JISTA

Journal of Intelligent Systems: Theory and Applications

SEPTEMBER 2022

ISSN: 2651-3927

VOL 5 NO 2

ARTIFICIAL INTELLIGENT > MACHINE LEARNING > DEEP LEARNING
https://dergipark.org.tr/en/pub/jista

https://dergipark.org.tr/en/pub/jista

Journal of Intelligent Systems: Theory and Applications
2022 Volume: 5 Issue: 2

Editorial Boards

Editorial Assistants

https://dergipark.org.tr/en/pub/jista

Journal of Intelligent Systems: Theory and Applications

2022 Volume: 5 Issue: 2

Contents

Research Articles

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 92-105
DOI: 10.38016/jista.1069541

Farklı Sınıflandırıcılar ve Yeniden Örnekleme Teknikleri Kullanılarak Kalp Hastalığı Teşhisine Yönelik Karşılaştırmalı Bir Çalışma

Onur Sevli[1]*

[1]Burdur Mehmet Akif Ersoy Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Burdur, Türkiye

onursevli@mehmetakif.edu.tr

Öz

Kalp hastalıkları dünya genelinde yaygın olarak görülmekte ve küresel ölümlerin üçte birlik kısmına neden olmaktadır. Kalp hastalığının semptomlarını ayırt etmedeki zorluk ve çoğu kalp hastasının kriz anına kadar semptomların farkında olmaması, hastalığın tanısını zorlaştırmaktadır. Bir yapay zekâ disiplini olan makine öğrenmesi bilinen verilerden yola çıkarak, yeni vakaların teşhisi konusunda uzmanlar için başarılı karar destek çözümleri sunmaktadır. Bu çalışmada kalp hastalıklarının erken teşhisine yönelik çeşitli makine öğrenmesi teknikleri kullanarak sınıflamalar gerçekleştirilmiştir. Çalışma literatürde yaygın olarak kullanılan UCI kalp hastalığı veri seti üzerinde gerçekleştirilmiştir. Sınıflandırma başarısını arttırmak için, eldeki veri setinin sınıf dengesini sağlamaya yönelik olarak yeniden örnekleme teknikleri kullanılmıştır. Naive Bayes, Karar Ağaçları, Destek Vektör Makinesi, K En yakın Komşu, Lojistik Regresyon, Rastgele Orman, AdaBoost ve CatBoost olmak üzere 8 farklı makine öğrenmesi tekniğinin her biri için örneklemesiz sınıflama yanında fazla örnekleme ve az örnekleme tekniklerinden 8 farklı yöntem kullanılarak toplam 72 sınıflandırma işlemi gerçekleştirilmiştir. Her bir sınıflandırma işleminin sonucu doğruluk, kesinlik, duyarlılık, F1 skoru ve AUC olmak üzere 5 farklı parametre ile raporlanmıştır. En yüksek doğruluk değeri Rastgele Orman ve InstanceHardnessThreshold az örnekleme tekniğinin kullanıldığı sınıflamada %98.46 olarak elde edilmiştir. Elde edilen ölçümlerin literatürde son yıllarda yapılan benzer çalışmalarda ulaşılan sonuçlardan daha yüksek olduğu görülmüştür.

Anahtar kelimeler: Kalp hastalığı teşhisi, Makine öğrenmesi, Yeniden örnekleme

A Comparative Study of Heart Disease Diagnosis using Various Classifiers and Resampling Techniques

Abstract

Heart diseases are common worldwide and cause one-third of global deaths. The difficulty in distinguishing the symptoms of heart disease and the fact that most heart patients are not aware of the symptoms until the moment of crisis make the diagnosis of the disease difficult. Machine learning, an artificial intelligence discipline, provides experts with successful decision support solutions in diagnosing new cases based on known data. In this study, classifications were made using various machine learning techniques for the early diagnosis of heart diseases. The study was carried out on the UCI heart disease dataset, which is widely used in the literature. In order to increase the classification success, resampling techniques were used to ensure the class balance of the dataset. For each of 8 different machine learning techniques, namely Naive Bayes, Decision Trees, Support Vector Machine, K Nearest Neighbor, Logistic Regression, Random Forest, AdaBoost, and CatBoost, in addition to no-sampling classification, 8 different methods from oversampling and undersampling techniques were used to make a total of 72 classification processes were carried out. The result of each classification process is reported with 5 different parameters: accuracy, precision, recall, F1 score, and AUC. The highest accuracy value was obtained as 98.46% in the classification using Random Forest and InstanceHardnessThreshold undersampling technique. It was observed that the measurements obtained were higher than the results obtained in similar studies conducted in the literature in recent years.

Keywords: Heart disease diagnosis, Machine learning, Resampling

* Sorumlu yazar.
E-posta adresi: onursevli@mehmetakif.edu.tr

Alındı : 7 Şubat 2022
Revizyon : 2 Mart 2022
Kabul : 11 Mart 2022

1. Giriş (Introduction)

Kalp sağlığının korunması yaşam için son derece kritik bir öneme sahiptir. Kalp, yaşamın temeli olan kan tedarik sistemini yönetme, kan basıncını sağlama, tek yönlü kan akışını güvence altına alma ve oksijence zengin kanın dokulara, dokulardaki kirli kanın akciğerlere akışını sağlama gibi hayati fonksiyonları yerine getirir. Kaslı bir yapıya sahip olan kalp dakikada ortalama 70 kez kasılıp gevşeyerek, vücuda dakikada yaklaşık 5, saatte 300 ve günde 7200 litre kan pompalar.

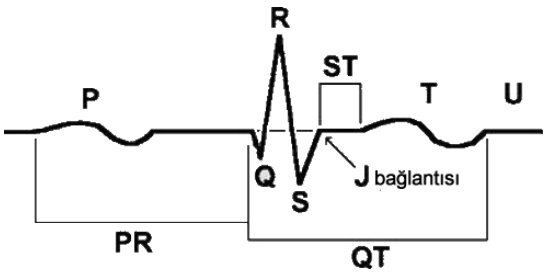
Kalp sağlığı üç temel sistemin birbiri ile uyum içinde çalışması ile mümkündür. Bunlar kardiyovasküler sistem, koroner arterler ve sinir ağıdır. Kardiyovasküler sistem, kanın tek yönde akışını sağlayan valflerle donatılmış kaslı bir pompalama sistemi ve vücudun en ince noktalarına kadar uzanan bir damar ağından oluşur. Kalp odacıkları sürekli kanla dolu olmasına rağmen kalp dokusu bu kanla beslenemez. Koroner arterler kalp yüzeyi boyunca yayılarak kalp kaslarına oksijence zengin kanın taşınmasını ve kalp dokusunun beslenerek düzenli şekilde çalışmasını sağlar. Kalbin kasılıp gevşemesi ise elektriksel sinyaller ve bu sinyalleri yöneten bir sinir ağı ile sağlanır.

Farklı sağlık problemleri bu sitemlerin çalışmasını olumsuz etkileyerek yaşamı tehdit eden kalp rahatsızlıkların oluşmasına sebep olur. Kalp sağlığını riske atan durumların ve belirtilerin erken tespiti hayati önem taşımaktadır. Kalp hastalıkları genetik etkenlere bağlı olabileceği gibi; sağlıksız yaşam, sigara kullanımı, diyabet, obezite, enfeksiyonlar, hipertansiyon gibi farklı sebeplerle kalp damar hastalıkları, kalp ritim bozuklukları, kalp yetmezliği ve kalp krizleri meydana gelebilir. Koroner arterlerdeki tıkanıklık veya daralma sonucu kalp kasını besleyen kan akışında kesilme meydana gelir. Bu durum kalp kasının yeterince oksijenlenmesine engel olarak kalp dokusunda hasar meydana getirebilir. Kalbi besleyen damarların duvarlarında biriken kolesterol gibi maddeler plaklar oluşturur. Zaman içinde çoğalan plaklar damarları tıkayarak kalp krizinin oluşumuna neden olur. Zamanında ve doğru müdahale ile tıkanan damarlar açılmazsa kalpte doku kaybı meydana gelir. Bu durum, kalbin pompalama gücünü azaltır ve kalp yetmezliğine sebep olur. Kalp dokusunun yeterince kanlanamadığı her an, kalıcı hasar oluşma riski artar.

Çağımızın en büyük sorunlarından biri olan stres, diğer tüm rahatsızlıklarda da olduğu gibi kalp hastalıkları ve krizlerinde tetikleyici bir role sahiptir. Ayrıca Kaliforniya merkezli Ailesel Hiperkolesterolemi Vakfı (FH Fountation) araştırmacıları tarafından yürütülen bir araştırmada, COVID-19'un genetik olarak yüksek kolesterol, kalp hastalığı veya her ikisi için risk taşıyanlarda kalp krizi oranını artırdığını ortaya konmuştur (Myers vd., 2021). Dünya Sağlık Örgütü'nün verilerine göre kalp hastalıklarının küresel ölçekte görülme oranı giderek artmakta olup, bulaşıcı olmayan hastalıklara bağlı ölümlerin %46'sı kalp damar hastalıklarından kaynaklanmaktadır. Kardiyovasküler hastalıklar son 15 yılın en yaygın sağlık sorunları olup, küresel ölümlerin tamamının %31'lik bölümü kalp ile ilgili hastalıklardan kaynaklanmaktadır (WHO, 2021). Türkiye İstatistik Kurumu verilerine göre ise son on yıldaki ölümlerin %40 oranı ile büyük çoğunluğunu kalp damar hastalıkları oluşturmaktadır (TUIK, 2021).

Kalp hastalıkları göğüste ağrı ve sıkışma, kısa aktiviteler sonrası nefes darlığı, baş dönmesi ve bayılmalar, kalbin çok sert ya da yavaş atması, kalp damarlarının sertleşmesine bağlı olarak bacak ve kollarda uyuşma, ani soğuk terlemeler, bulantı ve kusma gibi belirtiler verebilir. Bazı hastalarda belirgin şikâyetler görülmeden de kalp krizi oluşabilir. Kalp hastalıklarında erken müdahale son derece önemlidir. Erken müdahale can kaybı riskini ve kalp kasının zarar görme olasılığını azaltır.

Kalp hastalıklarının tanısı uzman hekimler tarafından yapılır. Tanı için fiziksel muayene, hastalığın durumuna bağlı olarak yapılacak tetkikler yanında elektrokardiyografi (EKG) ölçümleri kullanılır. EKG vücuda yapıştırılan elektrotlar ile kalbin elektriksel aktivitesini kaydeder. EKG kayıtları olan elektrokardiyogramlarda kalp atımları P,Q,R,S,T,U dalgalarından oluşan sinyaller şeklinde yorumlanır (Şekil 1). EKG dalgalarındaki değişimler, dalga düzeninin farklılaşması, dalgalar arasındaki sürelerde farklılıklar uzmanlara kalp hastalığı konusunda fikir verir (Kartal ve Köksal, 2020).



Şekil 1. EKG dalgası (ECG wave)

Kalp hastalığına sebep olan pek çok etken olmasından, belirtilerin bazen belirgin olmaması veya diğer hastalıklarla karıştırılabilmesinden dolayı teşhisi komplikedir. Pek çok kalp hastasında göğüs ağrısı ve yorgunluk gibi belirtiler görülürken, %50 gibi büyük bir bölümü kalp krizi geçirene kadar belirtilerin farkına varamamaktadır (Das vd., 2009). Bu durumda hastalığın erken tespiti uzmanlar için de zorlaşmaktadır. Çok sayıda değişken verinin yorumlanarak hastalık teşhisin zamanında ve doğru şekilde yapılabilmesi hayati öneme sahiptir. Bu konuda uzmanların doğru kararlar vermelerine yardımcı olacak destek sistemlere ihtiyaç duyulmaktadır.

Son yıllarda yapay zekâ teknolojisindeki gelişmeler sağlık alanında da alternatif çözümler sunmaktadır. Mevcut verilerden öğrenerek yeni vakalar hakkında sağlıklı tahminler üretmeyi sağlayan makine öğrenmesi, son zamanlarda popülerliği hızla artan bir teknolojidir. Makine öğrenmesinde belirli bir sonuca etki eden

parametreler üzerinden, daha önceki mevcut verilerden yola çıkılarak, yeni vakalar hakkında istikrarlı tahminler üretilebilmektedir.

Literatürde, kalp hastalıklarının teşhisine yönelik farklı makine öğrenmesi teknikleri kullanılarak gerçekleştirilen çalışmalar mevcuttur. Bu çalışmalarda Naive Bayes(NB), Destek Vektör Makinesi (DVM), Karar Ağaçları (KA), K En yakın Komşuluk (KNN), Rastgele Orman (RO) ve Yapay Sinir Ağları (YSA) gibi algoritmalar yaygın olarak kullanılmaktadır. Literatürde son yıllardaki benzer çalışmalar kronolojik sırada aşağıda özetlenmiştir.

Miranda vd. (2016) kalp hastalığı riskini tahminleme için açık uçlu sorulardan oluşan görüşmeler yapmışlar ve elde ettikleri 60589 kayıt ve 38 özellikten oluşan veri seti üzerinde gerçekleştirdikleri üç seviyeli risk sınıflamasında NB yöntemi ile en yüksek %87.98 doğruluk değerine ulaşmışlardır. Wiharto vd. (2016), 14 özellik ve 303 örnekten oluşan UCI veri seti üzerinde SMOTE aşırı örnekleme ve C4.5 tekniği ile gerçekleştirdikleri kalp hastalığı düzeyi sınıflamasında %84.2 AUC değeri elde etmişlerdir. Jabbar vd. (2016) ise RO yöntemi kullanarak gerçekleştirdikleri kalp hastalığı tahminleme çalışmasında %83.70 doğruluğa ulaşmış ve KA yöntemine göre daha yüksek bir başarı sağlandığını ortaya koymuşlardır.

Kim ve Kang (2017), Koreli 4146 adet bireyden elde edilen veriler üzerinde önce birbiriyle ilişkili nitelikleri tespit etmişler ve ardından YSA kullanarak gerçekleştirdikleri kalp hastalığı riski tahminleme çalışmasında %74.9 AUC değerine ulaşmışlardır. Arabasadi vd. (2017) sinir ağları kullanarak 303 örnek ve 54 özellikten oluşan Z-Alizadeh Sani kalp hastalığı veri seti üzerinde gerçekleştirdikleri sınıflama çalışmasında, ağ parametrelerinin genetik algoritmalar kullanılarak optimize edilmesi ile model başarısının arttığını ortaya koymuşlardır. Liu vd. (2017) 202 örnek ve 2 sınıftan oluşan Statlog kalp hastalığı veri seti üzerinde ReliefF and Rough Set (RFRS) özellik seçim yöntemi ve C4.5 sınıflayıcı kullanarak gerçekleştirdikleri çalışmada en yüksek %92.59 doğruluğa ulaşmışlardır.

David ve Belcy (2018), RO, KA ve NB sınıflayıcıları kullanarak UCI veri seti üzerinde gerçekleştirdikleri sınıflamada en yüksek doğruluğu %81 ile RO kullanımında elde etmişlerdir. Haq vd. (2018), aynı veri seti üzerinde farklı özellik seçim teknikleri ve yedi farklı makine öğrenmesi algoritması kullanarak gerçekleştirdikleri sınıflama çalışmasında en yüksek doğruluğu %86 ile DVM kullanarak elde etmişlerdir. Malav ve Kadam (2018), yine aynı veri seti üzerinde ilk olarak K-means uygulayarak elde ettikleri kümeleme sonucunu YSA'ya girdi olarak verip gerçekleştirdikleri sınıflamada %93.52 doğruluğa ulaşmışlardır. Poornima ve Gladis (2018), boyut indirgeme ve YSA kullanarak gerçekleştirdikleri sınıflama çalışmasında %94 doğruluk sağlamışlardır.

Ali vd. (2019b), UCI veri seti üzerinde istatistiksel özellik seçimi ve optimize edilmiş YSA kullanarak gerçekleştirdikleri sınıflamada %93.33 doğruluk elde etmişlerdir. Mohan vd. (2019), aynı veri seti üzerinde lineer model ile hibrit şekilde RO yöntemi kullanarak gerçekleştirdikleri sınıflandırma çalışmasında %88.7 doğruluğa ulaşmışlardır. Aynı veri seti üzerinde Ali vd. (2019a), bir yığın halinde iki ayrı DVM modelini kullanarak, lineer çekirdekli ilk model ile özellik seçimi gerçekleştirmiş, RBF çekirdekli model ile sınıflama gerçekleştirmişler ve %91.11 elde etmişledir. Latha ve Jeeva (2019) UCI veri seti üzerinde kolektif öğrenme algoritmaları ile gerçekleştirdikleri sınıflama çalışmasında zayıf sınıflayıcılara oranla maksimum %7 doğruluk artışı sağlandığını raporlamışlardır.

Mienye vd. (2020); Framingham, Massachusetts sakinlerinden toplanan 4238 adet örnekten oluşan kalp hastalığı veri seti üzerinde YSA kullanarak gerçekleştirdikleri sınıflama çalışmasında %90 doğruluğa ulaşmışlardır. Terrada vd. (2020), Z-Alizadeh Sani kalp hastalığı veri seti üzerinde YSA kullanarak gerçekleştirdikleri sınıflama çalışmasında %94 doğruluk elde etmişlerdir. Tama vd. (2020), UCI veri seti üzerinde kolektif öğrenme yöntemi ile gerçekleştirdikleri sınıflama çalışmasında %85.71 doğruluk elde etmişlerdir. Akalın vd. (2020), aynı veri seti üzerinde KNN, Gaussian Bayes ve RO olmak üzere üç farklı yöntem ile gerçekleştirdikleri sınıflamada %80, %80 ve %82 doğruluğa ulaşmışlardır.

Elhoseny vd. (2021), 13 adet özellik ve 270 örnekten oluşan kalp hastalığı veri seti üzerinde farklı makine öğrenmesi teknikleri kullanarak gerçekleştirdikleri sınıflama çalışmasında en yüksek doğruluk değerlerini %82.5, %81.5 ve %80.8 ile AdaBoost, LogitBoost ve NB yöntemleri ile elde etmişlerdir. Rani vd. (2021), yine aynı veri seti üzerinde SMOTE aşırı örnekleme yöntemi ve NB tekniği ile gerçekleştirdikleri sınıflamada %85.07 doğruluk değerine ulaşmışlardır. Aynı teknik ile aşırı örnekleme kullanmadan elde ettikleri doğruluk ise %84.79'dur. Katarya ve Meena (2021), farklı makine öğrenmesi teknikleri kullanarak UCI veri seti üzerinde gerçekleştirdikleri sınıflama çalışmasında RO yöntemi ile %95.60 doğruluğa ulaşmışlardır. Bharti vd. (2021) ise aynı veri seti üzerinde derin öğrenme yöntemi kullanarak %94.2 sınıflama doğruluğu sağlamışladır. Kavitha vd. (2021), KA ve RO yöntemlerini kapsayan hibrit bir model kullanarak UCI veri seti üzerinde gerçekleştirdikleri sınıflamada en yüksek %88.7 doğruluk elde etmişlerdir. Rajendran ve Vincent (2021), KA, DVM, KNN gibi farklı algoritmaların, RO meta sınıflayıcısı ile bir araya getirilmesi yoluyla gerçekleştirdikleri sınıflamada %87.64 doğruluğa ulaşmışlardır. Asif vd. (2021), UCI veri seti üzerinde kolektif öğrenme yöntemi kullanarak gerçekleştirdikleri sınıflamada en yüksek %92 doğruluk elde etmişlerdir. Maini vd. (2021), Güney Hindistan'da bir hastaneden elde edilen, 14 özellik 501 adet örnekten oluşan veri seti üzerinde RO yöntemi ile gerçekleştirdikleri kalp hastalığı tahminleme çalışmasında %93.8 doğruluğa ulaşmışlardır.

Bu çalışmada literatürde yaygın olarak kullanılan UCI veri seti üzerinde, yeniden örnekleme olmaksızın ve 8 farklı yeniden örnekleme tekniği ile kullanılarak, 8 farklı makine öğrenmesi yöntemi ile bireyin temel özellikleri ve klinik ölçümlere dayanarak, kalp hastalığı durumunun teşhisine yönelik sınıflandırma çalışmaları gerçekleştirilmiştir. Sınıflandırma işlemlerinin sonuçları doğruluk, kesinlik, duyarlılık, F1 skoru ve AUC olmak üzere 5 farklı parametre ile raporlanmış ve elde edilen 360 adet ölçümün sonuçları yorumlanmıştır.

2. Materyal ve Metot (Material and Method)

Bu çalışma, bireylerin kalp hastalığı riskini kişisel özellikler ve klinik ölçümlere bağlı olarak değerlendirmeyi amaçlamakta ve bu doğrultuda UCI kalp hastalığı veri seti üzerinde farklı makine öğrenmesi algoritmaları ile kalp hastalığının erken teşhisine yönelik sınıflandırma çalışmalarını içermektedir. Sınıflandırma için Naive Bayes, Karar Ağaçları, Destek Vektör Makinesi, K En yakın Komşuluk, Lojistik Regresyon, Rastgele Orman, AdaBoost ve CatBoost olmak üzere 8 farklı yöntem kullanılmıştır. Sınıflama çalışmalarındaki ana amaçlardan biri tahmin başarısını yükseltmektir. Bu amaçla, çalışmada kullanılan sekiz farklı sınıflandırıcının varsayılan durumundaki başarıları yanında, her biri için 8 ayrı yeniden örnekleme yöntemi bağımsız olarak uygulanmış ve ölçümleri raporlanmıştır.

2.1. Veri seti (Dataset)

Kalp hastalığının erken teşhisine yönelik bir çözüm sunmayı hedefleyen bu çalışmada, literatürde yaygın olarak kullanılan UCI kalp hastalığı veri seti ile gerçekleştirilmiştir. Kamuya açık olarak paylaşılan veri seti University of California Irvine çevrimiçi veri deposundan elde edilmiştir ("Heart Disease Data Set, UCI Machine Learning Repository," 1988). Macar ve İsveç bilim insanları tarafından derlenen veri seti 13 farklı tanı parametresi ile birlikte bireyin kalp hastalığı durumuna ilişkin bir adet tanı olmak üzere 14 adet özellik içermekte ve 303 örnekten oluşmaktadır. Veri seti içerisinde yer alan özellikler ve açıklamaları Tablo 1'de özet olarak verilmiştir. Veri setinde yer alan sayısal özelliklerin tanımlayıcı istatistikleri ise Tablo 2'de verilmiştir.

