, Samsun, Türkiye*
[2]*Ondokuz Mayis University, Engineering Faculty, Computer Engineering Department, 55139, Samsun, Türkiye*

Özet: NASA Metrics Data Program (MDP), NASA tarafından yürütülen ve çeşitli yazılım projelerinden elde edilen metrikleri ve hata bilgilerini içeren, araştırmalarda yaygın olarak kullanılan bir veri deposudur. Çok sayıda alt kümeye sahip NASA MDP verilerinde, çalışmanın kapsamını sınırlamak için uygun bir alt veri kümesinin seçilmesi uygun olacaktır. Bu amaçla, büyük ve dengesiz veriler içermesi nedeniyle JM1 alt kümesi tercih edilmiştir. JM1 üzerinde yeniden örnekleme tekniklerinin makine öğrenmesi modellerinin başarımına etkileri incelenmiştir. Bu kapsamda SMOTE, RUS, ROSE, ADASYN, Tomek Links, ENN, Near Miss ve Borderline-SMOTE gibi aşırı örnekleme, eksik örnekleme ve hibrit teknikler; Naive Bayes (NB), Destek Vektör Makineleri (DVM), Lojistik Regresyon (LR), Karar Ağacı (KA) ve Rastgele Orman (RO) sınıflandırıcıları ile birlikte değerlendirilmiştir. Sonuçlar, yeniden örnekleme uygulanmayan modellerin, özellikle azınlık sınıfı olarak tanımlanan hatalı modülleri tanımada düşük başarım sergilediğini; buna karşın genel doğruluk metriklerinde yanıltıcı şekilde yüksek değerler elde edebildiğini göstermektedir. Öte yandan, SMOTE+ENN gibi hibrit ve ROSE gibi aşırı örnekleme yöntemlerinin, özellikle Rastgele Orman ve Naive Bayes sınıflandırıcılarıyla birlikte kullanıldığında, AUC ve F1-ölçütü gibi dengesizliğe duyarlı metriklerde anlamlı iyileşmeler sağladığı gözlemlenmiştir. En iyi sonuç, SMOTE+ENN yöntemiyle birlikte kullanılan Rastgele Orman modeliyle elde edilmiş; 0,9350 doğruluk, 0,9837 AUC ve hatasız/hatalı modüller için sırasıyla 0,9126/0,9483 F1-ölçütü değerlerine ulaşılmıştır. Bu bulgular, yazılım hata tahmininde sınıf dengesizliğiyle mücadelede uygun yeniden örnekleme stratejilerinin seçiminin ve dengesizliğe duyarlı metriklerle değerlendirme yapılmasının önemini ortaya koymaktadır.

Anahtar Kelimeler: Yazılımda hata tahmini, Dengesiz veri seti, Yeniden örneklendirme yöntemleri, Ampirik yazılım mühendisliği

The Impact of Resampling Techniques on the Performance of Software Defect Prediction Using the NASA Metrics Data Program Dataset

Abstract: The NASA Metrics Data Program (MDP) is a widely used repository containing software metrics and defect data from various NASA projects. To narrow the study's scope, the JM1 subset was selected due to its large size and class imbalance. The effects of resampling techniques on the performance of machine learning models were examined using the JM1 dataset. We evaluate several oversampling, undersampling, and hybrid resampling methods-including SMOTE, RUS, ROSE, ADASYN, Tomek Links, ENN, Near Miss, and Borderline-SMOTE-in conjunction with classifiers such as Naive Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF). Our results indicate that models trained without any resampling often perform poorly in identifying faulty modules (the minority class), despite achieving deceptively high values in overall accuracy metrics. Conversely, hybrid methods such as SMOTE+ENN and oversampling techniques like ROSE significantly improve performance-particularly when used with Random Forest and Naive Bayes classifiers-based on metrics sensitive to class imbalance such as AUC and F1-score. The best performance is observed when combining Random Forest with SMOTE+ENN, achieving 0.9350 accuracy, an AUC of 0.9837, and F1-scores of 0.9126 and 0.9483 for non-defective and defective modules, respectively. Consequently, for imbalanced software defect datasets, the selection of appropriate resampling methods and the use of metrics truly reflecting real-world performance are critically important.

Keywords: Software defect prediction, Imbalanced dataset, Resampling methods, Empirical software engineering

***Sorumlu yazar (Corresponding author):** Ondokuz Mayis University, Institute of Graduate Studies, Computational Sciences Department, 55139, Samsun, Türkiye
E mail: emrecan@omu.edu.tr (E C. YILMAZ)
Emre Can YILMAZ https://orcid.org/0000-0003-4365-9131
Recai OKTAŞ https://orcid.org/0000-0003-3282-3549

Gönderi: 29 Eylül 2025l
Kabul: 07 Kasım 2025
Yayınlanma: 15 Ocak 2026
Received: September 29, 2025
Accepted: November 07, 2025
Published: January 15, 2026

Cite as: Yılmaz, E. C., & Oktaş, R. (2026). The impact of resampling techniques on the performance of software defect prediction using the NASA MDP dataset. *Black Sea Journal of Engineering and Science, 9*(1), 41-51.

This work is licensed (CC BY-NC 4.0) under Creative Commons Attribution 4.0 International License

1. Giriş

Yazılımda hatalar kaçınılmazdır. Bu hataların, son kullanıcıya ulaşmadan önce yazılım geliştirme yaşam döngüsü içerisinde tespit edilip giderilmesi isten hedeftir. Yazılımda hata tahmini (YHT), yazılım geliştirme yaşam döngüsü boyunca yazılım kalitesinin artırılmasında ve kaynak tahsisinin optimize edilmesinde kritik bir rol oynamaktadır. YHT, hataya eğilimli modüllerin erken tespiti sayesinde, ekiplerin test çalışmalarını önceliklendirmesine olanak tanır, böylece geç aşamada gerçekleşen hata tespiti ve düzeltmelere ilişkin maliyetleri azaltır (Mabayoje vd., 2019; Sơn vd., 2019; Li vd., 2018). Bu proaktif yaklaşım, yalnızca maliyeti azaltmakla kalmaz, aynı zamanda daha iyi risk değerlendirmesi ve kalite kontrolü sağlayarak etkin proje yönetimine de yardımcı olur (Wang vd., 2020; Hryszko vd., 2018).

Yazılımda hata tahmininin doğruluğunu ve verimliliğini artırma çabaları, özellikle makine öğrenmesi tekniklerini temel alan gelişmiş tahmin modellerinin kullanımını ön plana çıkarmıştır (Badvath vd., 2022; Deng vd., 2020). Bu modeller, geçmiş proje verilerini ve çeşitli yazılım metriklerini analiz ederek, yazılım bileşenlerini 'hatalı' veya 'hatasız' olarak sınıflandırma konusunda dikkate değer bir yetenek sergilemektedir (Ren vd., 2014; Balogun vd., 2019). Yapılan bu sınıflandırma, test kaynaklarının en yüksek riskli modüllere odaklanmasını sağlayan hedefe yönelik test stratejilerinin uygulanmasını kolaylaştırır. Dolayısıyla, bu gelişmiş YHT yaklaşımları, yazılım hatalarının ekonomik yükünü azaltmanın yanı sıra, bakım süreçlerini daha verimli hale getirerek ve test için ayrılan insan gücünü optimize ederek genel proje üretkenliğine önemli katkılar sunar (Tomar vd., 2016; Hall vd., 2012).

YHT alanında kullanılan veri setlerinin önemli bir özelliği, sıklıkla sınıf dengesizliği sergilemeleridir. Hatalı modüller genellikle hatasız modüllere kıyasla çok daha az sayıda bulunur. Bu durum, denetimli öğrenme algoritmaları için önemli zorluklar yaratır. Dengesiz veri setleri, modellerin çoğunluk sınıfına doğru yanlı olmasına ve özellikle kritik olan azınlık sınıfını (hatalı modüller) tespit etmede düşük başarım göstermesine neden olabilir. Bu yanlılık, modelin genelleme yeteneğini ciddi şekilde etkileyebilir (Zheng vd., 2022; Najeeb vd., 2024; Abdelmoumin vd., 2023; Loukili vd., 2024). Özellikle lojistik regresyon ve destek vektör makineleri gibi bazı sınıflandırıcıların, veri dengesizliğinin olumsuz etkilerine karşı daha hassas olduğu ve bu durumun tahmin doğruluğunu düşürebildiği rapor edilmiştir (Zheng vd., 2022; Bui, 2023; Mohapatra, 2024).

