T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDEKİ SINIF DENGESİZLİĞİ SORUNUNUN GİDERİLMESİ

AKKENZHE SARMANOVA

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. SONGÜL ALBAYRAK

İSTANBUL, 2013

ÖNSÖZ

Tezimin planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi, destek ve yardımlarını esirgemeyen, her zaman motive edici eleştirileri ile bana yön veren ve engin bilgi ve tecrübelerinden yararlandığım değerli hocam ve danışmanım Sayın Yrd. Doç. Dr. Songül ALBAYRAK' a sonsuz teşekkürlerimi sunarım.

En değerli hazinem sevgili aileme ve tüm dostlarıma manevi hiçbir yardımı esirgemeden yanımda oldukları için tüm kalbimle teşekkür ederim.

Eylül, 2013

Akkenzhe SARMANOVA

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

N^+	Azınlık sınıf örneklerinin sayısı
N^-	Çoğunluk sınıf örneklerinin sayısı
T	Topluluk sınıflandırıcılarının sayısı
Z_{min}	Azınlık sınıfı (Minority Class)
Z_{maj}	Çoğunluk sınıfı (Majority Class)
d	İki örnek arasındaki mesafe
m	Eğitim veri setindeki örnekler sayısı
x_i	Azınlık sınıfı örneği
x_j	Çoğunluk sınıfı örneği
x_k	x_i ve x_j arasındaki örnek
α	Rasgele float sayısı
δ	Çoğaltma yüzdesi
β	Dengesizlik oranı
γ	Çoğunluk sınıfı örneklerinin seçilmiş yüzdesi
ε_t	Hata oranı

KISALTMA LİSTESİ

ANN	Artificial Neural Networks
AUC	Area Under the Curve
BEV	Bagging Ensemble Variation
CFS	Correlation based Feature Subset
DT	Decision Tree (Karar Ağacı)
DVM	Destek Vektor Makinesi (Support Vector Machine)
ENN	Edited Nearest Neighbour
FN	False Negative
FP	False Positive
FPR	False Positive Rate
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
NBC	Naive Bayes Classifier
NCR	Neighbourhood Cleaning Rule
ROC	Receiver Operating Curve
ROS	Random Over Sampling
RUS	Random Under Sampling
SMOTE	Synthetic Minority Over-sampling Technique
TN	True Negative
TP	True Positive
TPR	True Positive Rate

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

VERİ MADENCİLİĞİNDEKİ SINIF DENGESİZLİĞİ SORUNUNUN GİDERİLMESİ

Akkenzhe SARMANOVA

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Songül ALBAYRAK

İki-sınıflı veri setlerindeki en önemli sorunlarından biri olan sınıf dengesizliği sorununu çözmek son yıllarda daha fazla önem kazanmıştır. Veri kümesinde sınıf dağılımı dengesiz olduğu zaman, geleneksel makine öğrenme yöntemleri genellikle azınlık sınıfının görülmemiş örnekleri için düşük sınıflama başarısı vermektedir. Çünkü çoğunluk sınıfına doğru kuvvetle yönlenme eğilimindedir. Literatürde sınıf dengesizliği sorununu gidermek için çeşitli algoritmalar mevcuttur. Bu tez, sınıf dengesizliği probleminin önemini ve problemin çözümünün veri madenciliğindeki geniş uygulama alanlarını değerlendirme ölçüleri ile tanıtır. Ayrıca dengesizlik sorununu değerlendirmek ve çözmek için mevcut yöntemleri, C4.5, DVM ve KNN gibi farklı sınıflandırıcıları temel öğrenici olarak kullanarak karşılaştırır. En iyi temel öğreniciyi ve çoğunluk ve azınlık sınıfları dağılımına göre en iyi performansa sahip algoritmayi bulmak amacıyla çeşitli deneyler yapılmıştır. Buna ek olarak, tez kapsamında geliştirilen yeni bir algoritma olarak RusAda önerilmiştir ve bu algoritma tezde incelenen diğer algoritmalarla karşılaştırılmıştır.

Anahtar Kelimeler: Sınıf dengesizliği, ikili sınıflandırma, yeniden örnekleme, boosting, topluluk öğrenme.

ABSTRACT

ALLEVIATING THE CLASS IMBALANCE PROBLEM IN DATA MINING

Akkenzhe SARMANOVA

Department of Computer Engineering

MSc. Thesis

Adviser: Assist. Prof. Dr. Songül ALBAYRAK

The class imbalance problem in two-class data sets which is one of the most important problems has got more and more emphasis in recent years. When the class distribution of a data set is imbalanced, a conventional machine learning method usually has poor classification accuracy for unseen examples from the minority class because it is strongly biased towards the majority class. There are several algorithms to alleviate the problem of class imbalance in literature. This paper introduces the importance of class imbalance problem and their broad application domains in data mining, and then summarizes the evaluation metrics and compares the existing methods using different classifiers like C4.5, DVM, and KNN as base learners to evaluate and solve the imbalance problem. Several experiments have been done in order to find the best base learner and the algorithm which has the best performance according to the distribution of majority and minority classes. In addition, we proposed a new algorithm RusAda and have done some investigations comparing the performance of RusAda with the other algorithms used in this paper.

Keywords: Class imbalance, binary classification, re-sampling, boosting, ensemble learning.

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

BÖLÜM 1

GİRİŞ

Son yıllarda, veri madenciliği ve makine öğrenmesi biyomedikal bilişim, örüntü tanıma, dolandırıcılık algılama, doğal dil işleme, tıbbi teşhis, yüz tanıma, metin sınıflandırma, arıza teşhis, anomali tespiti gibi benzeri birçok karmaşık ve önemli gerçek dünya problemlerini çözmek amacıyla geniş olarak incelenmiştir ve birçok alanlarda uygulanmıştır. Bu tür araştırmaların birkaç avantajları vardır, örneğin, veri madenciliği algoritmaları, insanların ham verileri verimli analiz etmelerine, bilinmeyen örnekleri ayıklamalarına, doğru kararlar almalarına ve yeni bilgi keşfetmelerine yardımcı olabilir.

İnsanlar geçmiş deneyimlerinden bilgileri öğrenir ve daha sonra bunları gelecekteki sorunları çözmek için kullanır. Veri madenciliği yöntemleri, benzer şekilde, bilinen ampirik verilerden bilgi keşfedebilirler ve sonunda öğrendikleri bilgileri gelecekteki bilinmeyen verilere dayalı kararları almak için kullanırlar.

Veri madenciliği alanlarındaki veri bilgileri, bilgi tablosu (information table) tarafından temsil edilir. Veri, bir çok örnek (ya da nesne) içerir ve her örnek, birçok ortak anlamlı özellik (ya da faktör, öznitelik) tarafından açıklanır, burada her bir özellik, geçerli bir nominal veya sayısal değer ile temsil edilir. Bilgi, veri özellikleri ve örnekleri ile dolaylı gösterilir, bu nedenle öğrenme algoritmaları bu bilginin, anlaşılır ve uygulanabilir şekilde açıklanmasını gerektirir.

Makine öğrenmesi yöntemlerinin iki ana türü vardır. Birincisi, öğreticisiz öğrenme: veri, bağımlı bir hedef sınıfını sağlamaz ve öğrenme yöntemi, veri örnekleri ve özellikleri arasındaki bazı ilişkilerin keşfedilmesini gerektirir. İkincisi, öğretici ile öğrenme: burada bağımlı sınıf önceden bilinmektedir; öğrenmenin ana görevi, diğer özellikler ile hedef sınıfı arasındaki ilişkiyi keşfetmektir. Ayrıca, öğreticiyle öğrenme, öğrenilmesi gereken

hedef sınıf özelliğine dayalı olarak iki kategoriye ayrılır. Eğer sınıf ayrık değerlerle veya tanımlayıcı etiketlerle verilmişse o zaman bu öğrenme görevine sınıflandırma denir, anlamı, örnekleri doğru sınıfa sınıflandırmaktır; aksi halde, eğer hedef sınıf sürekli sayısal (numerical continuous) bir değerle verilmişse, o zaman bu bir regresyondur, burada öğrenici her örnek için gerçek hedef değerini eşleştirmek için bir tahmin değeri üretir.

Bu tezde sınıflamada karşılaşılan sınıf dengesizliği problemi üzerinde durulmuştur. Özellikle ikili sınıflandırma problemi bu tezin özel ilgi alanıdır.

Aslında, bilim adamları tarafından çok sayıda sınıflandırma yöntemi önerilmiştir. Ünlü ve popüler olanlar arasında k-en Yakın Komşu yöntemi (KNN), Naive Bayes Sınıflandırıcısı (NBC), Lineer Diskriminant Analizi (LDA), Yapay Sinir Ağları (ANN), Karar Ağacı (DT), Destek Vektör Makinaları (DVM), boosting ve bagging yer almaktadır. Bu gelişmiş öğrenme algoritmalarının çoğu ve bunların çeşitleri, ampirik eğitim verileri ve yeni test verileri üzerinde yüksek sınıflandırma doğruluğunu vermektedir. Bu yüksek doğruluk eğilimi göz önüne alındığında, araştırmacılar için daha iyi sınıflandırma yöntemlerini tasarlamak zordur. Ayrıca, yüksek tahmin doğruluğu, bize makine öğrenme yöntemlerinin hemen hemen her sınıflandırma problemini doğru çözebildiği fikrini verir. Ancak gerçekte her problemin çözümünde yüksek tahmin doğruluğu görmek mümkün değildir. Bunun en önemli nedenlerinden biri de verinin sınıflara dağılımındaki dengesizlikten kaynaklanmaktadır. Son yıllarda dengesiz veri setleri ile veri madenciliği, hem teorik ve hem de pratik yönleriyle daha fazla dikkat çekmektedir. Çok büyük veriden veya gürültülü ve eksik değerli veriden verimli öğrenme gibi etkili çözümler bulmak için olağanüstü bir çaba gerektiren zor sorunlar hala vardır. Birçok araştırmacı tarafından hala geniş olarak araştırılan dengesiz veriden öğrenme sorunu en zor sorunlardan biridir ve sınıf dengesizliği olarak isimlendirilir. Dengesiz veri setlerinden öğrenme sorunu, bir sınıf örneklerinin sayısı diğer sınıf örnekleri sayısından önemli ölçüde daha fazla olduğu zaman oluşur. Bu tür problemlerde, bazı sınıflardaki veri miktarı diğer sınıf ile karşılaştırıldığında yeterince temsil edilemez. Veri miktarının az olduğu sınıflarda bulunan bu örnekler sınıfı temsil etmede çok önemli ve değerlidir. Bu sınıf dengesizliği olan veri setlerinde tahmin hatası, çok yüksek olur. Geleneksel sınıflandırma yöntemleri, bütün eğitim setinin genel hata oranını en aza indirmek

amacıyla, genelde nispeten dengeli sınıf dağılımı olduğunu varsayar ve azınlık sınıflarının katkısı az olduğundan dengesiz veriler üzerinde iyi performans vermez. Örneğin, bir sınıflandırıcı, bir sınıfı 98 örnek ve ikinci sınıfı sadece 2 örnek içeren bir veri setinden öğrensin. Eğer sınıflandırıcı tarafından tüm veri birinci sınıf olarak etiketlenmiş ise, 98% doğruluk elde ederdi [1]. Bu örnekten dengesiz veri probleminin çözülmesi gereken önemli bir problem olduğu net olarak görülmektedir. Bu tez çalışmasında, özellikle dengesiz ikili sınıflandırma problemi ile ilgilenilmiş ve bu sorunu çözmek için mevcut yöntemler araştırılmış ve ayrıca daha etkin bir yöntem geliştirilmeye çalışılmıştır.

Sınıflandırıcıların performansını ölçmek ve karşılaştırmak için doğru bir ölçünün kullanılması gerekir. Doğruluk veya hata oranı, öğrenilen karar fonksiyonu çıkışı ve gerçek sınıf etiketi arasındaki farkı hesaplayan klasik performans ölçüleridir. Genel olarak, yüksek doğruluk (veya düşük hata oranı), daha iyi bir sınıflandırma modeli demektir, ancak öğrenme ve tahmin sürecinin zaman ve uzay karmaşıklığı gibi diğer performansları da aynı zamanda önemlidir. Normalde, bir sınıflandırıcıyı eğitmek için sınıf etiketi olan veriye eğitim veri seti denir, sınıflandırıcıyı değerlendirmek için ayrılmış veri ise test veri seti olarak adlandırılır. *Dengesiz ikili sınıflandırma problemi:* Dengesiz ikili sınıflandırma probleminde iki sınıftan birinin diğer sınıfa göre daha çok sayıda örneği bulunur. Dengesiz veri setlerinde, daha fazla örneğe sahip olan sınıfa çoğunluk sınıfı ve daha az örneğe sahip olan sınıfa azınlık sınıfı denir. Verilerin "dengesiz" olarak adlandırılması için net bir eşik değer yok. Genelde araştırmacılar arasında dengesiz öğrenme sorunu incelenirken, çoğunluk sınıf örnekleri sayısının azınlık sınıf örneklerinden belirgin olarak daha fazla olması gerektiği düşünülmektedir. İkili sınıflandırmada veri dengesizliği derecesini ölçmek için standart bir terim tanımlanmıştır. *Dengesizlik oranı* olarak tanımlanan bu terim çoğunluk sınıf ve azınlık sınıf boyutu arasındaki oran olarak tanımlanır:

$\beta = N^- : N^+$

N^+, azınlık sınıf örneklerinin sayısı ve N^-, çoğunluk sınıf örneklerinin sayısını temsil eder.

Açıkçası, β her zaman 1'den daha büyüktür. Dengesizlik oranı ne kadar büyükse, veri o kadar çarpık demektir.

Veri dengesizliğinin büyük olduğu durumlarda, sınıflandırma başarımı için geleneksel değerlendirme ölçüleri (doğruluk veya hata oranı), dengesiz öğrenme sonuçlarını ölçmek için yeterli değildir. Örneğin, 19:1 dengesizlik oranı ile bir veri kümesi verilsin, öğrenme yöntemi her şeyi yalnızca çoğunluk sınıfına atayarak yine % 95 doğruluğa ulaşabilir. Açıkçası, böyle bir değerlendirme ölçüsü pratik durumda işe yaramaz. Dengesiz veriden öğrenme için hem azınlık hem de çoğunluk sınıflarının doğruluğunu dikkate alan yeni performans ölçüleri gereklidir. Neyse ki, bu tür ölçüler önceden literatürde sunulmuştur. Örneğin G-ortalama, AUC (Area Under the Curve) ve F-ölçü, bunlar sonraki bölümlerde daha detaylı olarak tartışılacaktır.

Dengesiz veri kümelerinden öğrenirken, makine öğrenme algoritmaları çoğunluk sınıfı üzerinde yüksek öngörü doğruluğunu üretme eğilimindedir, ancak esas ilgi alanı olan azınlık sınıfı üzerinde zayıf öngörü doğruluğu üretmektedir.

Sınıf dengesizliği sorunu için yaklaşımlar iki ana bölüme ayrılır: veri seviyesinde ve algoritma seviyesinde yaklaşımlar. Veri seviyesindeki yöntemler sınıf dengesizliğini dengeli hale getirmek için dengesiz veri kümeleri dağılımını değiştirir ve daha sonra, oluşan dengeli veri setlerini, azınlık sınıfının tespit oranını artırmak için öğreniciye verir. Veri seviyesindeki yöntemlere örnek olarak yeniden-örnekleme (re-sampling), boosting ve bagging verilebilir. Eğitim verilerini yeniden-örnekleme, sınıf dengesizliği sorunu için popüler bir çözümdür ve veri setinin sınıf dağılımını dengelemek için azınlık sınıfını aşırı-örnekler veya çoğunluk sınıfını alt-örnekler. İkinci yaklaşım algoritma seviyesinde olup, mevcut veri madenciliği algoritmalarını değiştirir veya dengesizlik sorununu gidermek ve sınıflandırma performansını artırmak için yeni algoritmalar geliştirir. Ana fikri, sınıflandırıcının endüktif önyargısını içten ayarlamaktır [2]. Bu kategori, maliyet-duyarlı öğrenme (cost-sensitive learning) [3], kernel-tabanlı algoritmalar (kernel-based algorithms) [4] ve tanıma tabanlı algoritmaları (recognition based algorithms) [5] içerir.

Yeniden-örnekleme tekniği, eğitim veri setlerini dengelemek için üç gruba kategorize edilir. Birincisi, azınlık sınıf örneklerinin aşırı-örneklenmesi, ikincisi, çoğunluk sınıf örneklerinin alt-örneklenmesi ve üçüncüsü, aşırı-örnekleme ve alt-örnekleme yöntemlerinin kombinasyonu olan sezgisel yeniden-örnekleme yöntemleridir. Alt-örnekleme yöntemleri eğitim setinden çoğunluk sınıf örneklerini rasgele kaldırır. Aşırı-

örnekleme ise alt-örnekleme yaklaşımının tam tersidir. Bu, dengesizliği azaltmak için azınlık sınıfı örneklerini çoğaltır.

Topluluk öğrenme (Ensemble learning) yöntemleri, makine öğrenmede yüksek başarı gösterdiğinden , sınıf dengesizliği sorunlarını ele alan başka bir önemli teknik haline gelmiştir. Topluluk öğrenmenin ana fikri, aynı görev üzerinde öğreniciler grubu oluşturmak ve daha sonra onları, daha iyi sınıflandırıcı oluşturmak için birleştirmektir. Özellikle, bagging ve boosting literatürdeki en popüler topluluk öğrenme yöntemleridir. Onlar sınıf dengesizliğini öğrenmede geliştirilmiş ve yaygın olarak kullanılmıştır, örneğin BEV [6], SMOTEBoost [7], RareBoost [8]. Bireysel açıdan bakıldığında, bir topluluk öğrenme, diğer sınıf dengesizliği öğrenme teknikleri ile kolayca entegre edilebilir. Topluluk açısından bakıldığında, birçok sınıflandırıcılar birleştirilerek öngörüyü stabilize edebilir ve daha iyi genelleme elde edebilir.

Boosting, sınıf dengesiz olduğunda herhangi bir zayıf sınıflandırıcının sınıflandırma performansını artırabilen topluluk-tabanlı öğrenme algoritmasıdır. Boosting'de topluluk sınıflandırıcıları, önceki sınıflandırıcıların performansına göre uyarlamalı ayarlanan eğitim örnekleri üzerindeki ağırlıklar ile seri olarak eğitilir. Ana fikri, sınıflandırma algoritmasının, öğrenmesi zor olan örnekler üzerine konsantre olmasıdır. Yanlış sınıflandırılmış örneklere, sonraki iterasyonun yeniden-örnekleme işlemi sırasında eğitim alt kümesinin üyesi olarak seçilmeleri için daha büyük bir ağırlık verilecektir. Son sınıflandırıcı, ağırlıklı oy şeması kullanılarak oluşturulur; her sınıflandırıcının ağırlığı, onu oluşturmak için kullanılan eğitim seti üzerindeki performansa bağlıdır.

Boosting sırasında meydana gelen ağırlık değişiklikleri iki şekilde uygulanabilir: yeniden-ağırlıklandırma veya yeniden-örnekleme. Yeniden ağırlıklandırma tarafında boosting'i gerçekleştirirken, her bir örnek için sayısal ağırlıklar temel öğreniciye doğrudan geçirilmektedir. Temel öğrenici, ağırlık bilgilerini kendi hipotezini oluştururken kullanır. Ancak, tüm öğrenme algoritmaları örnek ağırlığı bilgisini işlemek için tasarlanmamıştır. Bu nedenle, alternatif bir yaklaşım olan yeniden-örneklemeli boosting daha uygundur. Öğreniciye örnek ağırlıklarını geçirmek yerine eğitim verileri bu ağırlıkları yansıtacak şekilde yeniden-örneklenebilir. Orijinal ile aynı boyutta olan yeni bir eğitim veri kümesi, orjinal eğitim veri kümesinden örnekleme ile oluşturulur,

burada her örneğin seçilme olasılığı kendisine atanan ağırlık ile orantılıdır. Bu yeniden örneklenmiş eğitim verileri üzerinde oluşan bir model, bu nedenle daha yüksek ağırlıklar atanan örneklere daha çok önem verir, çünkü bunlar eğitim verilerinde daha sık görünür.

Birçok öğrenici kombinasyonu sayesinde, öngörülerdeki farklılıkların belli bir seviyesini korumak topluluğun başarısı için teorik ve ampirik açıdan önemlidir. Özdeş model kümelerini birleştirerek hiçbir kazanç elde edilemez olduğu açıktır. Buna ek olarak, sınıf dengesizliği öğrenmede, mevcut topluluk yöntemlerinin çoğu veri seviyesi teknikleri ve bazı bilinen sakıncalarından zarar gören maliyet-duyarlı öğrenme yöntemleri ile sınırlıdır. Bu nedenle, sınıf dengesizliği öğrenme bağlamında topluluklar etkisinin araştırılması için derinlemesine bir çalışma yapılması gereklidir.

1.1 Literatür Özeti

Genel olarak, sınıf dengesizliğini öğrenmek için önerilen çözümlerin büyük bir bölümü, literatürde yeniden-örnekleme teknikleri, tek-sınıf öğrenme algoritmaları, maliyet-duyarlı öğrenme yöntemleri ve topluluk yöntemlerini öğrenmede yapılmıştır. Bu tez çalışmasında, sınıf dengesizliği sorunları ile başa çıkmak için basit ve etkili bir yol oluşturmak amacıyla yeniden-örnekleme teknikleri ve topluluk öğrenme yöntemleri üzerinde odaklanılmıştır.

Sınıf dengesizliğini öğrenme son yıllarda birçok araştırmacının ilgisini çekmiştir. Bu alanla ilgili çeşitli özel çalışmalar yapılmıştır. Dengesiz ikili sınıflandırma ve çoklu sınıflandırma sorunlarını çözmek için literatürde birçok yöntem geliştirilmiştir. Dengesiz sınıflandırma algoritmaları ile ilgili kapsamlı bir karşılaştırma yapmak için, bu yöntemler dengesizlik sorununu çözebilme yeteneklerine göre iki ana sınıfa ayırılmıştır. Birinci grupta, dengesizlik, veri seviyesinde olup veri dağılımını yeniden dengeleme veya yeniden ağırlıklandırma ile çözülür. İkinci grupta önerilen yöntemlerde algoritma seviyesinde çözüm aranır ve, burada odaklanılan konu temel öğrenicinin içindeki indüksiyon veya öğrenme aşaması üzerinde olur.

Çalışmaların büyük bir bölümünde yeniden-örnekleme teknikleri ve bunların kombinasyonlarının etkinliği araştırılmıştır. Literatürde yeniden- örnekleme tekniklerini kullanan ve çok dengesiz veri setleri için geliştirilmiş sınıflandırma başarımını

arttırmaya yönelik çalışmalar mevcuttur [9]. Veri-seviyesinde yeniden- örnekleme teknikleri, öğrenme algoritmalarından bağımsız olarak çalışır, ve bu yöntemler literatürde daha yaygın olarak kullanılır. Ancak, veri-seviyesinde yeniden örnekleme işleminde veri miktarının ayarlanması kolay değildir. Çalışmalarda hangi yeniden-örnekleme yönteminin daha iyi performans verdiği ve hangi örnekleme oranının kullanılması gerektiği net değildir. Bazı ampirik çalışmalar tam denge olması için yeniden- örneklemenin her zaman iyi sonuç vermediğini, ve sınıflama başarımının yeniden-örnekleme oranı ve yeniden-örnekleme tekniklerine göre değiştiğini bulmuşlardır [10]. Onların performansı da farklı sınıflandırıcılar ve değerlendirme ölçülerine bağlıdır [11].

1.1.1 Yeniden - Örnekleme Yöntemleri

Sınıf dengesizliği öğrenme için temel yaklaşımlardan biri olan yeniden-örnekleme yöntemleri, eğitim verisi dağılımını doğrudan ayarlayan, veri-seviyesinde yer alan bir tekniktir. Yeniden- örnekleme, azınlık sınıfına yeni örnekleri ekleyerek aşırı-örnekleme veya çoğunluk sınıfından varolan örnekleri kaldırarak alt-örnekleme yapar. Alternatif olarak bu yöntemlerin ikiside kullanılarak veri dağılımı dengelenmeye çalışılır. Aşırı-örnekleme, dengesizlik ortadan kalkana kadar azınlık sınıfı örneklerinin sayısını arttırır. Alt-örnekleme, aynı şekilde veri seti nispeten dengeli olana kadar çoğunluk sınıfının bazı örneklerini kaldırır. Azınlık sınıfını büyütme ve çoğunluk sınıfını küçültmenin çeşitli yolları vardır. Bu yöntemler eğitim veri kümesindeki çoğunluk ve azınlık sınıflarının önceki dağılımlarını doğrudan değiştirir.