Tablo 1. Veri setine ait özellikler (Features of the dataset)

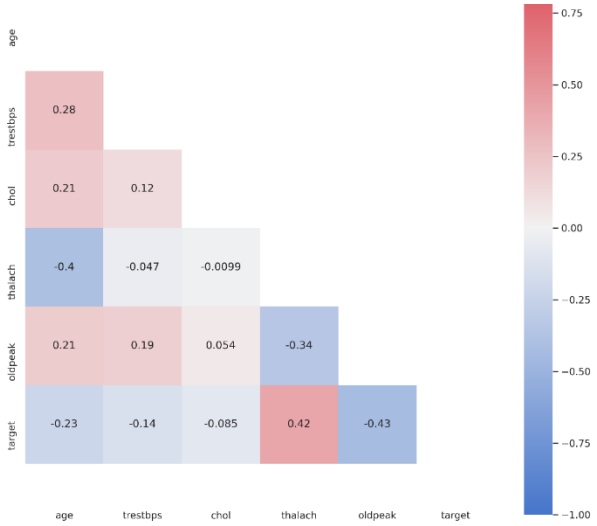
Özellik adı	Türü	Açıklama
age	Sayısal - sürekli	Hastanın yıl olarak yaşı
sex	Kategorik	Hastanın cinsiyeti (0 = kadın, 1 = erkek)
cp (chest pain)	Kategorik	Hastanın yaşadığı göğüs ağrısı türü (0 = tipik anjina, 1 = atipik anjina, 2 = anjinal olmayan ağrı, 3 = asemptomatik)
trestbps (resting blood pressure)	Sayısal - sürekli	Dinlenme durumundaki kan basıncı (mm Hg)
chol (cholestrol)	Sayısal - sürekli	Serumdaki kolesterol değeri (mg/dl)
fbs (fasting blood sugar)	Kategorik	Açlık kan şekeri fbs>120 ml/dl ise 1 (true) değilse 0 (false)
restecg	Kategorik	Dinlenme durumundaki elektrokardiyografik ölçüm (0 = normal, 1 = ST dalga anormalliği, 2 = olası sol ventrikül hipertrofisi)
thalach	Sayısal - sürekli	Ulaşılan maksimum kalp atış sayısı
exang	Kategorik	Egzersiz kaynaklı göğüs ağrısı (0 = yok, 1 = var)
oldpeak	Sayısal - sürekli	Egzersize bağlı dinlenmenin neden olduğu ST depresyonu (EKG'deki ST aralığı için)
slope	Kategorik	Egzersizin tepe noktasında ST segmentinin eğimi (0 = yukarı eğimli, 1 = düz, 2 = aşağı eğimli)
ca	Kategorik	Florosopi ile renklendirilen ana damarların sayısı (0-3)
thal	Kategorik	Talasemi durumu (1 = normal, 2 = sabit kusur, 3 = tersinir kusur)
target	Kategorik	Kalp hastalığı durumu (0 = sağlıklı, 1 = hasta)

Tablo 2. Veri setinin istatistiki karakteristiği (Statistical characteristic of the dataset)

Özellik	Minimum	Maksimum	Ortalama	Standart sapma
age	29	77	54.36	9.08
trestbps	94	200	131.62	17.53
chol	126	564	246.26	51.83
thalach	71	202	149.64	22.90
oldpeak	0	6.2	1.03	1.16

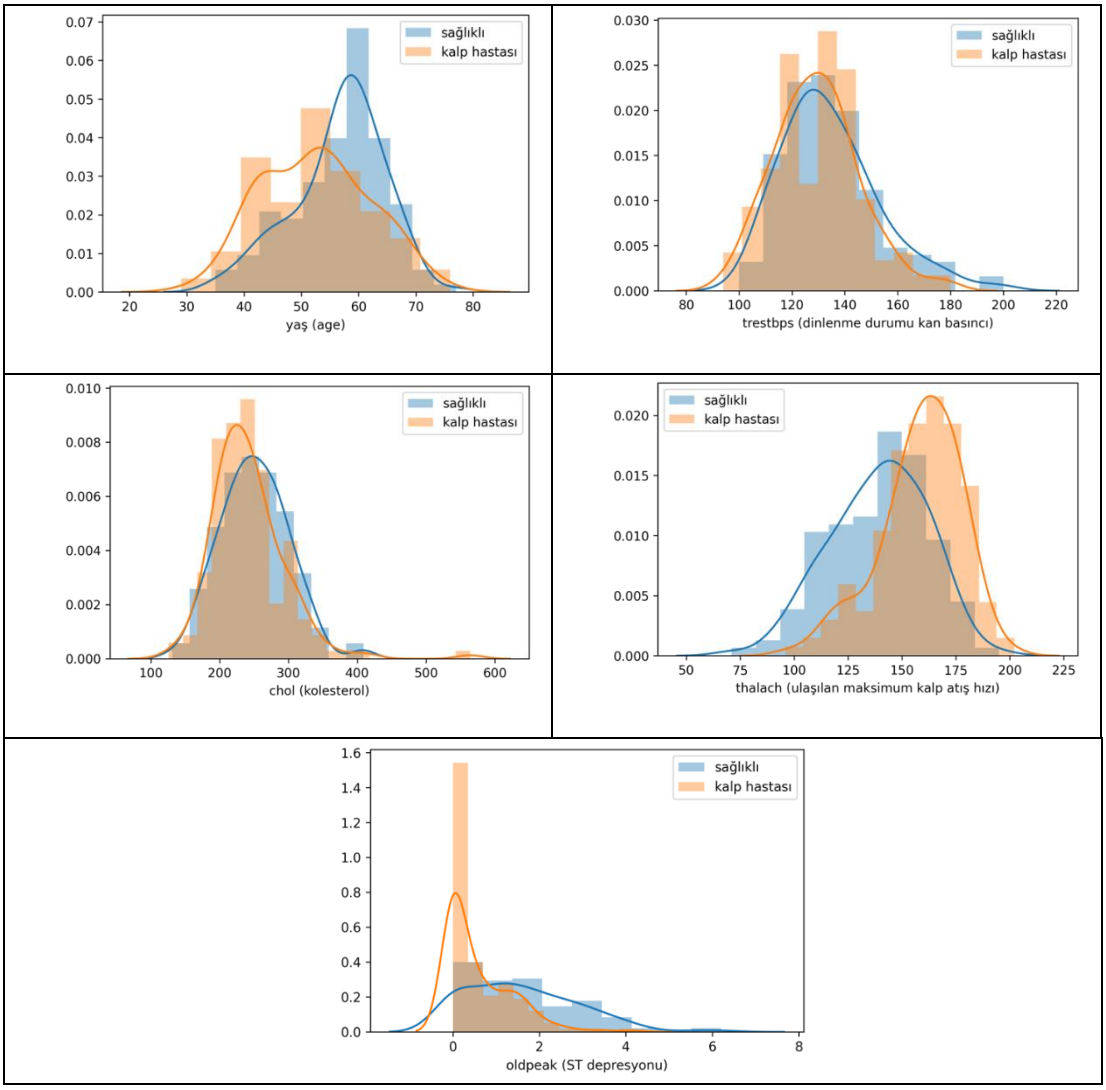
Veri setini oluşturan sayısal özellikler ve hedef değişken arasındaki korelasyonu gösteren matris Şekil 2'de verilmiştir.

Hedef özellik ile arasında 0.5 üzerinden korelasyon olan herhangi bir özellik bulunmadığından, tek başına hedefi tahminleme konusunda baskın değer yoktur. Hedef ile en yüksek korelasyona sahip özellik, ölçülen maksimum kalp atış hızı (thalac) değeridir. Matristeki her bir giriş özelliğinin, hedef değişkene göre dağılımını gösteren grafikler Tablo 3'te yer almaktadır.



Şekil 2. Korelasyon matrisi (Correlation matrix)

Tablo 3. Veri setindeki sayısal özelliklerinin hedef değişkene göre dağılımları (Distribution of numerical features in the dataset according to the target variable)



2.2. Kullanılan sınıflandırma yöntemleri (Classification methods used)

Bu çalışmada literatürde sıklıkla tercih edilen Naive Bayes, Karar Ağaçları (Decision Trees), Destek Vektör Makinesi (Support Vector Machine), K En yakın Komşuluk (K-Nearest Neighbor), Lojistik Regresyon (Logistic Regression), Rastgele Orman (Random Forest), AdaBoost (Adaptive Boosting) ve CatBoost (Categorical Boosting) sınıflandırma algoritmaları kullanılmıştır.

Naive Bayes (NB) algoritması, adını matematikçi Thomas Bayes'den alan bir sınıflandırma algoritmasıdır. NB sınıflandırıcı, olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile sınıfı bilinmeyen verilerin sınıfını tespit etmeyi amaçlar. Algoritma her durumun olasılığını hesaplar ve olasılık değeri en yüksek olana göre sınıflandırır.

Karar ağaçları (KA), sınıflama ve regresyon problemlerinden sıklıkla kullanılan ve karmaşık veri setleri ile çalışabilen, ağaç tabanlı bir algoritmadır. KA, özellikler ve hedefe bağlı olarak karar düğümleri ve yaprak düğümlerden oluşan bir ağaç yapısı formüle ederek bir sınıflandırma modeli oluşturur.

Destek vektör makinesi (DVM), Vapnik vd. (1997) tarafından literatüre kazandırılan, istatistiksel öğrenme teorisine dayalı, sınıflandırma ve regresyon analizi için kullanılan gözetimli bir makine öğrenmesi tekniğidir. DVM, her biri iki kategoriden birine ait olarak işaretlenmiş eğitim veri setinden öğrenerek, yeni örnekleri bu iki sınıftan birine olasılıklı olmayacak şekilde atayan bir model oluşturur. Veri örneklerinin yer aldığı düzlemde, sınıfları birbirinden ayırmak için, iki sınıfın üyelerinden en uzak mesafede olacak şekilde bir karar sınırının çizilmesi sağlanır. DVM'nin, aşırı uydurma problemi karşısındaki hassasiyetinin düşük olması ve yüksek doğruluk sağlaması kullanım yaygınlığını arttırmaktadır.

K En Yakın Komşuluk (KNN) algoritması, literatüre Fix ve Hodges (1952) tarafından kazandırılan, sınıflama ve regresyonda yaygın olarak kullanılan, parametrik olmayan bir yöntemidir. En temel makine öğrenmesi tekniklerinden biri olan KNN algoritmasındaki K değeri en yakın özellikleri seçmek amacıyla kullanılır (Bilgin, 2021). Sınıfı belirlenmek istenen nokta, K adet en yakın komşusuna bakılarak en yaygın olan sınıfa atanır. Sınıfı tahmin edilecek her bir örnek için, veri setindeki tüm örnekler arasında en yakın komşuluğun aranması nedeniyle veri setinin büyümesi durumunda işlem yükü artar. KNN algoritması mesafeye dayalı olduğundan, eğitim verilerinin normalizasyonu sınıflandırıcının doğruluğunu önemli ölçüde yükseltebilir.

Lojistik Regresyon (LR) bağımlı değişkenin süreksiz olduğu ikili sınıflama problemlerinde kullanılan istatistiksel bir modeldir. Bilgisayar bilimi, pek çok uygulamalı bilim ve gerçek dünya problemlerinde yaygın olarak kullanılmaktadır. Lojistik regresyon ikili bağımlı değişken ile bir dizi bağımsız değişken arasındaki ilişkiyi açıklamaya yönelik tahminleyici bir analizdir. Bu işi yerine getirmek için bir lojistik fonksiyon (logit fonksiyonu) kullanır. Bir olayın gerçekleşme olasılığı Eşitlik 1'deki formül ile ifade edilir. Olayın gerçekleşmeme olasılığı 1-p olmak üzere logit fonksiyonu ise Eşitlik 2'ye göre hesaplanır. Logistik regresyon logit dönüşümünü tahminlemek için bir formülün katsayılarını üretir.

$$p = \frac{e^{a+bx}}{1+e^{a+bx}} \qquad (1)$$

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \qquad (2)$$

Rastgele orman (RO), eğitim esnasında çok sayıda karar ağacı oluşturarak, her bir ağacın ürettiği sonuçların modu veya ortalamasını alarak çıktı sınıfı belirleyen bir kolektif öğrenme algoritmasıdır. Ho (1995) tarafından oluşturulan yönteme dayanan RO, daha sonra Breiman (2001) tarafından geliştirilerek literatüre kazandırılmıştır. RO, geleneksel karar ağaçlarında yaygın olan problemlerden biri olan aşırı uydurma (overfitting) sorununa hem veri seti, hem öznitelikleri çok sayıda parçaya bölüp birden çok ağaç üzerinde işleyerek çözüm getirir.

AdaBoost(AB), Freund ve Schapire (1996) tarafından formüle edilen adaptif bir meta algoritmadır. Bireysel öğrenicilerin ve kararlarının birleştirilmesi mantığına dayanan kolektif bir öğrenme yöntemidir. Eğitim sürecinde bireysel öğrenicilerin durumları ağırlıklandırılarak, yapılan güncellemelerle nihai modelin güçlü bir öğrenmeye yakınsaması sağlanır. AB, kaynak tüketiminin etkin ve tahmin hızının yüksek olması nedeni ile kolektif modeller içerisinde yaygın olarak tercih edilir.

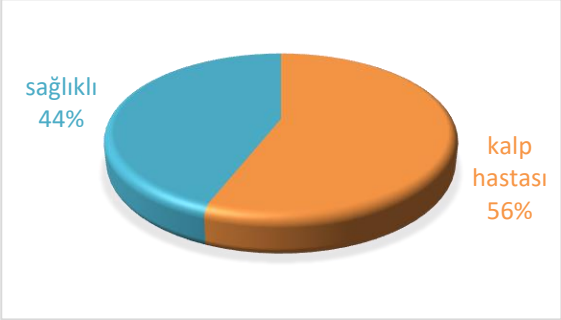
CatBoost(CB), Yandex mühendisleri Dorogush vd. (2018) tarafından formüle edilen açık kaynak kodlu bir algoritmadır. CB, klasik yönteme kıyasla kategorik özellikleri başarılı bir şekilde ele almak için permutasyona dayalı alternatif bir çözüm sunan yeni bir gradyan arttırma algoritmasıdır. Algoritmanın bir diğer avantajı, ağaç yapısını seçerken yaprak değerlerini hesaplamak için aşırı uydurmayı azaltmaya yardımcı olan yeni bir şema kullanmasıdır.

3. Kalp Hastalığı Teşhisine Yönelik Sınıflandırma (Classification for Diagnosis of Heart Disease)

Çalışmada kullanılan UCI kalp hastalığı veri seti 13 özellik ve 1 sınıf değeri ile tanımlanan, 303 örnekten oluşmaktadır. 13 adet girdi özelliğin 5 adedi sayısal, geriye kalan 8 adedi ise kategoriktir. Hedef sınıf değeri ise sağlıklı (0) ve kalp hastası (1) olmak üzere kategorik türdedir. Veri setinde içerisinde 165 adet hasta, 138 hasta olmayan birey kaydı yer almaktadır.

Sınıflandırma işlemi öncesinde veri seti üzerinde tahmin başarısını arttırmaya yönelik önişlemler uygulanmıştır. Kayıp veriler ilgili sütunun ortalama değeri ile doldurulmuştur. Ardından özelliklerin aykırı değerleri (outlier) tespit edilmiştir. Verilerin ilk ve

üçüncü çeyreği arasındaki farkın 1.5 katının alt sınırdan çıkarılması ve üst sınıra eklenmesi ile elde edilen aralık dışındaki değerler aykırı değer olarak kabul edilmiş ve bu değerleri içeren örnekler veri setinden temizlenmiştir. Daha sonra sayısal girdi özelliklerin tümü 0-1 aralığına ölçeklenmiştir. Son durumda veri setinde 159 adet hasta ve 125 adet sağlıklı birey verisi olmak üzere toplam 284 adet örnek yer almaktadır. Son durumda veri setindeki kalp hastası ve sağlıklı bireylerin dağılımlarını gösterir grafik Şekil 3'te yer almaktadır.



Şekil 3. Veri setinin sınıf dağılımı (Class distribution of the dataset)

Veri setindeki hasta ve sağlıklı bireylerin dağılımında kısmi bir dengesizlik söz konusudur. Dengesizlik durumunda, kullanılan sınıflama modeli baskın olan veriyi öğrenmeye yatkınlaşır. Veri setini daha dengeli hale getirerek bu soruna çözüm üretmek için yeniden örnekleme (resampling) tekniklerinden yararlanılır. Yeniden örnekleme, istatistiki yöntemlere dayanarak baskın olan sınıf verilerinin azaltılması ya da verinin istatistiki karakteristiğine uygun şekilde azınlık sınıf verilerinin çoğaltılması ile gerçekleştirilir.

Yeniden örnekleme, az örnekleme (undersampling) ve fazla örnekleme (oversampling) olmak üzere iki farklı yolla uygulanır. Baskın sınıfa ait olan verilerin azaltılması yoluyla dengeyi sağlamaya yönelik yaklaşım az örnekleme olarak adlandırılır. Fazla örnekleme yönteminde ise azınlık sınıfa ait veriler çeşitli tekniklerle arttırılır. Bu iş için, az olan veriyi rastgele olarak seçip kopyalama veya interpolasyon yöntemleri ile sentetik veri üretme yaklaşımları uygulanmaktadır. Bu çalışmada kullanılan veri setindeki dengesizlik büyük oranda olmasa da model başarısı üzerindeki etkisini ortaya koymak için varsayılan sınıflamaya ek olarak 8 farklı yeniden örnekleme tekniği kullanılmıştır. Çalışmada kullanılan fazla örnekleme ve az örnekleme teknikleri Tablo 4'te verilmiştir.

Tablo 4. Kullanılan yeniden örnekleme teknikleri (Resampling techniques used)

Fazla örnekleme	Az örnekleme
SMOTE	AllKNN
KMeansSMOTE	InstanceHardnessThreshold
ADASYN	NeighbourhoodCleaningRule
SVMSMOTE	OneSidedSelection

Tablo 4'te yer alan fazla örnekleme tekniklerinden SMOTE, Chawla vd. tarafından geliştirilen azınlık verilerden sentetik üretim yapan bir yaklaşımdır (Chawla vd., 2002). KMeansSMOTE, SMOTE tekniği üzerine, gürültü oluşumunu azaltmak için K-means kümeleme uygulayan bir tekniktir (Last vd., 2017). Benzer şekilde SVMSMOTE tekniği, SMOTE ile Destek Vektör Makinesini (SVM) birleştirir (Nguyen vd., 2011). ADASYN tekniği SMOTE tekniğine benzeyen ancak verideki sınıf dağılımlarına bağlı olarak değişen sayıda örnek üreten adaptif, sentetik bir fazla örnekleme tekniğidir (He vd., 2008).

Az örnekleme tekniklerinden AllKNN, en yakın komşuluk yöntemine göre karar sınırına yakın örnekleri kaldırmak suretiyle baskın sınıfı azaltma işlemini, komşuları çeşitlendirecek şekilde gerçekleştirir (Tomek, 1976a). InstanceHardnessThreshold veri setindeki örneklerin sınıflandırılma zorluğunun eşik değerlerine bağlı olarak gerçekleştirilen bir az örnekleme tekniğidir (Smith vd., 2014). NeighbourhoodCleaningRule, KNN ve genişletilmiş en yakın komşuluk (ENN) yöntemlerini kullanarak veri setindeki gürültüleri ortadan kaldırır (Laurikkala, 2001). Bir özellik uzayında, farklı sınıflara ait örneklerden birbirine en yakın Öklid mesafesine sahip olanlar, bu prosedürü ortaya koyan Ivan Tomek'e atfen Tomek's Link olarak adlandırılır. TomekLinks az örnekleme tekniği, veri seti içerisindeki Tomek's linkleri kaldırma üzerine kuruludur (Tomek, 1976b). OneSidedSelection tekniği, TomekLinks ve Yoğun En Yakın Komşuluk (CNN) kurallarını birleştiren bir az örnekleme tekniğidir (Kubat vd., 1997).

Veri seti üzerinde farklı sınıflandırıcıların her biri için yukarıda bahsi geçen yeniden örnekleme yöntemleri ayrı ayrı uygulanarak performans ölçümleri gerçekleştirilmiştir.

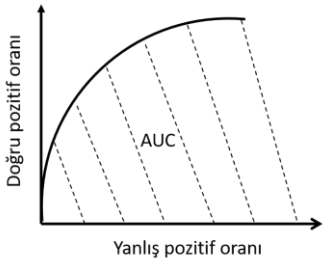
3.1. Performans Metrikleri (Performance Metrics)

Bir sınıflandırıcının başarısını ölçmek için karmaşıklık matrislerinden değerlerden yararlanılır. Karmaşıklık matrisleri sınıflandırma işlemi sonunda hangi sınıfların birbirinden ne düzeyde ayırt edilebildiği ile ilgili detaylı bilgiler sağlar. Gerçekte kalp hastası olan bir kayıt, sınıflandırıcı tarafından da hasta olarak doğru sınıflanmışsa Doğru Pozitif (DP), hastalığı yok şeklinde yanlış sınıflandırılmışsa Yanlış Negatif (YN) olarak adlandırılır. Benzer şekilde gerçekte kalp hastası olmayan kayıt, hastalığı yok şeklinde doğru sınıflandırılırsa Doğru Negatif (DN), hastalığı var şeklinde yanlış sınıflanırsa ise Yanlış Pozitif (YP) olarak adlandırılır. Karmaşıklık matrisinden elde edilen değerler kullanılarak farklı performans metrikleri üretilir. Bu çalışmada kullanılan metrikler Tablo 5'te verilmiştir.

Doğruluk metriği modelin genel başarısını ifade etmek için kullanılır. Doğruluk değeri, doğru sınıflanmış örnek sayısının, tüm örnek sayısına bölümü ile elde edilir. Kesinlik, pozitif olarak tahmin edilen değerlerin gerçekten ne kadarının pozitif olduğunu

gösteren metriktir. Aynı zamanda kesinlik metriği sınıflandırıcının yanlış pozitifleri eleme kabiliyetinin de göstergesidir. Duyarlılık, pozitif olarak tahmin edilmesi gereken değerlerin ne kadarının pozitif olarak tahmin edildiğini gösteren bir metriktir. Duyarlılık, sınıflandırıcının doğru pozitifleri tahmin etmedeki kabiliyetinin ölçütüdür. Kesinlik ve duyarlılık arasındaki dengeyi ifade etmek için F1 skoru kullanılır. F1 skoru, hesaplanan kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. Harmonik ortalama kullanılmasının nedeni uç durumların da göz ardı edilmemesinin gerekliliğidir. F1 skoru [0, 1] aralığında değer alır. Alıcı işlem karakteristiği (ROC) eğrileri, farklı sınıflar için bir olasılık eğrisidir. X ekseninde yanlış pozitif oranı, Y ekseninde ise doğru pozitif oranın yer aldığı bu eğri, kullanılan sınıflandırıcının tahminde ne kadar iyi olduğunu açıklar. Eğrinin altında kalan alan (AUC) [0,1] aralığında değer alır ve model performansının bir özeti kabul edilir. AUC değerinin 1'e yaklaşması veri setindeki sınıfların daha başarılı şekilde ayırt edilebildiğini gösterir.

Tablo 5. Performans metrikleri (Performance metrics)

Metrik	Matematiksel ifadesi
Doğruluk	(DP + DN) / (DP + YP + YN + DN)
Kesinlik	DP / (DP + YP)
Duyarlılık	DP / (DP + YN)
F1 Skoru	2 * kesinlik * duyarlılık / (kesinlik + duyarlılık)
ROC eğrisi ve AUC	

4. Bulgular ve Tartışma (Findings and Discussion)

Çalışmada kullanılan veri seti üzerinde NB, KA, DVM, KNN, LR, RO, AB ve CB sınıflandırıcıları ve her bir sınıflandırıcı için Tablo 4'te yer alan yeniden örnekleme teknikleri uygulanarak performans ölçümleri gerçekleştirilmiştir. Sınıflandırıcının hiperparametreleri ızgara arama yöntemi ile belirlenmiştir. Her bir yöntem için kullanılan parametreler Tablo 6'da verilmiştir.

Tablo 6. Sınıflandırıcıların parametreleri (Parameters of the classifiers)

Sınıflandırıcı	Kullanılan parametre ve değeri
NB	var_smoothing=1e-09
KA	min_samples_split=2, min_samples_leaf=1,
DVM	kernel='linear', C=3
KNN	n_neighbors=3
LR	C=0.1, penalty="l2", max_iter=250
RO	n_estimators =200
AB	n_estimators=100, learning_rate=0.01
CB	verbose=0, n_estimators=100, learning_rate=0.001

Yeniden örnekleme yapılarak ve yapılmadan gerçekleştirilen her bir sınıflama işleminin sonucu, Tablo 5'te verilen metrikler ile raporlanmıştır. Yeniden örnekleme oranı, azınlık ve baskın sınıf örnek sayılarını birbirine yaklaştıracak şekilde fazla örneklemede %30, az örneklemede %20 oranında uygulanmıştır. Fazla örnekleme sonucu veri seti örneklem büyüklüğü 321, az örnekleme sonucu 252'dir. Sınıflama işlemlerinde 10 kat çapraz doğrulama uygulanmıştır. Her bir aşamada %90 eğitim, %10 test seti olmak üzere, çapraz doğrulama işlemlerinden elde edilen sonuçların ortalamaları raporlanmıştır. Örnekleme tekniklerinin sınıflandırıcılar üzerindeki etkisini gösteren ölçümler NB için Tablo 7'de, KA için Tablo 8'de, DVM için Tablo 9'da, KNN için Tablo 10'da, LR için Tablo 11'de, RO için Tablo 12'de, AB için Tablo 13'te ve CB için Tablo 14'te verilmiştir.