Bu sorunların üstesinden gelmek ve modellerin başarımını iyileştirmek amacıyla literatürde çeşitli stratejiler önerilmiştir. Bu stratejiler genel olarak veri seviyesinde yaklaşımlar ve algoritma seviyesinde yaklaşımlar olarak ikiye ayrılabilir. Veri artırma (data augmentation), aşırı örnekleme (oversampling) ve eksik örnekleme (undersampling) gibi veri seviyesi teknikler, sınıf dağılımını dengeleyerek modelin azınlık sınıfını daha etkin bir şekilde tanımasını sağlamayı hedefler (Alam vd., 2022; Loukili vd., 2024; Byeon, 2021; Ayoub vd., 2023; Abdelmoumin vd., 2023; Patel vd., 2017). Algoritma seviyesinde ise, adaptif güçlendirme (adaptive boosting) veya maliyet duyarlı öğrenme (cost-sensitive learning) gibi yöntemler ve sınıf düzeltme kaybı (class rectification loss) gibi gelişmiş teknikler, modelin kendisini dengesiz senaryolarda daha iyi başarım gösterecek şekilde adapte etmesini amaçlar (Taherkhani vd., 2020; Huang vd., 2020). Dengesiz veri setlerinin yarattığı zorlukların ele alınması, çeşitli uygulamalarda doğru tahminler yapabilen sağlam makine öğrenmesi modelleri geliştirmek için kritik öneme sahiptir (Meysami vd., 2023; Zhang vd., 2023).

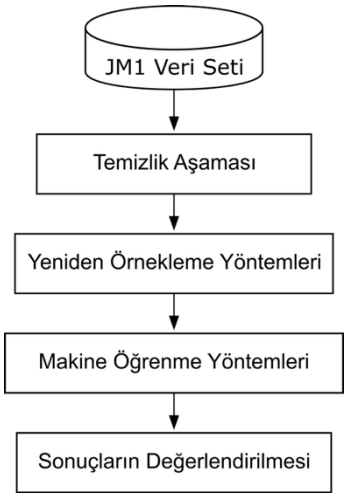
Aşırı örnekleme teknikleri, veri setini dengelemek amacıyla azınlık sınıfına ait yeni sentetik örnekler üretir. YHT bağlamında, Khuat vd. (2020), yaptıkları deneysel çalışmada, farklı örnekleme metotlarının sınıflandırıcı başarımını nasıl iyileştirdiğini incelemişler ve özellikle aşırı örneklemenin sınıflandırma doğruluğunu anlamlı ölçüde artırabildiğini göstermişlerdir. Benzer şekilde, Li vd. (2022), YHT veri setlerinin dengesiz doğasıyla mücadele etmek için parçacık sürüsü optimizasyonu (PSO) ile adaptif güçlendirmeyi (AdaBoost) birleştiren akıllı bir füzyon algoritması önermişler ve aşırı örnekleme stratejilerinin model başarımını artırmadaki etkinliğini ortaya koymuşlardır.

Diğer yandan, eksik örnekleme teknikleri, dengeli bir veri seti elde etmek amacıyla çoğunluk sınıfındaki örneklerin sayısını azaltmayı hedefler. Rahardian vd. (2020), yazılım hata tahmininde sınıf dağılımını iyileştirmek için eksik örnekleme stratejilerinin kullanımını ele almışlardır. Çalışmalarında, eksik örneklemenin, çoğunluk sınıfının model üzerindeki baskın etkisini azaltarak tahmin gücünü artırmada etkili olabileceğini belirtmişlerdir. Ayrıca, Kou vd. (2012), eksik örnekleme tekniklerini de içeren sınıflandırıcı değerlendirme yöntemlerini araştırmış ve bu tekniklerin hata tahminlerinin doğruluğunu iyileştirmedeki etkinliğini göstermişlerdir.

Mevcut literatür hem aşırı örnekleme hem de eksik örnekleme tekniklerinin YHT'nde sınıf dengesizliği ile mücadelede değerli yaklaşımlar olduğunu ortaya koymaktadır. Ancak, farklı yeniden örnekleme yöntemlerinin, belirli veri setleri (bu çalışmada JM1) ve farklı sınıflandırıcılar bağlamındaki karşılaştırmalı başarımları üzerinde daha fazla araştırmaya ihtiyaç duyulmaktadır. Bu çalışma, bu boşluğu doldurmaya yönelik bir adım atmayı ve JM1 veri seti üzerinde belirtilen yöntemlerin etkinliğini değerlendirmeyi amaçlamaktadır.

Bu bağlamda çalışmamızın temel amacı, YHT alanında sıkça karşılaşılan veri dengesizliği sorununun üstesinden gelmek için kullanılan farklı yeniden örnekleme yöntemlerinin etkisini, çalışmalarda sıklıkla kullanılan JM1 veri seti üzerinde incelemektir. Ayrıca, bu çalışma sınıf tabanlı yazılım hata tahmini problemine odaklanmaktadır. Bu kapsamda kullanılan veri seti, her bir yazılım modülünü veya sınıfını, hatalı ya da hatasız

olarak etiketlenmiş bağımsız örnekler şeklinde temsil etmektedir. Çalışma kapsamında, Sentetik Azınlık Aşırı Örnekleme Tekniği (Synthetic Minority Over-sampling Technique - SMOTE) gibi aşırı örnekleme ve Rastgele Azınlık Örneklemesi (Random Undersampling - RUS) gibi eksik örnekleme tekniklerinin, Naive Bayes (NB), Rastgele Orman (RO) ve Destek Vektör Makineleri (DVM) gibi yaygın makine öğrenmesi algoritmalarının hata tahmini başarımı üzerindeki etkileri karşılaştırmalı olarak analiz edilmiştir. Bu analizle, JM1 veri seti özelinde hangi yeniden örnekleme stratejisinin hangi sınıflandırıcı ile daha etkin sonuçlar verdiğinin belirlenmesi hedeflenmektedir.



Şekil 1. Çalışma akışı.

2. Materyal ve Yöntem

Bu bölümde, çalışmada kullanılan veri seti, uygulanan yeniden örnekleme teknikleri, değerlendirme metrikleri ve kullanılan makine öğrenmesi algoritmaları detaylandırılmaktadır. Çalışmada yer alan tüm makine öğrenmesi modellerinin geliştirilmesi, veri ön işleme adımlarının uygulanması ve yeniden örnekleme algoritmalarının yürütülmesi, Python programlama dilinin 3.12 sürümü ile scikit-learn kütüphanesinin 1.6.1 sürümü kullanılarak gerçekleştirilmiştir. Çalışmamızın baştan sona akışı Şekil 1'de gösterilmiştir.

Deneylerin metodolojik tutarlılığını ve tekrarlanabilirliğini sağlamak amacıyla, hem Bölüm 2.2'de açıklanan makine öğrenmesi sınıflandırıcıları hem de Bölüm 2.3'te ele alınan yeniden örnekleme teknikleri için standart bir yaklaşım benimsenmiştir. Yeniden örnekleme işlemleri için, Scikit-learn ekosistemiyle uyumlu çalışan Imbalanced-learn kütüphanesi de kullanılmıştır.

Çalışmanın odak noktası hiperparametre optimizasyonu olmadığından, farklı yeniden örnekleme stratejilerinin varsayılan modeller üzerindeki saf etkisini karşılaştırabilmek amacıyla, tüm sınıflandırıcılar ve yeniden örnekleme algoritmaları kütüphanelerin varsayılan hiperparametreleri ile çalıştırılmıştır. Ek olarak, tüm deneylerde algoritmaların dahili rastgeleliği kontrol altına almak ve sonuçların tutarlılığını güvence altına almak için "random_state" parametresi sabit bir değere (42) ayarlanmıştır.

2.1. Veri Seti

NASA Metrik Veri Programı (Metrics Data Program - MDP) veri setleri, YHT araştırmalarının temel taşlarından biri olarak geniş çapta kabul görmektedir. Çeşitli NASA projelerinden türetilen bu veri setleri, statik kod metrikleri ve hata verilerini sağlayarak, hataya eğilimli yazılım modüllerini tanımlamayı hedefleyen tahmin modellerinin geliştirilmesine ve değerlendirilmesine olanak tanır.

MDP koleksiyonu, her biri farklı NASA yazılım projelerinden elde edilen verileri temsil eden CM1, JM1, KC1, KC2 ve PC1 gibi iyi bilinen alt kümeleri içermektedir. Bu veri setleri, yazılım modüllerindeki hataların varlığını gösteren etiketlerin yanı sıra, satır sayısı, döngüsel karmaşıklık, Halstead karmaşıklık metrikleri, modül bağlantısı, yorum satırı oranı ve değişken/operatör yoğunluğu gibi 22 farklı statik kod özelliğini (özniteliği) içermektedir. Bu metrikler, yazılımın yapısal karmaşıklığı ve bakım zorluğunu nicel olarak ifade etmeleri nedeniyle hata tahmini modellerinde temel belirleyiciler olarak kullanılmaktadır (McCabe, 1976; Halstead, 1977; Menzies vd., 2010).

NASA MDP veri setlerinin popülerliğinin başlıca nedenlerinden biri, tekrarlanabilirliği teşvik eden ve çalışmalar arasında karşılaştırmalı değerlendirmelere olanak sağlayan kamuya açık olmalarıdır. Dahası, bu veri setleri, araştırmacıların çeşitli modelleme tekniklerini incelemesine olanak tanıyan zengin bir metrik çeşitliliği sunar. Alanda yaygın olarak kullanılmaları, bu metrikleri yeni yaklaşımların doğrulanması için standart bir referans noktası haline getirmiştir.