Rasgele örnekleme, iki sınıf arasındaki istenen dağılımı oluşturmak için basit bir rastgeleliği kullanır. Rasgele aşırı-örnekleme azınlık sınıfı örneklerini rasgele çoğaltır ve onları eğitim setine geri katar. Bir çoğaltma yüzdesi δ çoğaltılmış örnekler miktarını kontrol etmek için kullanılabilir, örneğin δ=100%, N^+ azınlık sınıf örneklerinin sayısını göstermek üzere, azınlık sınıfı için çoğaltma sayısı $2*N^+$ örnek sayısı anlamına gelir. Bu nedenle, genel dengesizlik oranı $\beta = N^- : (1+\delta) * N^+$, dengeli bir değere ulaşmak için δ tarafından ayarlanabilir. Diğer taraftan, rasgele alt-örneklemede çoğunluk sınıfından belli sayıda örnek rasgele seçilir, ve sonra bunlar yeni bir eğitim veri seti oluşturmak için azınlık sınıfı ile birleştirilir, sonra da standart sınıflandırma algoritmaları bu veri

setine uygulanır. Çoğunluk örneklerinin sadece γ yüzdesi seçildiğini varsayalım, yeni dengesizlik oranı $\beta = \gamma * N^- : N^+$ olur. Burada uygun örnekleme oranını seçmek dengeli veri setleri oluşturmayı mümkün kılar. Rasgele alt-örneklemenin getirdiği bilgi kaybını azaltmak için birçok gelişmiş strateji önerilmiştir.

1.1.1.1 Alt-örnekleme Yöntemleri

Rasgele alt-örnekleme (Random Under Sampling-RUS). Bu, azınlık ve çoğunluk sınıfları makul bir boyuta erişene kadar çoğunluk sınıfından verileri rasgele kaldıran sezgisel olmayan bir yöntemdir. Bu veri azaltma işlemi belki de sınıflama ile ilgili yaralı bilginin de atılmasına neden olabilir [12].

Tomek bağlantıları. Z_{min}'de x_i ve Z_{maj}'da x_j iki örnek verilmiş, d (x_i, x_j) onların arasındaki mesafe anlamına gelir. Eğer d (x_i, x_k) < d (x_i, x_j) veya d (x_j, x_k) < d (x_i, x_j) gibi herhangi bir x_k örneği yoksa, (x_i, x_j) çiftine Tomek bağlantısı denir. Alt-örnekleme yönteminde, bağlantı içindeki sadece çoğunluk sınıfına ait olan örnekler kaldırılır. Bu sınıflar arasındaki örtüşmeyi temizleyebilir ve iyi tanımlanmış sınıflama kurallarını oluşturabilir. Tomek bağlantılarını bulmak, hesaplama açısından işlem karmaşıklığı yüksek, zaman alan bir işlemdir [13].

Wilson'ın Düzenlenen en yakın komşu (Edited Nearest Neighbour- ENN) kuralı ve Komşuluk temizleme kuralı (Neighbourhood Cleaning Rule NCR). ENN, sınıf etiketi, üç yakın komşusunun en az ikisinden farklı olan herhangi bir örneği kaldırır. Bu fikre dayalı olarak, NCR, 3-NN tarafından yanlış sınıflandırılan çoğunluk sınıf örneklerini kaldırır ve gürültü olarak kabul eder. Bu arada, bir azınlık sınıfı örneği 3-NN tarafından yanlış sınıflandırılsa, onun çoğunluk sınıfına ait komşuları kaldırılır [14].

K. Nageswara Rao et al. [15] sınıf dengesizliği problemini çözmek için en iyi görselleştirme kümeleme tekniklerinden (visualization clustering technique) biri olan OPTICS'i kullanan alt-örnekleme yaklaşımını sunmuştur. Majority sınıfı ünlü OPTICS kümeleme tekniğini kullanarak alt-örneklenir. Çoğunluk veriseti üzerinde farklı kümeleri tanımlamak için kümeleme algoritması yugulanır. OPTICS sonucu çoğunluk verisetinde kümeler sayısın tanımlamak için kullanılır. Zayıf veya aykırı kümeleri tanımlamak ve onları çoğunluk alt kümesinden silmek gerekir. Silme işleminin miktarı

verisetinin benzersiz özelliklerine bağlıdır. Zayıf veya aykırı kümeleri çıkardıktan sonra yeni bir çoğunluk alt kümesi oluşur. Yeni çoğunluk alt kümesi ve azınlık alt kümesi yeni ve büyük bir olasılıkla dengeli veriseti oluşturmak için birleştirilir. Bu yeni oluşan dengeli veri seti temel algoritmaya uygulanır.

CILIUS. Naganjaneyulu Satuluri et al. [16] sınıf dendesizliğinin giderilmesi için yeni CILIUS (Class Imbalance Learning Using Intelligent Under-Sampling) algoritmasını geliştirmiştir. Alt-örneklemenin calışma tarzı zayıf veya gürültülü örnekler sayısını azaltmaya çalışır. Burada, iyi-kurulmuş (well-established) filtre ve akıllı tekniğe göre tanımlanan belirli 'k' özellikle ilgili zayıf örnekler ortadan kaldırılır. Önce sınıf içindeki gürültü niteliğindeki örnekler kaldırılır. Sonra sınır (borderline) örnekleri ve aykırı (outlier) örnekler kaldırılır. İlk olarak, güçlü örneklerden oluşmuş N_{strong} veri setinden bazı örnekler geçici silinir. Sonra, N_{strong} içindeki tüm örnekler sınıflandırılır. Eğer sınıflandırma doğru ise ve doğruluk artıyorsa, o zaman geçici silinmiş örnekler *sınıf etiket ğürültü özelliği* olarak sayılır. Sınıra yakın örnekler farklı sınıflar arasındaki sınıra yakın örneklerdir. Onlara güvenilmez, çünkü öznitelik gürültüsünün küçük bir miktarı bile örneği sınırın yanlş tarafına gönderebilir. Aykırı örnekler diğer örnekler kümesinden çok farklıdır nadir. Bu örnekler sınıflandırmada performansı arttırmak için kaldırılır. Zayıf örnekleri bulmanın bir yolu, sınıflamada en etkili öznitelikleri veya özellikleri bulmak ve sonra o özellikle ilgili gürültülü örnekleri kaldırmaktır. En etkiliyen öznitelik *öznitelik seçme filtresini* kullanarak bulunur, bu durumda CFS (correlation based feature subset) değerlendirmesi kullanılır. Azınlık kümesi ve güçlü çoğunluk alt kümesi dengeli ve güçlü verisetini oluşturmak için birleştirilir ve temel öğrenme algoritmasında kullanılır.

1.1.1.2 Aşırı-örnekleme Yöntemleri

Aşırı-örnekleme işleminde azınlık örneklerinin çoğaltılması gerekmektedir.

Rasgele aşırı-örnekleme (Random Over Sampling-ROS). Bu Z_{min} kümesindeki rasgele seçilen örnekleri çoğaltan ve onları orijinal azınlık kümesine ekleyen sezgisel olmayan bir yöntemdir. Sınıf çoğaltma yüzdesi, istenilen herhangi bir seviyeye ayarlanabilir. Bu basit, ama tam kopyaların aşırı uyuma (overfitting) yol açabileceği öne sürülmüştür [13].

Sentetik Azınlık aşırı-örnekleme tekniği (SMOTE). Azınlık sınıf örneklerini sadece çoğaltan rasgele aşırı örneklemenin aksine, sentetik aşırı-örnekleme yöntemleri, mevcut azınlık sınıf örneklerinin veri alanını analiz ederek yapay örnekler oluşturmayı dener. İdeal olarak, yeni azınlık sınıf örnekleri de aynı azınlık sınıf kavramlarını temsil etmelidir. Bir çok uygulamada büyük başarı gösteren gelişmiş ve iyi bilinen sentetik yöntemlerden biri SMOTE — yeni örnekler oluşturmak için enterpolasyon tekniği kullanan sentetik azınlık aşırı-örnekleme yöntemidir [19]. Bu, özellik uzayındaki orijinal azınlık sınıfı örnekleri arasındaki benzerliklere dayanarak azınlık sınıfı olarak etiketlenmiş yeni veri noktalarını oluşturur. Özellikle, SMOTE yapay örnekleri oluşturmak için aşağıdaki yolları kullanmaktadır: Önce, azınlık sınıfının bir x_i örneği için, öklid uzaklığına dayalı azınlık sınıfı içindeki k en yakın komşuları bulur; sonra bir x_j komşunu ve [0, 1] aralığında rasgele α float sayıyı seçer, sonra azınlık sınıfına ait yeni sentetik örneği bulur:

$$x_{yeni} = x_i + (x_j - x_i) * \alpha \tag{2.1}$$

Yeni örnekleri Iteratif olarak tek tek oluşturarak, SMOTE sentetik verinin denge oranını kontrol edebilir ve daha sonra öğrenme modellerini oluşturmak için genel sınıflandırmayı kullanır. SMOTE birkaç uygulamada en etkili aşırı-örnekleme yöntemlerinden biri olarak gösterilmiştir [19]. Ancak, aynı zamanda SMOTE 'in aşırı genelleştirme ve aşırı hesaplama gibi kendi dezavantajları da belirtilmiştir [20]. Bu yöntem, çoğunluk sınıfı ile ilgili komşusunu dikkate almaz, bu nedenle sınıflar arasında örtüşme olasılığı ile sonuçlanır. SMOTE'in çeşitli varyasyonları yan sorunların üstesinden gelmek için tasarlanmıştır, örneğin, Borderline-SMOTE [21], Adaptif Sentetik Örnekleme (Adaptive Synthetic Sampling ADA-SYN) [22], Kernel-tabanlı SMOTE (KSMOTE)[5], ve SMOTE ile Tomek link kombinasyonu [13].

Borderline-SMOTE. Sınıflandırma sınırları yakınındaki örneklerin (yani sınırdaki örnekler) doğru sınıflandırılmamış olma olasılığı daha yüksektir ve bu nedenle daha önemli olduğu varsayılır. Sadece sınırdaki örnekler yeni verileri oluşturmak için SMOTE'ı uygulayarak kullanılmıştır. Eğer azınlık sınıfı örneğinin en yakın komşularının çoğu çoğunluk sınıfına ait ise, sınır örneği "tehlike" olarak kabul edilir [21].

Adaptif Sentetik Örnekleme (ADASYN). Yöntemin ana fikri, kendi dağılımlarına göre azınlık sınıfı örneklerini üretmektir, burada daha fazla sentetik veri, öğrenilmesi "kolay" olan örneklere göre daha "zor" olanlar için oluşturulur. Borderline-SMOTE'e benzer olarak, öğrenilmesi "kolay / zor" örnekler, azınlık sınıfı örneğinin komşusu tarafından belirlenmektedir. Z_{min}'deki her bir x_i için, ADASYN k-NN grubu içindeki çoğunluk sınıfı örnekleri oranını Δ_i hesaplar. $\Sigma\Delta_i = 1$ olması için Δ_i' i normalize edilir. Daha sonra, x_i için oluşturulması gereken sentetik örneklerin sayısı Δ_i * L, buradaki L, yeni verilerin istenilen miktarı (|Smin| - |Zmin|). Δ, otomatik olarak sentetik örneklerin sayısına karar veren bir yoğunluk dağılımını açıklar [22].

EasyEnsemble ve BalanceCascade. Liu et al., çoğunluk sınıf örneklerinin kullanılmasını güçlendirmek için EasyEnsemble ve BalanceCascade denilen iki topluluk yöntemini önermiştir [17]. Onlar toplulukların bir alt topluluğunu oluşturmuştu, burada her bir sınıflandırıcı aynı zamanda AdaBoost topluluğudur, ancak farklı örnekleme stratejilerini kullanır. EasyEnsemble çoğunluk sınıf örneklerini rastgele örnekler. Seçilmiş olanları sonra eğitim altkümelerini oluşturmak için tüm azınlık sınıfı örnekleri ile birleştirilir. EasyEnsemble'in her altkümesi AdaBoost sayesinde daha iyi öğrenilmiş olduğuna inanılır. BalanceCascade, sınıflandırılmamış olanlara daha fazla dikkat vererek denetimli bir şekilde çoğunluk sınıfını araştırır. Burada, sınıflandırıcılar sırayla oluşturulur, bazı doğru olarak sınıflandırılmış çoğunluk sınıf örneklerinin "gereksiz" olduğu kabul edilerek eğitim setinden kaldırılır.

1.1.2 Sınıf Dengesizliğini Öğrenme İçin Boosting Yöntemleri

Geleneksel boosting (örneğin, AdaBoost) algoritması genel sınıflandırma görevleri için çok başarılı olduğundan sınıf dengesizliğini öğrenme için boosting stratejisinin kullanılması çok yaygınlaşmıştır [23, 24]. Dengesizliği öğrenmede bagging ve boosting arasındaki en büyük fark boosting'in gelişmiş sınıflandırıcıları sırayla oluşturmasıdır. AdaBoost, Boosting ailesindeki en tanınmış algoritmadır. Boosting, ardışık olarak dizili sınıflandırıcılardan oluşur ve, önceki sınıflandırıcılar tarafından yanlış sınıflandırılmış olan eğitim örnekleri üzerinde durur. Ağırlık seti, her iterasyon sonunda geçerli sınıflandırıcının doğruluğuna göre güncellenen eğitim seti içinde tutulur ve bir sonraki iterasyonda geri besleme için kullanılır [25]. Doğru sınıflandırılmamış örnekler

üzerindeki ağırlık, AdaBoost'ı doğruluk odaklı algoritma yapmaktadır. Yüksek doğruluğa sahip sınıflayıcılar yüksek ağırlıklar alır. Bu yöntem tüm sınıflara eşit davranır. Dengesiz veri durumunda, boosting'in öğrenme stratejisi çoğunluk sınıfına karşı önyargı eğilimindedir. Azınlık sınıfına ait örnekler genel sınıflandırma doğruluğuna daha az katkı sağlar [23]. Bu nedenle, orijinal AdaBoost azınlık sınıfına ait örnekleri tanımada iyi değildir. Buna ek olarak, bu aşırı uyum (overfitting) sorununu ortaya çıkarabilir. Azınlıık sınıfına ait örnekler genel olanlara göre fazla ağırlıklı olma eğilimindedir [2].

Bununla birlikte, AdaBoost, sınıf dengesizliğini öğrenme alanında halen popüler bir yöntemdir ve avantajları aşağıda açıklanmıştır. İlk olarak, boosting yöntemleri birçok sınıflandırma algoritmaları için geçerlidir. İkincisi, iyi sınıf dağılımı ve temsili örnekler, ekstra hesap maliyeti olmadan otomatik olarak incelenmektedir. Üçüncü olarak, çoğunluk sınıf örneklerininin hiçbirini ortadan kaldırmak gerekmez [26]. Bu ümit vadeden özellikler, AdaBoost'u yeniden-örnekleme teknikleri ile birleştirme [27] ve maliyet-duyarlı Boosting [12] özelliklerini içerecek şekilde evrimleştirmeye yöneltmiştir.

Boosting yeniden- örnekleme tekniği ile birleştiğinde veri üretme yöntemleri azınlık sınıfı için karar bölgesini genişletmek için kullanılır. Örneğin, N.V. Chawla et al. dengesiz veri setlerinden öğrenme için SMOTEBoost yaklaşımını sunmuştur [7]. Önerilen SMOTEBoost algoritması, standart boosting prosedürü ile SMOTE algoritmasının entegrasyonudur. SMOTEBoost, tüm yanlış sınıflandırılan örneklere eşit ağırlık veren standart boosting yönteminin aksine, nadir veya azınlık sınıfından sentetik örnekler oluşturmakta, böylece güncellenen ağırlıkları dolaylı değiştirmekte ve çarpık dağılımları dengelemektedir. Azınlık sınıfların tahminini iyileştirmek için SMOTE'u kullanmaktadır.

Mease et al. daha hızlı bir JOUS-Boost yöntemini önermiştir [28]. Azınlık sınıfı için yeni örnekler üretmek yerine, Boosting'in her iterasyonunda aşırı-örnekleme işlemi tarafından oluşan kopyalara (replicates) az miktarda rasgele gürültüler ekler. Bu, etkili sonuçlar göstermektedir, ancak "gürültünün" kullanılması nedeniyle aşırı genelleme sorunu bulunmaktadır.

Maliyet-duyarlı Boosting, sınıf dengesizliği öğrenme için topluluk çözümlerinin başka bir sınıfıdır. Ana fikri, ağırlığı güncelleme kuralını işleyerek her iterasyonda maliyetli

(costly) örneklerin seçilme olasılığını artırmaktır. Bunlar sınıflandırma hatası yerine toplam yanlış sınıflandırma maliyetini minimize etmeyi amaçlar, örneğin AdaCost [3], CSB1 ve CSB2 [29]. Maliyet matrisi veya ilişkili maliyet öğelerinin öğrenme öncesinde belirtilmesi gerekir. Ancak, birçok durumda belli bir maliyet belirlemek zordur. Bu sorunu aşmak için, bir maliyet duyarlı Boosting yöntemi RareBoost önerilmiştir [8]. Onun maliyet özelliği, TP (gerçek pozitif), FP (yanlış pozitif), TN (gerçek negatifler) ve FN (yanlış negatif) örneklerinin dört türünün oranlarına dayalı mevcut sınıflandırıcı performansıyla dinamik olarak belirlenir. TP> FP ve TN > FN kısıtı gereklidir. Bazen, bu azınlık sınıfı için oldukça güçlü bir durumdur. Maliyet-duyarlı Boosting da azınlık sınıfı örneklerinin aşırı uyum problemi ile karşılaşabilir.

RUSBoost. SMOTE-Boost karşısında RUSBoost stratejisi boosting ile rasgele alt-örneklemeyi birleştirir. Seiffert et al. göre, dengesizlik oranı çok büyük olduğu zaman SMOTE-Boost daha karmaşık ve zaman alıcıdır [30]. Bu nedenle, eğitim süresini azaltmak için alt-örnekleme ve bilgi kaybını önlemek amacıyla boosting'in kullanılmasını önerilmiştir. RUSBoost'ta her boosting döngüsünde, yeni bir eğitim setinin oluşturulması için azınlık sınıfı ile birleştirilmek üzere çoğunluk sınıfının küçük bir bölümü rastgele alt-örnekleme ile seçilir; sonra temel öğrenici eğitilir ve ağırlıkları öğrenme doğruluğunda güncellenir. Seiffert et al. RUSBoost, SMOTE-Boost, AdaBoost ve SMOTE ile kapsamlı bir karşılaştırma yapmış ve RUSBoost'in tipik olarak SMOTEBoost gibi iyi ya da daha iyi performans verdiğini göstermiştir [30].

DataBoost-IM. Hongyu Guo, Herna L Viktor sentetik aşırı örnekleme ile boosting yöntemini birleştiren hibrit bir yöntem olan DataBoost-IM'i sunmuştur [31]. SMOTE-Boost ile karşılaştırıldığında, DataBoost-IM, hem çoğunluk ve hem de azınlık sınıfları için yeni örnekler sentezlerler, ancak azınlık sınıfı için daha fazla örnek sentezlerler. DataBoost-IM yönteminde çoğunluk ve azınlık sınıflarında ki öğrenilmesi zor (hard) örnekler boosting algoritmasının yürütülmesi sırasında tespit edilmektedir. Daha sonra, zor örnekler çoğunluk ve azınlık sınıfları için ayrı ayrı sentetik örnekler üretmek için kullanılmaktadır. Sentetik veriler daha sonra orijinal eğitim kümesine ilave edilmekte ve yeni eğitim kümesinin farklı sınıflarının toplam ağırlıkları ve sınıf dağılımı dengelenmektedir. Sonra, temel öğrenici bu yeni eğitim setine uygulanır ve hata oranı ve ağırlık dağılımı buna göre yeniden hesaplanır.

1.2 Tezin Amacı

Sınıf dengesizliği sorunu tıbbi teşhis, dolandırıcılık tespiti, metin sınıflandırma, risk yönetimi dahil olmak üzere pek çok pratik uygulamada karşılaşılan yaygın bir problemdir. Veri setlerinin dengesiz olması, düşük sınıflandırma performansına neden olabilir. Yetersiz bilgi bazı alanlarda içseldir (intrinsic). Örneğin, belirli bir veri kümesinde, kanser olan hastaların sayısı kanser olmayan hastaların sayısı ile karşılaştırıldığında genelde çok azdır. İçsel olmayan (non-intrinsic) alanlar da vardır, bu gizlilik ve harcama (expenditure) gibi bazı nedenlerden dolayı veri toplamanın zor olduğu durumlarda görülür. Her iki durum da azınlık verisini analiz etmeye daha fazla zaman harcamanın gerekli olduğu durumlardır. Bu sebeplerle tezin ana konusu olan sınıf dengesizliğinin giderilmesi problemi ve bunun çözüm yollarını ortaya koymak amacı ile tezin hedefleri belirlenmiştir.

Tez şu noktaları hedeflemektedir:

- İki sınıflı sınıflandırma için dengesizlik sorunu üzerinde yeniden-örnekleme ve boosting tekniklerini incelemek.
- Sınıf dengesizliği sorununu hafifletmek için veri düzeyindeki mevcut RUSBoost, BalanceCascade ve EasyEnsemble algoritmaları karşılaştırarak analiz etmek.
- Temel öğrenici olarak C4.5, DVM ve KNN gibi çeşitli sınıflandırıcıları kullanarak deneyler yapmak ve en iyi performanslı temel öğreniciyi bulmak.
- Çoğunluk ve azınlık sınıfları dağılımına göre en iyi performansa sahip algoritmayı bulmak.
- Sınıflandırma başarımını değerlendirmek amacıyla sınıf dengesizliği sorunları için uygun AUC, F-ölçü ve G-ortalama ölçülerini kullanmak.
- Ek olarak sınıf dengesizliği sorununu gidermek için yeni bir yöntem önermeye çalışmak.

1.3 Hipotez

Azınlık sınıf örneklerini doğru tahmin etmek için topluluk modellerinin, dengesiz veri setlerinde nasıl bir katkıda bulunduğunu incelemek faydalı olacaktır. Topluluklar,

lıteratürde birçok çalışmada zayıf sınıflandırma algoritmalarının performansını artırma yeteneği ile umut verici bir teknik olarak ortaya çıkmıştır. Sınıflandırıcılar topluluğunun, genellikle tek bir sınıflandırıcıya göre daha iyi performans verdiği keşfedilmiştir. Sınıflandırıcılar topluluğunun nasıl oluştuğu ve neden topluluk tekniklerinin iyi çalıştığı üzerinde bir dizi araştırmalar vardır [33]. Ayrıca, topluluk teknikleri, temel öğrenici olarak farklı pratik uygulamalarını amaçlayan çesitli öğrenme algoritmalarını kullanır.

Sınıf dağılımını dengelemek için ortak tekniklerden biri olan yeniden-örnekleme teknikleri, uygun eğitim verisinin yokluğunda mevcut verilerin rasgele alt kümelerini oluşturmak için topluluk modelleri ile kolayca birleştirilebilir. Örneğin, Boosting'in veri altkümelerini seçmek için yeniden-örnekleme prosedürü vardır. Onların diğer yöntemlere göre avantaji, dişsal ve kolay taşınabilir olmasıdır. "Dışsal", algoritmaya-bağımsız veya algoritmaya özgü değil anlamına gelir. Bu nedenle, topluluk modeli sınıf dengesizliği için iyi bir seçim olur.

Sadelik (simplicity) ve doğruluk arasında her zaman bir denge vardır. Yeniden-örnekleme, basit ve algoritmadan-bağımsız olduğu için yaygın kullanılır ve sınıf dengesizliği sorununda etkilidir. Literatürde bu konuda önerilmiş çok sayıda teknik bulunmaktadır [7, 13]. Bir yöntemin diğerlerinden daha iyi bir yöntem olduğuna karar vermek kesinlikle zordur. Bu tezde, azınlık sınıfını daha doğru sınıflandırmak amacıyla yeniden-örnekleme ve boosting'i kullanmak düşünülmüştür.

Çalısmanın ikinci bölümünde veri madenciliği ve veritabanlarında bilgi kesfi süreci içinde yer alan temel kavramlar, yöntem ve teknikler hakkında bilgi verilmiştir. Üçüncü bölümde sınıf dengesizliğini öğrenme sorunu ve onu çözmede geliştirilmiş yöntemler hakkında bilgi verilmiştir. Dördüncü bölümde ise karşılaştırmada kullanılan algoritmalar tanımlanmış, deneysel düzen sunulmuş ve mevcut farklı algoritmalar ile elde edilen deneysel sonuçlar verilmiştir. Beşinci bölümde tez kapsamında bizim tarafımızdan önerilen RusAda algoritması sunulmuştur ve çalışmanın sonunda tez sonucuna varılmıştır.