Tablo 7. Naive Bayes (NB) sınıflandırıcıya ait ölçümler (Measurements of the Naive Bayes classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.912281	0.935484	0.906250	0.920635	0.913125
Fazla örnekleme	SMOTE	0.904762	0.931034	0.870968	0.900000	0.904234
	KMeansSMOTE	0.890625	0.875000	0.903226	0.888889	0.891007
	ADASYN	0.901639	0.933333	0.875000	0.903226	0.903017
	SVMSMOTE	0.904762	0.931034	0.870968	0.900000	0.904234
Az örnekleme	AllKNN	0.934783	0.950000	0.904762	0.926829	0.932381
	InstanceHardnessThreshold	**0.960000**	**0.958333**	0.920000	**0.938776**	**0.940000**
	NeighbourhoodCleaningRule	0.934783	0.950000	0.904762	0.926829	0.932381
	OneSidedSelection	0.927273	0.933333	**0.933333**	0.933333	0.926667

Tablo 8. Karar Ağacı (KA) sınıflandırıcıya ait ölçümler (Measurements of the Decision Tree classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.842105	0.870968	0.843750	0.857143	0.841875
Fazla örnekleme	SMOTE	0.936508	0.965517	0.903226	**0.933333**	**0.935988**
	KMeansSMOTE	**0.937500**	**1.000**	0.870968	0.931034	0.935484

	ADASYN	0.868852	0.900000	0.843750	0.870968	0.870151
	SVMSMOTE	0.888889	0.928571	0.838710	0.881356	0.888105
Az örnekleme	AllKNN	0.891304	0.900000	0.857143	0.878049	0.888571
	InstanceHardnessThreshold	0.92	0.956522	0.88	0.916667	0.92
	NeighbourhoodCleaningRule	0.888889	0.894737	0.850000	0.871795	0.885000
	OneSidedSelection	0.925926	0.931034	**0.931034**	0.931034	0.925517

Tablo 9. Destek Vektör Makinesi (DVM) sınıflandırıcıya ile ait ölçümler (Measurements of the Support Vector Machine classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.929825	0.937500	0.937500	0.937500	0.928750
Fazla örnekleme	SMOTE	0.936508	**0.965517**	0.903226	0.933333	0.935988
	KMeansSMOTE	0.920635	0.933333	0.903226	0.918033	0.920363
	ADASYN	0.885246	0.878788	0.906250	0.892308	0.884159
	SVMSMOTE	0.906250	0.882353	0.937500	0.909091	0.906250
Az örnekleme	AllKNN	0.913043	0.904762	0.904762	0.904762	0.912381
	InstanceHardnessThreshold	**0.961460**	0.925926	**1.000**	**0.961538**	**0.961460**
	NeighbourhoodCleaningRule	0.934783	0.950000	0.904762	0.926829	0.932381
	OneSidedSelection	0.909091	0.931034	0.900000	0.915254	0.910000

Tablo 10. K En yakın Komşuluk (KNN) sınıflandırıcı ile elde edilen ölçümler (Measurements of the K Nearest Neighbor classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.912281	0.935484	0.906250	0.920635	0.913125
Fazla örnekleme	SMOTE	0.888889	0.961538	0.806452	0.877193	0.887601
	KMeansSMOTE	0.904762	**1.000**	0.806452	0.892857	0.903226
	ADASYN	0.900000	0.906250	0.906250	0.906250	0.899554
	SVMSMOTE	0.890625	0.878788	0.906250	0.892308	0.890625
Az örnekleme	AllKNN	0.891304	0.833333	**0.952381**	0.888889	0.896190
	InstanceHardnessThreshold	**0.940000**	0.958333	0.920000	**0.938776**	**0.940000**
	NeighbourhoodCleaningRule	0.913043	0.947368	0.857143	0.900000	0.908571
	OneSidedSelection	0.909091	0.962963	0.866667	0.912281	0.913333

Tablo 11. Lojistik Regresyon (LR) sınıflandırıcı ile elde edilen ölçümler (Measurements of the Logistic Regression classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.894737	0.861111	0.968750	0.911765	0.884375
Fazla örnekleme	SMOTE	0.890625	0.878788	0.906250	0.892308	0.890625
	KMeansSMOTE	0.890625	0.878788	0.906250	0.892308	0.890625
	ADASYN	0.885246	0.878788	0.906250	0.892308	0.884159
	SVMSMOTE	0.888889	**0.961538**	0.806452	0.877193	0.887601
Az örnekleme	AllKNN	0.891304	0.900000	0.857143	0.878049	0.888571
	InstanceHardnessThreshold	**0.961688**	0.925926	**1.000**	**0.961538**	**0.961688**
	NeighbourhoodCleaningRule	0.911111	0.863636	0.950000	0.904762	0.915000
	OneSidedSelection	0.942308	0.900000	**1.000**	0.947368	0.940000

Tablo 12. Rastgele Orman (RO) sınıflandırıcı ile elde edilen ölçümler (Measurements of the Random Forest classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.894737	0.933333	0.875000	0.852941	0.813125
Fazla örnekleme	SMOTE	0.920635	**1.000**	0.83871	0.912281	0.919355
	KMeansSMOTE	0.921875	**1.000**	0.83871	0.912281	0.919355
	ADASYN	0.885246	0.931034	0.84375	0.885246	0.887392
	SVMSMOTE	0.904762	**1.000**	0.806452	0.892857	0.903226
Az örnekleme	AllKNN	0.956522	**1.000**	0.904762	0.950000	0.952381
	InstanceHardnessThreshold	**0.984600**	0.961538	**1.000**	**0.980392**	**0.984600**
	NeighbourhoodCleaningRule	0.933333	0.904762	0.950000	0.926829	0.935000
	OneSidedSelection	0.927273	0.933333	0.933333	0.933333	0.926667

Tablo 13. AdaBoost (AB) sınıflandırıcı ile elde edilen ölçümler (Measurements of the AdaBoost classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.929825	**0.937500**	0.937500	0.937500	0.928750
Fazla örnekleme	SMOTE	0.921875	0.909091	0.93750	0.923077	0.921875
	KMeansSMOTE	0.875000	0.900000	0.84375	0.870968	0.87500
	ADASYN	0.918033	0.909091	0.93750	0.923077	0.917026
	SVMSMOTE	0.921875	0.909091	0.93750	0.923077	0.921875
Az örnekleme	AllKNN	0.891304	0.833333	0.952381	0.888889	0.896190
	InstanceHardnessThreshold	0.960784	0.928571	**1.000**	0.962963	**0.960000**

	NeighbourhoodCleaningRule	0.844444	0.809524	0.850000	0.829268	0.845000
	OneSidedSelection	**0.963636**	**0.937500**	**1.000**	**0.967742**	**0.960000**

Tablo 14. CatBoost (CB) sınıflandırıcı ile elde edilen ölçümler (Measurements of the CatBoost classifier)

Örnekleme	Kullanılan teknik	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	AUC
Örneklemesiz	-	0.929825	0.937500	0.937500	0.937500	0.928750
Fazla örnekleme	SMOTE	0.921875	0.909091	0.937500	0.923077	0.921875
	KMeansSMOTE	0.920635	0.933333	0.903226	0.918033	0.920363
	ADASYN	0.934426	0.937500	0.937500	0.937500	0.934267
	SVMSMOTE	0.920635	**0.964286**	0.870968	0.915254	0.919859
Az örnekleme	AllKNN	0.913043	0.869565	0.952381	0.909091	0.916190
	InstanceHardnessThreshold	**0.960784**	0.928571	**1.000**	**0.962963**	**0.960000**
	NeighbourhoodCleaningRule	0.913043	0.904762	0.904762	0.904762	0.912381
	OneSidedSelection	0.945455	0.935484	0.966667	0.950820	0.943333

Elde edilen ölçümler değerlendirildiğinde, yeniden örneklemenin bazı durumlarda belirli sınıflandırıcıların başarısını örnekleme olmayan duruma göre düşürdüğü görülse de genel anlamda yeniden örnekleme tekniklerinin sınıflama başarısını arttırdığı gözlemlenmiştir. Performans metrikleri açısından yeniden örnekleme teknikleri kullanılarak %10'a varan performans artışı sağlandığı görülmüştür. Yeniden örnekleme ile elde edilen bu artış literatürdeki benzer çalışmaları destekler nitelikte olup, bu çalışmadaki artışın emsallerinden yüksek olduğu sonucuna varılmıştır. Az ve fazla örnekleme tekniklerinin başarı durumları değişkenlik göstermekle birlikte, bu çalışmada az örneklemenin daha başarılı sonuç verdiği ve InstanceHardnessThreshold tekniğinin genel anlamda daha yüksek başarı artışı sağladığı görülmüştür. Kullanılan tüm sınıflandırıcılar ve örnekleme yöntemlerinin en iyi sonuçları özet olarak Tablo 15'te verilmiştir.

Tablo 15. Sınıflandırıcıların en iyi sonuçları (Best results of the classifiers)

Sınıflandırıcı	Örnekleme Yöntemi	En İyi Doğruluk	En İyi Kesinlik	En İyi Duyarlılık	En İyi F1 Skoru	En İyi AUC
Naive Bayes	Örnekleme olmadan	0.912281	0.935484	0.906250	0.920635	0.913125
	Fazla Örnekleme	0.904762	0.933333	0.903226	0.903226	0.904234
	Az Örnekleme	**0.960000**	**0.958333**	**0.933333**	**0.938776**	**0.940000**
Karar Ağaçları	Örnekleme olmadan	0.842105	0.870968	0.843750	0.857143	0.841875
	Fazla Örnekleme	**0.937500**	**1.000**	0.903226	**0.933333**	**0.935988**
	Az Örnekleme	0.925926	0.956522	**0.931034**	0.931034	0.925517
Destek Vektör Makinesi	Örnekleme olmadan	0.929825	0.937500	0.937500	0.937500	0.928750
	Fazla Örnekleme	0.936508	**0.965517**	0.937500	0.933333	0.935988
	Az Örnekleme	**0.961460**	0.950000	**1.000**	**0.961538**	**0.961460**
K En yakın Komşuluk	Örnekleme olmadan	0.912281	0.935484	0.906250	0.920635	0.913125
	Fazla Örnekleme	0.904762	**1.000**	0.906250	0.906250	0.903226
	Az Örnekleme	**0.940000**	0.962963	**0.952381**	**0.938776**	**0.940000**
Lojistik Regresyon	Örnekleme olmadan	0.894737	0.861111	0.968750	0.911765	0.884375
	Fazla Örnekleme	0.890625	**0.961538**	0.906250	0.892308	0.890625
	Az Örnekleme	**0.961688**	0.925926	**1.000**	**0.961538**	**0.961688**
Rastgele Orman	Örnekleme olmadan	0.894737	0.933333	0.875000	0.852941	0.813125
	Fazla Örnekleme	0.921875	**1.000**	0.83871	0.912281	0.919355
	Az Örnekleme	**0.984600**	**1.000**	**1.000**	**0.980392**	**0.984600**
AdaBoost	Örnekleme olmadan	0.929825	**0.937500**	0.937500	0.937500	0.928750
	Fazla Örnekleme	0.921875	0.909091	0.93750	0.923077	0.921875
	Az Örnekleme	**0.963636**	**0.937500**	**1.000**	**0.967742**	**0.960000**
CatBoost	Örnekleme olmadan	0.929825	0.937500	0.937500	0.937500	0.928750
	Fazla Örnekleme	0.934426	**0.964286**	0.937500	0.937500	0.934267
	Az Örnekleme	**0.960784**	0.935484	**1.000**	**0.962963**	**0.960000**

Tablo 15'teki sonuçlar incelendiğinde, tüm parametreler açısından en yüksek ölçümlerin Rastgele Orman sınıflandırıcı ile elde edildiği görülmektedir. Bu sınıflandırıcı ile ulaşılan en yüksek doğruluk değeri ise InstanceHardnessThreshold az örnekleme tekniğinin kullanıldığı durumda elde edilmiştir. Rastgele Orman ile tüm sınıflamalarda elde edilen en iyi sonuçlar doğruluk için %98.46, kesinlik için %100, duyarlılık için %100, F1 Skoru için %98.03 ve AUC için %98.46 olmuştur. Rastgele Orman modelinden sonra, en yüksek başarı AdaBoost ile elde edilmiş ve bu sınıflandırıcı ulaşılan en yüksek %96.36 doğruluk değeri yine bir az örnekleme tekniği olan OneSidedSelection ile sağlanmıştır. Yeniden örnekleme ile birlikte yapılan sınıflandırmalar içerisinde elde edilen en iyi doğruluk değerlerinin en düşüğü %93.75 ile Karar Ağaçları ile elde edilmiştir. Tüm sınıflandırıcılar için en iyi kesinlik değerleri %93.75 ile %100 arasında değişim göstermektedir. Elde edilen en iyi duyarlılık değerlerinin en düşüğü %93.10 olup geneli ise %100 seviyesindedir. Bu durum tüm sınıflandırıcıların doğru pozitifleri tespit ve yanlışları eleme kabiliyetlerinin yüksek olduğunu göstermektedir.

Bu çalışmada elde edilen en iyi sonuçlar ile literatürde son dönemdeki benzer çalışmaların sonuçları karşılaştırılmalı olarak Tablo 16'da verilmiştir.

Tablo 16. Elde edilen bulguların literatürdeki çalışmalar ile karşılaştırılması (Comparison of the obtained findings with the studies in the literatüre)

Referans	Veri seti	Kullanılan Yöntem	En iyi doğruluk (%)
Liu vd., 2017	Statlog	C4.5	92.59
David ve Belcy, 2018	UCI	RO	81
Haq vd., 2018	UCI	DVM	86
Malav ve Kadam, 2018	UCI	K-Means & YSA	93.52
Poornima ve Gladis, 2018	UCI	YSA	94
Ali vd., 2019b	UCI	YSA	93.33
Mohan vd., 2019	UCI	Hibrit RO	88.7
Ali vd., 2019a	UCI	Yığınlanmış DVM	91.11
Mienye vd., 2020	Framingham	YSA	90
Terrada vd., 2020	Z-Alizadeh Sani	YSA	94
Tama vd., 2020	UCI	Kolektif öğrenme	85.71
Akalın vd., 2020	UCI	RO	82
Elhoseny vd., 2021	UCI	AdaBoost	82.5
Rani vd., 2021	UCI	SMOTE & NB	85.07
Katarya ve Meena, 2021	UCI	RO	95.60
Bharti vd., 2021	UCI	Derin öğrenme	94.2
Kavitha vd., 2021	UCI	Hibrit model	88.7
Asif vd., 2021	UCI	Kolektif öğrenme	92
Maini vd., 2021	14 özellik, 501 örnek	RO	93.8
		Literatür ortalaması	*89.67*
Bu çalışma	UCI	Önerilen NB	96
		Önerilen KA	93.75
		Önerilen DVM	96.15
		Önerilen KNN	94
		Önerilen LR	96.17
		Önerilen RO	**98.46**
		Önerilen AB	96.36
		Önerilen CB	96.08
		Önerilen yöntemlerin ortalaması	*95.87*

Bu çalışmada, uygulanan yeniden örnekleme yöntemleri ile birlikte kullanılan sınıflandırıcıların literatürdeki benzer çalışmaların genelinden daha yüksek başarı gösterdiği görülmektedir. Çalışmada kullanılan Rastgele Orman modelinin az örnekleme tekniği ile birlikte, %98.46 doğrulukla Tablo 16'da yer alan son beş yıldaki benzer çalışmaların tümünden daha yüksek doğruluk sağladığı görülmektedir. Tabloda yer alan benzer 19 çalışmada elde edilen en yüksek doğruluk değeri %95.60, en düşük doğruluk ise %81'dir. Literatürdeki çalışmalarda ulaşılan ortalama doğruluk değeri ise %89.67'dir. Bu çalışmada elde edilen en düşük doğruluk değeri %93.75, önerilen sekiz modelin ortalama doğruluğu ise %95.87'dir. Bu çalışmada uygulanan modeller ile elde edilen doğruluk değerlerinin geneli, literatürdeki diğer çalışmalarda elde edilen doğruluk değerlerinden daha yüksektir.

Literatürdeki çalışmalarda en yüksek doğruluğun Rastgele Orman modeli ve benzer kolektif öğrenme yöntemleri ile elde edildiği görülmektedir. Bu çalışmada da en yüksek doğruluk değeri Rastgele Orman ve bunu takiben AdaBoost ve CatBoost kolektif öğrenme yöntemleri ile birlikte az örnekleme yaklaşımı kullanılarak sağlanmıştır. Az örnekleme tekniklerinden InstanceHardnessThreshold yönteminin diğer yeniden örnekleme yöntemlerine nazaran daha başarılı olduğu görülmektedir. Kolektif öğrenme algoritmaları aşırı uydurma (overfitting) problemlerine karşı daha az

hassastır. Yeniden örnekleme yoluyla veri setinin dengelenmesi de belirli bir sınıfı aşırı öğrenmeye olan yatkınlığı azaltmaktadır. Bu anlamda yeniden örneklemenin kolektif öğrenme ile birleşiminin aşırı uydurma probleminin çözümünde daha güçlü bir etki oluşturduğu ve bu yolla modelin genellenebilir başarısını arttırdığı elde edilen sonuçlardan yola çıkılarak söylenebilir.

Bunun yanında çalışmada kullanılan veri seti dengesiz yapıda olmasına rağmen sınıf dağılımları arasındaki farklılık yüksek oranda değildir. Normal seviyede dengesiz kabul edilebilecek bir veri seti ile gerçekleştirilen bu çalışmada elde edilen bulgular, aşırı dengesiz veri setlerinde farklılık gösterebilir. Çalışmada kullanılan veri setinin dengesizliğinin yüksek olmaması çalışmanın bir sınırlılığı olarak belirtilebilir.

5. Sonuçlar (Conclusions)

Bu çalışmada dünya genelinde yaygın görülen sağlık sorunlarından biri olan ve başlıca ölüm nedenleri içerisinde yer alan kalp hastalığının erken teşhisine yönelik farklı makine öğrenmesi algoritmaları kullanılarak alternatif bir çözüm önerilmiştir. Kalp hastalığı teşhisinin komplike olması ve hastaların pek çoğunun kriz anına kadar belirtilerin farkına varamaması, durumu uzmanlar için de zorlaştırabilmektedir. Bu anlamda hastaya ait temel bilgiler ve klinik ölçümlerden yola çıkarak 13 farklı tanı parametresi ile kalp hastalığı durumunu tahminlemeye yönelik 303 örnekten oluşan UCI veri seti üzerinde çalışılmıştır. 8 farklı makine öğrenmesi algoritması ve veri setini dengeleyerek tahmin doğruluğunu arttırmaya yönelik 8 farklı yeniden örnekleme yöntemi kullanılarak kalp hastalığı tahminlemesi için ölçümler gerçekleştirilmiştir. 72 ayrı sınıflama ve doğruluk, kesinlik, duyarlılık, F1 skoru ve AUC olmak üzere 5 ayrı parametre ile karakterize edilen 360 ölçümün sonuçlarına göre en yüksek doğruluk %98.46 olarak InstanceHardnessThreshold az örnekleme tekniği ile birlikte Rastgele Orman sınıflandırıcısının kullanıldığı durumda elde edilmiştir. Çalışmada önerilen yöntemlerin tümü literatürdeki, çoğunluğu bu çalışma ile aynı veri setini kullanan, son beş yıldaki benzer çalışmaların genelinden daha yüksek başarı göstermiştir. Bu çalışmada ulaşılan en yüksek doğruluk değeri ise sözü geçen çalışmaların tümünden daha yüksektir. Veriden öğrenen makine öğrenmesi algoritmalarının başarılarını belirleyen temel etken üzerinde çalışılan veri setidir. Veri setindeki sınıf dengesizlikleri modelin başarısını, sonuçların genellenebilirliğini olumsuz etkiler. Eldeki veri setinin uygun örnekleme teknikleri kullanılarak dengeli hale getirilmesi kullanılan modelin daha başarılı sınıflamalar yapmasına olanak tanımaktadır. Bu çalışmada elde edilen sonuçlar yeninden örnekleme tekniklerinin performansları hakkında bir görüş oluşturmuştur. Ayrıca farklı makine öğrenmesi modellerinin başarılarını karşılaştırma olanağı tanımıştır. Bu çalışma ile kullanılan kısmi dengesiz veri seti üzerinde yapılan sınıflamalarda yeniden örneklemenin %10'a varan başarı artışı sağladığı görülmüştür.

Kaynaklar (References)

Akalın, B., Veranyurt, Ü., Veranyurt, O., 2020. Classification of individuals at risk of heart disease using machine learning. Cumhuriyet Medical Journal 42, 283–289.

Ali, L., Niamat, A., Khan, J.A., Golilarz, N.A., Xingzhong, X., Noor, A., Nour, R., Bukhari, S.A.C., 2019a. An optimized stacked support vector machines based expert system for the effective prediction of heart failure. IEEE Access 7, 54007–54014.

Ali, L., Rahman, A., Khan, A., Zhou, M., Javeed, A., Khan, J.A., 2019b. An Automated Diagnostic System for Heart Disease Prediction Based on x^2 Statistical Model and Optimally Configured Deep Neural Network. IEEE Access 7, 34938–34945. https://doi.org/10.1109/ACCESS.2019.2904800

Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., Yarifard, A.A., 2017. Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. Computer Methods and Programs in Biomedicine 141, 19–26. https://doi.org/10.1016/j.cmpb.2017.01.004

Asif, S., Wenhui, Y., Tao, Y., Jinhai, S., Jin, H., 2021. An Ensemble Machine Learning Method for the Prediction of Heart Disease, in: 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, pp. 98–103.

Bharti, R., Khamparia, A., Shabaz, M., Dhiman, G., Pande, S., Singh, P., 2021. Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning. Computational Intelligence and Neuroscience 2021, 8387680. https://doi.org/10.1155/2021/8387680

Bilgin, G., 2021. Makine öğrenmesi algoritmaları kullanarak erken dönemde diyabet hastalığı riskinin araştırılması. Journal of Intelligent Systems: Theory and Applications, 4(1), 55-64.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Das, R., Turkoglu, I., Sengur, A., 2009. Effective diagnosis of heart disease through neural networks ensembles. Expert Systems with Applications 36, 7675–7680. https://doi.org/10.1016/j.eswa.2008.09.013

David, H., Belcy, S.A., 2018. HEART DISEASE PREDICTION USING DATA MINING TECHNIQUES. ICTACT Journal on Soft Computing 9.

Dorogush, A.V., Ershov, V., Gulin, A., 2018. CatBoost: gradient boosting with categorical features support. CoRR abs/1810.11363.

Elhoseny, M., Mohammed, M.A., Mostafa, S.A., Abdulkareem, K.H., Maashi, Mashael S., Garcia-Zapirain, B., Mutlag, A.A., Maashi, Marwah Suliman, 2021. A new multi-agent feature wrapper machine learning approach for heart disease diagnosis. Comput. Mater. Contin 67, 51–71.

Fix, E., Hodges Jr, J.L., 1952. Discriminatory analysis-nonparametric discrimination: Small sample performance. California Univ Berkeley.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm, in: Icml. Citeseer, pp. 148–156.

Haq, A.U., Li, J.P., Memon, M.H., Nazir, S., Sun, R., 2018. A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. Mobile Information Systems 2018.

He, H., Bai, Y., Garcia, E., Li, S., 2008. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, in: Proceedings of the International Joint Conference on Neural Networks. pp. 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Heart Disease Data Set, UCI Machine Learning Repository [WWW Document], 1988. URL https://archive.ics.uci.edu/ml/datasets/Heart+Disease (erişim tarihi: 4.8.21).

Ho, T.K., 1995. Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition. IEEE, pp. 278–282.

Jabbar, M.A., Deekshatulu, B.L., Chandra, P., 2016. Prediction of Heart Disease Using Random Forest and Feature Subset Selection, in: Snášel, V., Abraham, A., Krömer, P., Pant, M., Muda, A.K. (Eds.), Innovations in Bio-Inspired Computing and Applications. Springer International Publishing, Cham, pp. 187–196.

Kartal, Mutlu, Köksal, Özlem, 2020. Akut Koroner Sendromlarda EKG.

Katarya, R., Meena, S.K., 2021. Machine learning techniques for heart disease prediction: a comparative study and analysis. Health and Technology 11, 87–97.

Kavitha, M., Gnaneswar, G., Dinesh, R., Sai, Y.R., Suraj, R.S., 2021. Heart Disease Prediction using Hybrid machine Learning Model, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT). pp. 1329–1333. https://doi.org/10.1109/ICICT50816.2021.9358597

Kim, J.K., Kang, S., 2017. Neural Network-Based Coronary Heart Disease Risk Prediction Using Feature Correlation Analysis. Journal of Healthcare Engineering 2017, 2780501. https://doi.org/10.1155/2017/2780501

Kubat, M., Matwin, S., others, 1997. Addressing the curse of imbalanced training sets: one-sided selection, in: Icml. Citeseer, pp. 179–186.

Last, F., Douzas, G., Bacao, F., 2017. Oversampling for imbalanced learning based on k-means and smote. arXiv preprint arXiv:1711.00837.

Latha, C.B.C., Jeeva, S.C., 2019. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Informatics in Medicine Unlocked 16, 100203. https://doi.org/10.1016/j.imu.2019.100203

Laurikkala, J., 2001. Improving Identification of Difficult Small Classes by Balancing Class Distribution, in: Quaglini, S., Barahona, P., Andreassen, S. (Eds.), Artificial Intelligence in Medicine. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 63–66.

Liu, X., Wang, X., Su, Q., Zhang, M., Zhu, Y., Wang, Qiugen, Wang, Qian, 2017. A Hybrid Classification System for Heart Disease Diagnosis Based on the RFRS Method. Computational and Mathematical Methods in Medicine 2017, 8272091. https://doi.org/10.1155/2017/8272091

Maini, E., Venkateswarlu, B., Maini, B., Marwaha, D., 2021. Machine learning–based heart disease prediction system for Indian population: An exploratory study done in South India. Medical Journal Armed Forces India. https://doi.org/10.1016/j.mjafi.2020.10.013

Malav, A., Kadam, K., 2018. A hybrid approach for heart disease prediction using artificial neural network and K-means. International Journal of Pure and Applied Mathematics 118, 103–10.