MDP veri setlerinin dikkat çeken bir diğer önemli özelliği, hatasız modüllerin hatalı olanlardan önemli ölçüde daha fazla olduğu doğal sınıf dengesizliğidir. Gerçek dünya yazılım sistemlerini yansıtan bu karakteristik özellik, aynı zamanda önemli bir zorluk teşkil etmekte ve aşırı örnekleme teknikleri ile örneklem tabanlı filtreleme gibi yenilikçi veri ön işleme yöntemlerinin geliştirilmesini teşvik etmektedir. Bu çalışmanın odak noktası da tam olarak bu dengesizlik sorununa yönelik yeniden örnekleme yöntemlerinin incelenmesidir.

MDP veri setleri, makine öğrenmesi ve derin öğrenme teknikleri için bir test ortamı görevi görerek YHT alanının ilerlemesinde kritik bir rol oynamıştır. Bu veri setlerinden yararlanan çalışmalar, aşırı uyum ve projeler arası tahmin gibi önemli zorlukları ele almış, hata tahmin modellerinin anlaşılmasına ve iyileştirilmesine ciddi katkılarda bulunmuştur (Goyal, 2022; Pandey vd., 2020; Arya vd., 2021).

Özetle, NASA MDP veri setleri, erişilebilirlikleri, kapsamlı metrikleri ve gerçek dünya yazılım zorluklarıyla uyumları nedeniyle YHT araştırmalarında hayati bir kaynaktır. Sürekli kullanımları, tahminsel metodolojilerin değerlendirilmesi ve ilerletilmesindeki önemlerini vurgulamaktadır. Çalışmada kullanılan JM1 veri seti tek bir sürüm (single-release) verisi olup, içerisinde 22

nitelik, 2106 hatalı modül (azınlık sınıfı) ve 8879 hatasız modül (çoğunluk sınıfı) bulunmaktadır.

2.2. Makine Öğrenme Algoritmaları

Bu çalışmada, JM1 veri seti üzerinde yazılım hata tahmini başarımı, farklı yeniden örnekleme teknikleri ile birlikte çeşitli makine öğrenmesi algoritmaları kullanılarak değerlendirilmiştir. Analizde kullanılan temel sınıflandırıcı algoritmalar aşağıda tanımlanmaktadır.

2.2.1. Naive Bayes

Naive Bayes (NB) sınıflandırıcıları, Bayes teoreminin, öznitelikler arasında güçlü bağımsızlık varsayımları yapılarak uygulanmasına dayanan olasılıksal bir sınıflandırıcı ailesidir. Basitlikleri ve hesaplama verimlilikleri, NB sınıflandırıcılarını yazılım mühendisliği dahil pek çok alanda değerli kılmaktadır; bu alanda özellikle güvenlik açığı sınıflandırması ve hata tahmini gibi görevlerde uygulamaları bulunmaktadır. İlgili literatür, NB modellerinin yazılım projelerindeki teknik borç sorunlarını sınıflandırmada ve yazılım güvenlik açıklarını tahmin etmede etkin bir şekilde kullanıldığını göstermektedir (Li vd., 2022; Fu vd., 2024). NB sınıflandırıcısının uygulama kolaylığı, bu sınıflandırıcıyı çeşitli makine öğrenmesi görevleri için sıklıkla tercih edilen bir seçenek haline getirmektedir (Agrawal vd., 2020).

2.2.2. Destek vektör makineleri

Destek Vektör Makineleri (DVM), güçlü matematiksel temellere dayanan bir makine öğrenmesi algoritmasıdır. Doğrusal olmayan örüntüleri ayırt edebilme, yüksek boyutlu uzaylarda çalışabilme ve gürültülü verilere karşı görece dayanıklı olma gibi özellikleri, DVM'leri YHT ve güvenilirlik kestirimi dahil olmak üzere çeşitli yazılım mühendisliği uygulamaları için uygun bir yöntem kılmaktadır. DVM'lerin ölçeklenebilirlik gibi bazı kısıtlamaları bulunmakla birlikte, devam eden araştırmalar bu kısıtlamaları gidermeye ve algoritmanın yorumlanabilirliğini artırmaya odaklanmakta ve sayede algoritmanın yazılım mühendisliği alanındaki güncelliğini korumasını sağlamaktadır.

2.2.3. Lojistik regresyon

Lojistik Regresyon (LR), bir bağımlı değişkenin belirli bir kategoriye ait olma olasılığını bir veya daha fazla bağımsız değişken aracılığıyla modellemek amacıyla kullanılan temel bir istatistiksel ve makine öğrenimi yöntemidir. Yazılım mühendisliği bağlamında, belirli metrikler veya faktörlerin bir sonucun olasılığı üzerindeki etkisini anlamak ve kategorik bir değişkeni (ör. yazılım hatasının varlığı/yokluğu, modülün risk durumu) tahmin etmek için yaygın olarak kullanılmaktadır. Yöntemin görece basitliği, katsayılarının yorumlanabilirliği ve özellikle ikili sınıflandırma problemlerindeki etkinliği, yazılım analizinde belirli sınıflandırma ve risk modelleme görevleri için sıkça tercih edilmesini sağlamaktadır. Lojistik regresyon, temel olarak ikili ve çoklu sınıflandırma problemleri için tasarlanmıştır. Yazılım hata tahmini, müşteri kaybı analizi gibi birçok yazılım mühendisliği problemi doğal olarak sınıflandırma görevi olduğundan, LR bu alanlarda temel bir modelleme ve tahmin aracı olarak kabul edilir ve genellikle daha karmaşık yöntemler için bir referans noktası görevi görür.

2.2.4. Karar ağacı

Karar Ağaçları (KA), sınıflandırma ve regresyon görevleri için kullanılan parametrik olmayan, kural tabanlı bir makine öğrenmesi yöntemidir. Veri setindeki özniteliklere dayalı olarak bir dizi karar kuralını hiyerarşik bir ağaç yapısında temsil eder. Ağacın her iç düğümü bir öznitelik üzerinde yapılan bir testi, her dal testin sonucunu ve her yaprak düğüm ise bir sınıf etiketini veya regresyon değerini gösterir. Bu yapısal özellik, karar ağaçlarının kolayca yorumlanabilmesine olanak tanır ve yazılım geliştirme efor tahmini gibi alanlarda kullanımlarını kolaylaştırır. Tekil karar ağaçları yorumlanabilirlik avantajı sunsa da genellikle aşırı uyum eğilimi gösterirler ve YHT'nde sıklıkla Rastgele Orman veya Gradyan Artırma ("Gradient Boosting") gibi topluluk yöntemlerinin temelini oluştururlar.

2.2.5. Rastgele orman

Rastgele Orman (RO), sınıflandırma ve regresyon problemleri için yaygın olarak kullanılan bir topluluk öğrenmesi (ensemble learning) yöntemidir. Leo Breiman tarafından 2001'de geliştirilen bu yöntem, yüksek tahmin doğruluğu sergilemesi ve çok sayıda öznitelik içeren büyük veri setlerini etkin bir şekilde işleyebilmesi nedeniyle tercih edilmektedir. RO, eğitim aşamasında çok sayıda karar ağacı oluşturur ve sınıflandırma için ağaç tahminlerinin modunu (en sık çıkan sınıf), regresyon için ise ortalamasını alarak nihai çıktıyı üretir. Özellikle öznitelik sayısının örneklem sayısından fazla olduğu durumlarda gösterdiği etkinlik, RO'ı yazılım hata tahmini gibi çeşitli uygulamalar için elverişli kılmaktadır.

2.3. Veri Dengeleme Algoritmaları

Bu çalışmada ele alınan JM1 veri setinin Bölüm 2.1'de belirtilen doğal sınıf dengesizliği özelliğinin, Bölüm 2.2'de tanımlanan makine öğrenmesi modellerinin başarımı üzerindeki potansiyel olumsuz etkilerini azaltmak amacıyla çeşitli yeniden örnekleme teknikleri uygulanmıştır. Bu bölümde, kullanılan temel veri dengeleme algoritmaları açıklanmaktadır.

2.3.1. Sentetik azınlık aşırı örnekleme tekniği

Sentetik Azınlık Aşırı Örnekleme Tekniği (Synthetic Minority Over-sampling Technique - SMOTE), veri setlerindeki sınıf dengesizliği sorununu gidermek için yaygın olarak kullanılan bir aşırı örnekleme yöntemidir. Mevcut örneklemleri basitçe kopyalamak yerine, azınlık sınıfı için sentetik örnekler üreterek çalışır. Bu yaklaşım, rastgele aşırı örnekleme ile ilişkili aşırı uyum sorunlarının hafifletilmesine yardımcı olur ve eksik örnekleme tekniklerinde kaybolabilecek değerli bilgilerin korunmasını sağlar (Byeon, 2021; Fernández vd., 2018).

2.3.2. Rastgele eksik örnekleme

Rastgele Eksik Örnekleme (Random Undersampling - RUS), çoğunluk sınıfından rastgele örneklemler çıkararak veri setindeki sınıf dengesizliğini gidermeyi amaçlayan basit bir eksik örnekleme tekniğidir. Bu yöntem, daha

dengeli bir veri seti oluşturarak makine öğrenmesi modellerinin azınlık sınıfının özelliklerini daha iyi öğrenmesine olanak tanımayı ve böylece model başarımını iyileştirmeyi hedefler (Fan, 2024; Guo vd., 2019; Jeon vd., 2020). RUS, basit ve hesaplama açısından verimli olmasına rağmen, potansiyel olarak değerli bilgileri içeren örneklemlerin atılması riskini taşır; bu durum önemli ölçüde veri kaybına yol açabilir ve nihayetinde model doğruluğunu olumsuz etkileyebilir (Guo vd., 2019; Jeon vd., 2020).