BÖLÜM 2

VERİ MADENCİLİĞİ

2.1 Veri Madenciliği Kapsamı

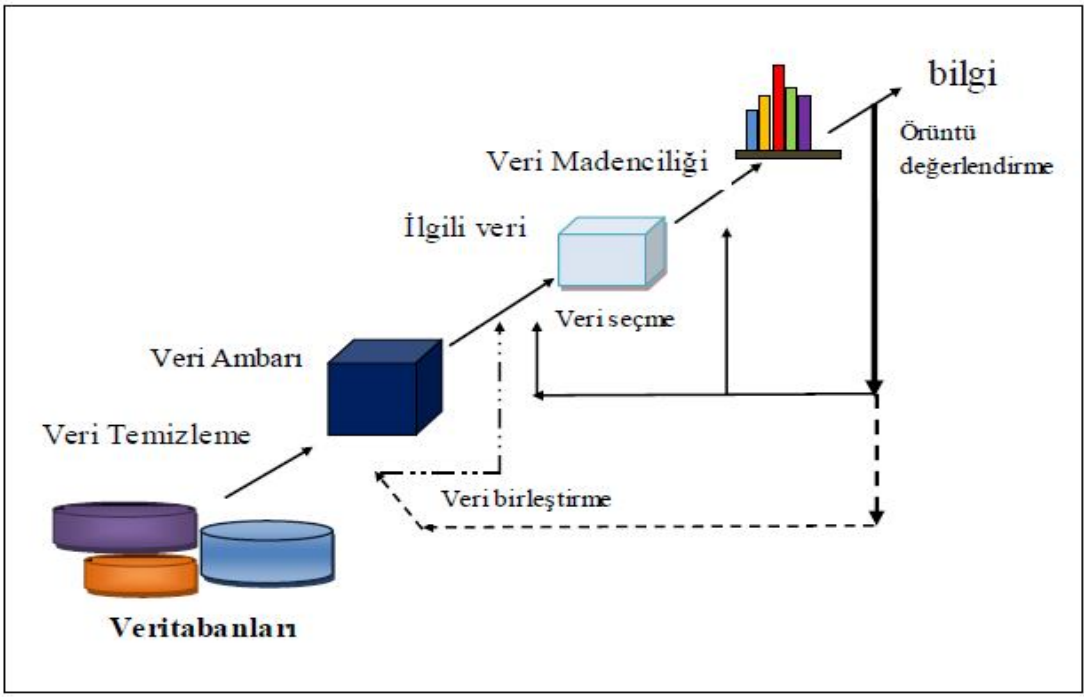
Sürekli büyüyen sayısal veri ortamlarında yararlı ve de gerekli olan bilgiye ulaşmayı sağlamak için çalışmalar her geçen gün artmaktadır. Bilgisayar ve iletişim teknolojilerindeki gelişmeler verilerin uzun süre depolanabilmesini, dolayısıyla büyük kapasiteli veritabanların oluşmasını sağlamıştır. Bu nedenle büyük veritabanlarında istenilen anlamlı ve kullanılabilinir bilgiye erişmek yeni bir disiplin olan Veri Madenciliği'nin doğmasına sebep olmuştur [34]. Veri madenciliği bu safhada öne çıkan ve önemi gitgide artan bir olgudur.

Veri madenciliği kendi başına bir çözüm değil çözüme ulaşmak için verilecek karar sürecini destekleyen, problemi çözmek için gerekli yöntemleri sağlamaya yarayan bir araçtır. Bilgiyi üreten ve kullanan yerler hızla gelişip sürekli ilerlediği için bilginin daha etkin bir şekilde yönetilmesi gerekir. Veri madenciliği de bu amaca ulaşmak için kullanılan bir tekniktir. Genel anlamıyla veri madenciliği, verileri analiz etme, büyük ölçekli veri setlerinden anlamlı bilgilere ulaşma ve ulaşılan bu bilgileri yararlı olacak şekilde özetleme işidir.

Veri madenciliği, makine öğrenimi, örüntü tanıma, istatistik veritabanları ve büyük veritabanlarından bilginin çıkarılması konusunun ele alınması için görselleştirme teknikleri ni bir araya getiren bilimler arası bir alandır.

Veri madenciliğinin temelinde, veri önişleme, verinin analiz edilmesi, veri ambarlarının benzerliklerinin ve ilişkilerinin çıkarılması için bilgisayar programlama teknikleri ve istatistiksel metotların uygulanması gibi işlemler vardır. Anlamsız verilerden anlamlı

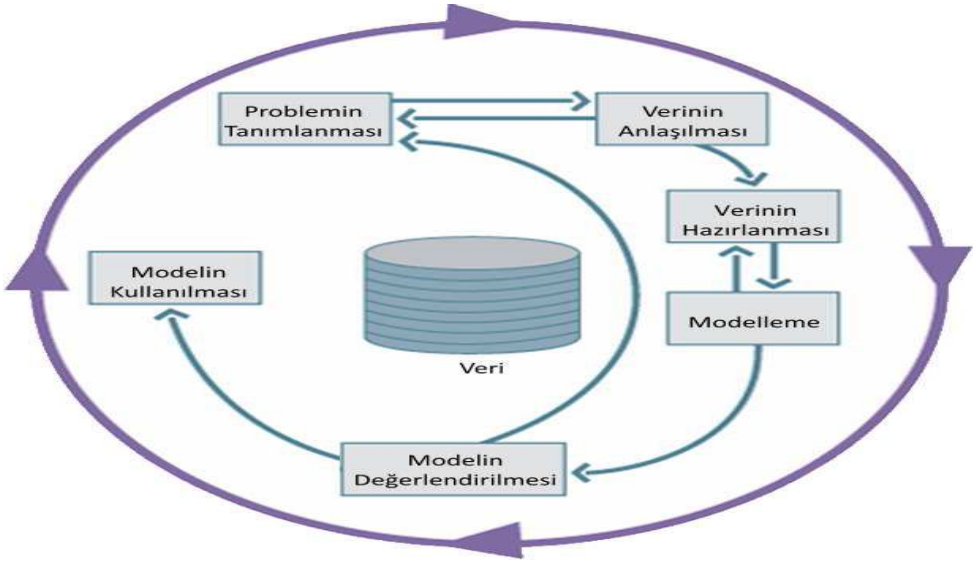
bilginin ortaya çıkarılması için veriler birçok işleme tabi tutulurlar ki bu işlemler veri madenciliğinin aşamalarını oluşturur. Veri madenciliği, bilgi keşfi işleminin bir parçasıdır ve onun basamaklarından birisi sayılır. Veritabanlarında özbilgi keşfinin adımları özet olarak Şekil 2.1'de verilmiştir.



Şekil 2.1 Veritabanlarındaki özbilgi keşfinin aşamaları

2.2 Veri Madenciliği Süreci

Veri madenciliği süreçleri konusunda 1996 yılında Daimler Chrysler, SPSS ve NCR'ı temsil eden bir analist grubu yoğun çalışmalar yapmış ve sonuçta CRISP DM ismini verdikleri bir veri madenciliği sureci geliştirmişlerdir. Veri madenciliğinin bir süreç olarak tanımlanabilmesi için, sürecin her bir aşamasının dikkatle izlenmesi gerekmektedir. Bir aşamanın sonucu, diğer bir aşamanın girdisidir. Bu sebeple her aşama bir önceki aşamanın sonuçlarına bağımlıdır. CRISP-DM süreci 6 aşamadan oluşmaktadır. Veri madenciliği, altı adımlı bir süreç olarak incelenebilir [35]. Şekil 2.2'de bu süreç görülmektedir.



Şekil 2. 2 Veri madenciliği süreci

2.2.1 Problemin Tanımlanması

Bu aşama veri madenciliği sürecinin en önemli aşamasıdır. Veri madenciliği çalışmalarında başarılı olmanın ilk şartı, uygulamanın hangi işletme amacı için yapılacağının açık bir şekilde tanımlanmasıdır. İlgili işletme amacı, işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçülebileceğinin tanımlanması gereklidir.

2.2.2 Verinin Anlaşılması

Veri anlama aşaması veri toplamakla başlamaktadır. Daha sonra benzer verileri bir araya getirme, veri niteliklerini tanımlama, verileri keşfetme, gizli bilgileri sınıflandırma ile sürece devam etmektedir.

2.2.3 Verinin Hazırlanması

Bu aşamada işlenmemiş verinin projede kullanılabilecek duruma getirilmesi amaçlanır. Günümüzde veritabanları, büyük boyutlardan ve birçok farklı kaynaktan geldiklerinden dolayı gürültülü, eksik, tutarsız veriler ile doludur. Veri hazırlama aşaması, ham veriden başlayarak son veriye kadar yapılması gereken bütün düzenlemeleri içermektedir. Veri hazırlama tablo, kayıt, veri dönüşümü ve modelleme araçları için veri temizleme gibi özellikleri içermektedir. Bu aşama en çok iş gücü gerektiren ve toplam süreç içinde en

fazla zaman alan aşamadır. Verinin hazırlanması; verinin toplanması, verinin birleştirilmesi ve temizlenmesi ile verinin dönüştürülmesi aşamalarından oluşmaktadır.

Veri Temizleme. Veri Temizleme ile eksik verilerin tamamlanması, aykırı ve gereksiz verilerin silinmesi, gürültülü ve uygunsuz verilerin veri tabanından temizlenmesi işlemleri yapılır.

Eksik Veri. Herhangi bir değişkene ilişkin eksik değerlerin doldurulması için farklı yollar vardır:

- Eksik değer içeren kayıt veya kayıtlar atılabilir.
- Eksik veri manüel olarak tamamlanabilir.
- Eksik veri genel bir sabit ile doldurulur.
- Eksik değer ortalama değer ile doldurulur.

Verilerin Düzeltilmesi. Veri temizleme işleminde gürültülü verilerin düzeltilmesi de gerekmektedir.

Veri Birleştirme. Veri Birleştirme çeşitli veri kaynaklarındaki verilerin birleştirilmesi işlemidir. Bu aşamada, farklı veri tabanlarındaki veriler birleştirilerek tek bir ambarda depolanır.

Veri Dönüştürme. Düzeltme, birleştirme, genelleştirme ve normalleştirme gibi işlemler kullanılarak verinin, veri madenciliği metotları için uygun biçimlere dönüşümünü sağlar.

Veri İndirgeme. Veri madenciliğinde en yüksek performansı elde edebilmek için büyük hacimli veri kümesinden daha küçük hacimli veri kümesinin oluşturulmasıdır. Bu sayede elde edilen indirgenmiş veri kümesine veri madenciliği teknikleri uygulanarak daha etkin sonuçlar elde edilebilir.

2.2.4 Modelleme

Bu aşama, uygun modelleme tekniğinin seçimi, test tasarımının üretimi, model geliştirme ve tahmin işlemlerini içermektedir. Veri madenciliği, her problem tipi için farklı yöntemler sunmaktadır. Bir veri madenciliği problemi için birden fazla teknik

kullanılabilir, problem için uygun olan teknik veya tekniklerin bulunabilmesi için birçok teknik oluşturulup bunların içinden en uygun olanları seçilir.

Genelde Veri Madenciliğinde kullanılan modeller, tanımlayıcı (descriptive) ve tahmin edici (predictive) olmak üzere iki ana başlık altında incelenmektedir. Tahmin edici modellerde amaç, sonuçları bilinen verileri kullanarak bir model geliştirmek ve bu model yardımıyla sonuçları bilinmeyen veri kümeleri ile sonuçların tahmin edilmesini sağlamaktır. Sınıflama (classification), regresyon ve kestirim (prediction) madenciliği tahmin edici tekniklerin bazılarıdır. Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerin tanımlanması sağlanmaktadır. Kümeleme (clustering), birliktelik kuralı (association rule) ve ardışıl örüntü (sequential pattern) madenciliği tanımlayıcı tekniklerden bazılarıdır.

Sınıflandırma (Classification): Bir verinin önceden tanımlanmış birçok sınıfa ayrılmasıdır. Sınıflandırma, daha önce görünmemiş veri kayıtlarına eski verilerden elde edilen modele dayanarak etiket atanmasını içerir. Sınıflandırma işlevinin iş ve araştırma alanlarında kullanım örnekleri şunlardır: Bir kredi kart işleminin sahte olup olmadığının belirlenmesi, ipotek (mortgage) işleminde kredi riskinin iyi veya kötü olduğunun değerlendirilmesi, tıpta belirli bir hastalığın olup olmadığının teşhis edilmesi, belirli bir mali ya da şahsi davranışın bir terörist eylem olup olmadığının belirlenmesi.

Sınıflama modelinde kullanılan başlıca teknikler şunlardır:

- Karar Ağaçları
- Yapay Sinir Ağları
- Genetik Algoritmalar
- K-En Yakın Komşu Metodu
- Regresyon Analizi
- Naive- Bayes Metodu

Kümeleme (clustering): Bir verinin benzerlik matrislerine ya da ihtimal yoğunluk modeline dayalı verilerin doğal gruplandırılması olan çeşitli kümelerden biriyle eşleştirilmesi ve benzer davranıştaki daha küçük alt yığınlara bölmektir. Kümeleme

işlemi hedef bir değişkeni sınıflandırma ya da tahmin etmeye çalışmaz. Bunun yerine bir veri yığınının birbirine benzer alt gruplara ayrılması için çalışır.

Kümeleme analizi, benzer özelliklere sahip bireylerin belirlenip gruplandığı çok değişkenli bir çözümleme tekniğidir. Kümeleme analizi sayesinde dağılımdaki yoğun ve seyrek alanlar belirlenebilir ve farklı dağılım örnekleri uygulanabilir. Kümeleme analizi benzer nesnelerin aynı grupta toplanması olarak da tanımlanabilir. Bu modelde dikkat edilmesi gereken unsur, hangi kriterleri kullanarak kümeleme yapılacağıdır. Bu işlemler konunun uzmanı olan kişiler tarafından tahmin edilir. Veriler kümeleme işleminde aynı grupta ya da farklı gruplarda yer alabilir. Kümeleme tekniği çok sayıda kayıt içeren veritabanlarında iyi bir şekilde uygulanabilir. Bu tür veritabanlarında her bir kayıt belirli bir grupta bir üye olarak sunulur. Kümeleme algoritması aynı gruplara uyan tüm üyeleri bulur. Bu üyeler içerisinde herhangi bir gruba uymayan üyeler de olabilir. Bu üyeler gürültü olarak nitelendirilir. Gürültüler kümeleme algoritmasının gücü açısından önemlidir.

Birliktelik (Association): Hangi niteliğin birleşeceğinin bilinmesi işidir. Iş dünyasında en yaygın olanı ilişki analizi ya da pazar sepet analizi olarak bilinen iki ya da daha fazla nitelik arasındaki ilişkinin belirlenmesi işlevidir.

Birliktelik kurallarının amacı, büyük veri kümeleri arasından birliktelik ilişkilerini bulmaktır. Verilerin sürekli artması nedeniyle şirketler, veritabanlarındaki birliktelik kurallarını ortaya çıkarmak isterler. Büyük miktarda eski verilerden değişik birliktelik ilişkileri bulmak, şirketlerin karar alma süreçlerini olumlu etkilemektedir. Birliktelik kuralları için verilebilecek örnek market sepeti uygulamasıdır. Bu kural, müşterilerin satın alma alışkanlıklarını analiz etmek için, müşterilerin satın aldıkları ürünler arasındaki ilişkileri bulur. Bu tür ilişkilerin analizi sonucunda, müşterilerin satın alma davranışları öğrenilebilir ve yöneticiler de öğrenilen bilgiler sonucunda daha etkili satış stratejileri geliştirebilirler. Örneğin bir müşterinin süt ile birlikte ekmek satın alma olasılığı nedir? Elde edilen müşteri bilgilerine dayanarak rafları düzenleyen market yöneticileri ürünlerindeki satış oranlarını arttırabilirler. Örneğin bir marketin müşterilerinin sütün yanında ekmek satın alma oranı yüksekse, market yöneticileri süt ile ekmek raflarını yan yana koyarak ekmek satışlarını arttırabilirler.

2.2.5 Modelin Değerlendirilmesi

Değerlendirme aşamasında, daha önce oluşturulmuş olan model, uygulamaya koyulmadan önce son kez tüm yönleriyle değerlendirilir, kalitesi ve etkinliği ölçülür. Bu aşama sonuçların değerlendirilmesi, veri madenciliği sürecinin gözden geçirilmesi ve sonraki adımların ne olacağı hususlarını içermektedir. Bu aşamanın sonunda, veri madenciliği sonuçlarının kullanımı üzerindeki karara varılmaktadır.

2.2.6 Modelin Kullanılması

Veri madenciliği sürecinin son aşaması, kurulan ve geçerliliği kabul edilen modelin kullanılmasıdır. Bu doğrudan bir uygulama olabileceği gibi bir başka modelin alt parçası olarak da kullanılabilir. Bu aşamada veri madenciliği süreciyle üretilen bilgiler, pratik işletme problemlerinin çözümünde kullanılmaktadır. Ayrıca elde edilen bilgilerin uygulanabilmesine yönelik bir plan hazırlama, gözden geçirme ve bakım faaliyetlerini içerir. Bu aşamada nihai araştırma raporunun yazılması ve projenin gözden geçirilmesi işlemleri yer almaktadır.

2.3 Veri Madenciliği Uygulama Alanları

Veri madenciliği bankacılık, finans, pazarlama, iletişim, astronomi, biyoloji, sigorta, tıp gibi birçok alanda kullanılmaktadır. Karar verme süreçlerinde başarılı sonuçlar elde edildiği için taşımacılık-ulaşım-konaklama, eğitim öğretim, perakendecilik gibi konularda da kullanılabilir. Örneğin pazarlamada mevcut müşterilerin satın alma davranışları, hangi müşteriye hangi promosyonların yapılacağı, sepet analizi gibi uygulamalarda kullanılmaktadır. Veri madenciliği hem özel hem de kamu işletmelerinde farklı amaçlarla kullanılabilmektedir. Örneğin sigortacılık ve bankacılık sektörü veri madenciliği uygulamalarını risk değerlemesine yardımcı olması ve sahtekarlıkları ortaya çıkarmak amacıyla kullanırlar. Uzun yıllar boyunca toplanan müşteri verileri kullanılarak, işletmeler bir müşterinin kredi riskini tahmin eden modeller geliştirebilirler. İlaç sanayi bazen bir ilacın etkinliğini tahmin etmede veri madenciliğini kullanabilir.

İletişim sektöründe veri transferi ve trafiğinin gittikçe artması ile sistem yükü, kaynak kullanımı, kullanıcı davranışları gibi konularda veri madenciliği teknikleri iletişim hizmetlerinin geliştirilmesine yardımcı olmaktadır.

Veri Madenciliği, tıpta, biyolojide ve diğer bilimsel uygulamalarda, bilgisayar ve ağların güvenliği konularında da kullanılmasının yanı sıra suçluların bulunmasında, olası terör saldırılarının tahmini gibi sosyal konularda da kullanılmaktadır.

Web Verileri: İnternet ve web üzerindeki veriler hem hacim hem de karmaşıklık olarak hızla artmaktadır. Web verilerinde düz metin ve resimden başka akıcı (streaming) ve nümerik veriler gibi farklı yapılarda veriler de yer alabilmektedir.

Kısacası büyük veri setlerinin bulunduğu ve analizinin gerektiği her sektörde veri madenciliği kullanım alanı bulunmaktadır.

BÖLÜM 3

SINIF DENGESİZLİĞİNİ ÖĞRENME

Sınıf dengesizliğini öğrenme, sınıflar arasında veya içinde önemli dengesizlikler gösteren veri setlerinden öğrenme anlamına gelir. Düzensiz veri dağılımlı herhangi bir veri kümesi dengesiz olarak kabul edilebilir. Sınıflar arasında dengesizlik olması durumda bazı veri sınıfları diğer sınıflarla karşılaştırıldığında oldukça az temsil edilir [12]. Geleneksel olarak, daha çok örneği olan sınıfa çoğunluk sınıfı ve az örneği olana azınlık sınıfı denir. Genellikle azınlık sınıfı özellikle ilgilenilen önemli sınıf olduğu için azınlık sınıfına ait bir örneğin yanlış sınıflandırılması daha büyük bir sorundur. Pratik uygulamalarda, sınıflar arasındaki oran 10:1, 100:1 ile 1000:1 gibi radikal olabilir [36]. İki sınıflı dengesizlik problemleri üzerine çok çalışma yapılmış olmasına rağmen birden fazla azınlık veya çoğunluk sınıfını içeren çok sınıflı dengesizlik problemi de eşit önem taşımaktadır [37].

Dengesizlik, sınıf içinde de olabilir ve sınıf-içi dengesizlik olarak adlandırılır. Bu durumda, sınıf alt-kümeler dizisinden oluşur ve bazı alt kümelerin örnekleri diğer alt kümelere göre daha azdır. Az temsil edilen alt-kümelerin hem azınlık hem de çoğunluk sınıflarında oluşabilmesine rağmen, onların azınlık sınıfında olma olasılığı daha yüksektir, bu örnekleri çoğunluk sınıfı için toplamanın kolay olmasından kaynaklanmaktadır [38].

Dengesizlik dağılımı gerçek dünya uygulamalarında yaygın bir şekilde bulunmaktadır. Bazı çalışmalar azınlık sınıfının basit sınıf dağılımları ile belirli dengesiz veri kümeleri için oldukça iyi öğrenmiş olduğunu göstermiştir [39, 40, 41]. Dengesizlik derecesinin, öğrenme algoritmaları performansının düşmesinden sorumlu tek faktör olmadığını

göstermektedir. Bir öğrenicinin sınıf dengesizliğine duyarlılığının, veriler karmaşıklığına ve eğitim setinin genel boyutuna bağlı olduğu görülmektedir.

Sınıf dengesizliği öğrenmede, azınlık sınıf örneklerinin nasıl daha iyi tanınabileceği sorunu önemli bir konudur. Bu sorunu gidermede, veri dağılımını yeniden dengelemek ya da algoritma içindeki öğrenme önyargısını ayarlamak yoluyla aşılabilir. Dengesizlik problemini öğrenme zorlukları ve azınlık sınıfı örneklerinin yüksek yanlış sınıflandırma maliyeti göz önüne alındığında, sınıf dengesizliğini öğrenme amacı genellikle çoğunluk sınıf doğruluğunu tehlikeye atmadan azınlık sınıfı için yüksek doğruluk sağlayan sınıflandırıcıyı elde etmektir [12].

Bu tez, sınıf dengesizliğini öğrenmenin temel sorununu çözmek ve çoğunluk sınıfı örneklerinden azınlık sınıfı örneklerını ayırmak için ve çoğu sınıf dengesizliği problemleri için geçerli bir etkili çözüm geliştirmeyi amaçlamaktadır.

Bu bölümde, gerçek uygulamalarda iyi gelişmiş ve popüler kullanılan birçok standart ve topluluk sınıflandırma algoritmaları kısaca gözden geçirilmiştir. Sonra, dengesiz veriler üzerinde birçok mevcut öğrenme yaklaşımları karşılaştırılmıştır. Ayrıca, topluluk öğrenme yaklaşımlarının sınıf dengesizliğini öğrenmede neden yaygın kullanıldığı ve bir topluluk sisteminin sınıf dengesizliği sorunları için kullanılması durumunda ortaya çıkan ana sorunlar incelenmiştir.

3.1 Sınıf Dengesizliği Sorunlarını Öğrenme

Birçok geleneksel öğrenme algoritmaları dengesiz sınıf dağılımlarını öğrenme için uygun değildir. Azınlık sınıfının genellikle daha yüksek yanlış sınıflandırma maliyetine sahip olduğunu bilmek, gelecekteki tahmin için yararlı değildir ve daha iyi sınıflama performansı elde etmek ister. Literatürde sınıf dengesizliğinin öğrenilmesinde topluluk sınıflayıcılar yerine sadece karar ağaçları [12], yapay sinir ağları [1], k-En Yakın Komşu (kNN) [42] ve Destek Vektör Makinaları-DVM [43] gibi sınıflandırıcıların kullanılmasının yeterli olmadığı bildirilmiştir.

Sorunu daha iyi anlamak için, bazı somut açıklamalar bu bölümde verilmiştir. 1-NN, bir örneği onun en yakın komşusuna ait sınıfa etiketleyen bir yöntemdir. Eğer dengesizlik

oranı çok yüksek ise, azınlık sınıfı örneğinin en yakın komşusunun çoğunluk sınıfının bir örneği olma olasılığı çok büyüktür. Bu nedenle yanlış sınıflanır.

Bir karar ağacını oluşturma, özyinelemeli ve kökten yaprağa doğru inilerek arama prosedürüdür. O her düğüm örneklerinin sınıflarını en iyi bölebilen özelliği seçer. Sonuç olarak, eğitim kümesi ard arda daha küçük alt kümelere bölünür. Fakat çoğunluk sınıfı yapraklara hükmetmekte ve azınlık sınıfı için çok spesifik kurallarda sonuçlar ardışık bölümlenmektedir. Dengesiz veri üzerinde karar ağaçlarının kullanımı konusunda çalışmalar mevcuttur [7].