Mienye, I.D., Sun, Y., Wang, Z., 2020. Improved sparse autoencoder based artificial neural network approach for prediction of heart disease. Informatics in Medicine Unlocked 18, 100307.

Miranda, E., Irwansyah, E., Amelga, A.Y., Maribondang, M.M., Salim, M., 2016. Detection of cardiovascular disease risk's level for adults using naive Bayes classifier. Healthcare informatics research 22, 196–205.

Mohan, S., Thirumalai, C., Srivastava, G., 2019. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. IEEE Access 7, 81542–81554. https://doi.org/10.1109/ACCESS.2019.2923707

Myers, K.D., Wilemon, K., McGowan, M.P., Howard, W., Staszak, D., Rader, D.J., 2021. COVID-19 associated risks of myocardial infarction in persons with familial hypercholesterolemia with or without ASCVD. American Journal of Preventive Cardiology 7, 100197. https://doi.org/10.1016/j.ajpc.2021.100197

Nguyen, H., Cooper, E., Kamei, K., 2011. Borderline over-sampling for imbalanced data classification. International Journal of Knowledge Engineering and Soft Data Paradigms 3, 4–21. https://doi.org/10.1504/IJKESDP.2011.039875

Poornima, V., Gladis, D., 2018. A novel approach for diagnosing heart disease with hybrid classifier. Biomed Res 29, 2274–2280.

Rajendran, N.A., Vincent, D.R., 2021. Heart Disease Prediction System using Ensemble of Machine Learning Algorithms. Recent Patents on Engineering 15, 130–139.

Rani, P., Kumar, R., Ahmed, N.M.S., Jain, A., 2021. A decision support system for heart disease prediction based upon machine learning. Journal of Reliable Intelligent Environments 1–13.

Smith, M.R., Martinez, T., Giraud-Carrier, C., 2014. An instance level analysis of data complexity. Machine Learning 95, 225–256. https://doi.org/10.1007/s10994-013-5422-z

Tama, B.A., Im, S., Lee, S., 2020. Improving an Intelligent Detection System for Coronary Heart Disease Using a Two-Tier Classifier Ensemble. BioMed Research International 2020, 9816142. https://doi.org/10.1155/2020/9816142

Terrada, O., Hamida, S., Cherradi, B., Raihani, A., Bouattane, O., 2020. Supervised machine learning based medical diagnosis support system for prediction of patients with heart disease. Advances in Science, Technology and Engineering Systems Journal 5, 269–277.

Tomek, I., 1976a. An Experiment with the Edited Nearest-Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics SMC-6, 448–452. https://doi.org/10.1109/TSMC.1976.4309523

Tomek, I., 1976b. Two modifications of CNN. IEEE Trans. Syst. Man Cybern. 6, 769–772.

TUIK (Türkiye İstatistik Kurumu), 2021.Ölüm Nedeni İstatistikleri. URL http://www.tuik.gov.tr/PreHaberBultenleri.do?id=27620 (erişim tarihi: 5.18.21).

Vapnik, V., Golowich, S.E., Smola, A., others, 1997. Support vector method for function approximation, regression estimation, and signal processing. Advances in neural information processing systems 281–287.

Wiharto, W., Kusnanto, H., Herianto, H., 2016. Interpretation of clinical data based on C4. 5 algorithm for the diagnosis of coronary heart disease. Healthcare informatics research 22, 186–195.

WHO (World Health Organization), 2021. Global status report on noncommunicable diseases. URL https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases (erişim tarihi: 6.21.21).

Makale Türü
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 106-118
DOI: 10.38016/jista.1054331

Ozon Konsantrasyonlarını Modellemek için Makine Öğrenmesi ve Derin Öğrenme Yöntemlerinin Karşılaştırılması

Şevket Ay[1]*, Ekin Ekinci[1],

[1]Sakarya Uygulamalı Bilimler Üniversitesi, Bilgisayar Mühendisliği Bölümü, Sakarya, Türkiye

sevketay09@gmail.com, ekinekinci@subu.edu.tr

Öz

Hava kirliliği günümüz için önemli bir problem olmakla birlikte sanayileşme, orman yangınları, egzoz gazları, kalitesiz yakıt kullanımı gibi sebepler gelecek nesilleri de tehdit edecek ciddi bir problem ile bizleri yüzleştirmektedir. Bu sebepler içerisinde ise yoğun sanayileşme hava kirliliğinde rol oynayan en önemli faktörlerden birisidir. Bölgesel sanayi gelişimi şehirlerde hava kalitesini etkilemektedir. Sanayinin gelişmesi ile birlikte bazı kirleticilerin miktarı azalmakta iken, ozon seviyelerinde artış yaşanmaktadır. Önümüzdeki yıllarda hava kirliliğini neden olacağı problemleri daha fazla hissetmemek, hava kalitesini yönetmek ve risklere karşı önlem almak için hava kirliliğinin tahmin edilmesi kaçınılmaz hale gelmektedir. Bu çalışmada sanayinin gelişmiş olduğu Kocaeli ve Sakarya illeri ile sanayinin çok fazla gelişmediği Çanakkale illeri için 2018-2021 arası saatlik ozon seviyelerini tahmin etmek amacıyla zaman serilerine dayalı makine öğrenmesi ve derin öğrenme yöntemleri uygulanmıştır. Uygulanan modeller Ortalama Mutlak Hata (MAE), Bağıl Mutlak Hata (RAE) ve R-kare (R^2) metrikleri kullanılarak karşılaştırılmış ve en etkin yöntemin belirlenmesi amaçlanmıştır.

Anahtar kelimeler: Makine Öğrenmesi, Derin Öğrenme, Aşırı Gradyan Arttırma (Xgboost), Yapay Sinir Ağları (YSA), Uzun Kısa Süreli Bellek (LSTM), Zaman Serileri

Comparison of Machine Learning and Deep Learning Methods for Modeling Ozone Concentrations

Abstract

Although air pollution is an important problem for today, reasons such as industrialization, forest fires, exhaust gases, poor quality fuel use confront us with a serious problem that will threaten future generations. Among these reasons, intensive industrialization is one of the most critical factors that play a role in air pollution. Regional industrial development affects air quality in cities. While the amount of some pollutants decreases with the development of the industry, there is an increase in ozone levels. In the coming years, it becomes inevitable to predict air pollution in order not to feel the problems that air pollution will cause more, to manage air quality, and to take precautions against risks. In this study, machine learning and deep learning methods based on time series were applied to predict hourly ozone levels between 2018 and 2021 for the provinces of Kocaeli and Sakarya, where the industry is developed, and Çanakkale, where the industry is not developed much. The applied models were compared using Mean Absolute Error (MAE), Relative Absolute Error (RAE), and R-square (R^2) metrics, and it was aimed to determine the most effective method.

Keywords: Machine learning, Deep learning, Extreme Gradient Boosting (Xgboost), Artificial Neural Network (ANN), Long-Short Term Memory (LSTM), Time Series

1. Giriş (Introduction)

Soluduğumuz hava kalitesinin sağlığımıza doğrudan etkisi vardır. Normal olarak havanın %78.084'ü Azot (N_2), % 20.946'sı Oksijen (O_2), %0.934'ü Argon (Ar), %0.035'i Karbondioksitten (CO_2) oluşturmaktadır. Günümüzün önemli tehditlerinden birisi olan hava kirliliği geçmişten günümüze çevresel değişiklikler, endüstriyel kirlilik, fosil yakıtların kontrolsüz tüketimi, kente göç vb. nedenlerden ötürü ortaya çıkmaktadır.

* Sorumlu yazar.
E-posta adresi: ekinekinci@subu.edu.tr

Alındı : 6 Ocak 2022
Revizyon : 27 Şubat 2022
Kabul : 20 Mart 2022

Hava kirliliği tehlikeli boyutlara ulaşırken hava kirliliği ile mücadele etmek elzem olmaktadır. Bu amaçla sürekli ölçümler yapılmaktadır. Kriter olarak ölçülmesi gereken kirleticiler ise, Karbon monoksit (CO), Kükürt dioksit (SO_2), Ozon (O_3), Partikül madde (PM), Azot oksitler (NOX) olarak belirtilmektedir.

Günümüzde yapay zekada yaşananan gelişmeler sadece bilgisayar bilimcileri değil diğer bilim dallarında çalışan araştırmacıların da ilgisini çekmeye başlamıştır. Hava kirliliği tahmininde yapay zekanın kullanılması literatürde önemli bir yer tutmaya başlamıştır.

Literatürde, O_3 konsantrasyonların modellenmesi için makine öğrenmesi ve derin öğrenme tabanlı farklı çalışmalar bulunmaktadır. Makine öğrenmesi doğrusal olmayan ve yüksek boyutlu veri kümeleri üzerinden kararlı ve performansı yüksek bilgi çıkarımı yapmaktadır (Bilgin, 2021; Yıldırım vd., 2021). Makine öğrenmesi yöntemlerinden çok katmanlı algılayıcı (ÇKA) (Paoli vd., 2011; Chattopadhyay vd., 2019; Yang vd., 2021; Bekesiene vd., 2021; Makarova vd., 2021), destek vektör makineleri (DVM) (Chelani, 2010; Tanaskuli, 2019; Mehdipour ve Memarianfard; 2019), lineer regresyon (Alipio, 2020; Allu vd., 2020; Matasović vd., 2021), Xgboost (Ding vd., 2020; Liu vd., 2020), rastgele orman (Liu vd., 2020; Ma vd., 2021) yöntemleri ile yapılmış çalışmalar mevcuttur.

Büyük veri analizi ve Grafik İşleme Biriminin (GPU) kullanılmasından bu yana, derin öğrenme büyük ilgi görmekte ve makine öğrenmesinin uygulandığı her alana uygulanmaktadır (Çağıl ve Yıldırım, 2020; Darendeli ve Yılmaz, 2021). Derin öğrenme yöntemleri ile yapılan çalışmalarda derin sinir ağları (DSA) (Wang vd., 2020; Felix vd., 2021), oto kodlayıcı (Nghiem vd., 2021), özyinelemeli sinir ağları (Adnane vd., 2021), Uzun Kısa Süreli Bellek (LSTM) (Alghieth vd., 2021; Ekinci vd. 2021; Zhang vd., 2021), konvolüsyonel sinir ağları (CNN) (Eslami vd., 2020; Sayeed vd., 2021) kullanılmıştır.

Bu çalışmanın amacı saatlik O_3 konsantrasyonlarını modellemede makine öğrenmesi ve derin öğrenme yaklaşımlarını etkinliğini değerlendirmektir. Bu amaçla kirliliğe sebep olan parametrelerden (PM10, SO_2, NO, NO_2 ve O_3) oluşan zaman serisi veri kümesi kullanılarak Xgboost, YSA ve Uzun Kısa Süreli Bellek (LSTM) yöntemleri karşılaştırılmıştır. Yapılan deneyler sonucunda ozon seviyesini tahmin etmede LSTM yönteminin diğer iki makine öğrenmesi yöntemine kıyasla daha başarılı olduğu gözlemlenmiştir.

Makalenin geri kalan kısmı ikinci bölümde veri kümesi ve uygulanan yöntemlerin anlatıldığı materyal ve yöntemler kısmıdır. Üçüncü bölümde veri önişleme, kullanılan hata metrikleri, modellerin tasarımı ve elde edilen deneysel sonuçlar ayrıntılı şekilde verilmiştir. Son bölümde ise sonuçlar ve öneriler yer almaktadır.

2. Materyal ve Yöntemler (Materials and Methods)

2.1. Veri kümesi (Dataset)

Marmara bölgesinde özellikle Kocaeli ve Sakarya illerinin sanayilerinin gelişmesi ile birlikte bu illerde hava kirliliği oldukça yüksektir. Bu çalışmada, Kocaeli ve Sakarya ile birlikte sanayinin yoğun olmadığı Çanakkale illeri için T.C. Çevre ve Şehircilik Bakanlığı Hava Kalitesi İzleme Ağı'ndan[1] sürekli ölçümler yapılarak elde edilen verilerden oluşan bir veri kümesi oluşturulmuştur.

İstasyonda ölçülen meteorolojik parametreler 10 μm'nin altındaki parçacıkları ifade eden PM10, azot oksitlerden NO, NO_2, NOX, SO_2 ve O_3 şeklindedir. Bu parametreler içerisinden O_3 konsantrasyonunu tahmin etmek için PM10, SO_2, NO, NO_2 ve O_3 parametrelerine dayalı bir tahmin yapılması hedeflenmiştir. Kocaeli, Sakarya ve Çanakkale illeri için 2018 Kasım ile 2021 Kasım arası saatlik ölçülen zaman serisi değerleri kullanılmıştır. 4 saatlik bir pencere boyutu ile (yani 4 zaman noktası) 5. saat için O_3 konsantrasyonlarının tahmini gerçekleştirilmiştir.

2.2. Yöntemler (Methods)

2.2.1. Xgboost (Xgboost)

Xgboost (Chen vd., 2016) karar ağacı temelli topluluk öğrenimi algoritmasıdır. Algoritmanın çalışma mantığı, değişkenlere farklı ağırlıklar vererek elde edilen ağaç topluluğundan çıkarımlar yapmaktır. İlk etapta tüm değişkenler eşit ağırlığa sahiptir. Ağaç topluluğu büyümeye başladıkça, problem bilgisine bağlı olarak ağırlıklar düzenlenmektedir. Yanlış sınıflandırılan gözlemlerin ağırlığı yükseltilirken, doğru sınıflandırılan gözlemlerin ağırlığı düşürülmektedir. Bu sayede ağaçlar zor durumlar karşısında kendini düzenleyebilme yeteneği kazanmaktadır. Fazla uyumu azaltan ve genel performansı artıran çeşitli düzenlemeler içermektedir (Ekinci vd., 2020). Bu özelliğinden dolayı "düzenli artırma" tekniği olarak da isimlendirilmektedir. Xgboost algoritması çeşitli düzenlemeler ile doğruluğu arttıran, paralel işleme ile hızlı sonuçlar verebilen, eksik değerlerin kullanımı için standart bir yapıya sahip olan, yükseltme işleminin yineleme aşamalarının her birinde çapraz doğrulama yapan bir algoritmadır.

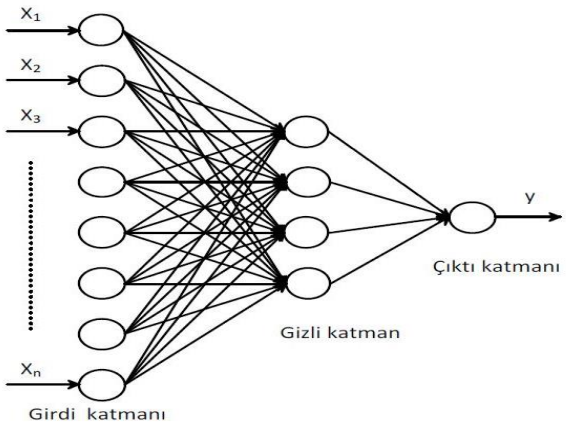
2.2.2. Yapay Sinir Ağları (Artificial Neural Networks)

Beyin insan vücudunun yapı taşı olup girdileri sinyal şeklinde alan, işleyen ve çıkış sinyallerini gönderen biyolojik sinir ağıdır. Beynin temel birimi nörondur. Beyin 200 milyar nörondan oluşmaktadır. Nöronlar dentrit, soma, akson ve sinapsis olmak üzere dört temel kısımdan oluşmaktadır. Nöron, dentritlerden sinyal

[1] http://sim.csb.gov.tr/Services/AirQuality/

toplamaktadır, soma hücreleri ise bu sinyallerin tümünü toplamaktadır ve toplam eşik değerine ulaştığında sinyal aksondan diğer nöronlara geçmektedir. Sinapslar ağırlıkları temsil etmektedir. İşte en sık tercih edilen yöntem olan YSA bu biyolojik sinir ağını taklit etmektedir ve bilgileri ağırlıklarda saklamaktadır (Garip vd., 2016; Şen, 2018). YSA'nın yapısı Şekil 1'de verilmektedir.

YSA doğrusal olmama, genelleme yapabilme, çok sayıda değişken ve parametre kullanabilme özelliklerine sahiptir. Her katman beynimizin nöronlarını taklit eden düğümlerden oluşmaktadır. Giriş katmanı, sinir ağının işleyebileceği bilgileri girdi olarak alan katmandır. Her düğüm bir özelliği yani bilgi parçasını temsil etmektedir.



Şekil 1. YSA Mimarisi (ANN Architecture)

Giriş katmanındaki her bir düğüm bir sonraki katmanda bulunan düğüme bağlanmaktadır. Ara katmanlar bir diğer deyişle gizli katmanlar giriş katmanından gelen bilgileri işlemektedir ve çıkış katmanına göndermektedir. Çıkış katmanı ise ağın son ara katmanındaki bilgileri bir araya getirmektedir ve bu şekilde gerekli tüm bilgileri çıkarmakta ve dış dünyaya göndermektedir.

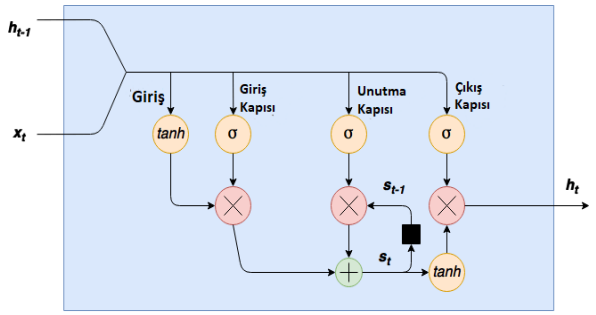
2.2.3. Uzun Kısa Süreli Bellek (LSTM)

Tekrarlı Yapay Sinir Ağları (RNN) aldıkları girdiyle ilgili önemli bilgileri hatırlayabilen ve bir sonraki adımın ne olacağını tahmin etmede çok hassas olan bir sinir ağıdır. Zaman serisi, konuşma, metin, finansal veri, ses, video, hava durumu ve daha fazlası gibi sıralı veriler için tercih edilen algoritma olmasının nedeni bu özelliğidir. RNN sadece çıktıları beslemez aynı zamanda kendi kendine geri besleme sağlamaktadır. Çünkü RNN'in dahili belleği vardır. LSTM ise RNN'in özelleşmiş bir yapısıdır (Sepp vd., 1997). Standart ileri beslemeli sinir ağlarının aksine, LSTM'in geri bildirim bağlantıları vardır. Yalnızca tek veri noktalarını değil aynı zamanda tüm veri dizilerini de işleyebilmektedir. RNN'den en önemli farkı uzun bir hafızası olmasıdır. RNN yakın geçmişi hafızasında saklayabilirken LSTM'de ise uzun bir hafıza vardır. Örneğin, LSTM ayrılmamış, bağlı el yazısı tanıma (Liwicki vd., 2009), konuşma tanıma (Sak vd., 2014) ve ağ trafiğinde veya izinsiz giriş tespit sistemlerinde (IDS) anormal durumları algılama gibi görevlere uygulanabilmektedir.

Ortak bir LSTM birimi bir hücreden, bir giriş geçidinden, bir çıkış geçidinden ve bir unutma kapısından oluşmaktadır. Hücre, keyfi zaman aralıkları boyunca değerleri hatırlamaktadır ve bu üç kapı, hücrenin içine ve dışına bilgi akışını düzenlemektedir.

Giriş kapısı, yeni bir değerin hücreye ne kadar aktığını kontrol etmektedir; unutma kapısı bir değerin hücrede ne kadar kalacağını kontrol etmektedir ve çıkış kapısı, hücredeki değerin LSTM ünitesinin çıkış aktivasyonunu hesaplamak için kullanılma derecesini kontrol etmektedir. LSTM kapılarının aktivasyon fonksiyonu genellikle lojistik sigmoid fonksiyondur. LSTM kapılarının bağlantıları vardır. Eğitim sırasında öğrenilmesi gereken bu bağlantıların ağırlıkları kapıların nasıl çalıştığını belirler.

Sigmoid katmanı, her bir bileşenden ne kadarının geçmesi gerektiğini tanımlayan sıfır ile bir arasında rakamlar vermektedir. Sıfır değeri geçiş izni yok demek iken, bir değeri geçişe izin var demektir. Aşağıdaki denklemlerde ifade edilen değişkenler vektörleri temsil etmektedir. LSTM mimarisi Şekil 2'de verilmiştir.



Şekil 2. LSTM Mimarisi (LSTM Architecture)

Şekil 2'de σ sigmoid katmanını, b_g girdi bias değerini, U_g girdi için ağırlık değerini, V_g önceki hücre çıktısı için ağırlık değerini, tanh ise aktivasyon fonksiyonunu temsil etmektedir.

$$g = \tanh\left(x_t U_g + h_{t-1} V_g + b_g\right) \quad (1)$$

$$i = \sigma(x_t U_i + h_{t-1} V_i + b_i) \quad (2)$$

$$f = \sigma(x_t U_f + h_{t-1} V_f + b_f) \quad (3)$$

$$s_t = f \circ s_{t-1} + g \circ i \quad (4)$$

$b_i$girdi kapısı için bias değerini, U_i girdi kapısı için ağırlık değerini, V_i önceki hücrenin çıktısının ağırlığını, $g \circ i$ girdi bölümünün çıktısını ifade etmektedir. $b_f$unutma kapısı için bias değerini U_f unutma kapısı için ağırlık değerini, V_f önceki hücrenin çıktısının ağırlığını temsil etmektedir. b_oçıktı kapısı için bias değerini, U_o çıktı kapısı için ağırlık değerini, V_o önceki hücrenin çıktısını ve h_t çıkışı ifade etmektedir.

$$h_t = \sigma(x_t U_o + h_{t-1} V_o + b_o) \circ \tanh(s_t) \quad (5)$$

3. Deneysel Çalışma (Experimental Study)

3.1. Veri Önişleme (Data Pre-processing)

Girdi olarak kullanılan değişkenler arasındaki değer farkının önemli ölçüde fazla olması modelde yanlılığa sebep olabilmektedir ve model verileri genelleştirememektedir. Bu sebeple eksik öğrenme durumu oluşabilmektedir. Çözüm olarak verilerin normalize edilmesi tavsiye edilmektedir. Normalleştirme yöntemlerinden Min-Maks normalizasyonu kullanılmıştır. Min-Maks normalizasyonu değişkenlerin değerini belirlenen bir aralığa dönüşmektedir. Bu çalışma kapsamında veri [0,1] aralığına çekilmiştir. Böylece herhangi bir bilgi kaybı yaşamadan modelin her özniteliğe eşit şekilde yaklaşması sağlanmıştır. Min-Maks normalizasyon formülü aşağıdaki gibidir (8):

$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (6)$$

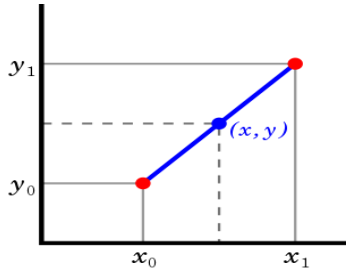
x_i değişkenin orijinal değeri, x_{min} değişkenin aldığı minimum değeri, x_{max} değişkenin aldığı maksimum değeri, y_i ise değişkenin normalize edildikten sonraki değerini temsil etmektedir.

Veriler üç il için eğitim ve test kümelerine bölünmüştür. Eldeki veri kümesinin %90'ı eğitim için %10'u ise test için kullanılmıştır. Tablo 1'de her il için eğitim ve test kümesinde bulunan örnek sayısı verilmiştir.

Tablo 1. Veri Dağılımı (Data Distribution)

Şehir 1	Eğitim Kümesi	Test Kümesi
Kocaeli	21500	4079
Sakarya	21500	4079
Çanakkale	21500	4079

Zaman serisi verilerinde eksik değerleri ortalama, ortanca vb. yöntemler ile doldurmak tehlikeli olabilmektedir. Veri seti incelendiğinde Kocaeli için %5, Sakarya için %4 ve Çanakkale için %5 oranında eksik değer bulunmaktadır. Veri adedi göz önüne alındığında bu oranlar düşük sayılabilir ve doldurulduğu zaman verilerin dağılımı bozulmayabilir. Enterpolasyon yöntemi komşular yardımıyla eksik değerleri doldurmaya yaramaktadır. Zaman serisi verilerinde eksik değerleri doldurmak için enterpolasyon yöntemi tercih edilmektedir. Çeşitli enterpolasyon metotları bulunmaktadır. Bunlar doğrusal, zamansal, polynomial gibi yöntemlerdir. Bu çalışmada doğrusal enterpolasyon kullanılmıştır. Doğrusal enterpolasyon, en yakın tanımlanmış iki veri noktası arasında eksik değerleri doğrusal olarak aralıklı değerlerle değiştirir. Şekil 3'te doğrusal enterpolasyon grafiği gösterilmiştir.