2.3.3. Rastgele aşırı örnekleme

Rastgele Aşırı Örnekleme Örnekleri (Random Over-Sampling Examples - ROSE), sentetik örnekler üreterek veri setlerindeki sınıf dengesizliğini gidermek için tasarlanmış istatistiksel bir tekniktir. Mevcut azınlık sınıfı örneklemlerini yalnızca çoğaltan geleneksel aşırı örnekleme yöntemlerinden farklı olarak ROSE, düzgünleştirilmiş bir bootstrap (smoothed bootstrap) yaklaşımına dayalı yeni örnekler oluşturur; bu da üretilen örneklemlerin çeşitliliğini artırır (Lunardon vd., 2014). Bu yöntem, özellikle bir sınıfın önemli ölçüde az temsil edildiği ikili sınıflandırma problemlerinde faydalıdır, çünkü daha dengeli bir eğitim seti sağlayarak makine öğrenmesi modellerinin başarımını iyileştirmeye yardımcı olur (Zhang vd., 2019).

2.3.4. Adaptif sentetik örnekleme

Adaptif Sentetik Örnekleme (Adaptive Synthetic - ADASYN), veri setlerindeki sınıf dengesizliğini gidermek için tasarlanmış gelişmiş bir aşırı örnekleme tekniğidir. SMOTE prensiplerine dayanarak, ADASYN azınlık sınıfı için sentetik örnekleri daha adaptif bir şekilde üretir. Spesifik olarak ADASYN, azınlık sınıfı örneklemlerinin öğrenilmesinin daha zor olduğu bölgelerde daha fazla sentetik örnek oluşturmaya odaklanır; böylece öznitelik uzayında sınıflandırıcılar için daha zorlayıcı olan alanları etkin bir şekilde hedefler (Zakariah vd., 2023; Rendón vd., 2020).

2.3.5. TOMEK bağlantıları

TOMEK Bağlantıları (Tomek Links), öncelikle dengesiz veri setleri bağlamında kullanılan bir veri temizleme tekniğidir. Farklı sınıflara ait olan ve öznitelik uzayında birbirine en yakın olan örneklem çiftlerini ("Tomek bağlantıları" olarak adlandırılır) tanımlar. Böyle bir çift bulunduğunda, genellikle çoğunluk sınıfına ait olan örneklem kaldırılır; bu sayede sınıflar arasındaki ayrım artırılır ve veri setindeki gürültü azaltılır (Branco vd., 2016; McKendrick vd., 2019; Aljawazneh vd., 2021). Bu yöntem, yanlış sınıflandırmaya yol açabilecek belirsiz örneklemleri ortadan kaldırmada özellikle etkilidir ve böylece dengesiz veriler üzerinde eğitilen sınıflandırıcıların başarımını iyileştirir (Boschi vd., 2014).

TOMEK Bağlantıları, tek başına bir teknik olarak veya SMOTE gibi aşırı örnekleme yöntemleriyle birleştirilerek SMOTE-Tomek olarak bilinen hibrit bir yaklaşım oluşturmak için de uygulanabilir. Bu kombinasyon, yalnızca azınlık sınıfı için sentetik örnekler üretmekle kalmaz, aynı zamanda sınıflar arasında gürültü veya örtüşme yaratabilecek örneklemleri kaldırarak veri setini temizler (Wu vd., 2021).

2.3.6. Düzenlenmiş en yakın komşular

Düzenlenmiş En Yakın Komşular (Edited Nearest Neighbor - ENN), özellikle dengesiz sınıflandırma problemleri bağlamında, eğitim veri setlerinin kalitesini artırmak için kullanılan bir veri ön işleme tekniğidir. ENN, gürültülü ve yanlış etiketlenmiş örneklemleri kaldıran bir süreç aracılığıyla veri setini iyileştirerek çalışır. Spesifik olarak, veri setindeki her bir örneklemi inceler ve en yakın komşularını dikkate alır; eğer bu komşuların çoğunluğu farklı bir sınıfa aitse, örneklem gürültülü kabul edilir ve eğitim setinden çıkarılır (Fu vd., 2022).

Bu yöntem, sınıf dengesizliğinin yanlış sınıflandırmaya yol açabileceği senaryolarda özellikle değerlidir, çünkü sınıflar arasındaki karar sınırlarının netleşmesine yardımcı olur. Öğrenme algoritmasını karıştırabilecek örneklemleri ortadan kaldırarak, ENN makine öğrenmesi modellerinin genel sınıflandırma doğruluğunu iyileştirebilir (Fu vd., 2022). ENN, eğitim için daha dengeli ve daha temiz bir veri seti oluşturmak amacıyla sıklıkla SMOTE gibi diğer yeniden örnekleme teknikleriyle birlikte kullanılır (Fu vd., 2022).

2.3.7. Near miss

Near Miss, makine öğrenmesi veri setlerindeki sınıf dengesizliğini gidermek için kullanılan bir veri yeniden örnekleme (eksik örnekleme) tekniğidir. Bu yöntem, azınlık sınıfı örneklemlerine en yakın olan çoğunluk sınıfı örneklemlerini tanımlayıp korumaya odaklanır; böylece mevcut örneklemleri basitçe çoğaltmadan azınlık sınıfının temsilini göreceli olarak artırır. Near Miss tekniği, azınlık sınıfı örneklemlerine en yakın olan çoğunluk sınıfı örneklemlerini seçerek çalışır ve çoğunluk sınıfının bilgilendirici özelliklerini korurken etkin bir şekilde daha dengeli bir veri seti oluşturur (Mqadi vd., 2021; Elsobky, 2023). (Not: Near Miss'in farklı versiyonları bulunmaktadır ve seçilen çoğunluk sınıfı örneklemleri, azınlık sınıfı örneklemlerine olan ortalama uzaklıklarına veya en yakın/en uzak komşularına göre belirlenebilir.)

2.3.8. Sınır Çizgisi Örneklemesi

Sınır çizgisi örneklemesi (Borderline), veri yeniden örneklemede sınıf dengesizliğini gidermek için kullanılan, özellikle sınıflar arasındaki karar sınırına yakın bulunan örneklemlere odaklanan bir tekniktir. Bu konsept, temel olarak, azınlık sınıfının özellikle sınır çizgisi örneklemlerini hedefleyerek geleneksel SMOTE yaklaşımını geliştiren Borderline-SMOTE yönteminde uygulanmaktadır.

Borderline-SMOTE'da, sentetik örnekler yalnızca çoğunluk sınıfına yakın olan azınlık sınıfı örneklemleri için üretilir. Bu, bu sınır çizgisi örneklemlerinin en yakın komşularını belirleyerek ve onları birbirine bağlayan çizgi segmentleri boyunca yeni sentetik örnekler oluşturarak yapılır. Bu yaklaşımın arkasındaki mantık, karar sınırına yakın örneklemlerin sınıflandırılmasının genellikle daha zor olması ve dolayısıyla sınıflandırıcının

başarımını iyileştirmek için daha kritik olmalarıdır (Nguyen vd., 2011).

Bu hedeflenmiş aşırı örnekleme, yalnızca azınlık örneklemlerini çoğaltırken oluşabilecek aşırı uyum riskini azaltmaya yardımcı olur ve dengesiz veri setleri üzerinde eğitilen sınıflandırıcıların genel sağlamlığını artırır (Salunkhe vd., 2018). Borderline-SMOTE1 ve Borderline-SMOTE2 gibi varyantlar, çevreleyen örneklemlerin yoğunluğuna göre üretilen sentetik örnek sayısını ayarlayarak bu yaklaşımı daha da geliştirir (Salunkhe vd., 2018).

3. Bulgular ve Tartışma

Bu bölümde, kullanılan sınıflandırma tekniklerinin başarımı incelenmektedir. JM1 veri seti üzerinde yapılan analizlerde, modellerin eğitimi ve değerlendirilmesi için standart bir yaklaşım izlenmiştir. Öncelikle, veri seti %80 eğitim ve %20 test verisi olacak şekilde rastgele bölünmüştür. Eğitim verisi üzerinde, model başarımının daha güvenilir bir şekilde değerlendirilmesi ve aşırı uyum riskinin azaltılması amacıyla 10-katlı çapraz doğrulama tekniği uygulanmıştır. Yeniden örnekleme teknikleri, çapraz doğrulama sürecinin her bir katında yalnızca eğitim bölümüne uygulanarak veri sızıntısının önüne geçilmiştir. Modellerin nihai başarımı, ayrılan %20'lik test seti üzerinde değerlendirilmiştir. Başarımın analizi ve değerlendirilmesi, temel olarak karmaşıklık matrisinden elde edilen çeşitli ölçütlere dayanmaktadır (Şekil 2). Karmaşıklık matrisi, bir sınıflandırma modelinin tahminlerinin gerçek sınıf etiketleriyle karşılaştırılmasını özetleyen ve aşağıdaki dört temel parametreyi içeren bir tablodur:

Doğru Pozitif (True Positive - TP): Gerçekte pozitif (örneğin, hatalı modül) olan ve model tarafından doğru bir şekilde pozitif olarak sınıflandırılan örneklemlerin sayısı.