Veri dengesizliğinde karşılaşılan çoğunluk sınıfına karşı önyargıyı gidermek için veri ve algoritma seviyesinde birçok çözüm yöntemi sunulmuştur. Veri seviyesindeki yöntemler, çarpık sınıf dağılımlarının düzeltilmesi için eğitim verilerini manipüle eden çeşitli yeniden- örnekleme tekniklerine başvurur. Onlar basit ve etkilidir ve onların doğru ayarlanması önemli bir konudur. Tek-sınıf öğrenme (one-class learning) ve maliyet-duyarlı öğrenme algoritmaları gibi çeşitli algoritma seviyesindeki yöntemler, azınlık sınıfı üzerinde daha iyi doğruluk elde etme amacı ile eğitim mekanizmasını değiştirerek sınıf dengesizliğini ortadan kaldırır. Tek-sınıf öğrenmede model, örneklerin yalnızca tek bir sınıfını kullanarak öğrenir. Tek-sınıf öğrenme, öğrenilmiş sınıfın yeni bir girişi için benzerliği ölçmek, yani bu sınıfa ait olup olmadığını bulmak amacıyla çalışır. Tek-sınıf öğrenme, çok boyutlu aşırı dengesiz veri setleri ile çalıştığında başarılı olduğu görülmüştür [44]. Maliyet-duyarlı yöntemler ile ilgili olarak, karar ağaçları ve yapay sinir ağları, sınıfların farklı yanlış sınıflandırma maliyetlerini oluşturabilmek için kullanılmıştır. Ayrıca, algoritma seviyesindeki çözümler, öğrenme algoritmalarının belirli türleri için farklı iyileştirmeler gerektirir. Biz çoğu durumda hangi algoritmanın daha iyi bir seçim olduğunu bilmediğimizden bu, pek çok uygulamada onların kullanımını kısıtlamaktadır. Iyi bir çözümün verilen sorunlar ile etkili bir şekilde başa çıkabilmesi beklenir [3]. Bu nedenle, bu tezde daha geniş bir kapsamda uygulanabilir çözümleri keşfetmek amaçlanmıştır.

Topluluk öğrenme, adı geçen veri seviyesi ve algoritma seviyesi çözümlerin yanı sıra, sınıf dengesizliği sorunlarını işleme yaklaşımlarının başka bir önemli kategorisi haline gelmiştir. Azınlık sınıfını vurgulamak için bu farklı yeniden-örnekleme tekniklerini entegre etmek veya farklı sınıflamaya ilişkin hata maliyetlerini uygulamak mümkündür

[17, 31, 45]. Topluluk düzeyinde bakıldığında, çoklu sınıflandırıcıları birleştirme fikri, hata varyansını azaltarak aşırı uyuma riskini aşağı düşürebilir ve her bir sınıflandırıcı zayıflıklarını telafi edebilir. Bu nedenle, bir topluluğunun, istikrarlı performanslı tek bir sınıflandırıcıya göre daha iyi performans verdiğine inanılmaktadır.

3.2 Standart Sınıflandırma Algoritmaları

Daha önce belirttiğimiz gibi makine öğrenmede sınıflandırma, yıllardır incelenmiştir ve pek çok öğrenme algoritması ikili ve çoklu veri sınıflandırma için önerilmiştir. Birçok temel algoritma kapsamlı ve teorik olarak incelenmiştir ve çeşitli başarılı uygulamalar klasik makine öğrenme konuları olmuştur. Popüler standart sınıflandırma yöntemlerine K-en yakın komşu algoritması, Naive Bayes sınıflandırıcısı, Yapay sinir ağları, Destek Vektör Makinesi, Karar Ağacı, Doğrusal ayırtaç analizi ve diğerleri dahildir. Daha teorik ve ampirik çalışmalar da bulunmaktadır.

3.3 Topluluk Öğrenme

Belirli bir problem üzerinde birçok öğrenicinin çalıştırılması fikrine topluluk öğrenme denir. Burada doğruluğu arttırmak için topluluk sınıflayıcılarının çıkışları ortak bir sonuç üretmek için birleştirilir. Bireysel komite üyesine temel öğrenici denir. Sınıflandırmada topluluk öğrenme, aynı zamanda çoklu sınıflandırıcı sistem, sınıflandırıcı füzyon, sınıflandırıcıların komitesi, sınıflandırıcı kombinasyonu, vb olarak da adlandırılır [33]. Üyelerin tahmini gerçek-değerli numaralar, sınıf etiketleri, gerideki olasılıklar (posterior probabilities) olabilir. Bireylerin güçlü taraflarını almak ve kendi zayıflıklarını telafi etmek için tahminleri birleştirmek önemlidir ve literatürde ortalama, oy çokluğu ve olasılıklı yöntemler dahil olmak üzere incelenmiştir [46, 47].

Topluluk öğrenme yöntemleri, ilgili araştırmaların hızlı büyümesini teşvik eden bazı istenen özelliklere sahiptir. Teorik nedenlerden dolayı [33], her öğrenme modelinin sınırlamaları vardır ve mevcut veri nedeniyle farklı uygulanabilir. Onların sonuçlarını ortalamak zayıf bir tahmin yapma riskini azaltır. Bu bir karar verilmeden önce diğerlerinin görüşlerine de danışmak için doğuştan gelen bir davranıştır. Topluluklar, her bir birey, sadece daha küçük ve daha basit bir alt problemden birini öğrenen veri

alanı bölümüne izin verir. Bunların kombinasyonu bütün sorunu daha iyi temsil edip, çözebilir.

Gerçek pratik uygulamalarda, çok büyük veya çok küçük sorunlarla karşılaşılabilir [33]. Büyük bir veri seti, birçok öğrenici tarafından paralel işlenecek şekilde birçok küçük alt kümeye ayrılabilir. Çok küçük veri olduğunda, mevcut verileri vurgulamak ve örtüşen alt kümeleri oluşturmak için yeniden- örnekleme teknikleri kullanılabilir.

Yukarıdaki avantajlarıyla topluluk öğrenme, yüz tanıma, konuşma tanıma, görüntü analizi ve elle yazılan rakamları tanıma gibi çeşitli alanlarda büyük katkı sağlamıştır.

1990'lardan bu yana öğrenme algoritmalarına etkin bir topluluk modelini oluşturmak için bir sürü çabalar ortaya konulmuştur. Özellikle, Bagging ve AdaBoost literatürde çok dikkat çekmiştir. Ayrıca Rastgele Altuzay, Rasgele Orman ve Negatif Korelasyon Öğrenme (NCL) gibi diğer popüler yöntemler de bulunmaktadır [48].

3.3.1 Sınıf Dengesizliğini Öğrenmede Topluluğun Farklılıkları

Son yıllarda, topluluk öğrenme yöntemleri dengesiz verileri sınıflandırmada tercih edilen bir yol haline gelmiştir. Topluluk öğrenme yöntemleri, veri seviyesindeki yeniden-örnekleme tekniklerini kullanarak ya da algoritma seviyesindeki farklı yanlış sınıflandırma maliyetlerini uygulayarak azınlık sınıfını vurgulamak için kolaylıkla adapte edilebilir [3, 17, 45]. Ayrıca, çoklu sınıflandırıcıları birleştirme fikri aşırı uyum riskini (overfitting) aşağı düşürebilir ve hata varyansını azaltabilir. Aşırı-örnekleme ve alt-örnekleme gibi özel teknikler, çoğu zaman azınlık sınıfı tahminini iyileştirmek için topluluklar ile birlikte kullanılır. Örneğin, SMOTEBoost çoğunluk sınıfına doğru öğrenme önyargısını ayarlamak ve azınlık sınıfı için yeni eğitim verisi üretmek için sık kullanılan bir topluluk yöntemidir ve tüm veri seti üzerinde doğruluğu sağlamak için boosting'i kullanır.

Varolan topluluk yöntemleri büyük bir öğrenme avantajına ve ampirik bir doğruluk başarımına sahiptir. Ancak literatürde topluluk yöntemlerinin sınıf dengesizliği sorununu çözümede neden iyi performans verdiğine dair kesin bir açıklama yoktur.Bir topluluk öğrenme yönteminin sınıflama başarısının ana nedenlerden biri farklılık (diversity) olarak kanıtlanmıştır, bu nedenle sınıf dengesizliğini öğrenmede topluluk

yöntemleri için farklılığın önemi ne kadar sorusu ortaya çıkar. Farklılığın azınlık/çoğunluk sınıflarının performansı üzerinde nasıl bir etkiye sahip olduğu belli değildir.

İyi bir topluluk öğrenme algoritmasının tasarım prensibi, bireysel doğruluklar ile komite üyeleri arasındaki "farklılığı" ("diversity") düşünmektir. Özdeş öğreniciler kümesini birleştirmenin hiçbir anlamı yoktur. Bütün topluluk algoritmaları farklılığı teşvik etmeye çalışırlar [49, 50]. Hata fonksiyonları ve birleştirme kurallarının değişik biçimleri sayesinde, topluluğun farklılığı, regresyon ve sınıflandırma görevleri için ayrı ayrı incelenmiştir. Farklılık, regresyon topluluklarında açıkça formüle edilmiştır, ancak sınıflandırma topluluklarındaki farklılık halen tam teorik olarak anlaşılmamıştır [49, 51].

Ayrıca, topluluk farklılığını üzerindeki mevcut çalışmalar, genel doğruluk veya hata oranı ile sınırlıdır. Diğer değerlendirme kriterleri ile ilişkisi çok dikkat çekmemiştir. Dengesizlik durumunda, sadece genel doğruluk yeterli değildir ve verideki değişikliklere son derece duyarlıdır[38]. Literatürde diğer performans ölçüleri sınıf dengesizliğini öğrenme için kabul edilmiştir. Bu tezde, belirli bir sınıf üzerinde performansı değerlendirmek için en sık kullanılan ölçüler F-ölçü, AUC ve G-ortalama kullanılmıştır [52, 53, 54].

3.3.2 Topluluk Öğrenme Yöntemleri

Makine öğrenme için topluluk yöntemi, öğrenme stratejisinin bir metodolojik düzeyidir. Topluluk sınıflandırmanın temel amacı, daha yüksek performansı elde etmek için birkaç bireysel sınıflandırıcının güçlerini birleştirmektir. Sınıflandırıcı üyelerini hem seçmek ve oluşturmak hem de bunları birleştirmek için çeşitli yaklaşımlar vardır. Topluluk sınıflandırmanın avantajları istatistiklerde önyargı ve varyans kavramı ile açıklanabilir [13]. Genel olarak, birkaç sınıflandırıcıyı birleştirmek, onların varyansını azaltır ve genelleştirme yeteneğini geliştirir.

Sınıflandırma topluluğunun başarısı ve topluluk üyelerinin farklılığına bağlıdır. Tüm bileşen sınıflandırıcılar aynı öğrenilen bilgiyi içeriyorsa, o zaman böyle bir entegrasyon iyileşme olmadan tek bir sınıflandırıcıya küçülecektir. Bu nedenle, her bir sınıflandırıcıyı doğru bir şekilde oluşturmak ve seçmek, topluluk sınıflandırma yöntemlerini uygulamada geliştirilmesi gereken bir stratejidir. Bireysel öğrenici doğruluğu, standart

sınıflandırma algoritmalarına kolayca manipüle edilebilir. Topluluk öğrenme için esas olan farklılığı (diversification) iyi tanıtmaktır.

Farklılık için birkaç basit strateji geliştirilmiştir: 1) tek tip algoritma ile eğitim verilerinin (örnek altkümeleri veya özellik altkümeleri) farklı alt kümelerini kullanarak; 2) aynı eğitim veri kümesi üzerinde tek bir yöntemin farklı parametrelerini kullanarak; 3) farklı öğrenme yöntemlerini kullanarak. Uygulamada sınıflandırıcı birleştime yapısı aşağıdaki gibi olabilir: i) paralel; her sınıflandırıcı bağımsız oluşturulur; ii) hiyerarşik veya kaskad; yeni sınıflandırıcıları oluşturmak için öncekilerden bilgi gerekir ve mevcut sınıflandırıcının eğitim doğruluğu, sonraki sınıflandırıcıda öğrenme için kullanılır. Literatürde sıkça incelenen üç topluluk stratejisi: bagging, boosting ve rasgele orman algoritmalarıdır [33].

Bagging. Bagging algoritması Bootstrap AGGregatING'in kısaltmasıdır, orijinal veriden düzgün örneklenen birçok veri alt kümesi üzerinde temel öğrenicileri oluşturur ve sonra onların sonuçlarının lineer bir kombinasyonunu kullanır. Kombinasyon stratejisi, eşit ağırlıklandırılmış (çoğunluk oylama) veya eğitim performansına göre ağırlıklandırılmış olabilir. Standart bagging, örnekleme işlemi için "değiştirme ile" ("with replacement") yöntemini kullanır ve örneklenen her alt küme, orijinal verilerin yalnızca %63.2 örneklerini içerir. Açıkçası, bagging, farklı örnekleme alt kümelerini oluşturarak farklılığı (diversity) korur ve tüm alt kümeler için düzgün öğrenme algoritmalarını kullanır.

Bagging yöntemi temel sınıflandırıcıların varyansını azaltarak genel hatayı geliştirir. Ancak, çalışmalarda bagging yönteminin, sınıflandırma başarımını artırmak için kararlı temel öğrenicilere göre kararsız olanları tercih ettiğini göstermiştir [55]. Kararsız öğreniciler, öğrenilmiş olan modellerde göze çarpan değişiklikleri olan ve eğitim verisinde küçük bir tedirgemesi olan algoritmaları ima eder. Dengesiz bir veri kümesini sınıflandırmak için orijinal Bagging algoritmasını uygulamak uygun değildir. Orijinal veriden önyüklenen (bootstrapped) her eğitim alt kümesi hala dengesiz bir dağılıma sahiptir. Bu orijinal eğitim seti ile karşılaştırıldığında çoğunluk sınıfına karşı daha önyargılı olabilir [58]. Azınlık sınıfı örneklerinin göz ardı edilmesinin veya aşırı uyum olmasının her temel sınıflandırıcı tarafından gerçekleşmesi mümkündür [12]. Sonuç olarak, son performans çoğunluk sınıfını tercih edecektir. Basit bir yol, bu çarpıklığı

yeniden örnekleyerek her alt küme içinde düzeltmektir ve böylece dengeli sınıf dağılımlı veriden bileşen sınıflandıncılar inşa edilebilir.

Basit bagging sınırlamalarını gidermek için, bagging'in çeşitli varyasyonları da literatürde geliştirilmiştir. Örneğin, Li dengesiz verileri sınıflandırmak için Bagging Topluluk Varyasyon'u (BEV) önermişti [6]; BEV, çoğunluk sınıfının altkümelerini, eşit boyutta bölerek yaratır, burada her alt kümede azınlık sınıf örneklerinin toplam sayısı ile çoğunluk sınıf örneklerinin sayısı aynıdır. Bagging tabanlı diğer yöntemleri, Tao et al.'ın AB-SVM [27], Chan et al.'in birleştirme modeli ve Yan et al.'ın DVM topluluğudur. Bu yöntemler, çoğunluk sınıfını azınlık sınıfı örnekleriyle eşit boyutta birkaç altkümeye alt-örnekler, alt kümelerin her biri daha sonra tüm azınlık sınıfı örnekleri ile birleştirilir. Her sınıflandırıcı böylece kesinlikle dengeli bir alt küme ile eğitilmiş olur. Veriler çok dengesiz olduğunda, tek başına alt-örneklemeyi kullanarak temsilci bir alt kümeni seçmek zordur. Bunun yanı sıra, aynı azınlık sınıfı örnekleri, her bir sınıflandırıcı tarafından giriş olarak alınır ve çıkan topluluk aşırı uyum sorunu ile karşılaşabilir [17].

Boosting. Boosting, önceki eğitim iterasyonlarında yanlış öğrenilen örnekleri vurgulamak için her yeni sınıflandırıcıyı eğiterek topluluk üyelerini adım adım oluşturur. Başlangıçta, öğrenme modelinde her örneğe eşit ağırlık atanır. Yanlış öğrenilen örnekler bir sonraki itersyonda daha yüksek ağırlıklar alır. Sınıflandırıcı, bu yüksek ağırlıklı örnekleri eğitme konusunda daha fazla çaba harcar. İteratif olarak, birkaç temel sınıflayıcı sırayla oluşturulur ve bunlar, ağırlıklı oy ile gelecek tahminlerde kullanılmak üzere birleştirir. Buradaki her temel modelin ağırlığı, eğitim seti üzerindeki doğruluk ile orantılıdır.

Boosting yöntemleri, bir çok uygulamada sınıflandırma performansını arttırırken etkili bir modelleme için varyansları ve önyargıları azaltabilir. En popüler boosting yöntemlerinden biri 1996 yılında Freund ve Schapire tarafından tanıtılan AdaBoost algoritmasıdır.

AdaBoost temel öğrenici olarak karar ağacını kullandığında, pratikte başarılı olduğu görülmüştür [44]. Fakat boosting yöntemi için bazı dezavantajlar da vardır: yeterli veri yoksa veya eğitim verileri çok fazla gürültü içeriyorsa, boosting yöntemi iyi performans vermeyebilir.

Random Forest. Rasgele orman, bagging ve karar ağacını birleştiren belirli bir topluluk sınıflandırıcısıdır [48]. Rasgele orman, birçok karar ağacı sınıflandırıcısını içerir, her ağaç orijinal veriden rasgele örneklenmiş özellik alt kümesi üzerinde eğitilir. Orijinal veri setinde, özelliklere sahip örneklerin ve maksimal iterasyon sayısının var olduğu varsayılır. Her iterasyonda, yeni bir veri kümesini oluşturmak için özellikler rasgele seçilir ve sonra, bir karar ağacını oluşturmak için kullanılır. Burada ağacın her bir düğümü, özellikler arasından rasgele seçilir. Ağaç herhangi bir budama yapılmadan oluşturulur ve son tahmin, çoğunluk oylama kullanılarak tüm ağaçlardan kombine edilerek alınır.

Rasgele orman, AdaBoost ile karşılaştırılabilir performans göstermektedir ve gürültülü veri için dayanıklıdır. Fakat, rasgele ormanlar, bazı veri setinde ağaç üyeleri tam oluşturulduğundan aşırı uyum (overfitting) vermeye daha eğilimlidir. Bunun yanı sıra, çok sayıda alakasız özelliğin işlenmesinden ve her ağaç için özellik alt kümesinin rasgele seçilmesinden dolayı iyi performans vermeyebilir. Yine de, rasgele ormanın ampirik olarak bir çok uygulamada hala oldukça doğru bir sınıflandırıcı olduğu gösterilmektedir [25].

3.4 Sınıf Dengesizliği İçin Topluluk Yöntemleri

Çoklu sınıflandırıcıları birleştirme fikri, birbirlerinin zayıflıklarını telafi etmek için ortaya çıkmıştır [9, 10, 26, 56]. Dolayısıyla topluluk öğrenme, sınıf dengesizliği öğrenmenin en önemli yöntemlerinden biri haline gelmiştir. Topluluk öğrenme, azınlık sınıfını vurgulamak ve daha iyi genelleme için aşırı uyum riskini azaltmak amacıyla bireysel sınıflandırıcılar kombinasyonunu kullanır. Mevcut birçok topluluk yöntemi, değişen başarı dereceleri ile sunulmuştur. Literatürde mevcut topluluk çözümleri, veri seviyesindeki her temel sınıflandırıcı için eğitim alt kümesini nasıl dengelemek ve topluluğu nasıl maliyet-duyarlı yapmak gerektiği üzerinde odaklanmıştır.

Bagging ve AdaBoost literatürdeki en popüler iki topluluk öğrenme tekniğidir [57]. Özellikle, AdaBoost'un zayıf bir öğrenme algoritmasının tahmin doğruluğunu arttırmada büyük yararları olduğu bildirilmektedir [25].

Bagging ve Boosting'e dayanan sınıf dengesizliği sorunları ile uğraşmak için topluluğun çeşitli yöntemleri sunulmuştur ve bunlar bazı uygulamalarda üstünlük göstermektedir.

Topluluk yöntemleri, küçük sınıfları vurgulamak için aşırı-örnekleme ve alt-örnekleme gibi diğer teknikler ile çalışmak üzere kolayca değiştirilebilir. Bagging tabanlı yöntemler için ortak strateji, çoğunluk sınıfı verilerini alt-örneklemek ve sonra dengeli eğitim alt kümeleri ile nispeten bileşen sınıflandırıcıları inşa etmektir. AdaBoost, boosting ailesinin en tanınmış üyesidir, çoğunlukla sıralı eğitimin her adımında gelişmiş aşırı-örnekleme stratejisi kullanılarak değiştirilir. Aynı zamanda, ağırlık-güncelleme kuralını işleyerek maliyet-duyarlı hale gelir [23].

Ancak, her iki yeniden-örnekleme tabanlı ve maliyet-duyarlı topluluk çözümlerinin, öngörülen bazı sakıncaları bulunmaktadır. Çoğunluk sınıf örneklerini alt-örnekleme, potansiyel olarak yararlı bilgileri kaldırabilir [12]. Ayrıca, ne kadar örnek kalacağına ve hangi örneklerin kaldırılması gerektiğine karar vermek bir sorundur. Sentetik örnekler oluşturmak için bazı veri üretme yöntemleri geliştirilmiştir [12]. Maliyet-duyarlı yöntemler veri düzeyinde çalışmazlar ve öğrenme öncesi sınıfların net maliyet bilgilerini talep ederler. Genel olarak, her zaman iyi sınıflama başarımı gösteren örnekleme tekniğini seçmek çok önemli bir konudur. Dolayısıyla, bu tez çalışmasında mevcut topluluk yöntemlerindeki sorunların üstesinden gelmek için alternatif bir yol aranmıştır.

3.5 Dengesiz Veri Üzerinde Maliyet - Duyarlı Öğrenme.

Bu bölümde, algoritma seviyesinde olan maliyet-duyarlı yaklaşımlar incelenmiştir [59]. Her sınıf için yanlış sınıflama maliyetleri verilir, maliyet-duyarlı bir yöntem maliyet bilgilerini dikkate alır ve eğitim prosedürü sırasında yanlış sınıflandırmalara farklı davranır. Azınlık sınıfına büyük bir maliyet atayarak sınıf dengesizliği sorunlarına, maliyet-duyarlı öğrenme yöntemlerini uygulamak doğaldır. Maloof, dengesiz veri setlerinden farklı maliyetler ile öğrenmenin ele alınabilir olduğunu göstermiştir [60]. Maliyet-duyarlı öğrenme ve sınıf dengesizliğini öğrenme arasındaki güçlü bağ araştırmalarda belirtilmiştir [51]. Maliyet matrisi, maliyet-duyarlı yöntemlerde temel rol alır [59].

Maliyet-duyarlı sınıflandırma, örnekler yanlış sınıflandırıldığında farklı maliyetler (veya ceza) kullanır. Bir maliyet matrisi her sınıf çifti arasındaki cezaları (penalties) temsil eden değerlerle inşa edilebilir. Cost (i, j), bir i sınıfı örneğinin diğer j sınıfı örneği olarak

tahmin edilme maliyetini temsil eder. Burada, Cost(+,-), azınlık sınıfı örneğinin çoğunluk sınıfı örneği olarak tahmin edilme maliyetidir ve Cost(-,+) tam tersidir. Sınıf dengesizliğini öğrenme sorununda, azınlık sınıfı örneklerini tanıma, çoğunluk sınıfı örneklerini tanımadan daha önemlidir. Bu nedenle, azınlık sınıfı örneğinin yanlış sınıflandırılma maliyeti, çoğunluk sınıfı örneğinin yanlış sınıflandırılma maliyetinden daha çok yüksek olmalıdır Cost(+,-)>>Cost(-,+). Bu arada, azınlık veya çoğunluk sınıfının bir örneğinin doğru sınıflandırılma maliyeti sıfır, Cost(+,+)= Cost(-,-)= 0. Sonuç olarak, maliyet-duyarlı öğrenmenin amacı, eğitim seti üzerinde genel maliyetleri en aza indirebilen bir sınıflandırıcıyı geliştirmektir. Birçok durumda, i sınıfındaki örnekler için yanlış sınıflandırma cezası diğer tüm örnekler için eşittir, böylece Cost(i,j), sadece Cost(i) olarak, ikili sınıflandırma için $Cost^{+}$ ve $Cost^{-}$ yazılabilir.

Birçok farklı yaklaşım sınıf dengesizliğini öğrenme için maliyet-duyarlı bilgileri kullanmak amacıyla geliştirilmiştir [12, 51, 60, 61]. Bu maliyet-duyarlı yöntemler üç sınıfa ayırılmaktadır [12]. İlk kategori, maliyet matrisini, veri alanını ağırlıklandırmak için uygular. Diğer bir deyişle, eğitim setinin dağılımı, yanlış sınıflandırma maliyetine dayalı bootstrap örnekleme kullanılarak değiştirilir. Değiştirilmiş veri dağılımı, kendi yüksek maliyetleri nedeniyle azınlık sınıfına karşı önyargılıdır. İkinci kategori, maliyeti minimize etme (cost-minimizing) tekniklerini topluluk programları ile birleştirir. AdaBoost gibi topluluk yöntemleri yeni maliyet-duyarlı meta-sınıflandırıcıları geliştirmek için maliyet katsayılarnı dahil edilmiştir. Sun et al. AdaC1, AdaC2, ve AdaC3 adlı üç maliyet-duyarlı AdaBoost algoritmasını önermiştir [23, 24]. Üçüncü kategori, maliyet cezalarının algoritma tasarımında gömülü olduğu doğrudan maliyet-duyarlı öğrenmedir. Bu tür yöntemler belirli bir makine öğrenme algoritması için çok özeldir (DT ve DVM gibi). Bu kategorinin örneklerine maliyet duyarlı karar ağaçları [60], maliyet duyarlı sinir ağları ve maliyet-duyarlı DVM (SVM) dahildir.