Şekil 3. Doğrusal Enterpolasyon (Linear Interpolation)

$$\frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0} \quad (7)$$

Formülde (x_0, y_0) ve (x_1, y_1) koordinat düzleminde bilinen iki noktayı temsil etmektedir. (x_0, x_1) aralığındaki x değeri için, y enterpolasyon sonucunda oluşan değerdir.

3.2 Hata Metrikleri (Error Metrics)

Çalışmada modellerin performansını değerlendirmek için MAE, R^2 ve RAE kullanılmıştır. MAE, tahmin edilen değer ile asıl değer arasındaki farkın mutlak değerinin ortalamasıdır. R^2 bir değişkenin varyansının ikinci değişkenin varyansını ne ölçüde açıkladığını gösterir. R^2 [0,1] arasında değer almaktadır. 1'e yakın bir R^2 değeri, model performansının iyi olduğunu gösterir. RAE ise tahmin edilen değer ile beklenen değerin mutlak farkının, her bir beklenen değer ile beklenen değerlerin mutlak farkına bölünmesi ile elde edilir. Sıfıra ne kadar yakınsa modelin başarılı tahminler yaptığı söylenebilir. Hata metriklerinin formülleri Eşitlik 9, 10 ve 11 ile verilmiştir.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (9)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)2}{\sum_{i=1}^{n}(y_i-\bar{y}_i)2} \quad (10)$$

$$RAE = \frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{\sum_{i=1}^{n}|y_i-\bar{y}_i|} \quad (11)$$

Formüllerde $\hat{y}_i$ tahmin edilen değeri, y_i beklenen değeri, $\bar{y}_i$ değişkenlerin ortalama değeri, n değişken sayısını ifade etmektedir.

3.3. Modellerin Gerçekleştirimi (Realization of Models)

Bu çalışmada her üç il için Xgboost, YSA ve LSTM modelleri kullanılarak performansları karşılaştırılmıştır. Modelleri geliştirirken Sckit-learn[2] ve Keras[3] kütüphaneleri kullanılmıştır.

XGBOOST modeli için ilk etapta GridSearch yöntemi ile ağacın derinliği, öğrenme oranı, ağaçları

[2] https://scikit-learn.org/stable/

[3] https://keras.io/

oluştururken alt örnekleme oranı ve ağaç sayısı parametreleri için en uygun değerler bulunmuştur. Parametre optimizasyonu sonucu ağaç derinliği değeri 4, öğrenme oranı 0.1, alt örnekleme oranı 1 ve ağaç sayısı 500 olarak belirlenmiştir.

YSA ve LSTM için ise katman sayısı, katmandaki nöronlar, gizli katman sayısı, çıktı katmanının boyutu, seyreltme değeri, optimize türü, dönem sayısı ve öğrenme tur sayısı gibi ayarlanması gereken parametreler ayarlanmıştır. YSA için iki adet yoğunluk katmanı kullanılmıştır. İlk yoğun katman için 128 nöron, ikinci yoğun katman için ise 64 nöron kullanılmıştır. Yoğunluk katmanlarında aktivasyon fonksiyonu relu tercih edilmiştir. Relu verilerdeki karmaşık ilişkilerin öğrenilmesine izin veren lineer olmayan bir fonksiyondur. Aşırı öğrenmeyi engellemek amacıyla 0.2 oranında seyreltme kullanılmıştır. Seyreltme işlemi rastgele bir şekilde bilgi azaltım yapmaktadır ve büyük oranlara sahip seyreltme işlemi önemli bilgilerin atılmasına sebebiyet verebilmektedir. LSTM için ise bir adet LSTM katmanı kullanılmıştır. LSTM katmanında 64 nöron tercih edilmiştir ve aktivasyon fonksiyonu tanh kullanılmıştır. YSA ve LSTM 0.001 öğrenme oranı, 16 parti boyutu, 500 öğrenme tur sayısı ve RMSProp kayıp fonksiyonu optimizasyonu kullanılmıştır. Kayıp fonksiyonu optimizasyonu derin öğrenme modelleri için oldukça büyük öneme sahiptir. RMSProp, gradyan tabanlı bir optimizasyondur. Kaybolan gradyan problemini önlemek için geliştirilmiştir. Dikey yönde salınımları kısıtlamaktadır ve yatay yönde hızlı yakınsama sağlamaktadır.

3.4. Deneysel Sonuçlar (Experimental Results)

Çalışma ile amacımız modellerin iller özelinde performansını karşılaştırmaktır. Sonuçlar Tablo 2, 3 ve 4 ile gösterilmiştir.

Sonuçlar incelendiğinde her üç modelinde birbirine yakın performans gösterdiği söylenebilir. Modeller Kocaeli ve Çanakkale illeri için yüksek performans göstermişlerdir. Kocaeli için R^2 skoru Xgboost için 0.93, YSA ve LSTM için ise 0.94'tür. Çanakkale için R^2 skoru Xgboost, YSA ve LSTM için 0.94'tür. Sakarya için R^2 skoru Xgboost için 0.88, YSA için 0.87 ve LSTM için ise 0.83'tür. R^2 değerlerine ait grafikler Şekil 5'te verilmiştir. Modeller Sakarya ili için daha düşük performans göstermiştir. Bunun sebebi Şekil 4 (c)'de görüldüğü üzere SO_2 değişkeni ile NO_2 ve NOX değişkeninin oldukça düşük bağımlılıkta olması olarak söylenebilir. Şekil 4 incelendiğinde Kocaeli ve Çanakkale illerindeki değişkenlerin birbiri ile daha iyi korelasyon içinde olduğu söylenebilir. İller bazında beklenen ve gerçekleşen değerlere ait grafikler ise Şekil 6, 7 ve 8 ile verilmiştir.

Tablo 2. XGBOOST modeli için performans değerlendirmesi (Performance evaluation for the XGBOOST model)

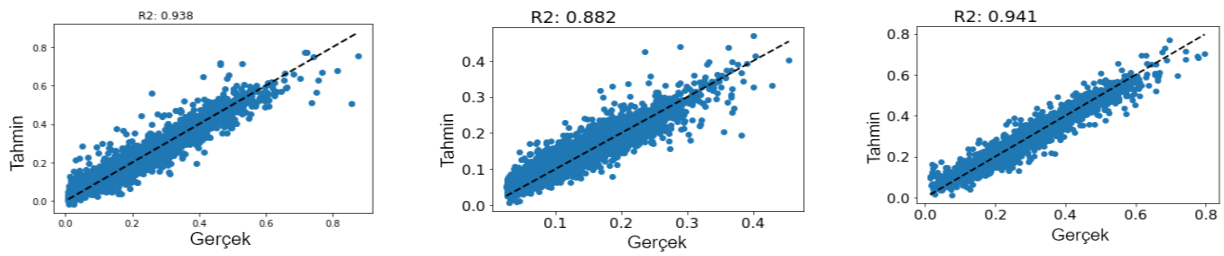
Metrik	Kocaeli	Sakarya	Çanakkale
R^2	0.93	0.88	0.94
MAE	0.026	0.019	0.020
RAE	0.19	0.30	0.21

Tablo 3. YSA Modeli için performans değerlendirmesi (Performance evaluation for the ANN model)

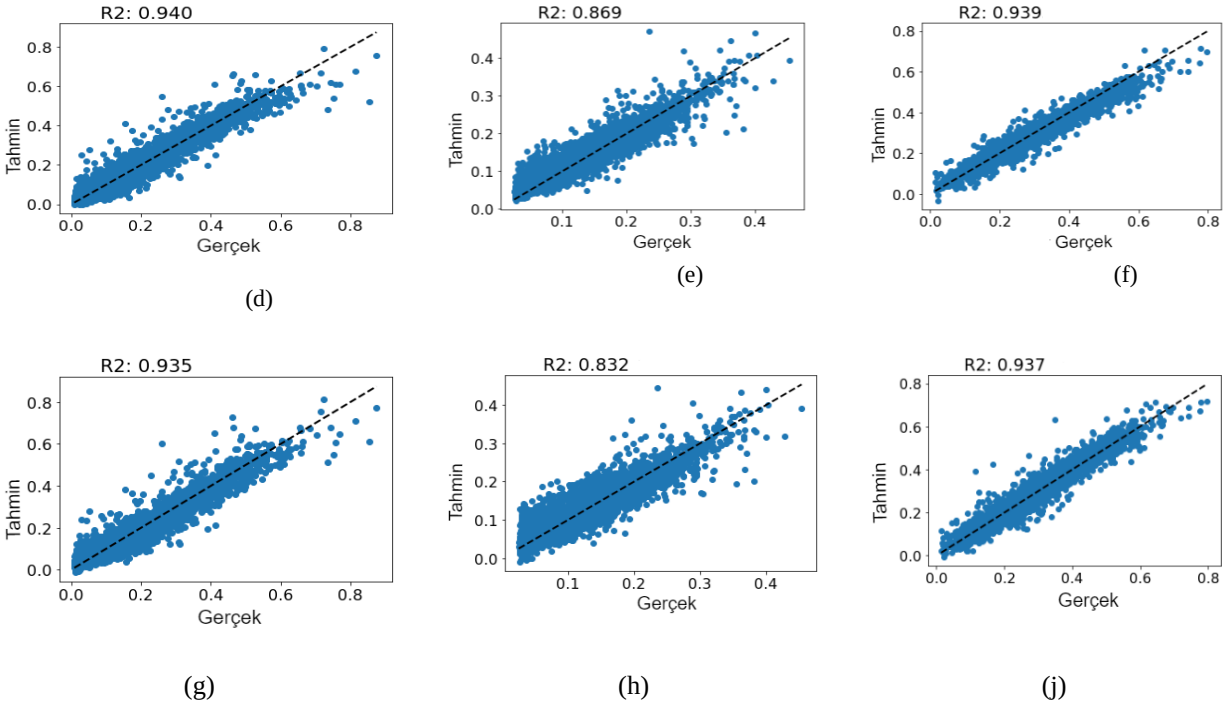
Metrik	Kocaeli	Sakarya	Çanakkale
R^2	0.94	0.87	0.94
MAE	0.026	0.021	0.021
RAE	0.19	0.32	0.22

Tablo 4. LSTM Modeli için performans değerlendirmesi (Performance evaluation for the LSTM model)

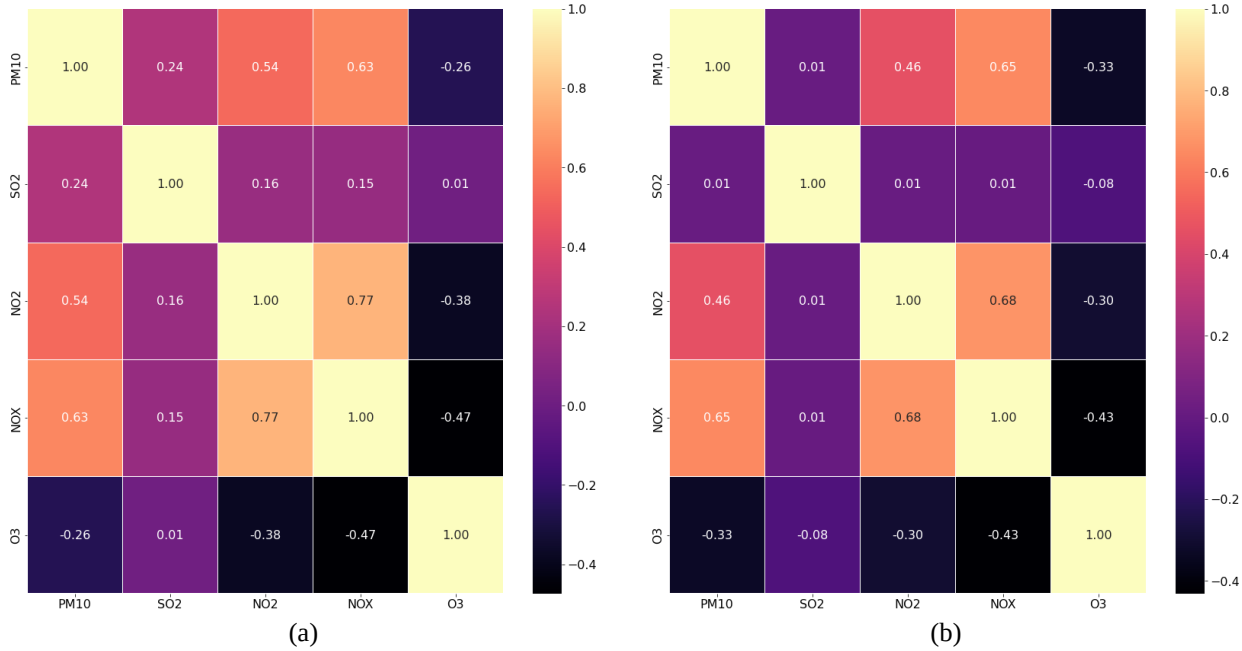
Metrik	Kocaeli	Sakarya	Çanakkale
R^2	0.94	0.83	0.94
MAE	0.027	0.022	0.020
RAE	0.20	0.34	

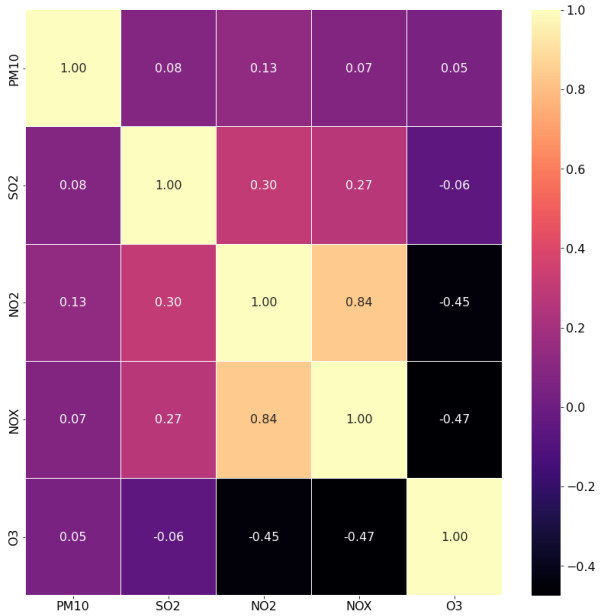


(a) (b) (c)



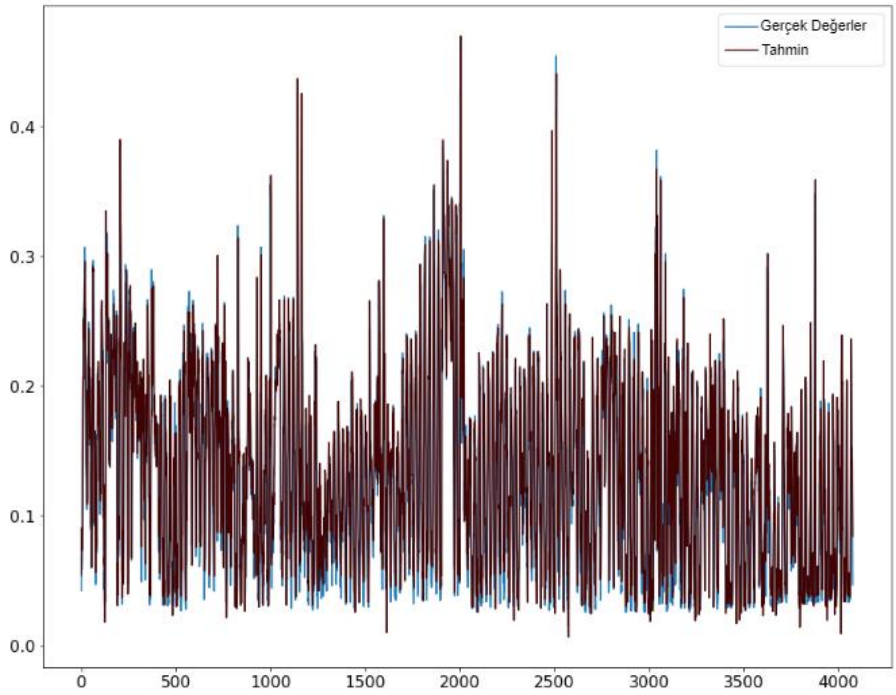
Şekil 5. Tahmin Edilen O_3 Değerleri ile Ölçülen O3 Değerleri XGBOOST; (a) Kocaeli (b) Sakarya (c) Çanakkale, ANN; (d) Kocaeli (b) Sakarya (c) Çanakkale, LSTM; (a) Kocaeli (b) Sakarya (c) Çanakkale. (Predicted O3 Values vs. Measured O3 Values XGBOOST; (a) Kocaeli (b) Sakarya (c) Çanakkale, ANN; (d) Kocaeli (b) Sakarya (c) Çanakkale, LSTM; (a) Kocaeli (b) Sakarya (c) Çanakkale)



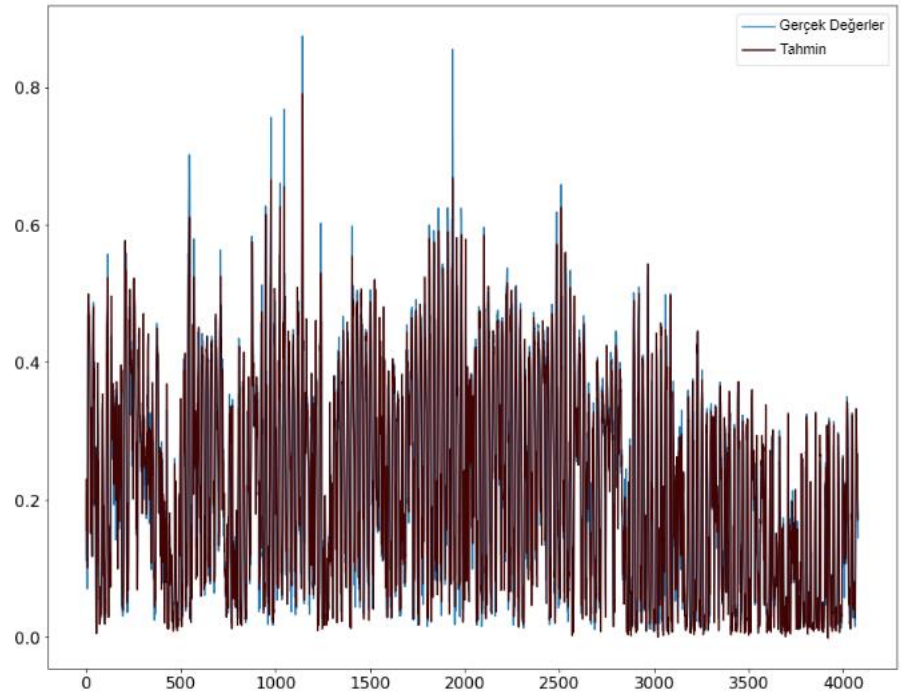


(c)

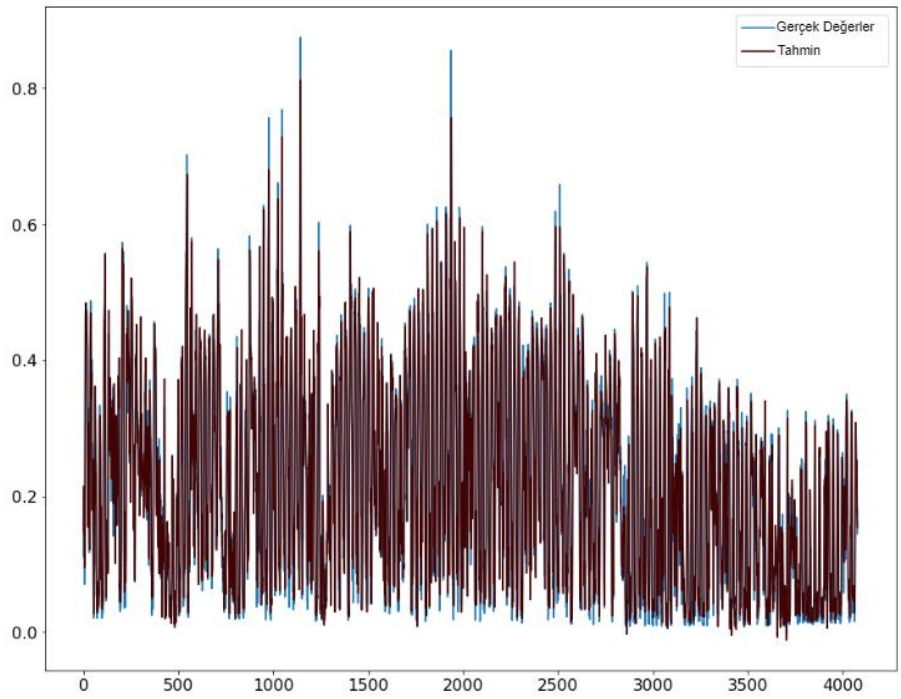
Şekil 4. (a) Kocaeli ili için değişkenlerin korelasyonu (b) Sakarya ili için değişkenlerin korelasyonu (c) Çanakkale ili için değişkenlerin korelasyonu ((a) Correlation of variables for Kocaeli province (b) Correlation of variables for Sakarya province (c) Correlation of variables for Çanakkale province)



(a)

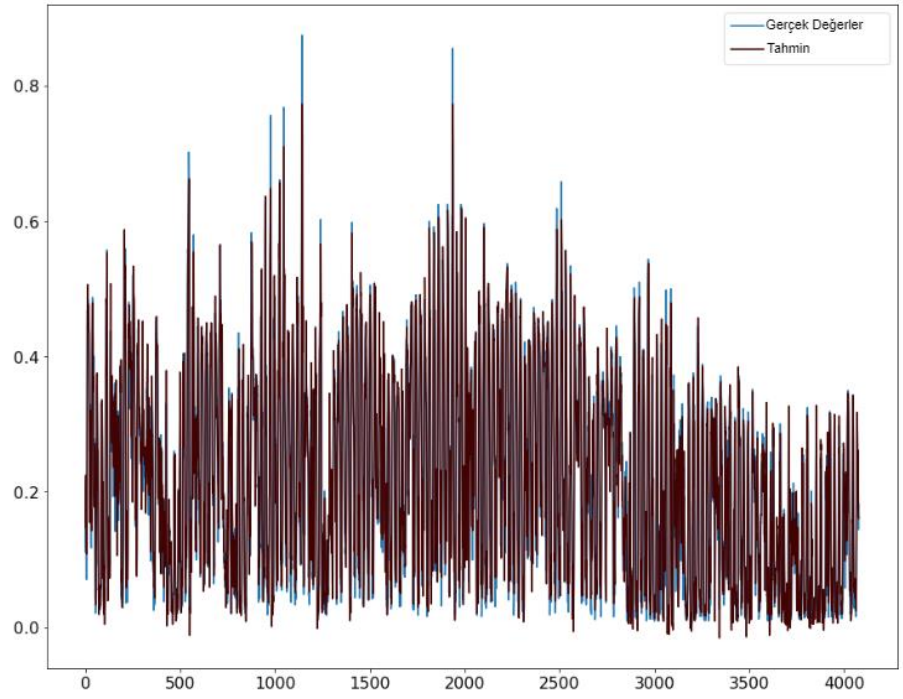


(b)

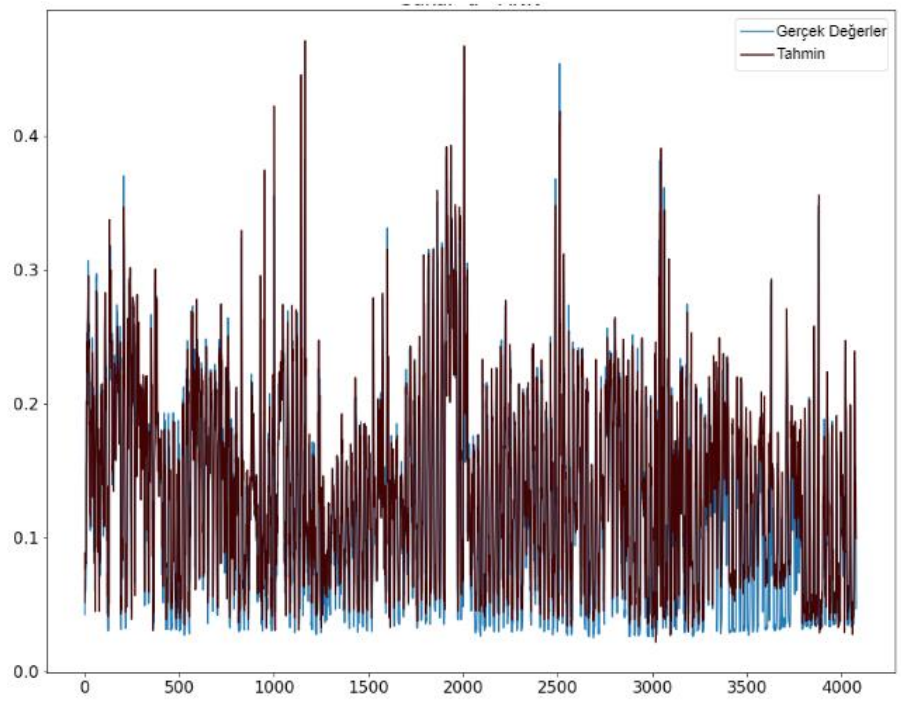


(c)