Yanlış Pozitif (False Positive - FP): Gerçekte negatif (örneğin, hatasız modül) olan ancak model tarafından yanlışlıkla pozitif olarak sınıflandırılan örneklemlerin sayısı.

Yanlış Negatif (False Negative - FN): Gerçekte pozitif olan ancak model tarafından yanlışlıkla negatif olarak sınıflandırılan örneklemlerin sayısı.

Doğru Negatif (True Negative - TN): Gerçekte negatif olan ve model tarafından doğru bir şekilde negatif olarak sınıflandırılan örneklemlerin sayısı.

		Gerçek Değerler	
		Hatalı Kod	Hatasız Kod
Tahmin Değerler	Hatalı Kod	TP	FP
	Hatasız Kod	FN	TN

Şekil 2. Karmaşıklık Matrisi.

Sınıflandırma modellerinin başarımı, karmaşıklık matrisinden türetilen ve literatürde yaygın olarak kullanılan aşağıdaki metrikler aracılığıyla değerlendirilmiştir: Kesinlik (Precision), F-ölçütü (F-measure), Doğruluk (Accuracy) ve ROC Eğrisi Altında Kalan Alan (AUC).

Kesinlik (Precision): Model tarafından pozitif olarak sınıflandırılan örneklemler içerisinde, gerçekte de pozitif olanların oranını ifade eder. Yanlış Pozitif sınıflandırmaların maliyetinin yüksek olduğu durumlarda özellikle dikkate alınan bir metriktir (eşitlik 1).

$$Kesinlik = \frac{TP}{TP+FP} \quad (1)$$

F-ölçütü (F-measure veya F1-Score): Kesinlik ve Duyarlılık metriklerinin harmonik ortalamasıdır. Bu iki metrik arasında bir denge sağlar ve özellikle sınıf dağılımının dengesiz olduğu veri setlerinde, Doğruluk metriğine kıyasla model başarımını daha anlamlı bir şekilde yansıtabilir (eşitlik 2).

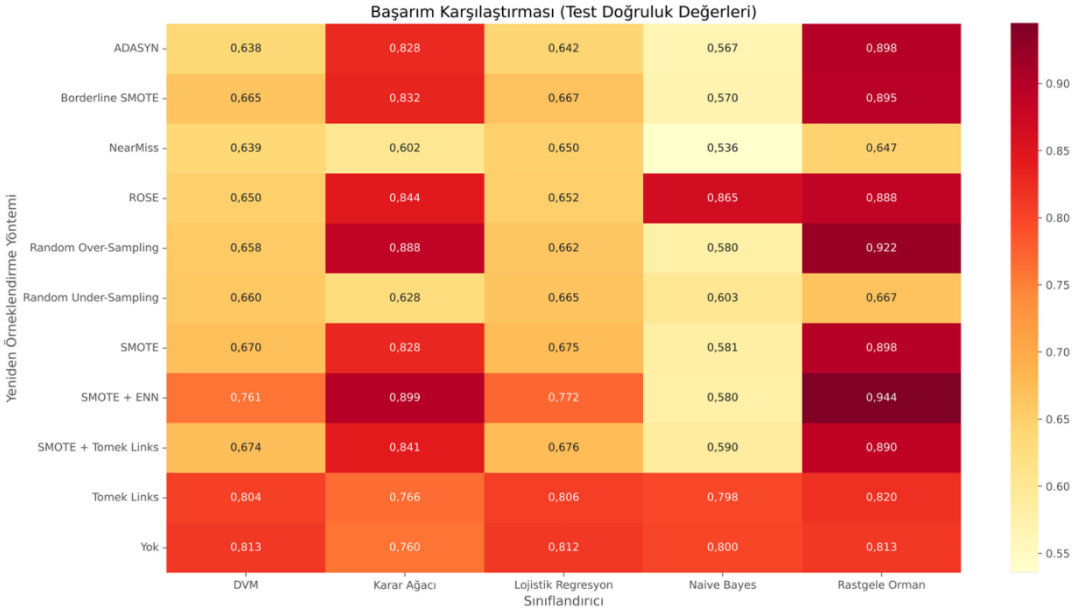
$$F - \text{Ölçütü} = 2x\frac{(Kesinlik+Duyarlılık)}{Kesinlik+Duyarlılık} = \frac{2xTP}{2xTP+FP+FN} \quad (2)$$

Doğruluk (Accuracy): Tüm sınıflandırmalar içinde doğru olarak yapılan tahminlerin (hem Doğru Pozitifler hem de Doğru Negatifler) toplam örneklem sayısına oranını belirtir. Yorumlanması en kolay metrik olmasına rağmen, sınıf dağılımının belirgin şekilde dengesiz olduğu durumlarda yanıltıcı olabilir; zira model, yalnızca çoğunluk sınıfını başarılı bir şekilde tahmin ederek yüksek bir doğruluk değeri elde edebilirken azınlık sınıfını tespit etmede yetersiz kalabilir (eşitlik 3).

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

ROC Eğrisi Altında Kalan Alan (AUC - Area Under the ROC Curve): ROC (Alıcı İşletim Karakteristiği - Receiver Operating Characteristic) eğrisi, modelin farklı sınıflandırma eşik değerlerinde Duyarlılık (TPR) oranının Yanlış Pozitif Oranına (FPR = FP / (FP + TN)) karşı çizilmesiyle elde edilir. AUC, bu eğrinin altında kalan alanı ifade eder ve [0, 1] aralığında bir değer alır. Modelin pozitif ve negatif sınıfları ne kadar iyi ayırt edebildiğinin genel bir ölçüsüdür. 0,5 değeri rastgele bir sınıflandırıcı başarımına işaret ederken, 1 değeri mükemmel bir ayırt etme yeteneğini gösterir. AUC, sınıf dağılımının dengesizliğinden etkilenmeyen bir metrik olarak değerlendirilir.

Hesaplamalarda her bir sınıf (bu çalışmada '0'- Hatasız ve '1' - Hatalı olarak temsil edilmiştir) için elde edilen F1-Ölçütü değerleri, genel Doğruluk ve AUC değerleri ile birlikte sunulan sonuç tablolarında detaylandırılmıştır. Sonuçların sunulduğu tablolarda, her bir metrik ve sınıflandırıcı kombinasyonu için elde edilen en yüksek değerler, okuyucunun kolayca ayırt edebilmesi amacıyla vurgulanmıştır.



Şekil 3. Sınıflandırıcı ve yeniden örnekleme yöntemlerinin ısı haritası.

3.1. Yeniden Örnekleme Yöntemlerinin Etkisi

Sonuç tabloları ve Şekil 3 incelendiğinde, yeniden örnekleme tekniklerinin farklı sınıflandırıcılar üzerindeki etkilerinin değişkenlik gösterdiği açıkça görülmektedir. İlk olarak, yeniden örnekleme yapılmayan durum ele alındığında, özellikle LR ve DVM en yüksek Doğruluk değerlerinin (0,8120 ve 0,8134) elde edildiği görülmektedir. Ancak bu yüksek doğruluk oranları, bu modellerin azınlık sınıfı (hatalı modüller, Sınıf 1) üzerindeki başarımını gizlemektedir. İlgili tablolara (Tablo 1-6) bakıldığında, LR ve DVM için yeniden örnekleme yapılmadığında Sınıf 1 F1-ölçütü değerlerinin sırasıyla 0,1996 ve 0,1880 gibi oldukça düşük seviyelerde kaldığı gözlemlenmektedir. Bu durum, modellerin büyük ölçüde çoğunluk sınıfını (hatasız modüller, Sınıf 0) doğru tahmin ederek yüksek doğruluk elde ettiğini, ancak hatalı modülleri tespit etmede ciddi şekilde başarısız olduğunu göstermektedir. Bu bulgu, literatürde belirtilen (Zheng vd., 2022; Bui, 2023; Mohapatra, 2024) LR ve DVM gibi algoritmaların sınıf dengesizliğine karşı hassasiyeti ile uyumludur ve yalnızca Doğruluk metriğine dayanarak model başarımını değerlendirmenin yanıltıcı olabileceğini vurgulamaktadır. Benzer bir durum, yeniden örnekleme yapılmayan NB, KA ve RO için de geçerlidir; bu modellerde de Sınıf 1 F1-ölçütü değerleri (sırasıyla 0,2762, 0,3793, 0,3477) düşüktür. Yeniden örnekleme teknikleri uygulandığında ise durum belirgin şekilde değişmektedir. Özellikle aşırı örnekleme ve hibrit yöntemler, birçok sınıflandırıcı için başarımı, özellikle azınlık sınıfı tespitini (Sınıf 1 F1-ölçütü) ve genel ayırt etme gücünü (AUC) önemli ölçüde iyileştirmiştir.