3.6 Dengesiz veri üzerinde Kernel-tabanlı örenme

Granüler destek vektör makineleri-tekrarlanan alt-örnekleme (Granular Support Vector Machines—Repetitive Under-sampling GSVM-RU), dengesiz veri öğrenme için kernel-tabanlı bir metodolojidir [4]. GSVM-RU, eğitilen DVM modellerinden destek vektörlerini ayıklar ve onları örneklenmiş veri alt kümeleri olarak kullanır. Başlangıçta,

ilk DVM orijinal veri kümesi üzerinde modellenir; sonra çoğunluk sınıfı için destek vektörler, tüm azınlık sınıf örnekleri ile birleştirilmiş yeni alt kümeyi oluşturmak için toplanır; sonra, bu toplanan destek vektörler orijinal veri kümesinden kaldırılır ve yeni bir DVM sınıflandırıcı modellenir. GSVM-RU, bu çoğunluk destek vektörleri iteratif kaldırarak ve yeni DVM modellerin oluşturarak, bilgilendirici örnekler içeren birçok alt küme oluşturur. Son olarak her bir alt küme topluluk veya en iyi bireysel sınıflandırıcıyı oluşturmak için kullanılır. Tang et al. GSVM-RU' ın farklı alanlarının birçok yüksek dengesiz veri kümesi üzerinde çok iyi performans verdiğini göstermiştir [4].

BÖLÜM 4

DENEYSEL ÇALIŞMALAR VE SONUÇLARI

Bu tezde sınıf dengesizliği sorunu ile ilgili mevcut üç algoritma: RUSBoost, EasyEnsemble ve BalanceCascade seçilmiş ve ayrıntıları aşağıda açıklanmıştır.

4.1 Karşılaştırmada Kullanılan Algoritmalar

4.1.1 RUSBoost.

C. Seiffert et al. çarpık eğitim verilerinden öğrenme için RUSBoost algoritmasını sunmuştur [30]. Bu algoritma basit ve hızlı bir alternatif sunmakta ve SMOTE-Boost karşısında boosting ile rasgele alt-örneklemeyi birleştirmektedir. C. Seiffert et al. iddia ettiği gibi, özellikle dengesizlik oranı çok büyük olduğu zaman SMOTE-Boost daha karmaşık ve zaman alıcıdır. Bu nedenle, eğitim süresini azaltmak ve alt-örneklemeden kaynaklanan bilgi kaybını önlemek için boosting'in kullanılması önerilmiştir.

RUSBoost, içerisinde istenilen denge sağlanana kadar çoğunluk sınıfından örnekleri rasgele kaldıran rasgele alt-örnekleme yöntemi RUS'u (random undersampling) uygular. RUS, sadeliğine rağmen çok iyi bir performans göstermektedir [11]. Basitlik, hız ve performans RUS'u boosting sürecine tanıtmak için kullanılan en önemli motivasyonlardandır. RUS, bir model oluşturmak için gerekli zamanı azaltır ki bu özelliği modeller topluluğu oluşturduğumuzda büyük yarar sağlar.

RUSBoost algoritmasını anlamak için x_i, özellik uzayı X'in bir noktası olsun ve y_i, Y sınıf etiketleri kümesindeki sınıf etiketi olsun. S veri setindeki m örneklerin her biri (x_i, y_i) ile temsil edilebilir. t değeri bir ile T (topluluk sınıflandırıcılarının sayısı) maksimum iterasyon sayısı arasındaki bir sayı olsun, h_t, t iterasyonunda eğitilmiş zayıf hipotez

(WeakLearn sınıflandırma algoritması kullanılarak eğitilen) ve $h_t(x_i)$ h_t hipotezlerinin çıkışı olsun. $D_t(i)$, t iterasyonun i'inci örneğinin ağırlığı olsun. [30]

İlk adımda her örnek ağırlıkları 1/m ile başlatılır, m - eğitim veri setindeki örnek sayısını gösterir. İkinci adımda zayıf hipotezler T defa iteratif olarak eğitilir. RUS çoğunluk sınıf örneklerini kaldırır. Örneğin, istenen sınıf oranı 50:50 ise çoğunluk ve azınlık sınıf örneklerinin sayıları eşit hale gelene kadar çoğunluk sınıf örnekleri rasgele kaldırılır. Sonuçta, S'_t geçici veri seti alınır. S'_t, yeni D'_t ağırlık dağılımına sahip olacaktır. S'_t ve D'_t, h_t zayıf hipotezini oluşturan temel öğrenici WeakLearn'e verilir. S orijinal eğitim veri setine ve D_t ağırlık dağılımına göre ε_t hata hesaplanır. Ağırlık güncelleme parametresi $\varepsilon_t/(1-\varepsilon_t)$ hesaplanır. Ardından, sonraki iterasyon için D_{t+1} ağırlık dağılımı güncellenir ve normalize edilir. T iterasyondan sonra sonuç H(x) hipotez, T zayıf hipotezlerinin ağırlıklı oyu olarak döndürülür. [30]

4.1.2 BalanceCascade

Alt-örnekleme, sınıf dengesizliği sorununun çözümünde kullanılan, yalnızca çoğunluk sınıfın bir alt kümesini kullanan ve bu nedenle çok verimli popüler bir yöntemdir. Ancak çoğunluk sınıf örneklerinin büyük bir bölümünü göz ardı etmektedir. Xu-Ying Liu et al. [17] bu eksikliği giderebilmek için EasyEnsemble ve BalanceCascade algoritmalarını önermiştir.

BalancedCascade, temel öğrenicileri sırayla oluşturur, yeni öğreniciler önceki öğreniciler tarafından filtre edilen örnekler üzerinde oluşturulur. Başlangıçta, çoğunluk sınıfının bir kısmını ve bütün azınlık sınıfını içeren bir örnek alt küme üzerinde ilk öğrenici eğitilir; sonra çoğunluk sınıfından örneklenen alt küme filtre edilir (doğru örnekler silinir ve sadece hatalı olanlar tutulur gibi); arıtılmış çoğunluk sınıfının bir alt kümesi ve azınlık kümesi ile yeni bir topluluk öğrenicisi oluşturulur. Filtrelenmiş veri seti örnekleri üzerinde yeni öğreniciler iteratif olarak oluşturulur, ve sonunda bütün öğreniciler birleştirilir. BalancedCascade, doğru modellenmiş örneklerin sonraki sınıflandırıcıları oluşturmada artık yararlı olmadığını kabul eder. Bu yöntemde Adaboost kullanılmaktadır. [17]

4.1.3 EasyEnsemble

EasyEnsemble çoğunluk sınıfından birkaç alt küme oluşturur ve bunların her biriyle bir öğreniciyi eğitir ve bu öğrenicilerin çıkışlarını birleştirir.

Azınlık eğitim seti Z_{min} ve çoğunluk eğitim seti Z_{maj} olsun, alt örnekleme yöntemiyle Z_{maj}'den rastgele Z_{maj}' alt küme örneklenir, $|Z_{maj}'| < |Z_{maj}|$. Bu yöntemde Z_{maj} setinden birbirinden bağımsız birkaç Z_{maj-1}, Z_{maj-2},..., Z_{maj-T} alt kümeler oluşturulur. Her Z_{maj-i} ($1 \leq i \leq T$) alt kümesi için, H_i sınıflandırıcı Z_{maj-i} ve tüm Z_{min} örneklerini kullanarak eğitilir. Tüm oluşturulan sınıflandırıcılar son karar için birleştirilir. AdaBoost H_i sınıflandırıcıyı eğitmek için kullanılır. [17]

4.2 Deneysel Çalışma

Bu deneyi yürütmek için, veri rasgele olarak eğitim (% 80) ve test (% 20) kümelerine bölünmüştür. Alt-örnekleme durumu için, tüm azınlık örnekleri kullanılarak ve çoğunluk örnekleri kaldırılarak eğitim setleri oluşturulmştur. Daha sonra sınıflandırıcılar oluşturulmuştur ve test seti örnekleri ile değerlendirilmiştir. Bu işlem on kez tekrarlanılmıştır ve bu tekrarlamalar için ortalama gerçek pozitif ve yanlış pozitif oranları bir ROC eğrisi olarak çizilmiştir. Karşılaştırılan algoritmaları değerlendirmek için AUC, F-ölçü ve G-ortalama kullanılmıştır.

4.2.1 Veri Setleri

Deneyler farklı uygulama alanlarından alınan sekiz gerçek veri seti kullanılarak yapılmıştır. Çalışmada kullanılan veri setleri ile ilgili bilgi Çizelge 4.1'de görülmektedir. Deneyler Matlab ve Weka kullanılarak gerçekleştirilmiştir.

WEKA, Yeni Zelanda Waikito Üniversitesi'nde, nesneye yönelik programlama dillerinden biri olan Java ile geliştirilmiş ve halen yeni sürümleri geliştirilmeye devam eden açık kaynak kodlu bir veri madenciliği yazılımıdır. Weka, sınıflandırma, kümeleme ve birliktelik kuralları analizinde hizmet sunabilen bir yazılımdır ve birçok genel algoritmanın Java gerçekleştirimlerini içerir. Örneğin, C4.5 algoritması Weka içerisinde "J48" olarak adlandırılarak kodlanılmıştır [62]. Veri sağlayıcısı, .csv ve .arff kütük biçimleri üzerinden veri girdisi kabul etmektedir. Aracın ücretsiz ve açık kaynak kodlu

olması en önemli avantajı olmaktadır. WEKA'nın Java ile geliştirilmiş olması, Linux, Unix, Windows ve Macintosh gibi başlıca işletim sistemi platformlarında kullanılabilirliğini sağlamıştır. Ayrıca WEKA programı, internet üzerinden ücretsiz olarak dağıtılmaktadır.

Çizelge 4. 1 Veri setleri hakkında temel bilgiler. **Boyut:** örnek sayısı. **Özellik:** veri setindeki özellikler sayısı. **min/maj:** azınlık ve çoğunluk sınıf örnekleri sayısı ve **%min/%maj** azınlık ve çoğunluk sınıfının yüzde oranı

Veri setleri	Boyut	Özellik	min/maj	%min/%maj
Breast	699	10	241/458	34/66
Bupa	345	7	145/200	42/58
Haberman	306	4	81/225	26/74
Hepatitis	155	18	32/123	21/79
İonosphere	351	35	126/225	36/64
Pima	768	9	268/500	35/65
Transfusion	748	5	178/570	24/76
Wpbc	198	35	47/151	24/76

4.2.2 Temel Sınıflandırıcılar

Karşılaştırılan algoritmaları doğrulamak için temel sınıflandırıcılar olarak C4.5 karar ağacı, destek vektor makinesi (DVM) ve k-en yakın komşu (KNN) algoritmaları kullanılmıştır. C4.5 olarak Weka'daki J48, DVM olarak SMO ve KNN olarak Ibk kullanılmıştır.

C4.5. Bu tezde tüm yazılım-kalite tahmin modelleri için tanınmış C4.5 algoritması kullanılmıştır. C4.5, ID3 algoritması gibi karar ağacı oluşturmayı sağlayan Quinlan'ın geliştirmiş olduğu algoritmalardan biridir. C4.5, ID3 algoritmasının gelişmiş versiyonudur. C4.5 tarafından oluşturulan karar ağaçları sınıflandırma için kullanılabildiği için literatürde C4.5 daha çok statiksel sınıflandırıcı olarak bilinmektedir [63]. C4.5 de aynı ID3 gibi belirli bir eğitim kümesini giriş olarak alıp buna göre bir karar ağacını entropi kavramını temel alarak oluşturur. C4.5 algoritması ID3 algoritmasının

bütün özelliklerini barındıran bir algoritmadır [64]. C4.5 algoritmasının geliştirilmesinde en önemli neden, ID3 algoritmasının bazı eksikliklerinin ve sorunlarının olmasıdır. C4.5 sisteminin ID3'e ek olarak getirdiği yaklaşımlar aşağıdaki gibi sıralanabilir:

- Sürekli ve sayısal değerli niteliklere sahip verilerle başarılı bir şekilde çalışmayı sağlamak.
- Nitelik değerleri eksik olan eğitim verileriyle başarılı bir şekilde çalışmayı sağlamak.
- Karar ağacı oluşturulduktan sonra ağacın budanmasını sağlamak.

Bahsedilen bu gelişmeler sonucunda C4.5'in özellikle sayısal veriler ve kayıp veriler üzerinde ID3 algoritmasına göre çok daha başarılı sonuçlar verdiği görülmektedir. ID3 algoritmasının değiskenleri birçok alt bölüme ayırma sırasında yaşanılan, aşırı-öğrenme durumunun önüne geçilebilmesi için C4.5 algoritmasında ID3'ten daha farklı bir 'kazanım' oranı kullanılmaktadır [65, 66]. J48, WEKA'daki C4.5 uygulamasıdır. C4.5 modelleri Weka tarafından sağlanan varsayılan parametreleri kullanarak inşa edilmiştir.

DVM. Destek Vektör Makineleri, güçlü teorik temeli, genelleme performansı ve yüksek boyutlu verileri işleme yeteneği ile tanınmıştır. Destek Vektör Makineleri (DVM) temel bir makine öğrenme teknolojisidir. DVM'in el yazısı tanıma, görüntü alma ve metin sınıflandırma gibi pek çok örüntü tanıma uygulamalarında güçlü teorik temelleri ve mükemmel ampirik başarıları vardır.

Temel DVM tabanlı aktif öğrenme, görünmeyen eğitim verilerinden önceki hiperdüzleme en yakın örneği seçer ve modeli yeniden eğitmek için onu eğitim setine ekler. Klasik aktif öğrenmede en bilgilendirici (yakın) örneği arama, tüm görünmeyen veriseti üzerinden yapılır [67]. Aktif öğrenmenin her iterasyonu, her örneğin yeni hiperdüzleme mesafesini yeniden hesaplar. Bu nedenle, büyük veri setleri için tüm eğitim setini arama, çok zaman alıcı ve hesaplama açısından pahalıdır.

Bugünün makine öğrenme uygulamalarında, destek vektör makinelerinin de alternatif olarak denenmesi gerektiği kabul edilmiştir. DVM, yaygın algoritmalar arasındaki sağlam ve hassas yöntemlerden biridir. Sağlam bir teorik temele sahiptir, sadece eğitim için bir düzine örnek gerektirir ve boyut sayısına duyarsızdır. Buna ek olarak, DVM eğitimi için etkin yöntemler hızla geliştirilmektedir.

DVM'nin İki sınıflı öğrenme görevinde amacı, eğitim verilerinde iki sınıf üyelerini birbirinden ayırmak için en iyi sınıflandırma fonksiyonunu bulmaktır. "En iyi" sınıflandırma fonksiyonu kavramı için ölçüm geometrik olarak gerçekleştirilebilir. Doğrusal ayrılabilen veri kümesi için, doğrusal bir sınıflandırma fonksiyonu, iki sınıf ortasından geçen ayırıcı bir düzleme (hyperplane) f (x) 'e karşılık gelir. Bu fonksiyon belirlendikten sonra, yeni x_n veri örneği, basitçe $f(x_n)$ fonksyonunun işaretini test ederek sınıflandırılabilir; eğer $f(x_n) > 0$ ise, x_n azınlık sınıfa aittır demektir.

Bu tür doğrusal hiperdüzlemler çok olduğundan, DVM en iyi fonksyonu iki sınıf arasındaki margini maksimize ederek bulunduğunu garanti eder. Sezgisel olarak, margin, hiperdüzlem tarafından tanımlanan iki sınıf arasındaki boşluk miktarı ya da ayrılma (seperation) olarak tanımlanır. Geometrik olarak, margin, hyperdüzlem üzerindeki bir noktaya en yakın veri noktaları arasındaki en kısa mesafeye karşılık gelir. Bu geometrik tanımın olması, marginin nasıl maksimize edileceğini keşfetmeyi sağlar, sonsuz sayıda hiperdüzlem olsa bile sadece birkaç DVM çözüm olarak nitelendirilir.

DVM'nin maksimum marginli hiperdüzlemleri bulma ısrarının nedeni, onun en iyi genelleme yeteneği sunuyor olmasıdır. Bu, eğitim verileri üzerinde en iyi sınıflandırma performansını (örneğin, doğruluk) sağlar, aynı zamanda geleceğe dair verilerin doğru sınıflandırılmasını sağlar.

KNN. Basit bir yaklaşım olan k-en yakın komşu (kNN) sınıflandırma, eğitim setinde test nesnesine en yakın k nesnelerin bir grubunu bulur, ve bu komşulukta belirli bir sınıfın baskın olan etiketini atar. Bu yaklaşımın üç temel unsuru vardır: (i) etiketli nesneler kümesi, örneğin, saklanan kayıtlar grubu, (ii) nesneler arasındaki mesafeyi hesaplamak için bir mesafe ya da benzerlik (similarity) olçümü, ve (iii) k değeri, en yakın komşu sayıları. Bir etiketsiz nesneyi sınıflandırmak için, bu nesnenin etiketli nesnelere mesafesi hesaplanır, onun k-en yakın komşuları tanımlanır, ve bu en yakın komşuların sınıf etiketleri daha sonra nesne sınıfı etiketlerini belirlemek için kullanılır.

4.2.3 Değerlendirme ölçüleri

Herhangi bir öğrenme yönteminin performansını kapsamlı bir şekilde değerlendirmek önemlidir. Farklı öğrenme yöntemlerinin sınıflandırma performanslarını dürüstçe karşılaştırmak basit bir görev değildir. İlk olarak, çeşitli alanı kapsayan ve önemli ölçüde

farklı veri özellikleri içeren standart veri kümesi (örneğin, örnek boyutu, özellik boyutu, özellik değerleri, vs gibi) test için dikkate alınmalıdır.

İkincisi, çoğu öğrenme yöntemleri, modelleri eğitmek için belli bilinen veri setlerini gerektirdiğinden, performansın test edilmesi ve bağımsız bilinmeyen veri kümeleri üzerinde karşılaştırılması gerekir. Çapraz doğrulama metodolojisi (cross validation-CV) bağımsız test verileri mevcut olmadığı zaman sıklıkla kullanılmaktadır.

Basit Doğrulama (Simple Validation): Buyuk veri setleri icin kullanılır. Verilerin %5 ile %33 luk kısmı test kumesi olarak ayrılır, geri kalanıyla model kurulur. Daha sonra kurulan model, test kumesi ile test edilir ve modelin doğruluk oranı hesaplanır. Eğer farklı eğitim kumesi ve test verileri kullanılmazsa modelin gecerliliği tahmin üstü olur.

Capraz Doğrulama (Cross Validation): Model kurmak icin az sayıda veri varsa, basit doğrulamada ki kadar veri alınmaz. Capraz doğrulama tum veriyi kullanmayı sağlayan bir metottur. Veri kumesi rastgele 2 eşit kısma ayrılır. İlk veri kumesiyle model kurulur, kurulan model diğer veri kümesiyle test edilir ve doğruluk oranı hesaplanır. Daha sonra ikinci veri kumesiyle model kurulur ve kurulan model birinci veri kumesiyle testi yapılıp, doğruluk oranı hesaplanır. En sonunda tüm veriler kullanılarak model oluşturulur. Daha önceden hesaplanmış olan 2 doğruluk oranının ortalamasıyla, son model karşılaştırılır.

Üçüncü olarak, sınıflandırma başarımını karşılaştırma ölçüleri doğruluk için çok önemlidir. Özellikle, azınlık sınıfı standart öğrenme yöntemleri tarafından önemli ölçüde göz ardı edildiğinden dengesiz veriler farklı ölçüler gerektirir.

Dengeli veri öğrenmede kullanılan birkaç standart değerlendirme ölçüsü incelenedikten sonra sınıf dengesizliği öğrenmede kullanılan bazı gelişmiş ölçüler tanıtılacaktır. Sınıflandırma başarımı için geleneksel değerlendirme ölçüleri olan doğruluk ve hata oranı, sınıf dengesizliğinden dolayı dengesiz öğrenme sonuçlarını ölçmek için uygun değildir. Hem azınlık hem de çoğunluk sınıflarının doğruluğunu dikkate alan yeni performans ölçümleri dengesiz veriden öğrenme için gereklidir. Bu tür ölçümler önceden literatürde sunulmuştur. Onlar: AUC, F-ölçü ve G-ortalamadır.

Temel ikili sınıf sınıflandırma problemi göz önüne alındığında pozitif ve negatif sınıfların gerçek etiket ile tahmin etiketleri arasındaki fark Çizelge 4.2' de olduğu gibi, sınıf

karışıklık matrisi ile temsil edilebilir. Sınıflandırıcı, test verileri üzerinde örneklerin dört tipini üretir,

- TP: pozitif sınıfa ait doğru sınıflandırılmış örnekler sayısı.
- TN: negatif sınıfa ait doğru sınıflandırılmış örnekler sayısı.
- FP: negatif sınıfa ait yanlış sınıflandırılmış örnekler sayısı.
- FN: pozitif sınıfa ait yanlış sınıflandırılmış örnekler sayısı.

Çizelge 4. 2 İkili sınıflandırma için sınıf karışıklık matrisi

	Tahmin Pozitif sınıf	**Tahmin Negatif sınıf**
Gerçek Pozitif sınıf	Gerçek Pozitif TP	Yanlış Negatif FN
Gerçek Negatif sınıf	Yanlış Pozitif FP	Gerçek Negatif TN

Bu karışıklık matrisine dayanarak, çeşitli ölçüler genel sınıflandırma performansını elde edebilirler. Geleneksel sınıflandırma görevi için genel doğruluk, performansı değerlendirmede en sık kullanılan kriterdir. Giriş bölümünde anlatıldığı gibi, bu, çoğunluk sınıfı lehine kuvvetli önyargısı olduğu için sınıf dengesizliği sorunları için uygun değildir ve yanıltıcı sonuçlar verebilir. Bir sınıflandırıcı çok yüksek doğruluk elde edebilir, ancak azınlık sınıf örneklerinin öngörülmesinde zayıf bir performans elde edebilir. Genel doğruluk, öğrenmenin ilgi alanı olan azınlık sınıfı örneklerini etkili bir şekilde bulmada başarısız olur. Bu nedenle, sınıf dengesizliği öğrenme için, her iki sınıf üzerinde özellikle azınlık sınıfı üzerinde doğruluğu dikkate alan yeni ölçüler gereklidir. Bu bölüm, sınıf dengesizliği öğrenme için en önemli değerlendirme ölçülerini tanıtır.

Geleneksel olarak, azınlık sınıfı pozitif olarak ve çoğunluk sınıfı negatif olarak işlenmiştir. Dört ölçüme dayanarak, aşağıda sınıf dengesizliği öğrenme için literatürde sıklıkla kullanılmakta olan ölçüler tanımlanmıştır (Çizelge 4.3).