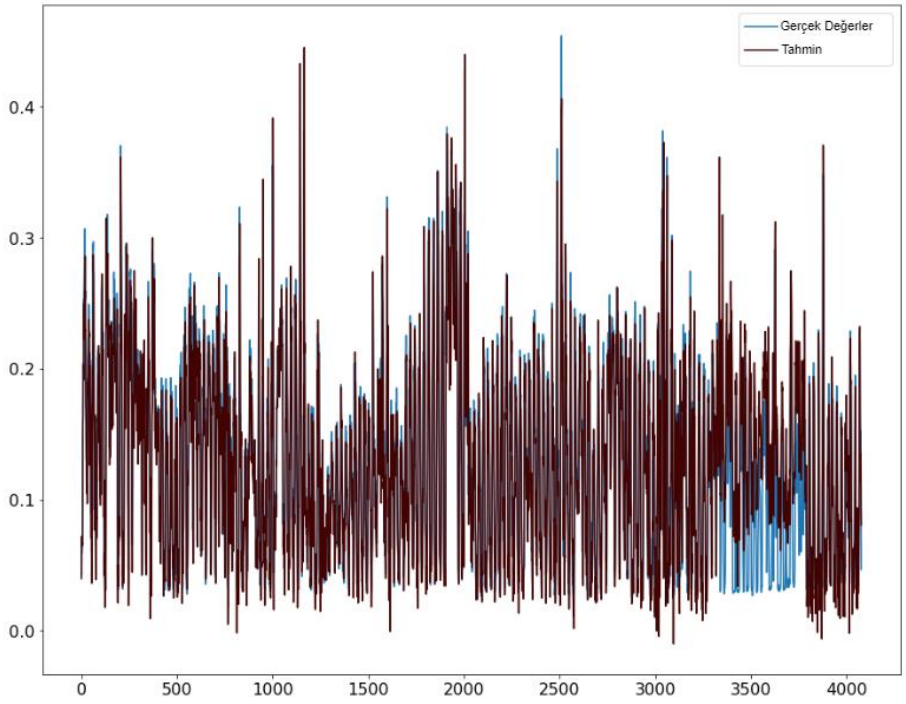
Şekil 6. Kocaeli ili için Ölçülen Değerler ve Tahmin Edilen Değerler (a) XGBOOST (b) ANN (c) LSTM (Measured Values and Estimated Values for Kocaeli Province (a) XGBOOST (b) ANN (c) LSTM)


Gerçek Değerler
Tahmin
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
3500
4000

(a)

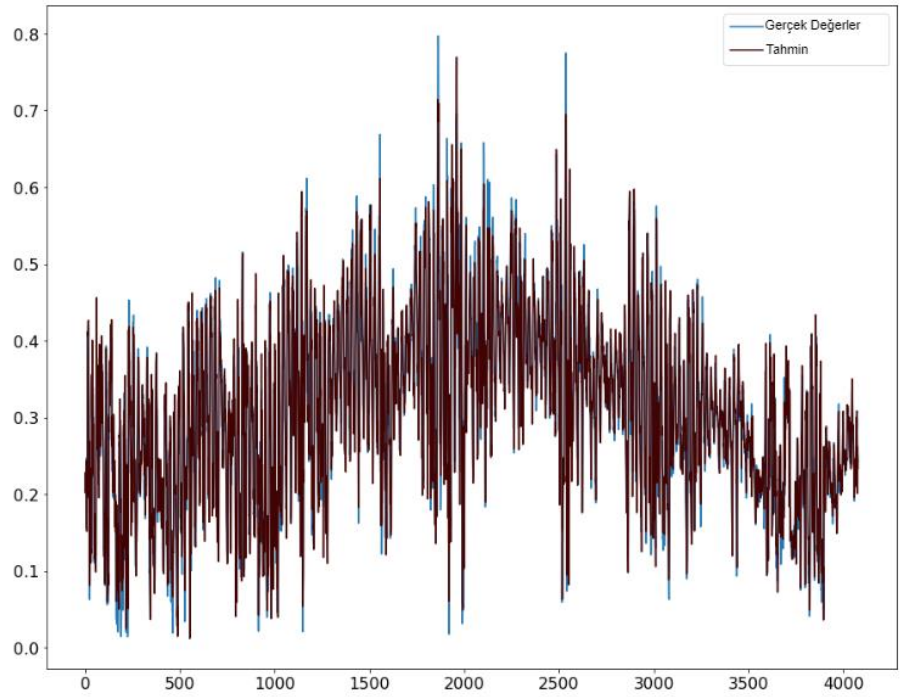

Gerçek Değerler
Tahmin
0.4
0.3
0.2
0.1
0.0
0
500
1000
1500
2000
2500
3000
3500
4000

(b)

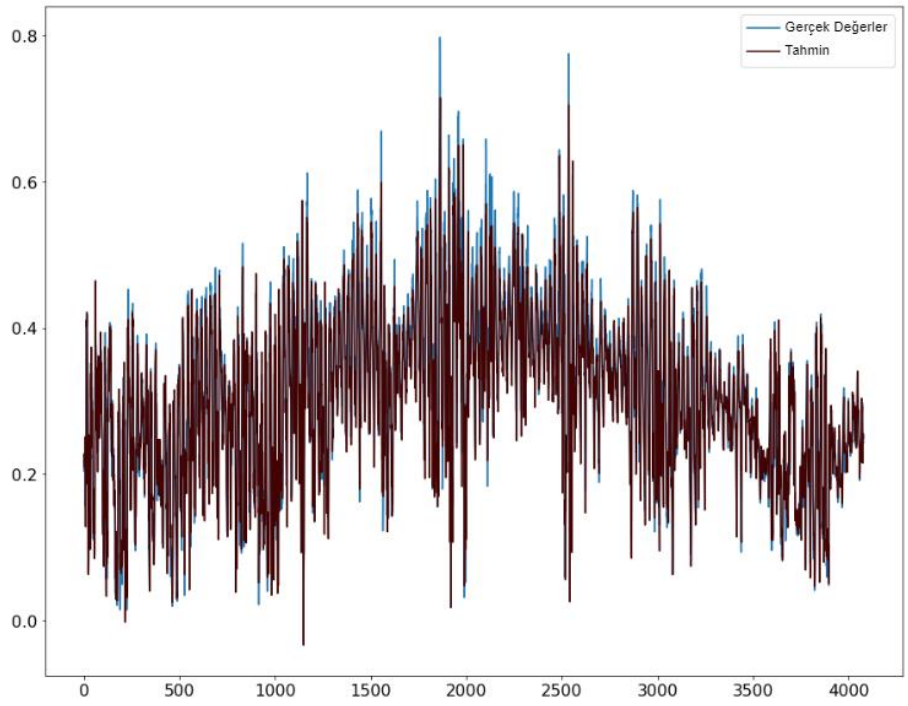


(c)

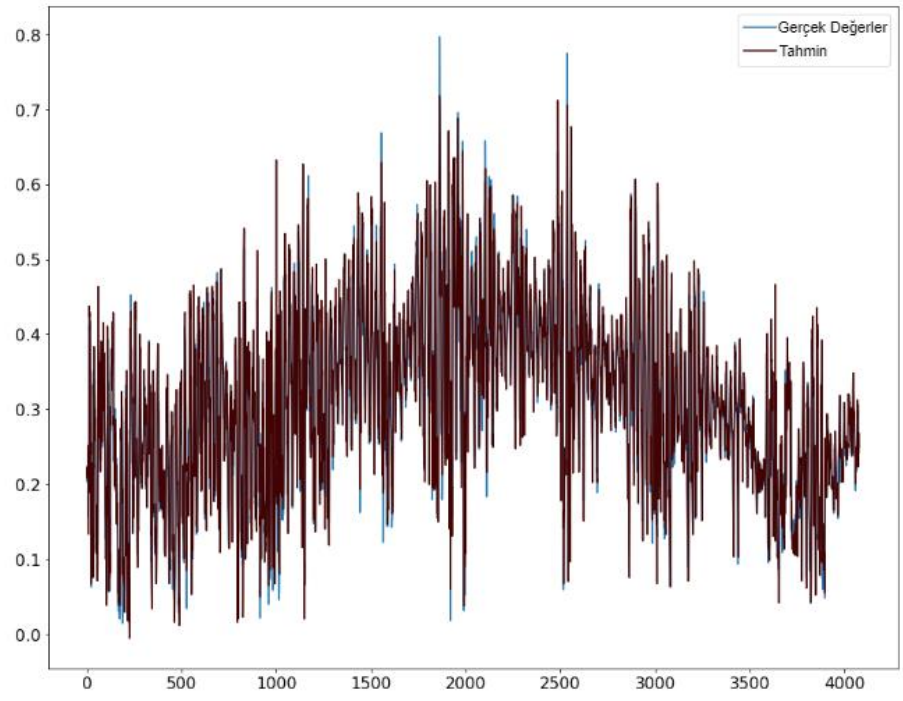
Şekil 7. Sakarya ili için Ölçülen Değerler ve Tahmin Edilen Değerler (a) XGBOOST (b) ANN (c) LSTM (Measured Values and Estimated Values for Sakarya Province (a) XGBOOST (b) ANN (c) LSTM)



(a)



(b)



(c)

Şekil 8. Çanakkale ili için Ölçülen Değerler ve Tahmin Edilen Değerler (a) XGBOOST (b) ANN (c) LSTM (Measured Values and Estimated Values for Çanakkale Province (a) XGBOOST (b) ANN (c) LSTM)

4. Sonuçlar (Results)

Bu çalışmanın amacı Türkiye'nin önde gelen sanayi şehirlerinden Sakarya, Kocaeli ile nispeten sanayisi daha az gelişmiş olan Çanakkale ilinin O_3 konsantrasyonlarını modellemektir. Bu amaçla makine öğrenmesi yöntemlerinden XGBOOST ve YSA ve derin öğrenme yöntemlerinden LSTM uygulanmıştır. Modelleme için girdi olarak kirliliğe sebep olan parametreler PM10, SO_2, NO, NO_2 ve O_3 kullanılmıştır. Test edilen modeller arasında herhangi bir ayrım yapmak zordur fakat karmaşık doğrusal olmayan sistemlerin modellemesinde ve zaman serisi problemlerindeki başarımından dolayı LSTM

kullanılması tavsiye edilir. Bu çalışmanın sonuçları, sanayilerinin gelişmişlik seviyelerine göre şehirlerin 0_3 seviyelerini tahmin etmek için bilgilendirici olabilir. Bundan sonraki çalışmalarda BiLSTM, CNN-LSTM ve Stacked LSTM gibi gelişmiş modeller denenebilir.

Kaynakçalar (References)

Adnane, A., Leghrib, R., Chaoufi, J., & Chirmata, A., 2020. The Use of a Recurrent Neural Network for Forecasting Ozone Concentrations in the City of Agadir (Morocco). Journal of Atomic, Molecular, Condensed Matter and Nano Physics, 7(3), 197-206.

Alghieth, M., Alawaji, R., Saleh, S. H., Alh, S., 2021. Air Pollution Forecasting Using Deep Learning. International Journal of Online & Biomedical Engineering, 17(14).

Alipio, M. M., 2020. Do latitude and ozone concentration predict Covid-2019 cases in 34 countries?. medRxiv.

Allu, S. K., Srinivasan, S., Maddala, R. K., Reddy, A., Anupoju, G. R., 2020. Seasonal ground level ozone prediction using multiple linear regression (MLR) model. Modeling Earth Systems and Environment, 6, 1981-1989.

Bekesiene, S., Meidute-Kavaliauskiene, I., Vasiliauskiene, V., 2021. Accurate prediction of concentration changes in ozone as an air pollutant by multiple linear regression and artificial neural networks. Mathematics, 9(4), 356.

Bilgin, G., 2021. Investigation of The Risk of Diabetes in Early Period using Machine Learning. Journal of Intelligent Systems: Theory and Applications, 4(1), 55-64.

Chattopadhyay, G., Midya, S. K., Chattopadhyay, S., 2019. MLP based predictive model for surface ozone concentration over an urban area in the Gangetic West Bengal during pre-monsoon season. Journal of Atmospheric and Solar-Terrestrial Physics, 184, 57-62.

Chelani, A. B., 2010. Prediction of daily maximum ground ozone concentration using support vector machine. Environmental monitoring and assessment, 162(1), 169-176.

Çağıl, G., Yıldırım, B., 2020. Detection of an Assembly Part with Deep Learning and Image Processing. Journal of Intelligent Systems: Theory and Applications, 3(2), 31-37.

Darendeli, B. N., Yılmaz, A., 2021. Convolutional Neural Network Approach to Predict Tumor Samples Using Gene Expression Data. Journal of Intelligent Systems: Theory and Applications, 4(2), 136-141.

Ding, J., Liu, M., Ma, Z., Liu, R., Bi, J., 2020. Spatial and temporal trends in the mortality burden of ozone pollution in China: 2005-2017. ISEE Conference Abstracts, 24-27 August 2020.

Ekinci, E., İlhan Omurca, S., Özbay, B., 2021. Comparative assessment of modeling deep learning networks for modeling ground-level ozone concentrations of pandemic lock-down period. Ecological Modelling, 457, 1-11.

Ekinci, E., İlhan Omurca, S., Sevim, S., 2020. Improve Offensive Language Detection with Ensemble Classifiers. International Journal of Intelligent Systems and Applications in Engineering, 8(2), 109-115.

Eslami, E., Choi, Y., Lops, Y., Sayeed, A., 2020. A real-time hourly ozone prediction system using deep convolutional neural network. Neural Computing and Applications, 32(13), 8783-8797.

Garip Batık, Z., Büyükbıçakçı, E., 2016. Klasik Enterpolasyon Yöntemleri ve Yapay Sinir Ağı Yaklaşımları ile Matematiksel Denklemlerin Karşılaştırılmalı Çözümü İçin Arayüz Tasarımı, 4th International Symposium on Innovative Technologies in Engineering and Science, 3-5 November 2016, Antalya, Turkey, pp. 1379-1383.

Kleinert, F., Leufen, L. H., Lupascu, A., Butler, T., Schultz, M. G., 2021. Representing chemical history for ozone time-series predictions-a method development study for deep learning models. EGU General Assembly Conference Abstracts, 19-30 April, pp. EGU21-12146.

Liu, H., Liu, J., Liu, Y., Ouyang, B., Xiang, S., Yi, K., Tao, S., 2020. Analysis of wintertime O3 variability using a random forest model and high-frequency observations in Zhangjiakou—an area with background pollution level of the North China Plain. Environmental Pollution, 262, 114191.

Liu, R., Ma, Z., Liu, Y., Shao, Y., Zhao, W., Bi, J., 2020. Spatiotemporal distributions of surface ozone levels in China from 2005 to 2017: A machine learning approach. Environment international, 142, 105823.

Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition". (IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855

Ma, R., Ban, J., Wang, Q., Zhang, Y., Yang, Y., He, M. Z., Li, S., Shi, W., Li, T., 2021. Random forest model based fine scale spatiotemporal O3 trends in the Beijing-Tianjin-Hebei region in China, 2010 to 2017. Environmental Pollution, 276, 116635.

Ma, Z., Liu, R., Bi, J., 2019. Spatiotemporal distributions of ground ozone levels in China from 2005 to 2016: a machine learning approach. AGU Fall Meeting Abstracts, 9-13 December 2019, San Francisco, USA, pp. A41J-2709.

Makarova, A., Evstaf'eva, E., Lapchenco, V., Varbanov, P. S., 2021. Modelling tropospheric ozone variations using artificial neural networks: A case study on the Black Sea coast (Russian Federation). Cleaner Engineering and Technology, 5, 100293.

Matasović, B., Pehnec, G., Bešlić, I., Davila, S., Babić, D., 2021. Assessment of ozone concentration data from the northern Zagreb area, Croatia, for the period from 2003 to 2016. Environmental Science and Pollution Research, 1-11.

Mehdipour, V., Memarianfard, M., 2019. Ground-level O3 sensitivity analysis using support vector machine with radial basis function. International Journal of Environmental Science and Technology, 16(6), 2745-2754.

Nghiem, T. D., Mac, D. H., Nguyen, A. D., Lê, N. C., 2021. An integrated approach for analyzing air quality monitoring data: a case study in Hanoi, Vietnam. Air Quality, Atmosphere & Health, 14(1), 7-18.

Paoli, C., Notton, G., Nivet, M. L., Padovani, M., Savelli, J. L. 2011. A neural network model forecasting for prediction of hourly ozone concentration in Corsica. 2011 10th International Conference on Environment and Electrical Engineering, 1-7 May 2011, Rome, Italy, pp. 1-4.

Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling"

Sayeed, A., Choi, Y., Eslami, E., Jung, J., Lops, Y., Salman, A. K., Lee, J. B., Park, H. J., Choi, M. H. (2021). A novel CMAQ-CNN hybrid model to forecast hourly surface-ozone concentrations 14 days in advance. Scientific reports, 11(1), 1-8.

Sepp H., Jürgen S., 1997. Long short-term memory.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Şen, Z., 2018. Significance of Artificial Intelligence in Science and Technology. Journal of Intelligent Systems: Theory and Applications, 1(1), 1-4.

T. Chen, C. Guestrin, M. Assoc Comp, XGBoost: a scalable tree boosting system, 2016.

Tanaskuli, M., Ahmed, A. N., Zaini, N., Abdullah, S., Borhana, A. A., Mardhiah, N. A., 2020. Ozone prediction based on support vector machine. Indonesian Journal of Electrical Engineering and Computer Science, 17(3), 1461-1466.

Wang, H. W., Li, X. B., Wang, D., Zhao, J., & Peng, Z. R., 2020. Regional prediction of ground-level ozone using a hybrid sequence-to-sequence deep learning approach. Journal of Cleaner Production, 253, 119841.

Yang, X., Zhang, M., Zhang, B., 2021. A Generic Model to Estimate Ozone Concentration from Landsat 8 Satellite Data Based on Machine Learning Technique. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 7938-7947.

Yıldırım, A. E., Kadıoğlu, Ö. F., Kavak, H., Salman, K., Uçar, M. K., Uçar, Z., Bozkurt, M. R., 2021. Gender-Based Artificial Intelligence Based Detection of Basal Metabolic Rate by Electrocardiography Signal. Journal of Intelligent Systems: Theory and Applications, 4(2), 168-176.

Research Article
Journal of Intelligent Systems: Theory and Applications 5(2) 2022: 119-126
DOI: 10.38016/jista.1033354

A New Instance Selection Method for Enlarging Margins Between Classes

Fatih Aydın[1*]

[1] Balıkesir University, Department of Computer Engineering, Balıkesir, Turkey
fatih.aydin@balikesir.edu.tr

Abstract
As discarding superfluous instances in data sets shortens the learning process, it also increases learning performance because of eliminating noisy data. Instance selection methods are commonly utilized to undertake the abovementioned tasks. In this paper, we propose a new supervised instance selection algorithm called Border Instances Reduction using Classes Handily (BIRCH). BIRCH considers k-nearest neighbors of each instance and selects instances that have neighbors from the only same class, namely, but not having neighbors from the different classes. It has been compared with one traditional and four state-of-the-art instance selection algorithms by using fifteen data sets from various domains. The empirical results show BIRCH well delivers the trade-off between accuracy rate and reduction rate by tuning the number of neighbors. Furthermore, the proposed method guarantees to yield a high classification accuracy. The source code of the proposed algorithm can be found in https://github.com/fatihaydin1/BIRCH.

Keywords: Machine learning, nearest neighbors, instance reduction, instance selection, big data.

Sınıflar Arası Kenar Payını Genişletmek İçin Yeni Bir Örnek Seçim Algoritması

Öz
Veri kümelerindeki gereksiz örneklerin atılması öğrenme sürecini kısalttığı gibi gürültülü verileri ortadan kaldırdığı için öğrenme performansını da arttırmaktadır. Örnek seçim yöntemleri, yukarıda belirtilen görevleri yerine getirmek için yaygın olarak kullanılmaktadır. Bu makalede, "Border Instances Reduction using Classes Handily (BIRCH)" adlı yeni bir denetimli örnek seçim algoritması öneriyoruz. BIRCH, her örneğin k-en yakın komşularını dikkate alarak, sadece aynı sınıftan komşuları olan, yani farklı sınıflardan komşuları olmayan örnekleri seçer. BIRCH, çeşitli alanlardan on beş veri kümesi kullanılarak biri geleneksel ve dördü son teknoloji örnek seçim algoritması ile karşılaştırılmıştır. Ampirik sonuçlar, BIRCH'in komşu sayısının ayarlanmasıyla doğruluk oranı ve azaltma oranı arasındaki dengeyi iyi sağladığını göstermektedir. Ayrıca önerilen yöntem, yüksek bir sınıflandırma doğruluğunu sağlamayı garanti eder. Önerilen algoritmanın kaynak kodu https://github.com/fatihaydin1/BIRCH web adresinde bulunabilir.

Anahtar Kelimeler: Makine öğrenmesi, en yakın komşular, örnek azaltma, örnek seçimi, büyük veri.

1. Introduction

Machine Learning (ML) is a discipline, which intends to redound learning capability for automata to discover patterns in real-world data. But Some ML algorithms such as Support Vector Machine (SVM) and k-Nearest Neighbors (kNN) suffer from big data in terms of running time. Instance selection is a process of getting rid of unnecessary data (Olvera-López *et al.*, 2010). In other words, the common goal of the instance selection methods is to discard redundant data from the data set. After the instance selection stage, the desired end is the classification performances over the original data set and the selected subset are close to each other. Instance selection would be beneficial at reducing the training and test time for lazy learners and function learners such as SVM and Neural Networks (NN). Besides, instance reduction methods are used to address the challenges in the different areas such as class-imbalanced data sets, time series, distributed learning, monotonic data sets, noise sensitiveness, and lazy learners. In the literature review, it is seen that the nearest neighbor, evolutionary methods, meta-approaches, computational strategies, probabilistic approaches, cluster-based approach, geometrical

* Corresponding Author
E-mail: fatih.aydin@balikesir.edu.tr

Recieved : 6 Dec 2021
Revision : 29 Dec 2021
Accepted : 23 Mar 2022

approaches, and ranking approach have been utilized to develop instance selection algorithms. There exist several joint characteristics in instance selection methods: type of selection, the direction of search, and evaluation of search (Olvera-López *et al.*, 2010; García-Pedrajas, 2011; Garcia *et al.*, 2012). Furthermore, the criteria such as storage requirement, noise resistance, classification accuracy, and running time have been used to compare instance selection algorithms (Garcia *et al.*, 2012).

In the literature, Condensed Nearest Neighbor (CNN) is the first approach that has been designed to discard irrelevant or noisy data (Hart, 1968). CNN is an iterative method and begins with a blank subset. In the next stage, CNN indiscriminately selects a point from the training data and joins it to the subset if the instance is misclassified while using the subset as training data. The halt rule is that there remain no more instances. CNN does not promise to attain the optimal subset. Besides, it forms different subsets at each run because of selecting instances arbitrarily (Alpaydin, 1997). Modified Condensed Nearest Neighbor (MCNN) has been proposed to enhance CNN. MCNN produces the subset by regarding the centroid of the misclassified instances in each class. MCNN achieves better performance if the data is normally distributed (Susheela Devi and Murty, 2002). Edited Nearest Neighbor (ENN) is one of the first algorithms that focus on eliminating noisy instances (Wilson, 1972).

Wilson and Martinez proposed six reduction algorithms abbreviated DROP1-DROP5 (i.e., Decremental Reduction Optimization Procedure Family), and DEL (Wilson and Martinez, 2000). DROP3-DROP5 methods are hybrid methods that fuse condensing and editing techniques.

In respect of meta approaches, Alpaydin introduced a voting approach combining predictions from a sequence of models after training multiple subsets by using two voting schemes such as simple voting and weighted voting (Alpaydin, 1997).

As for the use of local-sensitive hashing family (LSH), LSH-IS-S and LSH-IS-F methods proposed based on LSH are with quadratic and log-linear complexities and rely on unveiling similarities between instances (Arnaiz-González *et al.*, 2016). Data Reduction with Locality-Sensitive Hashing (DR.LSH) is a new instance selection method using LSH. The proposed method tries to rapidly detect similar and redundant data and discard them from the original data set (Aslani and Seipel, 2020). The Border Point extraction based on Locality-Sensitive Hashing (BPLSH) that has been suggested as a novel instance selection method holds instances that are close to the decision borders and eliminates interior instances (Aslani and Seipel, 2021).