Naive Bayes için en iyi başarımı ROSE tekniği sergilemiştir. ROSE uygulandığında Doğruluk 0,8647'ye, AUC değeri ise 0,9075'e yükselmiş ve Sınıf 1 F1-ölçütü 0,8523 gibi dengeli bir değere ulaşmıştır. Bu, basit bir aşırı örnekleme tekniği olan ROSE'un NB'in olasılıksal yapısıyla iyi çalıştığını ve dengesizliği gidermede etkili olduğunu göstermektedir.

DVM ve LR için, yeniden örnekleme uygulanmayan duruma göre Doğruluk düşse de SMOTE + ENN gibi hibrit yöntemler AUC değerlerini (sırasıyla 0,8172 ve 0,8223) ve Sınıf 1 F1-ölçütü değerlerini (sırasıyla 0,7919 ve 0,7978) dramatik şekilde artırmıştır. Bu, modellerin yeniden örnekleme ile hatalı modülleri çok daha iyi tespit edebildiğini, ancak bunun genel doğruluktan bir miktar ödün vererek gerçekleştiğini göstermektedir. Hata tespitinin öncelikli olduğu durumlarda bu yöntemler tercih edilebilir.

Ağaç tabanlı topluluk yöntemleri olan KA ve RO için yeniden örnekleme tekniklerinin etkisi oldukça olumlu olmuştur. Her iki sınıflandırıcı için de en yüksek başarımı SMOTE + ENN hibrit yöntemi sağlamıştır. RO ile SMOTE + ENN kombinasyonu, 0,9350 Doğruluk ve 0,9837 AUC gibi çalışmadaki en yüksek değerlere ulaşmış, ayrıca her iki sınıf için de dengeli ve yüksek F1-ölçütü değerleri (Sınıf 0: 0,9126, Sınıf 1: 0,9483) elde etmiştir. DT ile SMOTE + ENN kombinasyonu da benzer şekilde 0,9052 Doğruluk ve 0,8981 AUC ile oldukça başarılı sonuçlar vermiştir. Bu durum, SMOTE'un sentetik azınlık örnekleri oluşturarak ve ENN'nin sınıf sınırlarına yakın gürültülü veya örtüşen örnekleri temizleyerek ağaç tabanlı modellerin karar sınırlarını daha etkin bir şekilde öğrenmesine yardımcı olduğunu düşündürmektedir. Random Over-Sampling, ADASYN, Borderline SMOTE gibi diğer aşırı örnekleme ve hibrit yöntemler de RO ve DT için yeniden örnekleme yapılmayan duruma göre belirgin iyileşmeler sağlamıştır.

Eksik örnekleme teknikleri olan RUS ve NearMiss, genel olarak çoğu sınıflandırıcı için düşük başarım göstermiştir. Bu yöntemler, Doğruluk, AUC ve F1-ölçütü değerlerini sıklıkla diğer yöntemlerin veya hatta yeniden örnekleme yapılmayan durumun altına düşürmüştür. Bu durum, çoğunluk sınıfından rastgele veya belirli bir

kritere göre örnek silmenin, modelin öğrenmesi için gerekli olan önemli bilgilerin kaybına yol açabileceği hipotezini desteklemektedir. Benzer şekilde, bir veri temizleme/eksik örnekleme tekniği olan Tomek Links, tek başına kullanıldığında genellikle yeniden örnekleme yapılmayan duruma kıyasla marjinal bir iyileşme sağlamış veya başarımı değiştirmemiştir. Ancak, SMOTE ile birleştirildiğinde (SMOTE + Tomek Links), bazı durumlarda saf SMOTE'a göre küçük iyileşmeler göstermiştir.

3.2. Sınıflandırıcı Başarımlarının Karşılaştırılması

Yeniden örnekleme yöntemleri uygulandıktan sonraki başarımlar dikkate alındığında:

- RO, özellikle SMOTE + ENN ile birleştirildiğinde, JM1 veri seti üzerinde yazılım hata tahmini için en yüksek ve en dengeli başarımı sergileyen sınıflandırıcı olmuştur. Yüksek Doğruluk, çok yüksek AUC ve dengeli F1-ölçütü değerleri, bu kombinasyonun hem genel tahmin gücünün yüksek olduğunu hem de hatalı modülleri etkin bir şekilde tespit edebildiğini göstermektedir.
- KA da SMOTE + ENN ile birlikte oldukça iyi bir başarım göstermiş, RO'ya yakın sonuçlar elde etmiştir.
- NB, uygun yeniden örnekleme tekniği (ROSE) ile kullanıldığında oldukça rekabetçi sonuçlar vermiş, özellikle AUC ve dengeli F1-ölçütü açısından başarılı olmuştur.
- DVM ve LR, yeniden örnekleme olmadan yüksek doğruluk göstermelerine rağmen, hatalı modül tespitinde zayıf kalmışlardır. SMOTE + ENN gibi yöntemlerle bu zayıflık giderilmiş ve AUC değerleri önemli ölçüde artırılmış olsa da en yüksek doğruluğa ulaşamamışlardır. Bu durum, bu modellerin dengesiz veriye karşı yapısal hassasiyetini ve yeniden örneklemenin bu hassasiyeti gidermedeki rolünü ortaya koymaktadır.

Özetle, bulgular JM1 gibi dengesiz yazılım hata tahmini veri setlerinde yeniden örnekleme tekniklerinin kritik bir rol oynadığını göstermektedir. Özellikle SMOTE + ENN gibi hibrit yöntemler ve ROSE gibi aşırı örnekleme teknikleri, modellerin azınlık sınıfını (hatalı modüller) daha etkin bir şekilde öğrenmesini sağlayarak F1-ölçütü ve AUC gibi metriklerde önemli iyileşmeler sağlamıştır. En iyi sonuçlar, güçlü bir topluluk öğrenme yöntemi olan Rastgele Orman ile SMOTE + ENN hibrit yeniden örnekleme tekniğinin birleştirilmesiyle elde edilmiştir. Eksik örnekleme yöntemleri ise bu spesifik veri seti ve problem bağlamında genellikle bilgi kaybına yol açarak daha düşük başarıma neden olmuştur. Bu sonuçlar, SDP modelleri geliştirilirken sınıf dengesizliği sorununun dikkatle ele alınması ve uygun veri ön işleme stratejilerinin seçilmesinin önemini vurgulamaktadır.

Tablo 1. Sınıflandırıcıların en iyi sonuç verdiği yeniden örnekleme yöntemleri

Sınıflandırıcı	En İyi Yeniden Örnekleme Yöntemi	Doğruluk
LR	Örneklendirme yok	0,812
DVM	Örneklendirme yok	0,813
NB	ROSE	0,865
KA	SMOTE + ENN	0,905
RO	SMOTE + ENN	0,935

Tablo 2. NB için yeniden örnekleme yöntemleri sonuçları

Yeniden Örnekleme Yöntemi	Doğruluk	AUC	F1 (Hatasız)	F1 (Hatalı)
ROSE	0,865	0,908	0,875	0,852
Örneklendirme Yok	0,800	0,694	0,884	0,276
Tomek Links	0,798	0,708	0,883	0,293
RUS	0,603	0,683	0,712	0,363
SMOTE + Tomek Links	0,588	0,699	0,697	0,357
SMOTE	0,581	0,691	0,693	0,341
Random Over-Sampling	0,580	0,683	0,693	0,337
SMOTE + ENN	0,576	0,772	0,634	0,496
Borderline SMOTE	0,568	0,680	0,683	0,323
ADASYN	0,564	0,679	0,682	0,307
NearMiss	0,536	0,625	0,663	0,255

Tablo 3. DVM için yeniden örnekleme yöntemleri sonuçları

Yeniden Örnekleme Yöntemi	Doğruluk	AUC	F1 (Hatasız)	F1 (Hatalı)
Örneklendirme Yok	0,813	0,723	0,895	0,188
Tomek Links	0,809	0,731	0,891	0,220
SMOTE + ENN	0,745	0,817	0,672	0,792
SMOTE + Tomek Links	0,676	0,729	0,670	0,648
SMOTE	0,670	0,723	0,692	0,645
Borderline SMOTE	0,661	0,708	0,679	0,642
RUS	0,660	0,705	0,691	0,623
Random Over-Sampling	0,659	0,709	0,686	0,626
ROSE	0,650	0,676	0,673	0,622
ADASYN	0,644	0,701	0,655	0,632
NearMiss	0,624	0,639	0,659	0,580

Tablo 4. RO için yeniden örnekleme yöntemleri sonuçları

Yeniden Örnekleme Yöntemi	Doğruluk	AUC	F1 (Hatasız)	F1 (Hatalı)
SMOTE + ENN	0,935	0,984	0,912	0,948
Random Over-Sampling	0,924	0,975	0,921	0,926
ADASYN	0,893	0,948	0,892	0,894
Borderline SMOTE	0,892	0,949	0,892	0,892
SMOTE	0,892	0,949	0,891	0,893
SMOTE + Tomek Links	0,888	0,953	0,889	0,888
ROSE	0,888	0,943	0,895	0,880
Tomek Links	0,825	0,793	0,897	0,391
Örneklendirme Yok	0,818	0,751	0,894	0,347
RUS	0,668	0,737	0,670	0,666
NearMiss	0,658	0,709	0,685	0,625