Çizelge 4. 3 Değerlendirme ölçüleri

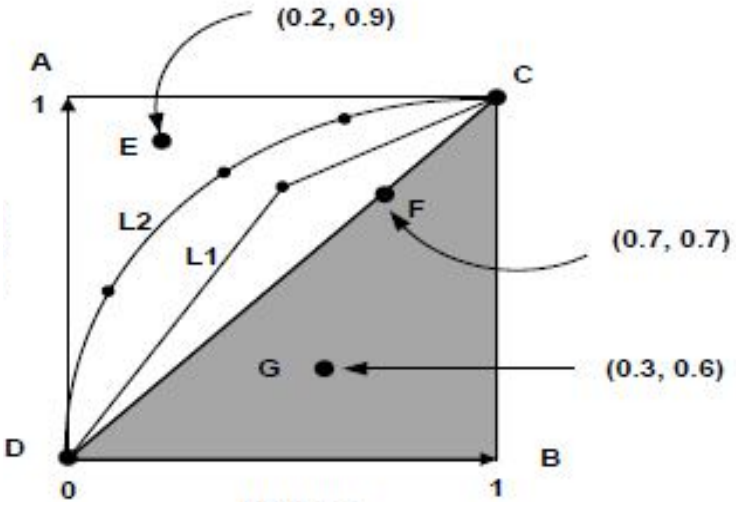
AUC	AUC = (1- TPR- FPR)/2
F–ölçü	F- ölçü = (2× Kesinlik × Anma)/(Kesinlik + Anma)
G-ortalama	G– ortalama =√Acc_+ × Acc_-
Gerçek Pozitif Oranı	TPR (Acc_+) = TP/(TP+FN)
Yanlış Positif Oranı	FPR = FP/(FP+TN)
Gerçek Negatif Oranı	TNR (Acc_-) = TN/(TN+FP)
Kesinlik	Kesinlik = TP/(TP+FP)
Anma	Anma = Acc_+

Anma (recall), kesinlik (precision) ve F-ölçü, sınıf dengesizliği sorununda sınıflandırıcıları değerlendirilmek için kabul edilmiş ölçülerdir. Anma bütünlük ölçüsüdür - tüm pozitif sınıf örnekleri için doğru sınıflandırılmış pozitif sınıf örneklerinin oranı. Kesinlik hatasızlık ölçüsüdür- sınıflandırıcı tarafından pozitif olarak tahmin edilen örneklerin doğru olarak sınıflandırılmış pozitif sınıf örnekleri oranı [12]. İki veya daha fazla temel ölçüleri birleştiren birçok yeni metrik önerilmiştir. Örneğin, F-ölçü, anma ve kesinlik aralarındaki ödünleşimi (trade-off) ifade etmek için ikisini de içermektedir. F-ölçüsü sınıflandırıcı seçiminde tercih edilen bir ölçü olarak kullanılmaktadır [69]. F-ölçüsü hem Gerçek Pozitif Oranı (True Positive Rate TPR) hem de kesinliği değerlendirir, özellike öğrenme doğruluğu pozitif sınıf üzerinde odaklanır. Diğer bir deyişle, F-ölçü anma ve kesinlik arasındaki bir harmonik ortalamadır.

G-ortalama, hem pozitif sınıf hem de negatif sınıf performansını dikkate alır ve onları birleştirmek için geometrik ortalamayı kullanır. Yüksek G-ortalama değeri, hem pozitif sınıfı hem de negatif sınıf için yüksek tahmin doğruluğuna sahip olduğu zaman elde edilebilir.

AUC-ROC. Tüm ölçüler TP, TN, FP, FN sabit değerlerine dayanır. Bu gibi değerler, sınıf etiketleri ve tahmin değerlerinin ikisi de ayrık (discrete) olduğunda kolayca toplanabilir.

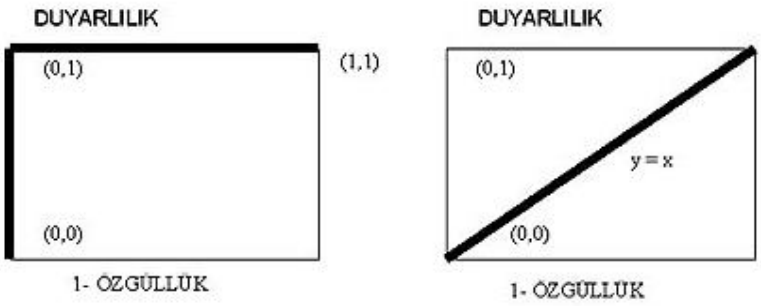
TP ve FP değerlerini birbirine ortak bağlayan ve bir 2-D ekseni üzerinde çizilen, ROC (Receiver Operating Curve) grafiği Şekil 4.1'de gösterilmiştir. ROC eğrisi, modellerin doğruluk değerlendirilmesi ve kıyaslanması için yaygın olarak kullanılan bir yöntemdir. ROC eğrisi, duyarlılık ve özgürlük (anma ve kesinlik) oranlarını kullanarak birimleri sınıflarına ayıran en uygun kesim noktasını belirler. Sınıflandırma doğruluğu, ROC eğrisi altında kalan alanın büyüklüğüne bağlıdır.



Şekil 4. 1 AUC (ROC eğrisi altındaki alan)

Amacı A Pozisyonda -TPR değeri 1 ve FPR değeri 0 olan sol üst köşesinde bir nokta üretmektir; ve en kötü model sağ alt köşedeki B noktasıdır. Bu nedenle, iyi bir sınıflandırma modeli mümkün olduğunca sol üst köşeye yakın olmalıdır. Bu arada, rasgele bir tahmin yapan model, TPR ve FPR birbirine eşit olduğu diyagonal üzerinde yer alır. Sol alt köşesindeki D noktası sınıflandırıcının her örneğinin negatif sınıf olarak sınıflandırılmasıdır ve sağ üstteki C noktası tüm tahminlerin pozitif sınıf olarak öngörüldüğü anlamına gelir. ROC eğrisi, D ve C noktalarının ve TPR ve FPR değerlerinin tüm gruplarını bağlayarak oluşturulur.

Yanlış değerlere sahip olmayan ideal bir testte ROC eğrisi (0,0)-(0,1)-(1,1) noktalarını birleştirmektedir. Buna karşın ROC çizimi y=x fonksiyonuna yaklaştıkca başarısız bir test ortaya çıkarır. Çünkü bu testte yanlış değerlerin oranı yükselmektedir. Bu fonksiyonun altındaki ROC eğrisine sahip test başarısızdır.



Şekil 4. 2 İdeal ve kötü performans göstergesi olan ROC eğrileri [70]

Şekil 4.2'de görüldüğü gibi ROC eğrisi altında kalan alan 1 ile 0.5 aralığında değer almaktadır. Alan değeri ne kadar büyük ise modelin o kadar iyi ayrım yeteneğine sahip olduğunu gösterir.

ROC eğrisi sol üst köşeye ne kadar yakın olursa, sınıflandırma başarımı o kadar iyi olur. Ancak, iki ya da daha fazla ROC eğrilerini doğrudan karşılaştırmak zor ve kullanışsızdır. Örneğin, iki eğri birlikte içiçe yapraklanmış olabilir ve iyi olanı iddia etmek zor olabilir. Bu nedenle ROC etkinliğini göstermek için ROC eğrileri altındaki alan AUC (Area Under the ROC Curve) tekil sayısal değer gösterir. Açıkçası, AUC 0 ile 1 arasında değişmektedir ve ne kadar yüksek olursa, o kadar iyi sınıflandırıcı üretilmiş demektir.

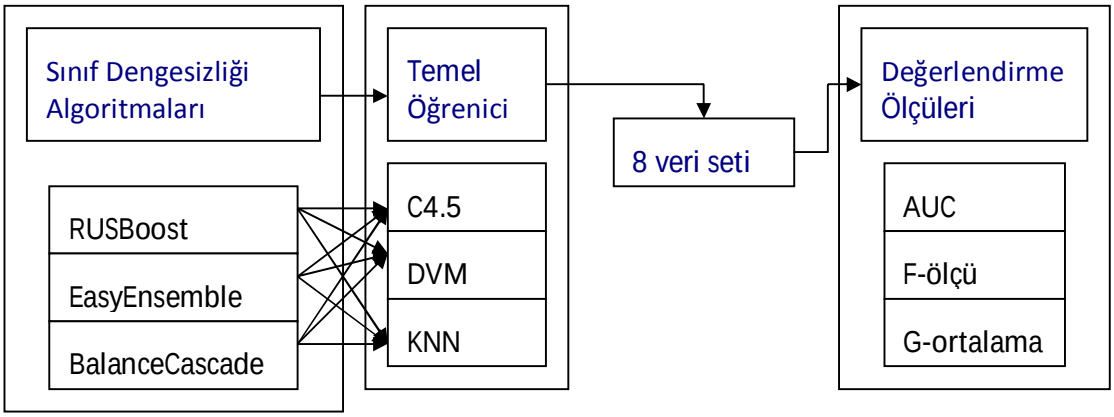
Doğruluk dışında, yukarıdaki tüm ölçüler esas olarak verinin gerçek sınıfı ya da tahmin sınıfı üzerindeki performansını dikkate alır ve verinin diğer parçalarını göz ardı eder. Bu nedenle, sınıf dengesizliği öğrenme için her iki sınıf üzerinde, özellikle azınlık sınıfı üzerindeki doğruluğu dikkate alan metrikler gereklidir.

Bu tezde, performans değerlendirme ölçüleri olarak, sınıf dengesizliği öğrenmede etkili olan F-ölçü, G-ortalama, AUC-ROC kullanılmıştır.

4.3 Deneysel Ayarlar

Önerilen çeşitli veri seti üzerinde çeşitli algoritmaları gerçekleştirmek için Şekil 4. 3 te gösterilen basit bir karşılaştırma çerçevesi oluşturulmuştur. İlk olarak, dengesiz verilerden öğrenme için üç farklı algoritma, yani RUSBoost, BalanceCascade ve EasyEnsemble deneylere dahil edilmiştir. Daha sonra, üç farklı temel sınıflandırıcı, C4.5, DVM ve KNN topluluklar için temel öğreniciler olarak kullanılmıştır. Veri setleri üzerinde bu algoritmalar uygulandıktan sonra, F-ölçü, G-ortalama ve AUC gibi

dengesizlik-odaklı değerlendirme ölçüleri karşılaştırılmıştır. Her bir öğrenme algoritmasının performansını bulmak için kod 10 kez çalıştırılmıştır ve bu çalıştırmalar sonucu üzerinde ortalama değer hesaplanmıştır.



Şekil 4. 3 Çeşitli veri seti üzerinde çeşitli algoritmaları gerçekleştirmek için basit bir karşılaştırma çerçevesi

4.4 Deneysel Sonuçlar

Bu bölümde RUSBoost, BalanceCascade ve EasyEnsemble algorıtmalarını kullanarak deneyler yapılmıştır. 4.4.1 alt bölümündeki deneyler C4.5, DVM ve KNN arasından en iyi temel öğreniciyi bulmak amacıyla yapılmıştır. 4.4.2 alt bölümünde en başarılı algoritma bulunmaya çalışılmıştır. 4.4.3 alt bölümünde deney farklı azınlık ve çoğunluk sınıf dağılımına göre bir önceki bölümde belirlenen en iyi performanslı algoritma ile yapılmıştır.

4.4.1 Temel Öğreniciler Performansı

Deneyler için ilk ilgilenilen konu, topluluk için uygun bir temel öğreniciyi seçmektir. Bilindiği gibi, topluluk yöntemleri, özellikle AdaBoost algoritması, normalde temel öğrenici zayıf ise, daha iyi çalışır. Literatürde sıkça kullanılan temel öğreniciler, C4.5 karar ağacı, CART, NN ve Naive Bayes sınıflandırıcısıdır. Bu tezde, temel öğreniciler olarak C4.5, DVM ve KNN kullanılmış ve sonuçları karşılaştırılmıştır.

Bu bölümde RUSBoost, BalanceCascade ve EasyEnsemble algoritmaları için C4.5, DVM ve KNN farklı sınıflandırıcıları temel öğrenici olarak kullanılarak incelenmiştir. Karşılaştırılan algoritmaları değerlendirmek için AUC, F-ölçü ve G-ortalama

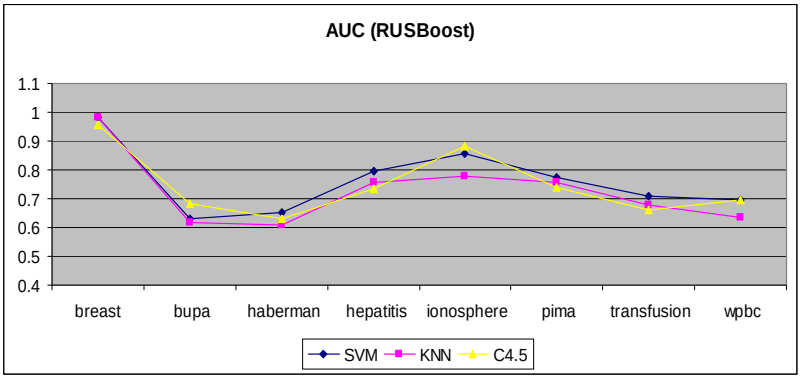
kullanılmıştır. Şekil 4. 4 - 4. 12'de karşılaştırılan üç yöntemin tüm temel öğreniciler ve tüm veri seti üzerindeki AUC, F-ölçüsü ve G-ortalama değerlendirme ölçüleri sonuçları verilmektedir. Üç algoritma için de en iyi sonuç C4.5 temel öğrenici olarak kullanıldığında elde edilmiştir.

RUSBoost algoritmasının temel öğrenici olarak C4.5 karar ağacı, destek vektör makinesi ve k en yakın komşu sınıflandırıcıları kullandığındaki elde ettiği sonuçlar Çizelge 4. 4 te verilmiştir.

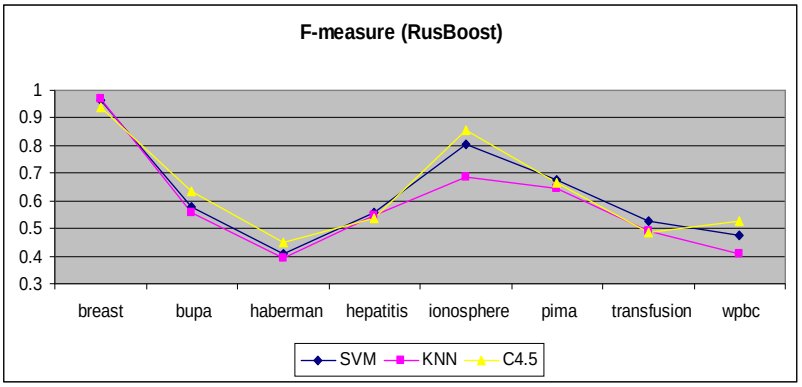
Çizelge 4. 4 RUSBoost algoritmasının C4.5, DVM ve KNN kullandığında tüm veri seti üzerindeki sonuçları

RUSBoost	**C4.5**			**DVM**			**KNN**		
	AUC	**F- ölçü**	**G-ort**	**AUC**	**F-ölçü**	**G-ort**	**AUC**	**F-ölçü**	**G-ort**
Breast	0.9569	0.9380	0.9567	0.9805	0.9637	0.9733	**0.9828**	**0.9682**	**0.9786**
Bupa	**0.6807**	**0.6362**	**0.6778**	0.6322	0.5769	0.611	0.6162	0.556	0.5910
Haberman	0.6303	**0.4487**	**0.5889**	**0.6538**	0.4068	0.5341	0.6101	0.3925	0.5592
Hepatitis	0.7353	0.5375	0.7242	**0.797**	**0.5590**	**0.7602**	0.7573	0.5487	0.7512
İonosphere	**0.8845**	**0.8567**	**0.8808**	0.8546	0.8039	0.8402	0.7787	0.6842	0.7285
Pima	0.7413	0.6634	0.7360	**0.7758**	**0.6767**	**0.7488**	0.7556	0.6428	0.7197
Transfusion	0.6598	0.4832	0.6254	**0.7097**	**0.5255**	**0.6982**	0.6765	0.4906	0.6583
Wpbc	**0.6978**	**0.529**	**0.6858**	0.6967	0.4743	0.6495	0.6356	0.4072	0.5965
mean	0.7483	**0.6366**	**0.7345**	**0.7625**	0.6234	0.7269	0.7266	0.5863	0.6979

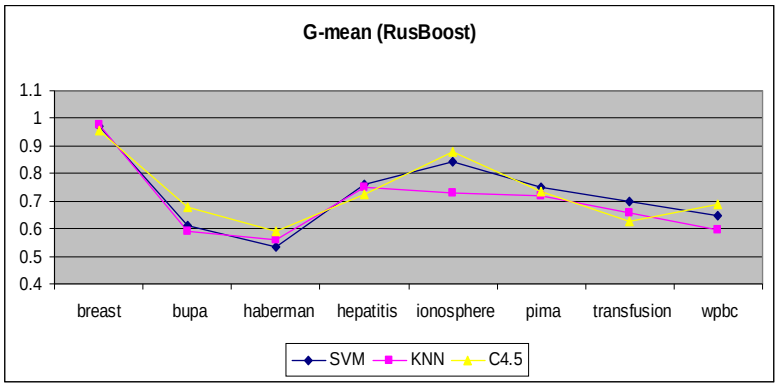
Şekil 4. 4 te RUSBoost algoritmasının AUC değeri sonuçları gösterilmekte. Temel öğrenici olarak C4.5 kullanıldığında sekiz veri setinin üçünde, DVM dört veri seti üzerinde ve KNN bir veri seti üzerinde başarılı olmuştur. F-ölçü ve G-ortalama (Şekil 4.5 – 4.6) hesaplandığında C4.5 dört veri seti ve DVM üç veri seti üzerinde, KNN ise bir veri seti üzerinde iyi sonuç vermiştir.



Şekil 4. 4 RUSBoost algoritmasının AUC sonuçları

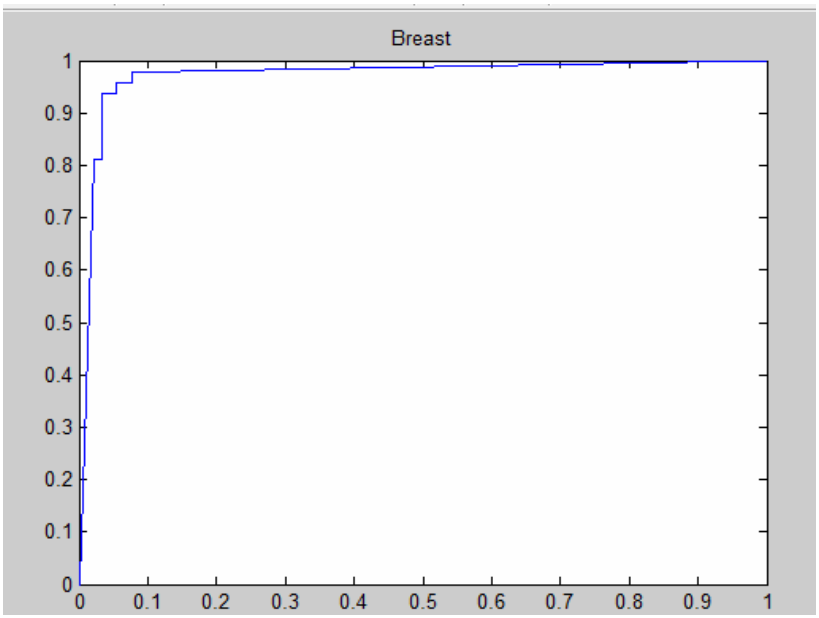


Şekil 4. 5 RUSBoost algoritmasının F-ölçü sonuçları

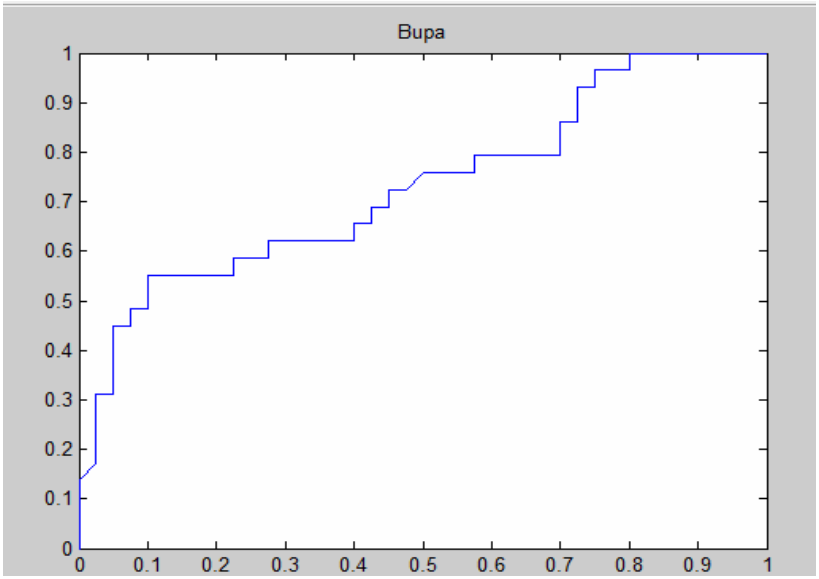


Şekil 4. 6 RUSBoost algoritmasının G-ortalama sonuçları

RUSBoost algoritması çalıştırıldığında Breast ve Bupa veri seti için ROC eğrisi Şekil 4.7 ve 4.8'de verilmiştir.



Şekil 4. 7 Breast veri seti için ROC eğrisi



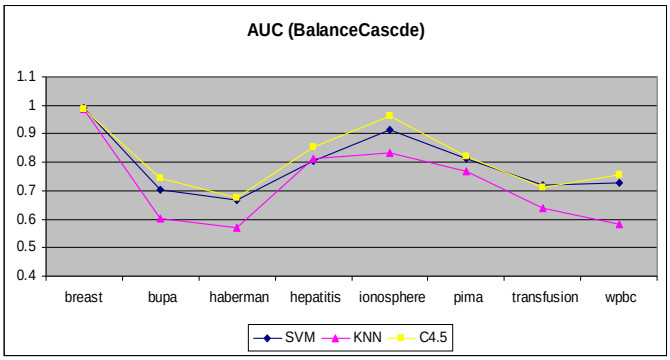
Şekil 4. 8 Bupa veri seti için ROC eğrisi

BalanceCascade algoritmasının temel öğrenici olarak C4.5 karar ağacı, destek vektör makinesi ve k en yakın komşu sınıflandırıcıların kullandığındaki elde ettiği sonuçlar Çizelge 4. 5 te verilmiştir.

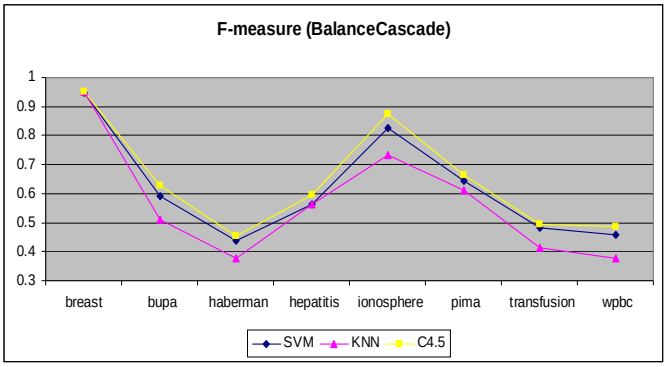
Çizelge 4. 5 BalanceCascade algoritmasının C4.5, DVM ve KNN kullandığında tüm veri seti üzerindeki sonuçları

Balance Cascade	C4.5			DVM			KNN		
	AUC	F-ölçü	G-ort	AUC	F-ölçü	G-ort	AUC	F-ölçü	G-ort
Breast	0.9869	**0.9507**	**0.9687**	**0.9897**	0.9487	0.9638	0.9876	0.9502	0.9663
Bupa	**0.7454**	**0.6260**	**0.6729**	0.7023	0.5931	0.6302	0.6007	0.5123	0.5659
Haberman	**0.6767**	**0.4544**	**0.6120**	0.6686	0.4360	0.5893	0.5696	0.3750	0.5378
Hepatitis	**0.8520**	**0.5968**	**0.7971**	0.8027	0.5618	0.7611	0.8117	0.5634	0.7413
İonosphere	**0.9617**	**0.8747**	**0.8970**	0.9126	0.8255	0.8577	0.8319	0.7317	0.7686
Pima	**0.8207**	**0.6659**	**0.7395**	0.8135	0.6432	0.7209	0.7669	0.6114	0.6938
Transfusion	0.7117	**0.4930**	**0.6677**	**0.7213**	0.4825	0.6576	0.6369	0.4120	0.5930
Wpbc	**0.7544**	**0.4881**	**0.6569**	0.7277	0.4574	0.6438	0.5822	0.3762	0.5562
mean	**0.8137**	**0.6437**	**0.7515**	0.7923	0.6185	0.7281	0.7234	0.5665	0.6779

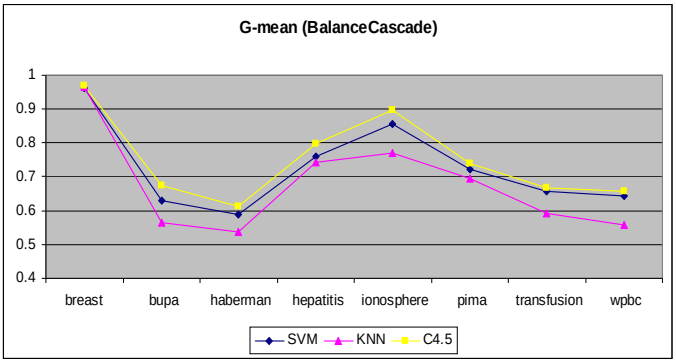
Şekil 4.9'da BalanceCascade algoritmasının AUC değerinin sonuçları verilmektedir. Temel öğrenici olarak C4.5 kullanıldığında sekiz veri setinin altısında, DVM iki veri seti üzerinde başarılı olmuştur. F-ölçü ve G-ortalama (Şekil 4.10 – 4.11) hesaplandığında C4.5 tüm veri seti üzerinde iyi performans vermiştir. DVM ve KNN temel öğrenici olarak kullanıldığında iyi sonuç vermemektedir.