Rico-Juan et al proposed two instance selection algorithms based on the ranking approach. The goal of the first extension is to obtain greater robustness against noise according to the nearest neighbors in the selection process. The second method employs a new parameter-free approach to select instances (Rico-Juan, Valero-Mas and Calvo-Zaragoza, 2019). Ruiz and Gómez-Nieto proposed a novel instance selection algorithm to build Quantitative Structure-Activity Relationship (QSAR) classification models by using the Rivality Index NeighborHood (RINH) algorithm. The method can get significant reduction rate in the size of the training data as maintaining the classification performance (Ruiz and Gómez-Nieto, 2020). Fast Data Reduction with Granulation-based Instances Importance Labeling (FDR-GIIL) has been proposed as a fast instance selection method using granular computing to select the instances that contribute to the classification performance (Sun *et al.*, 2019).

For the solution suggestions to data sets with different properties, Wang et al introduced two data cleaning algorithms to address class-imbalanced data sets. The former examines whether realizing instance selection to eliminate several noisy data from the majority class can improve the performance of one-class classifiers. The latter handles instance selection and missing value problems jointly for incomplete data sets (Wang, Tsai and Lin, 2021).

Constraint Nearest Neighbor-based Instance Reduction (CNNIR) has been proposed as a novel instance selection algorithm based on the concept of natural neighbor, removes noises, and searches core instances. It defines a constraint nearest-neighbor chain that only consists of three instances to choose boundary instances that can build a smooth decision boundary, next the subset is obtained by merging boundary and inner instances (Yang *et al.*, 2019).

Shell Extraction (SE) is a new instance selection method, which considers an unbalanced distribution of instances and a strategy with self-adaption from the geometrical perspective (Liu *et al.*, 2017). Akinyelu and Adewumi introduced two novel instance selection methods for SVM Speed Optimization: FFA-based Instance Selection (FFA_IS) and Edge Instance Selection Algorithm (EISA). FFA_IS is inspired by the flashing behavior of fireflies. EISA relies on the idea of edge detection in image processing (Akinyelu and Adewumi, 2017). Akinyelu and Ezugwu suggested two instance selection methods for SVM speed optimization called the Flower Pollination Instance Selection Algorithm (FPISA) and the Social Spider Instance Selection Algorithm (SSISA), which are respectively a nature-inspired metaheuristic algorithm and normal individual-based swarm intelligence algorithm (Akinyelu and Ezugwu, 2019).

In this paper, we propose a condensing approach that performs to eliminate the boundary instances instead of preserving them. Thus, large margins between classes are formed. The reason for applying to the first stage is to reduce the error that a model makes due to variance. This approach especially supports the learners that suffer from high variance. The time complexity of our proposed method is log-linear in the best case and

quadratic in the worst case. Besides, the proposed algorithm has obtained remarkable results on the data sets used in the experiments. The main contributions of the proposed method are as follows:

- The proposed algorithm can faster process big data compared to the similar approaches.
- The algorithm is easy to implement.
- The proposed method guarantees a high accuracy rate.
- The algorithm has only two parameters to adjust.

The rest of the paper is organized as the following. In Section 2, we introduce the proposed method. In Section 3, we explain the experimental setup. In Section 4, we present the experimental results. Finally, we put forth the conclusions of the paper in Section 5.

2. The related work

In this section, we provide the description of the proposed method and calculate the time and space complexities of the proposed algorithm.

2.1. The description of the proposed method

The proposed algorithm performs to remove boundary instances and thus, enlarges the margin between the classes. In this end, the proposed method selects instances that have neighbors from the only same class, namely, but not having neighbors from the different classes by considering the k-nearest neighbors of each instance. The contributions of removing boundary instances are: (i) keeping up with streaming data that changes over time, (ii) increasing resistance against noise, and (iii) reinforcing learners that suffer from variance. As a result of filtering up boundary instances by using 1-Nearest Neighbors (1NN), the removed error rate corresponds to at most twice the Bayes error rate as proved by Cover and Hart (Cover and Hart, 1967) in (1):

$$R^* \leq R \leq 2R^*(1 - R^*) \leq 2R^* \quad (1)$$

where R^* denotes Bayes error rate (i.e., irreducible error) and R denotes 1NN error rate. The Bayes classifier is optimal since its risk is the minimum expected error rate R^*. For a data set with two classes (c_1 *and* c_2) and any point x, let $P(c_1|x)$ and $P(c_2|x)$ be error rates for each class. Accordingly, the n-sample 1NN risk is shown in (2).

$$R = E[P(c_1|x)P(c_2|x) + P(c_2|x)P(c_1|x)]$$
$$= E[2P(c_1|x)P(c_2|x)] \quad (2)$$

Since $P(c_1|x) + P(c_2|x) = 1$, we have

$$R = E\big[2P(c_1|x)\big(1 - P(c_1|x)\big)\big]$$

Since $R^* = E[P(c_1|x)]$, we have

$$R = 2R^*(1 - R^*) - 2 \times Var\big(P(c_1|x)\big)$$

Considering the case in which the variance of $P(c_1|x)$ is zero, we arrive at (3).

$$R \leq 2R^*(1 - R^*) \quad (3)$$

Consequently, removing the boundary instances on the training set decreases the generalization error since it removes the noisy instances or the instances that can cause errors due to high variance.

The proposed method runs according to the number of the nearest neighbors to eliminate instances. We propose a new instance selection algorithm called Border Instances Reduction using Classes Handily (BIRCH) and describe it in Algorithm 1.

Algorithm 1: BIRCH

Input:
T = {(x_1, y_1), …, (x_m, y_m)} ∈ $\mathbb{R}^{m\times d}$: Data set
δ: The distance metric (by default, 'cityblock')
k: The number of neighbors (by default, 1)

Output:
S = {(x_1, y_1), …, (x_t, y_t)} ∈ $\mathbb{R}^{t\times d}$: Selected points

1: **S** ← **T**

2: $\mathbf{N}^{m\times k}$ ← Find the k-nearest neighbors of each instance

3: $\mathbf{C}^{m\times k}$ ← Find the class of **N**

4: **A** = {x: x ∈ **X**, x is neighbor instance from the different class in **C**}

5: **B** = {x: x ∈ **X**, x is neighbor instance from the same class in **C**}

6: **S = B\A**

2.2. The time and space complexities

Accordingly, we carry out the calculation of the time and space complexities of the algorithm. In the first stage, BIRCH searches for the k-nearest neighbors and removes instances, depending on the case that they are from the same or different class. The determination of the k-nearest neighbors is calculated with time complexity $O(kmlog_2m)$ and space complexity $O(md)$. Finding the classes of the neighbors is calculated with time complexity $O(mk)$ and space complexity $O(mk)$. The upper bound time and space complexities that are needed to search unique instances are $O(2mlog_2m)$ and $O(mk)$, respectively. The time and space complexities of difference between two sets are $O(mk)$. As a result, the total time complexity of the first stage is $O\big((k+2)mlog_2m + 2mk\big)$ and the total space complexity of the first stage is $O(md)$. We neglect the expressions owing less effect on high order terms.

Accordingly, the time complexity of BIRCH is found as $O(kmlog_2m + mk)$. Consequently, the time complexity of BIRCH is log-linear in the best case and log-quadratic in the worst case (i.e., $k \approx m$).

3. Experimental Setup

In this section, we explain the experimental setup, including experimental data sets, instance selection algorithms used in the experiments, evaluation metrics, and implementations.

3.1. Data sets

BIRCH has been compared with the state-of-the-art instance selection algorithms to measure its efficiency by using fifteen data sets from the UCI database[1], OpenML[2], and MATLAB[3]. The data sets have been picked from the various domains. Besides, the selected data sets contain the different number of instances, features and classes. The descriptive information belonging to those data sets is shown in Table 1. The imbalance ratio denotes the ratio of the number of classes with the most instances to the number of those with the least instances.

3.2. Instance selection methods

Table 1. The characteristics of the data sets used experiments

#	Data set	Instances	Features	Classes	Imbalance ratio
1	Arrhythmia	452	279	13	122.50
2	Avila	20867	10	12	857.20
3	BostonHousing2	506	18	92	30.00
4	EEG_EyeState	14980	14	2	1.23
5	Electricity	45312	8	2	1.36
6	HTRU2	17898	8	2	9.92
7	HumanActivity	24075	60	5	2.34
8	LetterRecognition	20000	16	26	1.11
9	Madelon	2000	500	2	1.00
10	MAGIC Gamma Telescope	19020	10	2	1.84
11	Mozilla4	15545	5	2	2.04
12	Nomao	34465	118	2	2.50
13	Ovariancancer	216	4000	2	1.27
14	Seeds	210	7	3	1.00
15	Shuttle	58000	9	7	4558.60

The proposed method has been compared with one conventional and four state-of-the-art instance selection algorithms in Table 2. The parameter values and other characteristics of the algorithms used in the experiments are also shown Table 2. In addition, we have conducted all the experiments by the default values of the algorithms. All the methods used in the experiments benefit from class information and they adopt the filter approach.

3.3. Implementations

The baseline method means that the 1NN algorithm applies to the original data set. Additionally, we apply 10-fold cross-validation to all the experiments and repeat each experiment five times to select the training data with different combinations. The experiments have been conducted in the MATLAB R2021a on an i5-8265U CPU at 1.6 GHz with 8 GB of RAM on Windows 11 Pro (64-bit). Further, we use the default number of neighbors and default distance metric as 1 and 'city block', respectively for BIRCH.

3.4. Evaluation metrics

We have used three criteria such as classification accuracy, reduction rate, and running time have been used to compare instance selection methods.

Table 2. The instance selection methods used in the experiments

Algorithm	Supervision	Type	Technique	Parameter(s)
BPLSH[4]	✓	Filter	Condensation	M=30, L=10, W=1
DR.LSH[5]	✓	Filter	Hybrid	M=25, L=10, W=1, ST=9
LSH-IS-S[6]	✓	Filter	Hybrid	L=0, Y=10, O=4, W=1, S=1
LSH-IS-F[6]	✓	Filter	Hybrid	L=0, Y=10, O=4, W=1, S=1
Wilson's ENN[7]	✓	Filter	Edit	k=3

4. Results and Discussion

In this section, we report the results regarding the comparative results of the instance selection methods.

The illustration in which the proposed algorithm reduces the boundary instances on the seeds data set is shown in Figure 1. The seeds data set is a data set related

[1] http://archive.ics.uci.edu/ml
[2] https://www.openml.org/
[3] https://www.mathworks.com/help/stats/sample-data-sets.html
[4] https://github.com/mohaslani/BPLSH
[5] https://github.com/mohaslani/DR.LSH
[6] https://github.com/alvarag/LSH-IS
[7] https://github.com/LucyKuncheva/Instance_selection

to Life sciences. According to the results, the reduction rates of the instances are respectively 27.14%, 39.05%, and 43.33% for k = 1, k = 2, and k = 3.

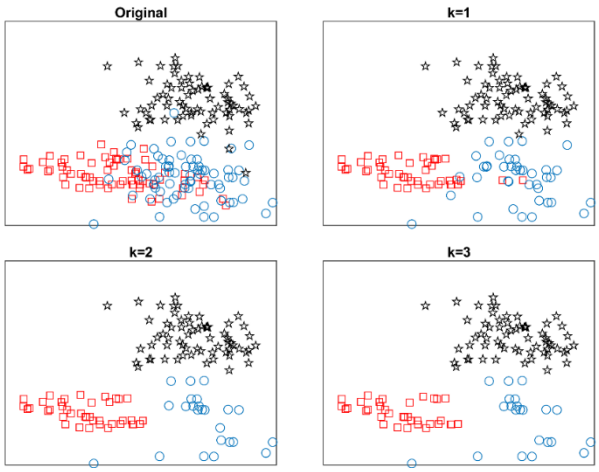


Figure 1. The illustration of reduction of the boundary instances on the seeds data set (3rd and 7th features), according to nearest neighbors (i.e., the value k)

Figure 2 shows the illustration in which the proposed algorithm reduces the boundary instances on the HTRU2 data set. The HTRU2 data set is a data set related to Physical sciences. According to the results, the reduction rates of the instances are respectively 4.34%, 7.96%, and 10.99% for k = 1, k = 3, and k = 5.

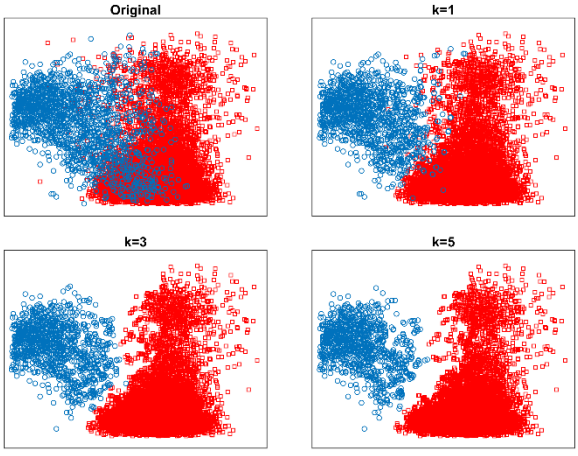


Figure 2. The illustration of reduction of the boundary instances on the HTRU2 data set (1st and 6th features), according to nearest neighbors (i.e., the value k)

Figure 3 shows the illustration in which the proposed algorithm reduces the boundary instances on the Human Activity data set. The Human Activity data set is a data set related to Health sciences. According to the results, the reduction rates of the instances are respectively 6.95%, 16.10%, and 21.09% for k = 1, k = 3, and k = 5.

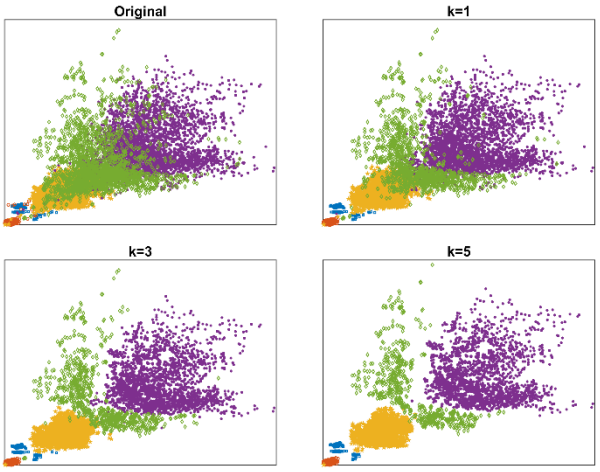


Figure 3. The illustration of reduction of the boundary instances on the human activity data set (4th and 6th features), according to nearest neighbors (i.e., the value k)

The illustration in which the proposed algorithm reduces the boundary instances on the madelon data set is shown in Figure 4. The madelon data set is an artificial data set. According to the results, the reduction rates of the instances are respectively 50.95%, 77.85%, and 90.25% for k = 1, k = 2, and k = 3.

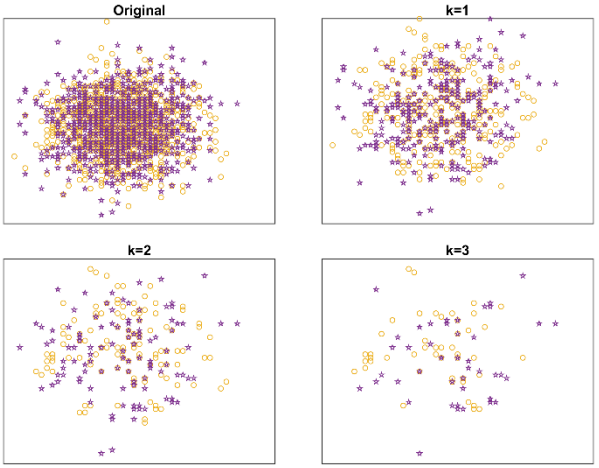


Figure 4. The illustration of reduction of the boundary instances on the madelon data set (1st and 4th features), according to nearest neighbors (i.e., the value k)

The comparative results of the algorithms in terms of accuracy rate are shown in Figure 5. BIRCH yields the highest rate with 87.01% classification accuracy after the baseline with 87.45%. Besides, BIRCH delivers the highest classification accuracy on two data sets (#1 and #5) in comparison to the other methods. The lowest and highest classification accuracy of the proposed method are 58.08% and 99.82%, respectively. BIRCH ranks third in terms of average reduction rate. This situation demonstrates that there is a trade-off between accuracy rate and reduction rate. According to Kruskal-Wallis test results, accuracy rates do not have mean ranks significantly different from each other.

Figure 6 shows the comparative results of the algorithms in terms of average reduction rate. According

to the results, DR.LSH has the highest average reduction rate. According to Kruskal-Wallis test results, reduction rates do not have mean ranks significantly different from each other. BIRCH ranks third. Wilson's ENN ranks last, as well. Accordingly, it is apparent that BIRCH is comparable to the hybrid, edition, or condensation methods. Although BIRCH and Wilson's ENN search nearest neighbors of an instance, BIRCH is faster than the well-known similar approaches. As is known to all, Wilson's ENN is faster than CNN and so on.

The comparative results of the algorithms in terms of average running time are shown in Figure 7. According to Kruskal-Wallis test results, running time of Wilson's ENN has mean ranks significantly different from other methods. It is obvious that the slowest method is Wilson's ENN. LSH-IS-S and LSH-IS-F are the fastest methods. BIRCH ranks fourth. Considering these three criteria, we would like to remark that BIRCH is more balanced compared to the other methods.

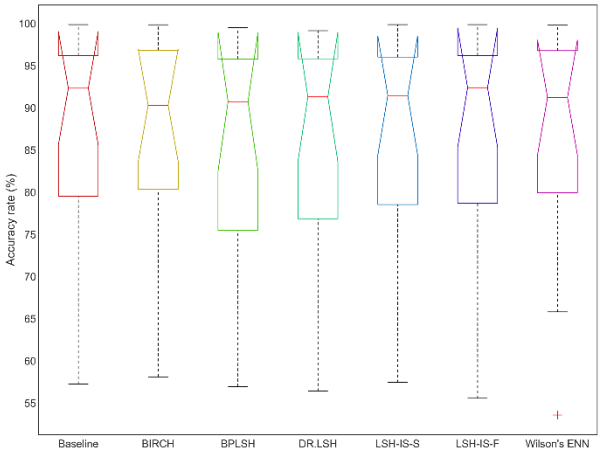


Figure 5. The comparative results of the algorithms in terms of accuracy rate (%)

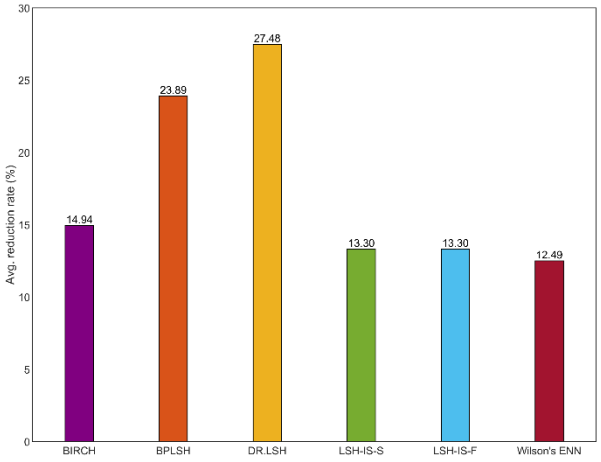


Figure 6. The comparative results of the algorithms in terms of reduction rate (%)

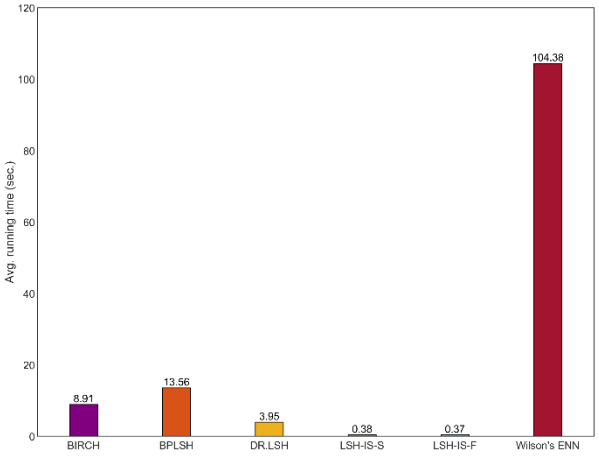


Figure 7. The comparative results of the algorithms in terms of running time

The trade-off between accuracy rate and reduction rate according to the number of neighbors is shown in Figure 8. Apart from the Boston Housing data set, while the accuracy rates on the other data sets decrease to an insignificant extent their reduction rates increase to a remarkable extent. We draw attention to the Boston housing data sets has 92 classes. Accordingly, as the k value increases on data sets where have the many classes the accuracy rate decreases quickly. BIRCH can attain a good trade-off between accuracy rate and reduction rate by adjusting the number of neighbors. Finally, we take the number of the nearest neighbors as 1 by default to obtain the maximum accuracy rate. In general, as the number of neighbors increases the accuracy rate decreases. Hence, it is more suitable to empirically determine the appropriate value of k. Thus, the accuracy rate also maintains as possible while the reduction rate increases. Further, the geometric average of k-value for maximum classification accuracy on the data sets is calculated as approximately 1.49. Hence, we set the k-value as 1.

Figure 9 shows the variation in the running time of BIRCH in terms of the number of neighbors. Considering the results, the running time of BIRCH does not rise excessively as the number of neighbors increases. This situation shows that BIRCH maintains a stable runtime performance. Thereby, the suitable accuracy rate-reduction rate balance can be obtained by increasing the k-value without performance loss. Consequently, the proposed method provides a satisfactory trade-off between classification accuracy and the reduction rate. The time complexity of the proposed method is log-linear, and it can achieve both high classification accuracy and reduction rates over many data sets by tuning the number of neighbors. Finally, the proposed method promises to remove more boundary instances by providing to reach high accuracy rates over data sets.

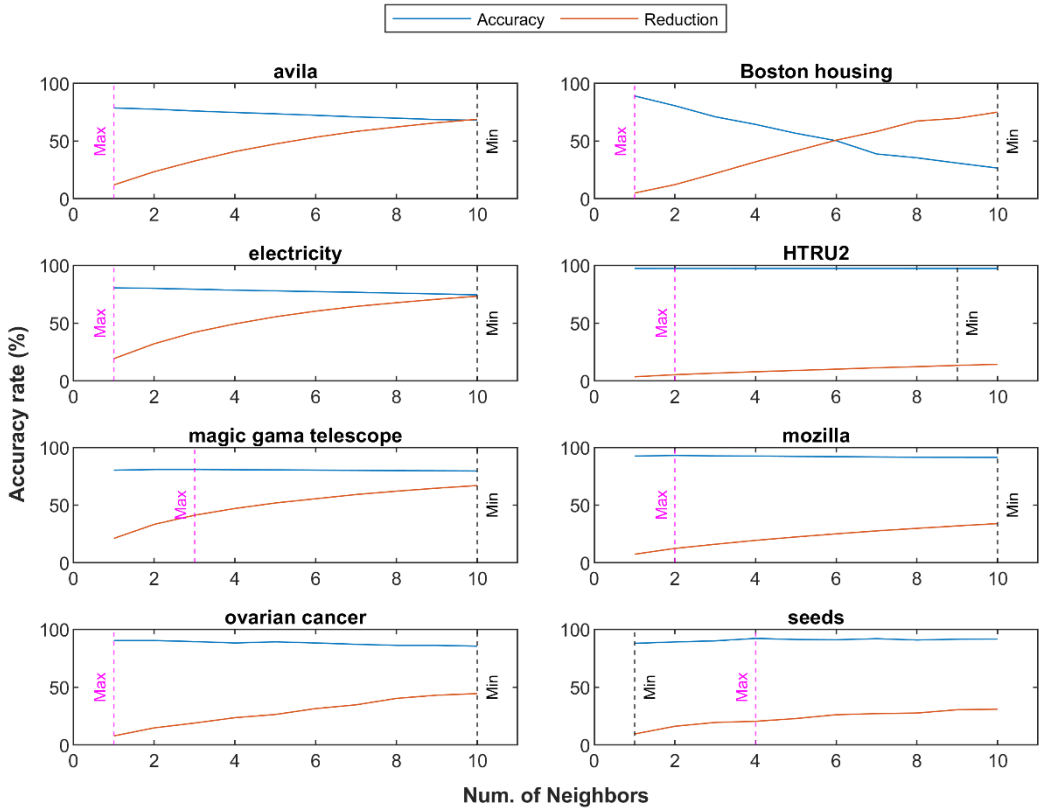


Figure 8. The trade-off between accuracy rate and reduction rate according to the number of neighbors

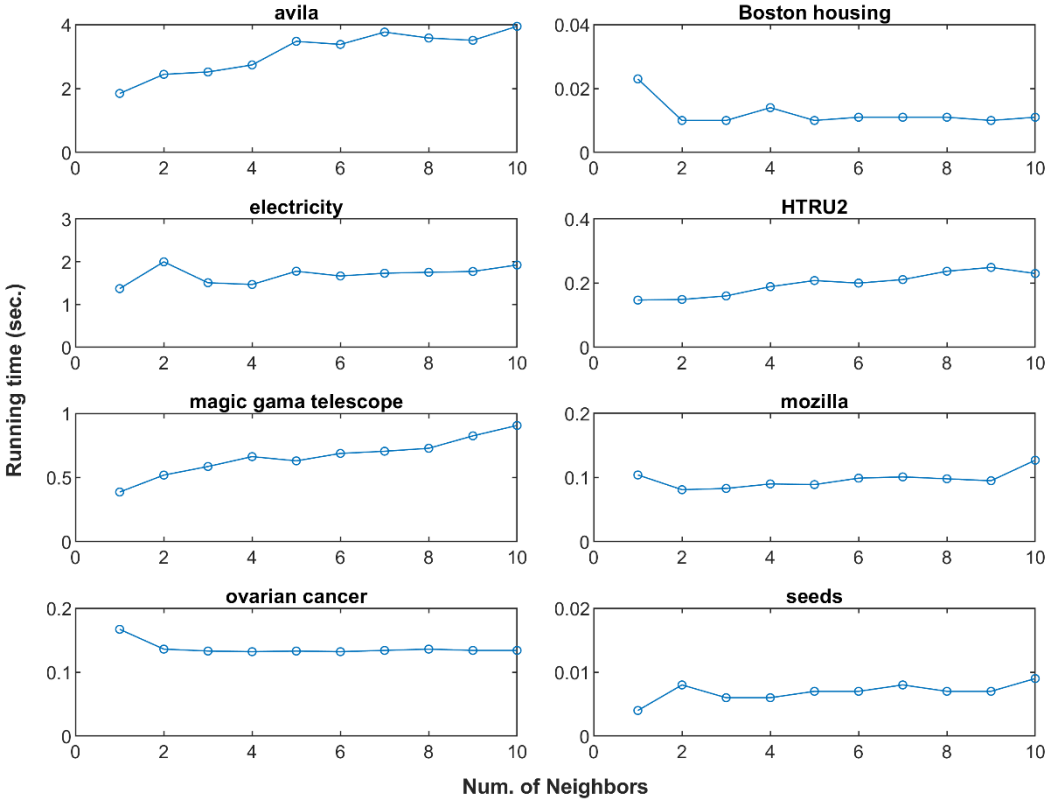


Figure 9. The variation in the running time of BIRCH in terms of the number of neighbors

5. Conclusions

In this paper, we propose a new instance selection algorithm called Border Instances Reduction using Classes Handily (BIRCH). We have tested the performance of BIRCH by using fifteen data sets from different domains and compared it with one traditional and four state-of-the-art instance selection methods in recent literature. Accordingly, BIRCH delivers a better trade-off between the accuracy rate and the reduction rate in comparison to the other methods. The time complexity of BIRCH is log-linear in the best case and log-quadratic in the worst case. BIRCH can acquire both high accuracy rates and reduction rates over lots of data sets by adjusting the number of neighbors. Principally, the reduction rate decreases as the accuracy rate increases. The proper tradeoff between accuracy rate, reduction rate, and speed-up is what is supposed to be focused on. BIRCH guarantees to discard more boundary instances by allowing to attain high classification accuracy over data sets. The future work of this study is to develop an unsupervised extension of BIRCH.