Tablo 5. LR için yeniden örnekleme yöntemleri sonuçları

Yeniden Örnekleme Yöntemi	Doğruluk	AUC	F1 (Hatasız)	F1 (Hatalı)
Örneklendirme Yok	0,812	0,719	0,893	0,199
Tomek Links	0,807	0,727	0,890	0,221
SMOTE + ENN	0,754	0,822	0,686	0,797
SMOTE + Tomek Links	0,679	0,731	0,704	0,651
SMOTE	0,675	0,726	0,697	0,650
Borderline SMOTE	0,665	0,710	0,684	0,644
RUS	0,665	0,709	0,695	0,628
Random Over-Sampling	0,661	0,710	0,688	0,630
ROSE	0,651	0,676	0,673	0,626
ADASYN	0,648	0,703	0,659	0,637
NearMiss	0,633	0,641	0,665	0,594

Tablo 6. KA için yeniden örnekleme yöntemleri sonuçları

Yeniden Örnekleme Yöntemi	Doğruluk	AUC	F1 (Hatasız)	F1 (Hatalı)
SMOTE + ENN	0,905	0,898	0,875	0,923
Random Over-Sampling	0,884	0,894	0,875	0,891
ROSE	0,845	0,841	0,846	0,844
Borderline SMOTE	0,838	0,838	0,837	0,839
ADASYN	0,829	0,828	0,828	0,831
SMOTE	0,826	0,826	0,825	0,828
SMOTE + Tomek Links	0,825	0,825	0,825	0,824
Tomek Links	0,766	0,623	0,853	0,429
Örneklendirme Yok	0,757	0,598	0,849	0,379
RUS	0,628	0,627	0,637	0,618
NearMiss	0,605	0,610	0,603	0,607

4. Sonuçlar

Çalışmada elde edilen bulgular, JM1 veri setinin doğal dengesiz yapısının, yeniden örnekleme uygulanmadığında özellikle DVM ve LR gibi bazı sınıflandırıcıların başarımını olumsuz etkilediğini doğrulamıştır. Bu modeller, yüksek genel doğruluk oranlarına ulaşsalar da azınlık sınıfı olan hatalı modülleri tespit etmede oldukça düşük F1-ölçütü değerleri sergilemişlerdir. Bu durum, yalnızca doğruluk metriğine dayalı değerlendirmelerin yanıltıcı olabileceğini ve sınıf dengesizliğinin YHT modelleri üzerindeki belirgin etkisini bir kez daha göstermiştir.

Yeniden örnekleme tekniklerinin uygulanması, özellikle aşırı örnekleme ve hibrit yöntemlerin, sınıflandırıcı başarımını, bilhassa azınlık sınıfı tespiti (F1-ölçütü Sınıf 1) ve genel ayırt edicilik (AUC) açısından önemli ölçüde iyileştirdiği gözlemlenmiştir. Çalışmanın öne çıkan bulguları şunlardır:

1. Hibrit Yöntemlerin Başarısı: SMOTE ile ENN'nin birleştirildiği hibrit yöntem (SMOTE + ENN), özellikle ağaç tabanlı topluluk modelleri olan RO ve KA ile kullanıldığında üstün başarım göstermiştir. RO ve SMOTE + ENN kombinasyonu, 0,9350 Doğruluk ve 0,9837 AUC ile çalışmadaki en iyi sonuçları elde etmiş, aynı zamanda her iki sınıf için dengeli ve yüksek F1-ölçütü değerleri sağlamıştır.
2. Aşırı Örneklemenin Etkinliği: ROSE tekniği, NB sınıflandırıcısı için en iyi sonuçları vermiş, AUC ve F1-ölçütü değerlerini önemli ölçüde artırmıştır. SMOTE, Borderline-SMOTE ve ADASYN gibi diğer aşırı örnekleme teknikleri de birçok sınıflandırıcı için yeniden örnekleme yapılmayan duruma göre belirgin iyileşmeler sağlamıştır.
3. Eksik Örneklemenin Sınırlılıkları: Rastgele Eksik Örnekleme (RUS) ve NearMiss gibi eksik örnekleme yöntemleri, incelenen senaryoda genellikle diğer yöntemlere kıyasla daha düşük başarım sergilemiştir. Bu durum, bu yöntemlerin çoğunluk sınıfından veri silerken potansiyel olarak önemli bilgileri kaybetme riski taşıdığını düşündürmektedir.
4. Sınıflandırıcı Başarımı: Yeniden örnekleme teknikleri uygulandıktan sonra, RF genel olarak en güçlü ve dengeli başarımı sunan sınıflandırıcı olarak öne çıkmıştır.

Bu çalışma, JM1 veri seti özelinde çeşitli yeniden örnekleme tekniklerinin farklı sınıflandırıcılar üzerindeki etkisini sistematik ve karşılaştırmalı bir şekilde ortaya koymuştur. Bulgular, YHT'de sınıf dengesizliğiyle mücadele için uygun yeniden örnekleme stratejisi seçiminin kritik öneme sahip olduğunu ve özellikle SMOTE+ENN gibi hibrit yöntemlerin ve RF gibi topluluk öğrenmesi algoritmalarının bu tür problemler için güçlü adaylar olduğunu göstermektedir. Ayrıca, dengesiz veri setlerinde model başarımını değerlendirirken Doğruluk metriğinin ötesinde AUC ve F1-ölçütü gibi metriklere odaklanmanın gerekliliğini vurgulamaktadır.

Çalışmanın bazı sınırlılıkları bulunmaktadır. Bulgular yalnızca JM1 veri seti ile sınırlıdır ve diğer yazılım

projelerinden elde edilen farklı özelliklere sahip veri setleri üzerinde genellenebilirliği test edilmemiştir. İncelenen yeniden örnekleme teknikleri ve sınıflandırıcılar kapsamlı olmakla birlikte, literatürdeki tüm alternatifleri içermemektedir. Ayrıca, algoritmaların ve yeniden örnekleme yöntemlerinin hiperparametre optimizasyonu bu çalışmanın odak noktası olmamıştır.

Gelecek çalışmalarda NASA MDP koleksiyonundaki JM1 gibi diğer alt veri setleri ve farklı programlama dilleri veya uygulama alanlarından gelen endüstriyel veri setleri üzerinde tekrarlanarak bulguların genellenebilirliği araştırılabilir. Yeni ve gelişmiş yeniden örnekleme teknikleri ve derin öğrenme modellerinin YHT ve sınıf dengesizliği bağlamındaki başarımları incelenebilir. Kapsamlı hiperparametre optimizasyonunun ve öznitelik seçimi tekniklerinin yeniden örnekleme ile birleştirilmesinin etkileri de değerli araştırma konularıdır. Son olarak, farklı yeniden örnekleme ve modelleme yaklaşımlarının pratik uygulamadaki maliyet-etkinlik analizleri, bu tekniklerin endüstriyel benimsenmesini teşvik edebilir.

Katkı Beyan Oranı

Yazarların katkı yüzdeleri aşağıda verilmiştir. Yazarlar makaleyi incelemiş ve onaylamıştır.

	E.C.Y.	R.O.
K	50	50
T	50	50
Y	50	50
VTI	50	50
VAY	50	50
KT	50	50
YZ	50	50
KI	50	50
GR	50	50

K= kavram, T= tasarım, Y= yönetim, VTI= veri toplama ve/veya işleme, VAY= veri analizi ve/veya yorumlama, KT= kaynak tarama, YZ= Yazım, KI= kritik inceleme, GR= gönderim ve revizyon

Çatışma Beyanı

Yazarlar bu çalışmada hiçbir çıkar ilişkisi olmadığını beyan etmektedirler.

Etik Onay Beyanı

Bu araştırmada hayvanlar ve insanlar üzerinde herhangi bir çalışma yapılmadığı için etik kurul onayı alınmamıştır.

Kaynaklar

Abdelmoumin, G., Rawat, D. B., & Rahman, A. (2023). Studying imbalanced learning for anomaly-based intelligent IDS for mission-critical Internet of Things. *Journal of Cybersecurity and Privacy, 3*(4), 706–743.

Agrawal, A., Menzies, T., Minku, L. L., Wagner, M., & Yu, Z. (2020). Better software analytics via "DUO": Data mining algorithms using/used-by optimizers. *Empirical Software Engineering, 25*(3), 2099–2136.

Alam, T. M., Shaukat, K., Khan, W. A., Hameed, I. A., Almuqren, L. A., Raza, M. A., Aslam, M., & Luo, S. (2022). An efficient deep learning-based skin cancer classifier for an imbalanced dataset. *Diagnostics, 12*(9), 2115.

Aljawazneh, H., Mora, A. M., García-Sánchez, P., & Castillo-Valdivieso, P. A. (2021). Comparing the performance of deep learning methods to predict companies' financial failure. *IEEE Access, 9*, 97010–97038.

Arya, D. M., Nassif, M., & Robillard, M. P. (2021). A data-centric study of software tutorial design. *IEEE Software, 39*(3), 106–115.