Şekil 4. 9 BalanceCascade algoritmasının AUC sonuçları

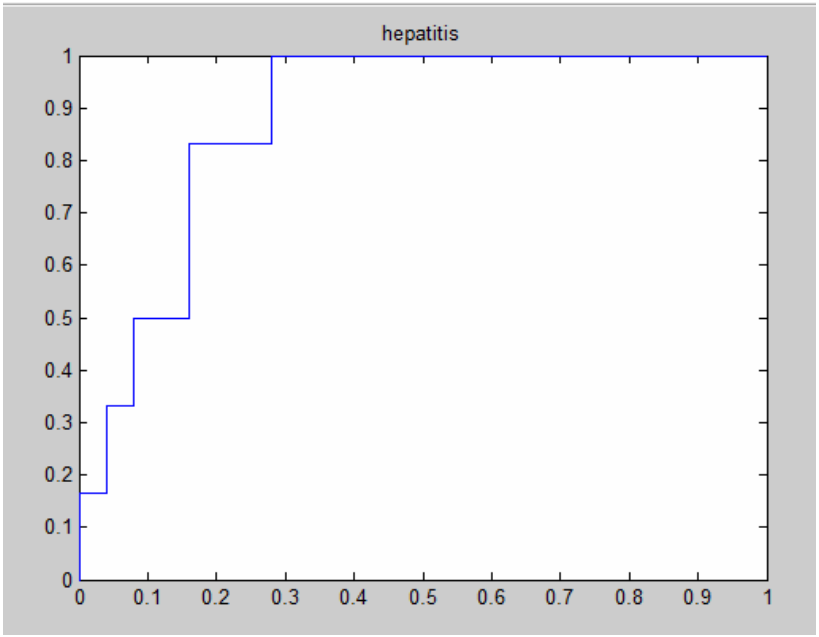


Şekil 4. 10 BalanceCascade algoritmasının F-ölçü sonuçları

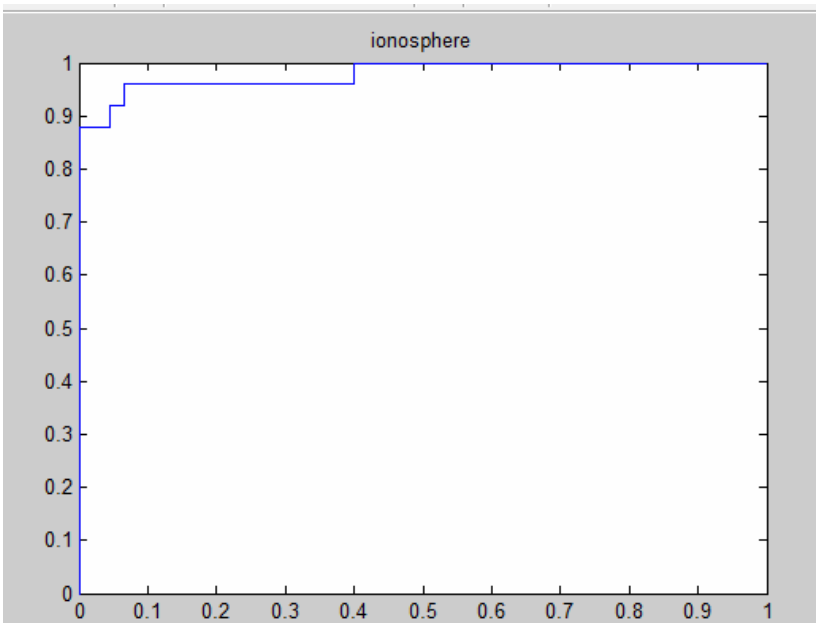


Şekil 4. 11 BalanceCascade algoritmasının G-ortalama sonuçları

BalanceCascade algoritması çalıştırıldığında Hepatitis ve İonosphere veri seti için ROC eğrisi Şekil 4.12 ve 4.13'te verilmiştir.



Şekil 4. 12 Hepatitis veri seti için ROC eğrisi



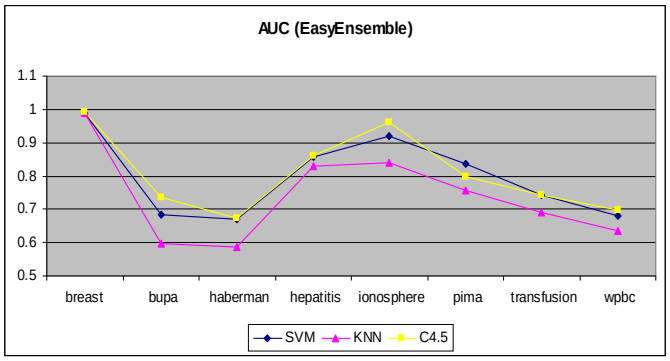
Şekil 4. 13 İonosphere veri seti için ROC eğrisi

EasyEnsemble algoritmasının temel öğrenici olarak C4.5 karar ağacı, destek vektör makinesi ve k en yakın komşu sınıflandırıcıların kullandığındaki elde ettiği sonuçlar Çizelge 4.6'da verilmiştir.

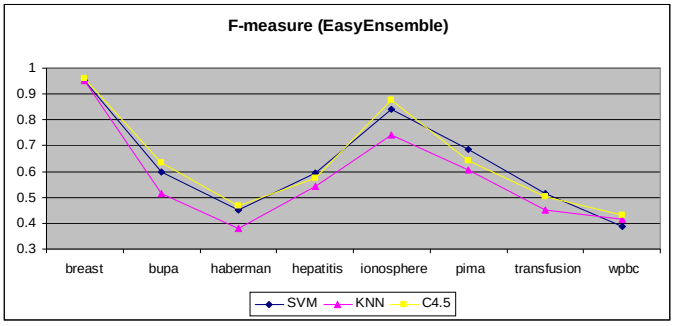
Çizelge 4. 6 EasyEnsemble algoritmasının C4.5, DVM ve KNN kullandığında tüm veri seti üzerindeki sonuçları

Easy Ensemble	**C4.5**			**DVM**			**KNN**		
	AUC	**F-ölçü**	**G-ort**	**AUC**	**F-ölçü**	**G-ort**	**AUC**	**F-ölçü**	**G-ort**
Breast	**0.9932**	**0.9605**	**0.9752**	0.9877	0.9538	0.9680	0.9904	0.9521	0.9679
Bupa	**0.7375**	**0.6356**	**0.6721**	0.6833	0.5996	0.6412	0.5955	0.5133	0.5654
Haberman	**0.6736**	**0.4682**	**0.6226**	0.6711	0.4527	0.6011	0.5860	0.3792	0.5395
Hepatitis	**0.862**	0.5734	**0.7892**	0.8580	**0.5944**	0.7882	0.8293	0.5419	0.7439
İonosphere	**0.9624**	**0.8775**	**0.9044**	0.9205	0.8405	0.8685	0.8411	0.7432	0.7828
Pima	0.7967	0.6438	0.7201	**0.8357**	**0.6862**	**0.7572**	0.7571	0.6082	0.6849
Transfusion	0.7412	0.5010	0.6736	**0.7414**	**0.5165**	**0.6896**	0.6919	0.4504	0.6292
Wpbc	**0.6974**	**0.4312**	**0.6178**	0.6789	0.3866	0.5728	0.6356	0.4154	0.6004
mean	**0.808**	**0.6364**	**0.7469**	0.7971	0.6288	0.7358	0.7409	0.5755	0.6893

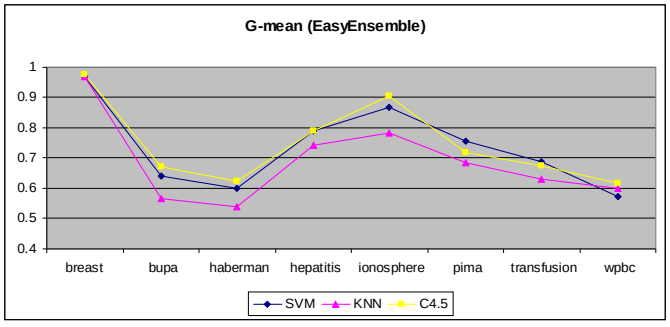
Şekil 4.14 ve Şekil 4.16'da EasyEnsemble algoritmasının AUC ve G-ortalama değerleri sonuçları verilmektedir. C4.5, sekiz veri setinin altısında, DVM iki veri seti üzerinde başarılı olmuştur. F-ölçüsü (Şekil 4.15) C4.5 kullanarak hesaplandığında beş veri seti üzerinde ve DVM üç veri seti üzerinde iyi sonuç vermiştir. KNN temel öğrenici olarak kullanıldığında iyi sonuç vermemektedir.



Şekil 4. 14 EasyEnsemble algoritmasının AUC sonuçları

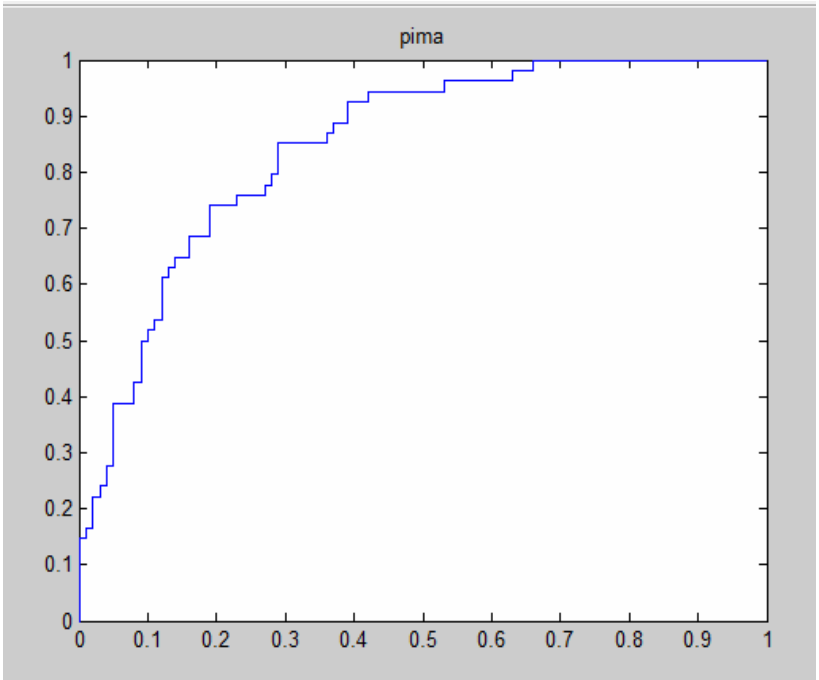


Şekil 4. 15 EasyEnsemble algoritmasının F-ölçü sonuçları

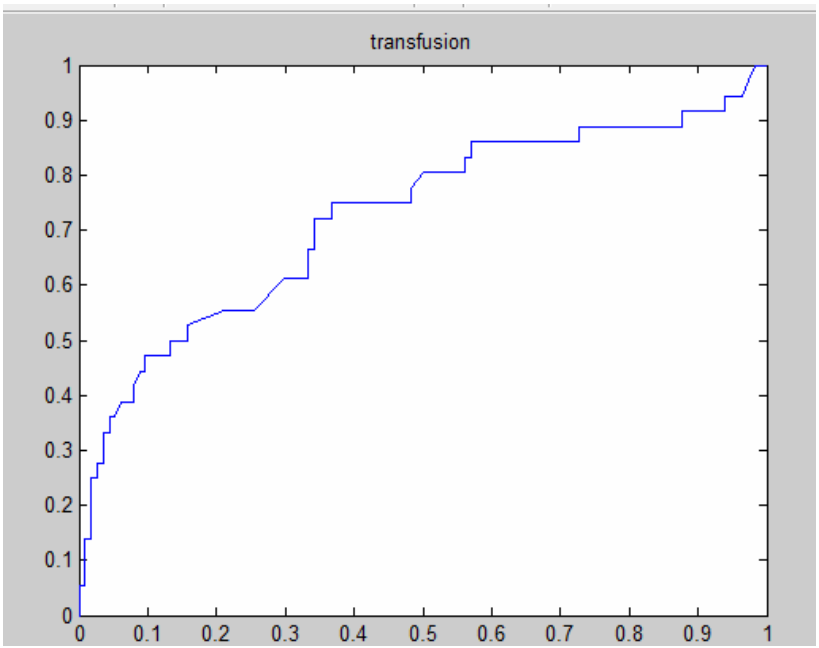


Şekil 4. 16 EasyEnsemble algoritmasının G-ortalama sonuçları

EasyEnsemble algoritması çalıştırıldığında Pima ve Transfusion veri seti için ROC eğrisi Şekil 4.17 ve 4.18'de verilmiştir.



Şekil 4. 17 Pima veri seti için ROC eğrisi



Şekil 4. 18 Transfusion veri seti için ROC eğrisi

Çizelge 4.7 – 4.9 C4.5, DVM ve KNN temel öğrenicilerin RUSboost, BalanceCascade ve EasyEnsemble algoritmaları için AUC, F-ölçü ve G-ortalama değerlerinin tüm veri setleri üzerindeki ortalama değeri verilmiştir.

Çizelge 4.7'de RUSBoost algoritmasının temel öğrenici olarak C4.5 kullanıldığında yüksek AUC ve G-ortalama değerlerini ve DVM kullanıldığında daha başarılı F-ölçü değerini elde ettiği görülmektedir.

Çizelge 4.8 -4.9'da BalanceCascade ve EasyEnsemble algoritmalarının C4.5 kullanarak iyi AUC, F-ölçü and G-ortalama değerlerini elde ettiği görülmektedir.

Bu tablolara dayanarak C4.5'nin DVM ve KNN göre daha iyi AUC, F-ölçü and G-ortalama sonuçlarını verdiği görülmektedir.

Çizelge 4. 7 RUSBoost algoritmasının ortalama değerleri

Algorithm	**Base learners**	**AUC**	**F-measure**	**G-mean**
RUSBoost	C4.5	**0.796**	0.627	**0.736**
	DVM	0.774	**0.636**	0.733
	KNN	0.732	0.586	0.695

Çizelge 4. 8 BalanceCascade algoritmasının ortalama değerleri

Algorithm	**Base learners**	**AUC**	**F-measure**	**G-mean**
EasyEnsemble	C4.5	**0.808**	**0.636**	**0.747**
	DVM	0.797	0.628	0.735
	KNN	0.741	0.576	0.689

Çizelge 4. 9 EasyEnsemble algoritmasının ortalama değerleri

Algorithm	**Base learners**	**AUC**	**F-measure**	**G-mean**
Balance Cascade	C4.5	**0.814**	**0.644**	**0.752**
	DVM	0.792	0.619	0.728
	KNN	0.723	0.567	0.678

4.4.2 Algoritma Performansı

Önceki yapılmış deneylere göre üç algoritma da temel öğrenici olarak C4.5 kullanıldığında iyi sonuçlar vermişti. Bu bölümdeki bizim amacımız C4.5'i temel öğrenici

olarak kullanarak en iyi algoritmayı bulmaktır. Çizelge 4.10'da RUSBoost, BalanceCascade ve EasyEnsemble algoritmalarının AUC, F-ölçü ve G-ortalama performansları verilmektedir.

- AUC hesaplandığında EasyEnsemble sekiz veri setinin dördünde, BalanceCascade üçünde ve RUSBoost bir veri seti üzerinde başarılı olmuştur.
- F-ölçü hesaplandığında EasyEnsemble algoritması dört veri setinde RUSBoost ve BalanceCascade algoritmalarına göre başarılı performans vermiştir. RUSBoost ve BalanceCascade algoritmaları iki veri setinde iyi sonuç vermiştir.
- G-ortalama hesaplandığında EasyEnsemble ve BalanceCascade üç ve RUSBoost bir veri seti üzerinde başarılı olmuştur.

Yapılan deneylere göre C4.5 kullanarak EasyEnsemble algoritmasının RUSBoost ve BalanceCascade göre daha başarılı olduğu tespit edilmiştir.

Çizelge 4. 10 RUSBoost, BalanceCascade ve EasyEnsemble algoritmalarının C4.5 kullandığında AUC, F-ölçü and G-ortalama sonuçları

	AUC			**F-ölçü**			**G-ortalama**		
	RUSBoost	**Balance Cascade**	**Easy Ensemble**	**RUSBoost**	**Balance Cascade**	**Easy Ensemble**	**RUSBoost**	**Balance Cascade**	**Easy Ensemble**
Breast	0.977	0.987	**0.993**	0.918	0.951	**0.961**	0.945	0.969	**0.975**
Bupa	0.718	**0.745**	0.738	0.617	0.626	**0.636**	0.664	**0.673**	0.672
Haberman	**0.702**	0.677	0.674	**0.509**	0.454	0.468	**0.651**	0.612	0.623
Hepatitis	0.806	0.852	**0.862**	0.532	**0.597**	0.573	0.742	**0.797**	0.789
Ionosphere	0.930	0.961	**0.963**	0.832	0.875	**0.878**	0.863	0.897	**0.904**
Pima	0.807	**0.821**	0.797	**0.676**	0.666	0.644	**0.745**	0.739	0.720
Transfusion	0.698	0.712	**0.741**	0.469	0.493	**0.501**	0.629	0.668	**0.674**
Wpbc	0.728	**0.754**	0.697	0.465	**0.488**	0.431	0.649	**0.657**	0.618
	1/8	3/8	**4/8**	2/8	2/8	**4/8**	2/8	***3/8***	***3/8***
İyi sonuç veren veri seti sayısı									

4.4.3 Sınıf Dağılımı Analizi

Sınıf dengesizliği probleminde azınlık sınıfına ait örnek sayısının çoğunluk sınıfına ait örneklerden daha az olduğu durmda, sorunu çözmenin bir yolu sınıf dağılımlarını değiştirmektir. Dengeli dağılımlar, çoğunluk sınıfını alt-örnekleyerek, azınlık sınıfını aşırı-örnekleyerek veya her ikisini birleştirerek bir diğer gelişmiş örnekleme yoluyla elde edilebilir. Weiss, doğruluk ve AUC açısından çeşitli oranlarda sınıf dağılımlarını değiştirerek, karar ağacı üzerindeki sınıf dağılımlarının etkisini araştırmıştır.

Bu tezde yapılan deneylerde EasyEnsemble diğerlerine göre dahabaşarılı algoritma olarak tespit edilmişti. Yukarıda EasyEnsemble tanımlamasına göre algoritmada azınlık eğitim seti Z_{min} ve çoğunluk eğitim seti Z_{maj} verilmişti, alt-örnekleme yöntemi Z_{maj} setinden Z'_{maj} setini rastgele örneklemişti, burada $|Z'_{maj}| < |Z_{maj}|$. Daha önceki deneylerde, Z_{min} azınlık ve Z_{maj} çoğunluk sınıf örnekleri eşit olana kadar yeniden örneklenmişti (50-50). Bu bölümde, deneyler çoğunluk ve azınlık sınıfı örnekleri dağılımını 55-45 60-40 ve 65-35alarak yapılmıştır. Çizelge 4.11 deney sonuçlarını göstermektedir. Sonuçlara göre, EasyEnsemble çoğunluk ve azınlık sınıfı örnekleri dağılımı 55-45 olduğunda her üç değerlendirme ölçüsü için de daha başarılı sonuçlar vermiştir.

Çizelge 4. 11 EasyEnsemble algoritmasının tüm veri seti üzerinde sınıf dağılımına göre ortalama performansı

	Sınıf dağılımı (çoğunluk – azınlık %)			
	50-50	**55-45**	**60-40**	**65-35**
AUC	0.794	**0.801**	0.788	0.79
F-measure	0.596	**0.61**	0.595	0.59
G-mean	0.733	**0.738**	0.724	0.716

BÖLÜM 5

ÖNERİLEN RUSADA ALGORİTMASI VE UYGULAMA SONUÇLARI

5.1 RusAda Tanımı

Çeşitli çarpık veri setleri üzerinde, sınıf dengesizliğini gidermek üzere bu çalışma kapsamında önerdiğimiz, RusAda yönteminin performansı incelenmiştir. Karşılaştırmalar tezimizde kullanılan sınıf dengesizliği öğrenme algoritmaları ile yapılmıştır. RUSBoost, BalanceCascade ve EasyEnsemble deneyler ve karşılaştırmalar için kullanılmıştır. Bu topluluk yöntemleri için deneyler, C4.5 karar ağacını etkili toplulukları oluşturmak için temel öğrenici olarak kullanmıştır. 8 veri seti üzerindeki ortalama sonuçlara göre önerilen RusAda algoritması, diğer gelişmiş sınıf dengesizliği öğrenme yöntemleri ile karşılaştırılabilir veya daha üstün sonuçlar elde etmiştir.

Çizelge 5. 1 Önerilen RusAda algoritması

Bilinen*:*

Eğitim seti S: (x_1,y_1), …, (x_m,y_m), $y^r \in Y, |Y| = 2$

Adaboost iterasyon sayısı T

Boosting prosedürünün iterasyon sayısı K

Temel öğrenici, *C4.5*

1. t=1,2, …, T
    a. Rastgele alt-örneklemeyi kullanarak geçici S_t eğitim kümesi oluşturun
    b. k=1,2, …, K
        i. Boosting prosedürüne göre geçici $S_{t,k}$ eğitim kümesi oluşturun

        Başlatın $D_1(i)=1/n$, i = 1, …, n (n- $S_{t,k}$kümesindeki örnekler sayısı)

Çizelge 5. 1 Önerilen RusAda algoritması (devam)

ii. $S_{t,k}$ örneklerini ve onların $D_{t,k}$ ağırlıklarını C4.5 temel öğreniciye verin

iii. Hipoteze geri dönün $h_{t,k}$: $X \times Y \rightarrow [0,1]$

iv. Hata oranını hesaplayın (S_t ve $D_{t,k}$ için):

$$\varepsilon_{t,k} = \sum_{(i,y):y_i \neq y} D_{t,k}(i)(1 - h_{t,k}(x_i, y_i) + h_{t,k}(x_i, y))$$

v. Ağırlığı güncelleme parametresini hesaplayınız:

$$\alpha_{t,k} = \varepsilon_{t,k} / (1 - \varepsilon_{t,k})$$

vi. Ağırlığı güncelleyin $D_{t,k}$:

$$D_{t,k+1}(i) = D_{t,k}(i)\alpha_{t,k}^{1/2(a + h_{t,k}(x_i, y_i) - h_{t,k}(x_i, y: y \neq y_i))}$$

vii. Normalize edin $D_{t,k+1}$:

$$D_{t,k+1}(i) = D_{t,k+1}(i) / Z_{t,k}$$

$$Z_{t,k} = \sum_i D_{t,k+1}(i)$$

2. Son hipotezin çıkışı:

$$\mathrm{H}_x = \arg\max_{y \in \mathrm{Y}} \sum_{t=1}^{T} \sum_{k=1}^{K} h_{t,k}(x, y) \log 1 / \alpha_{t,k}$$

Çizelge 5.1'de RusAda algoritması gösterilmektedir. RusAda RUSBoost algoritmasını esas almaktadır. Yukarıda bahsettiğimiz gibi RUSBoost, veriyi yeniden alt-örnekleme tekniği RUS'u (Rastgele Alt-örnekleme) AdaBoost algoritmasının içerisinde tanıtır. Yeniden örneklenmiş eğitim veri kümesi iterasyon modelini oluşturmak için kullanılır. Topluluktaki modellerin T sayısı iteratif eğitilmiştir. Bu alt bölümde mevcut RUSBoost algoritmanın performansını artırmak için AdaBoost içinde boosting prosedürünü tanıtan yeni bir RusAda algoritması önerilmiştir. Boosting, sınıf dengesiz olduğu zaman tercih edilen bir algoritmadır. Boosting prosedürü, sınıflandırılması zor olan örneklere odaklanarak sınıflandırma performansını artırır. Önceden yanlış sınıflandırılmış örnekler sonraki turda yeniden örnekleme sırasında eğitim alt kümesinin üyesi olarak seçilmesi için daha büyük ağırlık alacaktır.

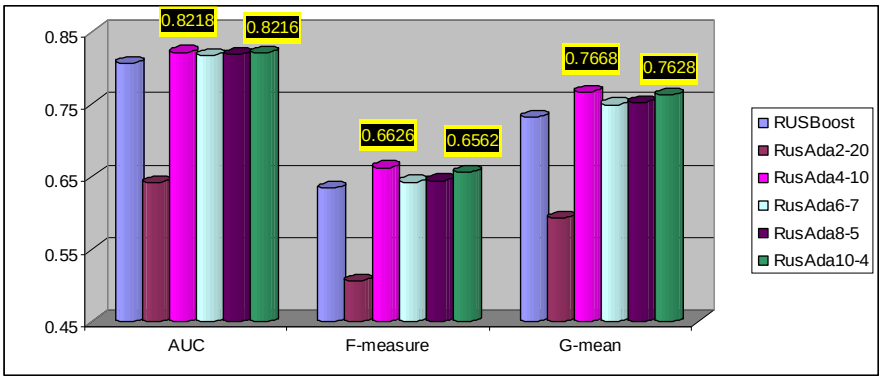
k, bir ve maksimum iterasyon sayısı K (AdaBoost topluluğu sınıflandırıcıların sayısı) arasında bir artırma iterasyonu, $h_{t,k}$ k iterasyonunda eğitilen temel öğreniciyi

kullanarak eğitilen zayıf hipotez ve $h_{t,k}(x_i)$, $h_{t,k}$ hipotezinin çıkışı olsun. Adaboost iterasyonlarında üretilen tüm sınıflandırıcılar son karar için birleştirilir.

Bu çalışmada, yeni RusAda algoritmasının Adaboost iterasyon sayılarını ve boosting prösedürü iterasyon sayılarını değiştirerek çeşitli deneyler yapılmıştır. İlk deneyde, boosting prösedürü iterasyon sayılarına 20, 10, 7, 5, 4 ve AdaBoost iterasyon sayılarına sırasıyla 2, 4, 6, 8, 10 değerleri verilmiştir. Ayrıca, RUSBoost ile karşılaştırılmıştır. Çizelge 5.2'de deney sonuçları gösterilmektedir. RusAda2-20, 2 Adaboost iterasyonunu ve 20 boosting prosedürü iterasyonunu çalıştırır ve diğerleri de aynı şekilde gösterilir. Deneylere göre RusAda4-10 algoritması, 4 AdaBoost iterasyonu ve 10 boosting prosedürü iterasyonu ile diğerlerinden daha iyi sonuç vermiştir. RusAda4-10, RUSBoost ile karşılaştırıldığında da daha iyi performans göstermiştir. Şekil 5.1'de AUC, F-ölçü ve G-ortalamaya göre RusAda4-10 allgoritmasının grafiksel performansı gösterilmektedir.

Çizelge 5. 2 RusAda algoritmasının tüm veri seti üzerinde ortalama AUC, F-ölçü ve G-ortalama sonuçları

	AUC	**F-measure**	**G-mean**
RUSBoost	0.8067	0.6341	0.7321
RusAda2-20	0.6424	0.5068	0.5925
RusAda4-10	**0.8218**	**0.6626**	**0.7668**
RusAda6-7	0.8176	0.6428	0.7496
RusAda8-5	0.819	0.6446	0.7526
RusAda10-4	0.8216	0.6562	0.7628

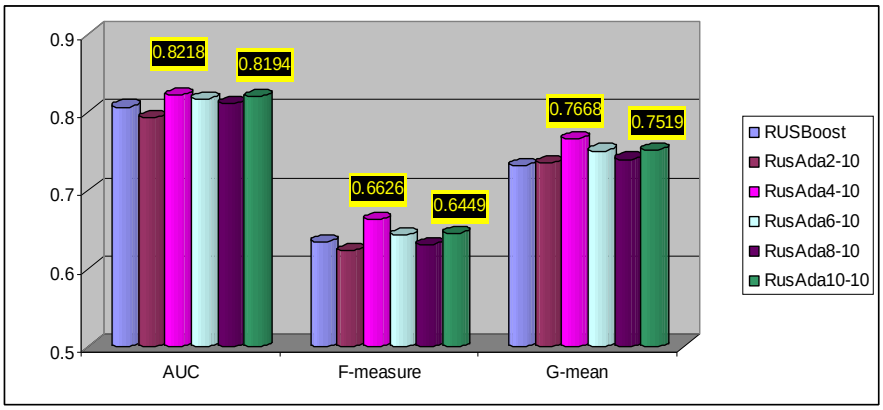


Şekil 5. 1 RusAda algoritmasının AUC, F-ölçü ve G-ortalama sonuçları

İkinci deneyde boosting iterasyon sayısı değiştirilmemiştir ve ona sabit 10 sayısı verilmiştir. Ancak, AdaBoost iterasyon sayısına 2, 4, 6, 8, ve 10 gibi farklı iterasyon sayıları verilmiştir ve aynı zamanda RUSBoost ile karşılaştırılmıştır. Çizelge 5.3' te RusAda4-10 algoritmasının 4 AdaBoost iterasyonu ve 10 boosting iterasyonu ile iyi bir performans verdiği görülmektedir. Görüldüğü gibi, RusAda4-10 bu deneyde de iyi sonuç vermiştir. Şekil 5.2, RusAda4-10 algoritmasının AUC, F-ölçü ve G-ortalamaya göre grafiksel performansını göstermektedir.

Çizelge 5. 3 RusAda algoritmasının tüm veri seti üzerinde ortalama AUC, F-ölçü ve G-ortalama sonuçları

	AUC	**F-ölçü**	**G-ortalama**
RUSBoost	0.8067	0.6341	0.7321
RusAda2-10	0.793	0.6224	0.7346
RusAda4-10	**0.8218**	**0.6626**	**0.7668**
RusAda6-10	0.8167	0.6426	0.7506
RusAda8-10	0.8109	0.6305	0.7397
RusAda10-10	0.8194	0.6449	0.7519



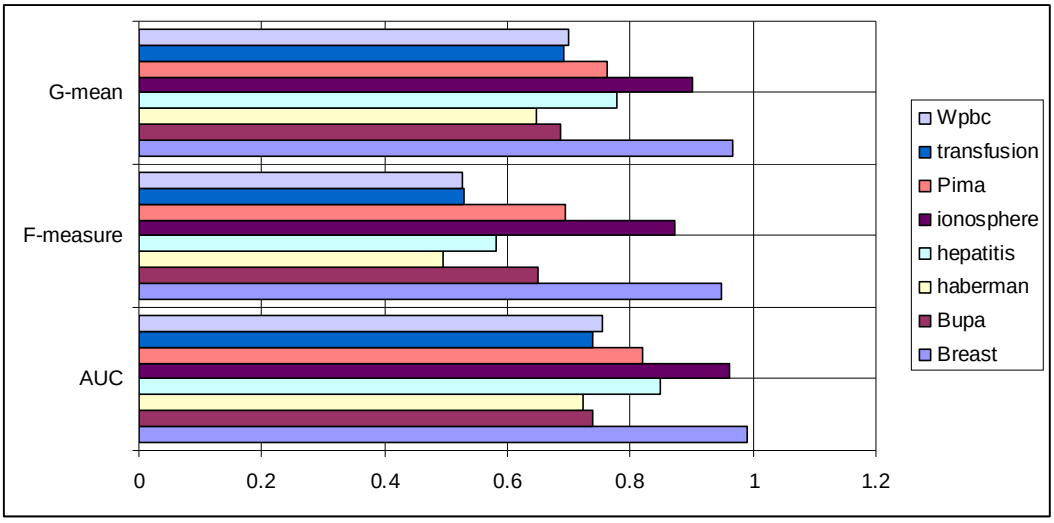
Şekil 5. 2 RusAda algoritmasının AUC, F-ölçü ve G-ortalama sonuçları

Önceki deneylerde de iyi performans verdiğinden dolayi sonraki deneylerde RusAda4-10 kullanılmıştır. Çizelge 5.4, RusAda4-10 algoritmasının tüm veri seti üzerindeki AUC, F-ölçü ve G-ortalama değerlerin vermektedir. Şekil 5.3' te onların grafiği verilmektedir.

RusAda4-10, Breast ve İonosphere veri setlerinde güzel sonuçlar vermekte, ancak Haberman veri setinde kötü performans göstermektedir.

Çizelge 5. 4 RusAda algoritmasının tüm veri setindeki AUC, F-ölçü VE G-ortalama değerleri

RusAda4-10	**AUC**	**F-measure**	**G-mean**
Breast	0.9896	0.9495	0.9672
Bupa	0.7384	0.6501	0.6875
Haberman	0.723	0.4954	0.6479
hepatitis	0.8487	0.5828	0.7769
İonosphere	0.9605	0.8729	0.9014
Pima	0.8207	0.6934	0.7615
Transfusion	0.7393	0.5298	0.6921
Wpbc	0.7545	0.5265	0.6997
Mean	0.8218	0.6626	0.7668



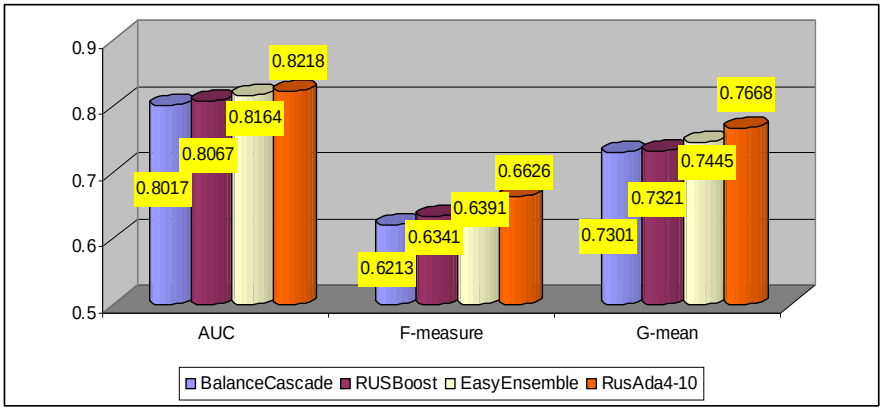
Şekil 5. 3 RusAda algoritmasının tüm veri seti üzerindeki AUC, F-ölçü VE G-ortalama değerleri

Üçüncü deneyde BalanceCascade, RUSBoost, EasyEnsemble ile RusAda4-10 karşılaştırılmıştır ve onun diğerlerinden daha başarılı sonuçlar verdiği görülmüştür. Çizelge 5.5, bu dört algoritmanın AUC, F-ölçütü ve G-ortalama sonuçlarını

göstermektedir. Şekil 5.4, BalanceCascade, RUSBoost, EasyEnsemble ve RusAda4-10 sonuçlarını AUC, F-ölçü ve G-ortalama açısından gözlemlemektedir. Deney sonucundan görüldüğü gibi RusAda4-10 daha iyi performanslı algoritma olarak bulunmuştur.

Çizelge 5. 5 BalanceCascade, RUSBoost, EasyEnsemble ve RusAda algoritmalarının tüm veri seti üzerinde ortalama AUC, F-ölçü ve G-ortalama sonuçları

	AUC	**F-ölçü**	**G-ortlama**
BalanceCascade	0.8017	0.6213	0.7301
RUSBoost	0.8067	0.6341	0.7321
EasyEnsemble	0.8164	0.6391	0.7445
RusAda4-10	**0.8218**	**0.6626**	**0.7668**



Şekil 5. 4 BalanceCascade, RUSBoost, EasyEnsemble ve RusAda algoritmalarının AUC, F-ölçü ve G-ortalama sonuçları

BÖLÜM 6

SONUÇ VE ÖNERİLER

Bu çalışmada veri madenciliği alanında karşılaşılan önemli sorunlardan biri olan sınıf dengesizliği sorununu gidermek için yaygın kullanılan RUSBoost, EasyEnsemble ve BalanceCascade algoritmaları, temel öğrenciler seçimi ve algoritma performansına göre karşılaştırılmıştır. Deneyler temel sınıflandırıcı olarak C4.5, DVM ve KNN algoritmalarını kullanarak 8 ayrı gerçek dünya veri setleri üzerinde gerçekleştirilmiştir ve AUC, G-ortalama ve F-ölçüleriyle sınıflama başarımları karşılaştırılmıştır. C4.5, temel öğrenici olarak alındığında diğer DVM ve KNN algoritmalarına göre daha iyi sonuç vermiştir. Yapılan deney sonuçlarına göre EasyEnsemble yönteminin sınıf dengesizliği sorununu gidermek için en iyi algoritma olduğu tespit edilmiştir. Çalışmamızın sonuçlarına göre, EasyEnsemble 55-45 (çoğunluk - azınlık) sınıf dağılımı ile RUSBoost ve BalanceCascade göre daha iyi sonç vermiştir. Ayrıca, bu çalışma kapsamında EasyEnsemble dahil olmak üzere karşılaştırdığımız diğer algoritmalardan daha iyi performans gösteren RusAda algoritması önerilmiştir. Tez kapsamında geliştirdiğimiz RusAda sınıf dengesizliği sorununu hafifletmek için en iyi algoritma olarak tespit edilmiştir.

KAYNAKLAR

[1] Visa, S. ve Ralescu, A., (2005). "Issues in mining imbalanced data sets - a review paper", In Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, 67-73.

[2] Guo, X., Yin, Y., Dong, C., Yang, G ve Zhou, G., (2008). "On the class imbalance problem", In ICNC'08: Proceedings of the 2008 Fourth International Conference on Natural Computation, 192-201.

[3] Fan, W., Stolfo, S.J. ve Zhang. J., (1999). "AdaCost: misclassification cost-sensitive boosting," Proc.Int. Conf. Machine Learning, Bled, Slovenia, 97-105.

[4] Tang, Y., Zhang, Y. ve Chawla, N.V., (2009). "SVMs modeling for highly imbalanced classification," IEEE Trans. Syst., Man, and Cybern. - Part B, 39(1):281 - 288.

[5] Zeng, Z. ve Gao, J., (2009). "Improving SVM classification with imbalance data set," Proc. 16th Int.Conf. Neural Information Processing, Bangkok, Thailand, 389-398.

[6] Li, C., (2007). "Classifying imbalanced data using a bagging ensemble variation", In ACM-SE 45: Proceedings of the 45th Annual Southeast Regional Conference, 203-208.

[7] Nitesh V.C., (2003). "C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure", In Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 1-8.

[8] Mahesh V.J., Kumar, V., ve Ramesh, C.A., (2001). "Evaluating boosting algorithms to classify rare classes: Comparison and improvements", In IBM Research Report, 257-264.

[9] Seiffert, C., Taghi M.K., Hulse, J.V. ve Napolitano, A., (2008). "Improving learner performance with data sampling and boosting", In 20th IEEE International Conference on Tools with Artificial Intelligence, 452-459.

[10] Estabrooks, A., Jo, T. ve Japkowicz, N., (2004). "A multiple resampling method for learning from imbalanced data sets", In Computational Intelligence 20, 20: 18-36.

[11] Hulse, J.V., Taghi M. K., ve Napolitano, A., (2007). "Experimental perspectives on learning from imbalanced data", In ICML '07: Proceedings of the 24th international conference on Machine learning, 935-942.

[12] He, H. ve Garcia, E. A., (2009). "Learning from imbalanced data", IEEE Transactions on Knowledge and Data Engineering, 21(9):1263-1284.

[13] Batista, G. E. A. P. A., Prati, R. C. ve Monard, M.C., (2004). "A study of the behavior of several methods for balancing machine learning training data", Special issue on learning from imbalanced datasets, Sigkdd Explorations, 6(1):20-29.

[14] Jorma, L., (2001). "Improving identification of dificult small classes by balancing class distribution", In Conference on artificial intelligence in medicine in Europe No8, Cascais , PORTUGAL, 2101: 63-66.

[15] Rao, K. N., Rao, T.V. ve Lakshmi, D. R., (2012). "A Novel Class Imbalance Learning using Ordering Points Clustering", International Journal of Computer Applications (0975 – 8887) 51(16).

[16] Satuluri, N. ve Kuppa, M.R., (2012). "A Novel Class Imbalance Learning Using Intelligent Under-Sampling", International Journal of Database Theory and Application 5:(3).

[17] Liu, X.Y., Wu, J. ve Zhou, Z.H., (2009). "Exploratory undersampling for class imbalance learning", IEEE Transactions on Systems, Man and Cybernetics, 39(2):539-550.

[18] Panov, P. ve Džeroski, S., (2007). "Combining bagging and random subspaces to create better ensembles," Advances in Intelligent Data Analysis VII (Springer, LNCS 4723), 118-129.

[19] Chawla, N.V., Bowyer, K.W., Hall, L.O. ve Kegelmeyer, W. P., (2002). "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research 16, 321–357.

[20] Wang, B.X. ve Japkowicz, N., (2004). "Imbalanced data set learning with synthetic examples", In IRIS Machine Learning Workshop.

[21] Han, H., Wang, W.Y. ve Mao, B. H., (2005). "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," Proc. Int. Conf. Intelligent Computing, Hefei, China, 878-887.

[22] He, H., Bai, Y. ve Garcia, E.A., (2008). "ADASYN: adaptive synthetic sampling approach for imbalanced learning", Proc. Int. J. Conf. Neural Networks, Hong Kong, China, 1322-1328.

[23] Sun, Y. Kamel, M.S. ve Wong, A.K.C., (2007). "Cost-sensitive boosting for classification of imbalanced data", Pattern Recognition, 40(12):3358-3378.

[24] Sun, Y., (2007). "Cost-sensitive boosting for classification of imbalanced data", Ph.D. Dissertation, Dept.Electrical and Comput. Eng., Univ. of Waterloo, Ontario, Canada.

[25] Díaz-Uriarte, R. ve Andrés, S. A., (2006). "Gene selection and classification of microarray data using random forest", BMC Bioinformatics, 7(3): 10.1186/1471-2105-7-3.

[26] Sotiris B. K., Kanellopoulos, D. ve Pintelas, D., (2006). "Handling imbalanced datasets: A review", GESTS International Transactions on Computer Science and Engineering, 30(1):25-36.

[27] Tao, D., Tang, X., Li, X. ve Wu, W., (2006). "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(7):1088-1099.

[28] Mease, D., Wyner, A.J. ve Buja, A., (2007). "Boosted classification trees and class probability/quantile estimation", The Journal of Machine Learning Research, 8: 409-439.

[29] Ting, K.M., (2000). "A comparative study of cost-sensitive boosting algorithms", In Proc. 17th International Conf. on Machine Learning, 983-990.

[30] Seiffert, C., Khoshgoftaar, T. M. Ve Hulse, J. V., (2010). "RUSBoost: A hybrid approach to alleviating class imbalance", IEEE Transactions Syst., Man, And Cybern. -Part A: Syst. and Humans, 40(1):185-197.

[31] Guo, H. ve Viktor, H. L., (2004). "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach", SIGKDD Explor. Newsl., 6(1):30-39.

[32] Chawla, N.V., Japkowicz, N. ve Kotcz, A., (2004). "Editorial: Speficial issue on learning from imbalanced data sets", SIGKDD Explor. Newsl., 6(1):1-6.

[33] Polikar, R., (2006). "Ensemble based systems in decision making", IEEE Circuits and Syst. Mag., 6:21–45.

[34] Kaya, H. ve Köymen, K., (2008). "Veri Madenciliği Kavramı ve Uygulama Alanları", Doğu Anadolu Bölgesi Araştırmaları.

[35] Larose, D.T., (2005). "Discovering Knowledge In Data An Introduction to Data Mining", New Jersey: John Wiley and Sons Ltd.

[36] Pearson, R.K., Gonye, G.E. ve Schwaber, J.S., (2003). "Imbalanced clustering for microarray time-series", In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets.

[37] Zhao, X.M., Li,X., Chen, L. ve Aihara, K., (2008). "Protein classification with imbalanced data", Proteins: Structure, Function, and Bioinformatics, 70:1125-1132.

[38] Weiss, G.M., (2004). "Mining with rarity: a unifying framework", SIGKDD Explor. Newsl., 6 (1):7-19.

[39] Japkowicz, N., (2003). "Class imbalances: are we focusing on the right issue", In Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets, 17-23, 2003.

[40] Prati, R.C., Batista, G.E.A.P.A. ve Monard, M. C., (2004). "Class imbalances versus class overlapping: An analysis of a learning system behavior", Lecture Notes in Computer Science, 2972:312-321.

[41] Batista, G.E.A.P.A., Prati, R.C. ve Monard, M.C., (2005). "Balancing strategies and class overlapping", Advances in Intelligent Data Analysis, 3646:24-35.

[42] Zhang, J. ve Mani, I., (2003). "knn approach to unbalanced data distributions: A case study involving information extraction", In Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 42-48.

[43] Yan, R., Liu, Y., Jin, R. ve Hauptman, A., (2003). "On predicting rare classes with svm ensembles in scene classification", In IEEE International Conference on Acoustics, Speech and Signal Processing.

[44] Lee, H.J ve Cho, S., (2006). "The novelty detection approach for different degrees of class imbalance", Lecture Notes in Computer Science Series as the Proceedings of International Conference on Neural Information Processing, 4233:21-30.

[45] Chawla, N.V. ve Sylvester, J., (2007). "Exploiting diversity in ensembles: Improving the performance on unbalanced datasets", Multiple Classifier Systems, 4472:397-406.

[46] Kuncheva, L.I., (2004). "Combining Pattern Classifiers: Methods and Algorithms", Wiley Press.

[47] Rao, N.S.V., (2001). "On fusers that perform better than best sensor", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(8):904-909.

[48] Breiman, L., (2001). "Random forests," Machine Learning, 45(1): 5-32.

[49] Kuncheva, L.I., (2003). "That elusive diversity in classifier ensembles", Pattern Recognition and Image Analysis, 2652:1126-1138.

[50] Kuncheva, L.I. ve Whitaker, C.J., (2003). "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy", Machine Learning, 51: 181-207.

[51] Liu, X.-Y. ve Zhou, Z.-H., (2006). "The influence of class imbalance on cost-sensitive learning: An empirical study", Proc. IEEE 6th Int. Conf. Data Mining, Hong Kong, China, 970-974.

[52] Fawcett, T., (2003). "Roc graphs: Notes and practical considerations for researchers", HP Labs, Palo Alto, CA, Technical Report HPL-2003-4.

[53] Fawcett, T., (2006). "An introduction to roc analysis", Pattern Recognition Letters, 27(8):861-874.

[54] Ling, C.X., Huang, J. ve Zhang, H., (2003). "Auc: a statistically consistent and more discriminating measure than accuracy", In Proceedings of 18th International Conference on Artificial Intelligence (IJCAI2003), 329-341.

[55] Tan, P., Steinbach, M. ve Kumar, V., (2006). "Introduction to Data Mining", Boston, MA: Pearson Addison Wesley.

[56] Seiffert, C. Khoshgoftaar, T.M. ve Hulse, J.V., (2008). "Hybrid sampling for imbalanced data", In IEEE International Conference on Information Reuse and Integration, 202-207.

[57] Brown, G., (2010). Ensemble learning. Encyclopedia of Machine Learning, 1-24.

[58] Zhu, X., (2007). "Lazy bagging for classifying imbalanced data", In Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007., 763-768.

[59] Elkan, C., (2001). "The foundations of cost-sensitive learning," Proc. 17th Int. Joint Conf. Artificial Intelligence, Seattle, WA, 973-978.

[60] Maloof, M. A., (2003). "Learning when data sets are imbalanced and when costs are unequal and unknown", Proc. Int. Conf. Machine Learning (ICML) 2003 Workshop on Learning from Imbalanced Datasets, Washington, DC.

[61] McCarthy, K., Zabar, B. ve Weiss, G.M., (2005). "Does cost-sensitive learning beat sampling for classifying rare classes," Proc. 1st Int. Workshop Utility-Based Data Mining, Chicago, IL, 69-77.

[62] Bozkır, A.S., (2009). Olap ve Veri Madenciliği Teknolijelerinden Yararlanılarak Web Tabanlı Bir Karar Destek Sisteminin Gerçekleştirilmesi, Hacettepe Üniversitesi, Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin Bilgisayar Mühendisliği Anabilim Dalı, Ankara.

[63] İnternet: C4.5 Algorithm , "C4.5 algorithm", http://en.wikipedia.org/wiki/C4.5 algorithm (2009).

[64] Özkan, Y., (2008). "Veri madenciliği yöntemleri", Papatya Yayıncılık Eğitim, 1-88.

[65] Silahtaroğlu, G., (2008). "Kavram ve Algoritmalarıyla Temel Veri Madenciliği", Papatya Yayıncılık, İstanbul, 174.

[66] Dunham, M.H., (2003). "Data Mining Introductory and Advanced Topics", Prentice Hall, New Jersey.

[67] Tong, S. ve Koller, D., (2002). "Support vector machine active learning with applications to text classification", Journal of Machine Learning Research (JMLR), 2:45–66.

[68] Muslu, D., (2009). Sigortacılık Sektöründe Risk Analizi: Veri Madenciliği Uygulaması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.

[69] Sun, Y., Kamel, M.S. ve Wang, Y., (2006). "Boosting for learning multiple classes with imbalanced class distribution", In ICDM '06: Proceedings of the Sixth International Conference on Data Mining, 592-602.

[70] Dirican A., (2001). "Tanı Testi Performanslarının Değerlendirilmesi ve Kıyaslanması",Cerrahpaşa Tıp Dergisi, 32(1): 25-30.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Akkenzhe SARMANOVA
Doğum Tarihi ve Yeri	:25.11.1988 KAZAKİSTAN
Yabancı Dili	:İngilizce, Rusça, Kazakça
E-posta	:akkenzhe1988@mail.ru

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Bilgisayar Mühendisliği	Avrasya Milli Üniversitesi	2010
Lise	Fizik ve matematik	Yetenekli Öğrenciler İçin Milli Lise	2006

YAYINLARI

Konferans Bildiri

1. SİU	21.IEEE Sinyal İşleme ve İletişim Uygulama Kurultayı, "Veri Madenciliğindeki Sınıf Dengesizliğinin Giderilmesi", Kıbrıs, 2013
2.DMIN	The 9th International Conference on Data Mining, "Alleviating the Class Imbalance Problem in Data Mining" Las Vegas, USA, 2013