Ayoub, S., Gulzar, Y., Rustamov, J., Jabbari, A., Reegu, F. A., & Turaev, S. (2023). Adversarial approaches to tackle imbalanced data in machine learning. *Sustainability, 15*(9), 7097.

Badvath, D., Miriyala, A. S., Gunupudi, S. C. K., & Kuricheti, P. V. K. (2022). Prediction of software defects using deep learning with improved cuckoo search algorithm. *Concurrency and Computation: Practice and Experience, 34*(26).

Balogun, A. O., Basri, S., Abdulkadir, S. J., & Hashim, A. S. (2019). Performance analysis of feature selection methods in software defect prediction: A search method approach. *Applied Sciences, 9*(13), 2764.

Boschi, R. S., Rodrigues, L. H. A., & Lopes-Assad, M. L. R. C. (2014). Using classification trees to evaluate the performance of pedotransfer functions. *Vadose Zone Journal, 13*(8), vzj2013-11.

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys, 49*(2), 1–50.

Bui, M. T. T. (2023). Incremental ensemble learning model for imbalanced data: A case study of credit scoring. *Journal of Advanced Engineering and Computation, 7*(2), 105.

Byeon, H. (2021). Predicting the depression of the South Korean elderly using SMOTE and an imbalanced binary dataset. *International Journal of Advanced Computer Science and Applications, 12*(1).

Deng, J., Lu, L., & Qiu, S. (2020). Software defect prediction via LSTM. *IET Software, 14*(4), 443–450.

Elsobky, A. M., Keshk, A. E., & Malhat, M. G. (2023). A comparative study for different resampling techniques for imbalanced datasets. *International Journal of Computers and Information, 10*(3), 147–156.

Fan, Z., Sohail, S., Sabrina, F., & Gu, X. (2024). Sampling-based machine learning models for intrusion detection in imbalanced dataset. *Electronics, 13*(10), 1878.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research, 61*, 863–905.

Fu, M., Tantithamthavorn, C., Le, T., Kume, Y., Nguyen, V., Phung, D., & Grundy, J. (2024). AIBugHunter: A practical tool for predicting, classifying and repairing software vulnerabilities. *Empirical Software Engineering, 29*(1).

Fu, T., Gao, W., Coley, C., & Sun, J. (2022). Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems, 35*, 12325–12338.

Goyal, S. (2022). Effective software defect prediction using support vector machines (SVMs). *International Journal of System Assurance Engineering and Management, 13*(2), 681–696.

Guo, H., Diao, X., & Liu, H. (2019). Improving undersampling-based ensemble with rotation forest for imbalanced problem.

Turkish Journal of Electrical Engineering and Computer Sciences, 27(2), 1371–1386.

Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. (2012). A systematic literature review on fault prediction performance in software engineering. *IEEE Transactions on Software Engineering, 38*(6), 1276–1304.

Halstead, M. H. (1977). *Elements of software science*. Elsevier North-Holland.

Hryszko, J., & Madeyski, L. (2018). Cost effectiveness of software defect prediction in an industrial project. *Foundations of Computing and Decision Sciences, 43*(1), 7–35.

Huang, C., Li, Y., Loy, C. C., & Tang, X. (2020). Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(11), 2781–2794.

Jeon, Y. S., & Lim, D. J. (2020). Psu: Particle stacking undersampling method for highly imbalanced big data. *IEEE Access, 8*, 131920–131927.

Khuat, T. T., & Gabrys, B. (2020). A comparative study of general fuzzy min-max neural networks for pattern classification problems. *Neurocomputing, 386*, 110–125.

Kou, G., Peng, Y., Shi, Y., & Wu, W. (2012). Classifier evaluation for software defect prediction. *Studies in Informatics and Control, 21*(2), 118.

Li, Y., Soliman, M., & Avgeriou, P. (2022). Identifying self-admitted technical debt in issue tracking systems using machine learning. *Empirical Software Engineering, 27*(6).

Li, Z., Jing, X. Y., & Zhu, X. (2018). Progress on approaches to software defect prediction. *IET Software, 12*(3), 161–175.

Loukili, M., Messaoudi, F., & El Ghazi, M. (2024). Enhancing customer retention through deep learning and imbalanced data techniques. *Iraqi Journal of Science*, 2853–2866.

Lunardon, N., Menardi, G., & Torelli, N. (2014). ROSE: a package for binary imbalanced learning. *The R Journal, 6*(1), 79–89.

Mabayoje, M. A., Balogun, A. O., Jibril, H. A., Atoyebi, J. O., Mojeed, H. A., & Adeyemo, V. E. (2019). Parameter tuning in KNN for software defect prediction: An empirical analysis. *Jurnal Teknologi dan Sistem Komputer, 7*(4), 121–126.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering, 2*(4), 308–320.

McKendrick, R., Feest, B., Harwood, A., & Falcone, B. (2019). Theories and methods for labeling cognitive workload: Classification and transfer learning. *Frontiers in Human Neuroscience, 13*, 295.

Menzies, T., Milton, Z., Turhan, B., Cukic, B., Jiang, Y., & Bener, A. (2010). Defect prediction from static code features: Current results, limitations, new approaches. *Automated Software Engineering, 17*(4), 375–407.

Meysami, M., Kumar, V., Pugh, M., Lowery, S. T., Sur, S., Mondal, S., & Greene, J. M. (2023). Utilizing logistic regression to compare risk factors in disease modeling with imbalanced data: A case study in vitamin D and cancer incidence. *Frontiers in Oncology, 13*, 1227842.

Mohapatra, U. (2024). An efficient convolutional neural network-based classifier for an imbalanced oral squamous carcinoma cell dataset. *IAES International Journal of Artificial Intelligence, 13*(1), 487.

Mqadi, N. M., Naicker, N., & Adeliyi, T. (2021). Solving misclassification of the credit card imbalance problem using near miss. *Mathematical Problems in Engineering, 2021*(1), 7194728.

Najeeb, M. A., & Alariyibi, A. (2024). Imbalanced dataset effect on CNN-based classifier performance for face recognition. *International Journal of Artificial Intelligence & Applications, 15*(1), 25–41.

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms, 3*(1), 4–21.

Pandey, S. K., Mishra, R. B., & Tripathi, A. K. (2020). BPDET: An effective software bug prediction model using deep representation and ensemble learning techniques. *Expert Systems with Applications, 144*, 113085.

Patel, H., & Thakur, G. S. (2017). Classification of imbalanced data using a modified fuzzy-neighbor weighted approach. *International Journal of Intelligent Engineering and Systems, 10*(1), 56–64.

Rahardian, H., Faisal, M. R., Abadi, F., Nugroho, R. A., & Herteno, R. (2020). Implementation of data level approach techniques to solve unbalanced data case on software defect classification. *Journal of Data Science and Software Engineering, 1*(01), 53–62.

Ren, J., Qin, K., Ma, Y., & Luo, G. (2014). On software defect prediction using machine learning. *Journal of Applied Mathematics, 2014*, 1–8.

Rendón, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., & Granda-Gutiérrez, E. E. (2020). Data Sampling Methods to Deal With the Big Data Multi-Class Imbalance Problem. *Applied Sciences, 10*(4), 1276.

Salunkhe, U. R., & Mali, S. N. (2018). A hybrid approach for class imbalance problem in customer churn prediction: A novel extension to under-sampling. *International Journal of Intelligent Systems and Applications, 14*(5), 71.

Sơn, L. H., Pritam, N., Khari, M., Kumar, R., Phuong, P. T. M., & Thong, P. H. (2019). Empirical study of software defect prediction: A systematic mapping. *Symmetry, 11*(2), 212.

Taherkhani, A., Cosma, G., & McGinnity, T. M. (2020). AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing, 404*, 351–366.

Tomar, D., & Agarwal, S. (2016). Prediction of defective software modules using class imbalance learning. *Applied Computational Intelligence and Soft Computing, 2016*, 1–12.

Wang, S., Liu, T., Nam, J., & Tan, L. (2020). Deep semantic feature learning for software defect prediction. *IEEE Transactions on Software Engineering, 46*(12), 1267–1293.

Wu, S., Yau, W. C., Ong, T. S., & Chong, S. C. (2021). Integrated churn prediction and customer segmentation framework for telco business. *IEEE Access, 9*, 62118–62136.

Zakariah, M., AlQahtani, S. A., & Al-Rakhami, M. S. (2023). Machine learning-based adaptive synthetic sampling technique for intrusion detection. *Applied Sciences, 13*(11), 6504.

Zhang, D., Tian, W., Cheng, X., Shi, F., Qiu, H., Liu, X., & Chen, S. (2023). FedBIP: A federated learning-based model for wind turbine blade icing prediction. *IEEE Transactions on Instrumentation and Measurement, 72*, 1–11.

Zhang, J., & Chen, L. (2019). Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. *Computer Assisted Surgery, 24*(sup2), 62–72.

Zheng, M., Wang, F., Hu, X., Miao, Y, Cao, H., & Tang, M. (2022). A method for analyzing the performance impact of imbalanced binary data on machine learning models. *Axioms, 11*(11), 607.