References

Akinyelu, A. A. and Adewumi, A. O. (2017) 'Improved Instance Selection Methods for Support Vector Machine Speed Optimization', *Security and Communication Networks*, 2017, pp. 1–11. doi: 10.1155/2017/6790975.

Akinyelu, A. A. and Ezugwu, A. E. (2019) 'Nature Inspired Instance Selection Techniques for Support Vector Machine Speed Optimization', *IEEE Access*, 7, pp. 154581–154599. doi: 10.1109/ACCESS.2019.2949238.

Alpaydin, E. (1997) 'Voting over Multiple Condensed Nearest Neighbors', *Artificial Intelligence Review*, 11(1/5), pp. 115–132. doi: 10.1023/A:1006563312922.

Arnaiz-González, Á. *et al.* (2016) 'Instance selection of linear complexity for big data', *Knowledge-Based Systems*, 107, pp. 83–95. doi: 10.1016/j.knosys.2016.05.056.

Aslani, M. and Seipel, S. (2020) 'A fast instance selection method for support vector machines in building extraction', *Applied Soft Computing*, 97, p. 106716. doi: 10.1016/j.asoc.2020.106716.

Aslani, M. and Seipel, S. (2021) 'Efficient and decision boundary aware instance selection for support vector machines', *Information Sciences*, 577, pp. 579–598. doi: 10.1016/j.ins.2021.07.015.

Cover, T. and Hart, P. (1967) 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory*, 13(1), pp. 21–27. doi: 10.1109/TIT.1967.1053964.

García-Pedrajas, N. (2011) 'Evolutionary computation for training set selection', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(6), pp. 512–523. doi: 10.1002/widm.44.

Garcia, S. *et al.* (2012) 'Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), pp. 417–435. doi: 10.1109/TPAMI.2011.142.

Hart, P. (1968) 'The condensed nearest neighbor rule (Corresp.)', *IEEE Transactions on Information Theory*, 14(3), pp. 515–516. doi: 10.1109/TIT.1968.1054155.

Liu, C. *et al.* (2017) 'An efficient instance selection algorithm to reconstruct training set for support vector machine', *Knowledge-Based Systems*, 116, pp. 58–73. doi: 10.1016/j.knosys.2016.10.031.

Olvera-López, J. A. *et al.* (2010) 'A review of instance selection methods', *Artificial Intelligence Review*, 34(2), pp. 133–143. doi: 10.1007/s10462-010-9165-y.

Rico-Juan, J. R., Valero-Mas, J. J. and Calvo-Zaragoza, J. (2019) 'Extensions to rank-based prototype selection in k-Nearest Neighbour classification', *Applied Soft Computing*, 85, p. 105803. doi: 10.1016/j.asoc.2019.105803.

Ruiz, I. L. and Gómez-Nieto, M. Á. (2020) 'Prototype Selection Method Based on the Rivality and Reliability Indexes for the Improvement of the Classification Models and External Predictions', *Journal of Chemical Information and Modeling*, 60(6), pp. 3009–3021. doi: 10.1021/acs.jcim.0c00176.

Sun, X. *et al.* (2019) 'Fast Data Reduction With Granulation-Based Instances Importance Labeling', *IEEE Access*, 7, pp. 33587–33597. doi: 10.1109/ACCESS.2018.2889122.

Susheela Devi, V. and Murty, M. N. (2002) 'An incremental prototype set building technique', *Pattern Recognition*, 35(2), pp. 505–513. doi: 10.1016/S0031-3203(00)00184-9.

Wang, Z., Tsai, C.-F. and Lin, W.-C. (2021) 'Data cleaning issues in class imbalanced datasets: instance selection and missing values imputation for one-class classifiers', *Data Technologies and Applications*, ahead-of-p(ahead-of-print). doi: 10.1108/DTA-01-2021-0027.

Wilson, D. L. (1972) 'Asymptotic Properties of Nearest Neighbor Rules Using Edited Data', *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3), pp. 408–421. doi: 10.1109/TSMC.1972.4309137.

Wilson, D. R. and Martinez, T. R. (2000) 'Reduction techniques for instance-based learning algorithms', *Machine Learning*, 38, pp. 257–286.

Yang, L. *et al.* (2019) 'Constraint nearest neighbor for instance reduction', *Soft Computing*, 23(24), pp. 13235–13245. doi: 10.1007/s00500-019-03865-z.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 127-136
DOI: 10.38016/jista.1082310

Covid-19 Hastalarının Ölüm Oranlarının ve Yüksek Ölüm Riskine Sahip Hastaların Belirlenmesi için Temel Bileşen Analizinin Kullanılması

Ebru Efeoğlu[1*]

[1] Dumlupınar Üniversitesi, Yazılım Mühendisliği, Kütahya, Türkiye

ebru.efeoglu@dpu.edu.tr

Öz

Covid-19 virüsü 2019 yılında ortaya çıktı ve kısa bir sürede tüm dünyaya yayıldı. Milyonlarca insanın enfekte olmasına ve yüz binlerce insanın ölümüne neden oldu. Her geçen gün vaka sayısı artmakta ve virüsün yeni varyantlar meydana gelmektedir. Bu hastalığa sahip kişileri tespit etmek için Polimeraz Zincir Reaksiyonu (PCR) testleri uygulanmaktadır. Hastalığı tespit edilen kişilerin durumlarının incelenmesi yoğun bakım ve ölüm oranlarının önceden tespiti oldukça önemlidir. Bu çalışmada Covid-19 hastalarından ölüm oranlarının tespitinde özellik çıkarımı yöntemi olarak Temel Bileşen Analizi (PCA) kullanılmış ve yöntemin başarılı sonuçları en popüler makine öğrenmesi teknikleri ile gösterilmiştir. Çalışmada kullanılan makine öğrenmesi teknikleri K-En Yakın Komşu (KNN), Doğrusal Ayrımcılık Analizi (LDA), Extra Ağaçlar, Random Tree, Rep Tree ve Naive Bayes algoritmalarıdır. Bu tekniklerin performans değerlendirmesinde Doğruluk, Kesinlik, Duyarlılık, Rms, F-skoru değerleri hesaplanmıştır. Ayrıca ROC Eğrileri ve Karışıklık matrisleri incelenerek sonuçlar karşılaştırılmıştır. Sonuç olarak, en iyi performansın Temel bileşenler analizi uygulandıktan sonra Doğrusal Ayrım Analizi (PCA+LDA) kullanımı ile elde edildiği görülmüştür. PCA+LDA uygulaması ile %96,39 Doğruluk oranı elde edilmiştir. Makalede ayrıca özellik çıkarımının kullanılmasıyla Covid-19 virüsünden Zatürre, Şeker, KOAH ve Astım hastalarının, hamile, yaşlı ve entrube insanların daha çok etkilendiği ve ölüm riskinin daha yüksek olduğu ortaya çıkmıştır. Virüsün varyantlarının ölümcüllüğünün incelenmesi, riskli hastaların tedavisi, ölüm riski bulunan hastaların izolasyonu için gereken önlemlerin alınması ve hastane kapasite planlamasının iyileştirilmesi açısından bu çalışma önem arz etmektedir.

Anahtar kelimeler: Covid-19, Performans analizi, Sınıflandırma algoritmaları, Temel Bileşen Analizi.

Using Principal Component Analysis to Identify Mortality Rates of Covid-19 Patients and Patients at High Risk of Death

Abstract

The Covid-19 virus emerged in 2019 and spread all over the world in a short time. It caused millions of people to be infected and hundreds of thousands to die. The number of cases is increasing day by day and new variants of the virus are emerging. Polymerase Chain Reaction (PCR) tests are used to detect people with this disease. It is very important to examine the conditions of the people with the disease and to determine the intensive care and mortality rates in advance. In this study, Principal Component Analysis (PCA) was used as a feature extraction method to determine mortality rates from Covid-19 patients, and the successful results of the method were demonstrated with the most popular machine learning techniques. Machine learning techniques used in the study are K-Nearest Neighbor (KNN), Linear Discrimination Analysis (LDA), Extra Trees, Random Tree, Rep Tree and Naive Bayes algorithms. In the performance evaluation of these techniques, Accuracy, Precision, Sensitivity, Rms, F-score values were calculated. In addition, ROC Curves and Confusion matrices were examined and the results were compared. As a result, it was seen that the best performance was obtained with the use of Linear Discrimination Analysis (PCA+LDA) after applying Principal component analysis. With the PCA+LDA application, an accuracy rate of 96.39% was obtained. In the article, it has also been revealed that Pneumonia, Diabetes, COPD and Asthma patients, Pregnant, Elderly and Intubated people are more affected and the risk of death is higher from the Covid-19 virus by using feature extraction. This study is important in terms of examining the lethality of virus variants, taking the necessary precautions for the treatment of risky patients isolation of patients at risk of death, and improving hospital capacity planning.

Keywords: Covid-19, Performance Analysis, Classifications, Principal Component Analysis.

* Sorumlu yazar.
E-posta adresi: ebru.efeoglu@dpu.edu.tr

Alındı : 3 Mart 2022
Revizyon : 13 Mayıs 2022
Kabul : 28 Mayıs 2022

1. Giriş (Introduction)

Çin'de ortaya çıkan koronavirüs hastalığı (Covid-19) dünya çapında bir pandemi haline geldi (Velavan & Meyer, 2020). Bu pandeminin en başından itibaren, vaka tespiti konusu her zaman bilimsel ve kamusal söylemin merkezinde yer aldı. Salgının kontrol altına alınabilmesi için Popülasyonda gerçekten kaç enfeksiyonun bulunduğunu bilmek büyük önem taşımaktadır. Ancak hastalığa dair belirti göstermeyip taşıyıcı olarak hastalığı başkasına bulaştıran, koronavirüs testi pozitif çıkan asemptomatik bireyler ve ilk önce belirti göstermeyip ilerleyen süreçlerde hastalık belirtisi gösteren presemptomatik bireylerin, özellikle genç popülasyonda bir pandemik hastalığın yayılmasında önemli bir etkiye sahiptir (Stella, Martínez, Bauso, & Colaneri, 2020). Hastalığın yayılmasının dinamiklerini tahmin etmek için epidemiyolojik modeller (de León, Pérez, & Avila-Vales, 2020) (Chinazzi et al., 2020), enfekte hastaları ve yeni vakaları izlemek için mobil cihaz uygulamaları geliştirilmiştir (Zens, Brammertz, Herpich, Südkamp, & Hinterseer, 2020) (Drew et al., 2020).

Ayrıca bu vakalar gerçek ölüm oranını ortaya çıkarma sorunuyla sıkı sıkıya iç içedir. Covid-19 enfeksiyonlarının gerçek sayılarını tahmin etme sorunu, ölüm oranıyla ilişkili olduğu tamamen istatistiksel bir bakış açısıyla da tartışılmıştır (Manski & Molinari, 2021). Bu bağlamda, farklı ulusal eksik raporlama oranları karşılaştırılmış (Rahmandad, Lim, & Sterman, 2020; Jagodnik, Ray, Giorgi, & Lachmann, 2020) ve enfeksiyon ölüm oranının değerlendirilmesine ilişkin genel bir tartışma ve anket yapılmıştır (Levin, Cochran, & Walsh, 2020).

Son zamanlarda Covid-19 hastalığı ile ilgili farklı amaçlar için çeşitli makine öğrenimi yaygın olarak uygulanmıştır (Albahri et al., 2020). Örneğin Röntgen görüntülerinden, (Kassania, Kassanib, Wesolowskic, Schneidera, & Detersa, 2021), tam kan sayımından makine öğrenimi algoritmaları kullanılarak hastalık teşhisi yapılmıştır (Akhtar et al., 2021). Çin'de Covid-19 vaka ölüm oranının erken tahmini için veriye dayalı bir analiz yapılmıştır (Yang et al., 2020). Makine öğrenimi yaklaşımını kullanarak COVİD-19 bulaşması ve ölümle ilişkili yeni faktörlerin belirlenmesi ile ilgili çalışmalar yapılmıştır (M. Li et al., 2021). Hastanede yatan Covid-19 hastalarının ölümcül risk tahmini için klinik ve inflamatuar özelliklere dayalı makine öğrenimi kullanılmış (Guan et al., 2021) (Quiroz-Juárez, Torres-Gómez, Hoyo-Ulloa, León-Montiel, & U'Ren, 2021) ve risk faktörlerini değerlendirilmiştir (Gansevoort & Hilbrands, 2020), (Parra-Bracamonte, Lopez-Villalobos, & Parra-Bracamonte, 2020). Ayrıca pozitif ve negatif Covid-19 vakaları için epidemiyoloji etiketli veri seti kullanılarak lojistik regresyon, karar ağacı, destek vektör makinesi, Naive Bayes ve yapay sinir ağları içeren öğrenme algoritmaları ile Covid-19 enfeksiyonu için denetimli makine öğrenimi modelleri geliştirilmiştir (Muhammad et al., 2021). Hipertansiyon, diyabet, koroner kalp hastalığı, kronik obstrüktif akciğer hastalığı, kronik böbrek hastalığı (Escobedo-de la Peña et al., 2021), obezite (Bello-Chavolla et al., 2020) ve Hamilelik durumunun (Ríos-Silva, Murillo-Zamora, Mendoza-Cano, Trujillo, & Huerta, 2020) Covid-19 mortalitesi üzerindeki etkisi incelenmiştir.

Ölüm riski yüksek hastaların tespiti onlara gereken özenin gösterilmesi ve önlemlerin alınması ölüm oranlarının düşmesine önemli bir katkı sağlayacaktır. Ayrıca bu hastaların tespiti hastanelerdeki kaynakları ve kapasiteleri yönetmek (Singh et al., 2021), (Zawiah et al., 2020) hastalara zamanında tedavi sağlamak (Nemati, Ansary, & Nemati, 2020) için oldukça önemlidir.

Çalışmanın amacı, Covid-19 hastalarının ölüm oranı tespitini en yüksek başarı oranı ile en kısa işlem süresine sahip bir yöntem önermek ve ölüm riski yüksek olan hastaların hangi özelliklere sahip olduklarının tespiti ile hem bu hastalara gereken izolasyon sağlanarak ölüm oranlarının düşürülmesine hem de hastane kapasite planlamasının iyileştirilmesine yardımcı olmaktır. Bu hastaların tespiti için özellik seçiminde sıklıkla kullanılan ve başarılı bir algoritma olan PCA yöntemi tercih edilmiştir. Bu yöntem sayesinde hem yüksek ölüm riskine sahip hastaların özellikleri tespit edilebilecek hem de daha kısa sürede ve daha yüksek başarıyla ölüm oranları tespiti yapılabilecektir. PCA yöntemine ek olarak ölüm oranları tespitinde sınıflandırma algoritmalarından yararlanılmıştır. En başarılı algoritmanın belirlenebilmesi için Algoritmaların performans analizi yapılmıştır. Analizde PCA yöntemi kullanılmadan ve PCA yöntemi kullanıldıktan sonra veri setinden ölüm oranı tahmin etme başarıları dikkate alınmıştır. Analiz sonucunda elde edilen sonuçlar karşılaştırılmıştır.

2. Materyal ve Yöntem (Material and Method)

2.1. Veri seti (Dataset)

Makalede kullanılan veri seti, Kaggle sitesinde bulunan ve "COVID-19 Mexico Patient Health Dataset" isimli veri setidir. Bu veri setindeki Covid-19 hastalığına ilişkin veriler Meksika hükümeti tarafından 15 Ocak 2020 ile 3 Mayıs 2020 tarihleri arasında kaydedilmiştir. Bu veri seti daha önce ölüm oranı tespiti için (Yavuz & Dudak, 2020) makalesinde kullanıldı. Bu çalışmada hem ölüm oranları tahmin edilmiş hem de hangi özelliklerin hastada ölüm riskinin artmasında daha etkili olduğu incelenmiştir.

Veri seti 19 özellikten oluşan 95805 örnekten oluşmaktadır. (Kaggle.com, 2020). Veri setindeki özellikler hastanın cinsiyeti, hastalığın tipi, entübe olma durumu, zatürre olma durumu, hastanın yaşı, hamilelik durumu, diyabet olma durumu, Kronik Obstrüktif Akciğer Hastalığı olması durumu (KOAH), astım olma durumu, Bağışıklık sistemi baskılanması durumu (İmmünosupresyon), Hipertansiyon durumu, diğer

hastalık durumu, Kardiyovasküler durum, Obezite olma durumu, Kronik böbrek yetmezliği durumu (Chronic_kidneyfailure), sigara içme durumu, başka bir vaka durumu, yoğun bakım durumu (icu), ölüm tarihi bulunmaktadır. Çalışmada ölüm tarihi özelliğinin yerine ölü tarihi yazan hastanın öldüğü, tarih yoksa hastanın yaşadığı kabul edilmiştir. Veri seti sayısal değerlerden oluşmaktadır. Hastalarda sayılan özelliklerin bulunması durumu 1, bulunmama durumu 2 ile bu özelliğe ait veri yoksa 99 ile gösterilmiştir. 98/97 değeri ise bu özellik için kişinin uygun olmadığını göstermektedir.

2.2. Sınıflandırma algoritmaları (Classification algorithms)

K-En Yakın Komşu Algoritması: Bu algoritmanın amacı, bir veri setinde en yakın komşuları bulmaktır. Bu komşuları bulmak için farklı mesafe metrikleri kullanır. Algoritmanın başarısı bu mesafe metrikleri ve k ile gösterilen komşu sayısına göre değişkenlik gösterir (B. Li, Yu, & Lu, 2003; Xia, Xiong, Luo, Dong, & Zhang, 2015).

Naive Bayes : Bayes teoremine dayanır. Sınıfı belli olan örnek verileri kullanarak yeni verinin sınıflara ait olma olasılığını hesaplar. Bulunan değerlerden en yüksek olasılık değerine sahip sınıf, örneğinin sınıfıdır. (Bermejo, Gámez, & Puerta, 2011).

Doğrusal Ayrımcılık Analizi (LDA): Bu algoritma ilk olarak 1936 yılında ikili sınıflandırmalar için R. A. Fisher tarafınca geliştirilmiştir. Daha sonra C. R. Rao tarafından ikiden fazla sınıflandırmalar için formülüze edilmiştir. Diskriminant analizinde ilk olarak sınıfları birbirinden ayırmayı sağlayan diskriminant fonksiyonları bulunur. Daha sonra bulunan fonksiyonlar kullanılarak yeni örneğin hangi sınıfa dahil edilmesi gerektiğine karar verilir (Ünsal, Bileşenler, _Faktür, & Mali, 1996).

Random tree: Her düğümde belirli sayıda özellik kullanılır ve ağaç oluşturulur. Seçilen bu özellikler rasgele seçilmiş özelliklerdir. Budama işlemi yapılmaz. Ayrıca, tutulan veri kümesine dayalı olarak sınıf olasılıklarının tahmin edilmesini sağlayan bir seçeneğe de sahiptir.

Rep Tree algoritması: İlk olarak Quinlan tarafından (Quinlan, 1999) önerildi. Eğitim örneklerinde gürültünün etkilerini azaltmak istenir ve bunun için budama işlemi yapılarak bir karar ağacı oluşturulur. Hızlı makine öğrenmesi algoritmalarındandır (J. Li, Zhang, Lu, & Yan, 2008). Rep Tree algoritması varyanstan kaynaklanan hatayı en aza indirme ilkesine ve entropi (Amasyali & Ersoy, 2009) ile bilgi edinme ilkesine dayanır ve sadece sayısal verilerle çalışır.

Ekstra Ağaçlar algoritması: Klasik yukarıdan aşağıya prosedüre göre budanmamış bir karar ağaçları topluluğu oluşturur. Diğer ağaç tabanlı topluluk yöntemleriyle arasındaki iki temel fark, kesme noktalarını tamamen rastgele seçmeleri, düğümleri ayırmaları ve ağaçları büyütmek için tüm öğrenme örneğini kullanmalarıdır (Freund & Mason, 1999).

Hoefding tree: Hoeffding ağacı, veri dağılımının zaman içinde değişmediğini varsayan büyük veri akışı için artan bir karar ağacı öğrenicisidir. Hoeffding sınırına dayanan bir karar ağacı aşamalı olarak büyür (Zhang, Ding, & Wang, 2011).

Random Forest: Torbalama yöntemine rastgelelik özelliği eklenerek oluşturulan bir algoritmadır (Breiman 2001). Regresyon ve sınıflandırma yöntemi olarak kullanılabilir.

2.3. Özellik Seçimi ve PCA Yöntemi (Feature Selection and PCA Method)

Özellik seçimi kısaca veri setini temsil edebilecek en iyi altkümenin seçilmesidir. Bu işlem, çözülmek istenen problem için en faydalı ve en önemli özelliklerin seçilmesiyle veri kümesindeki özellik sayısının azaltılmasıdır. Birçok özellik seçim yöntemi bulunmaktadır. Bir örneği tekrar tekrar örnekleyerek ve aynı ve farklı sınıfın en yakın örneği için verilen özniteliğin değerini göz önünde bulundurarak bir özniteliğin değerini değerlendiren Relief yöntemi (Kira & Rendell, 1992), Sınıfa göre simetrik belirsizliği ölçerek bir özelliğin değerini değerlendiren simetrik belirsizlik katsayısı (Novaković, Strbac, & Bulatović, 2011). Sınıfa göre kazanç oranını ölçerek bir özelliğin değerini değerlendiren Kazanç oranı yöntemi, Bir öğrenme şeması kullanarak öznitelik kümelerini değerlendiren Sarmalayıcı Alt Kümes (Wrapper Subset Eval) yöntemi. (Kohavi & John, 1997).Veri setinde negatif olmayan bir şekilde lineer olarak temsil edilmesini sağlayan ve Negatif olmayan sinyallerin bulunduğu alanlarda başarılı olan Negatif olmayan matris yöntemi bu yöntemler arasında yer almaktadırlar.

PCA yöntemi, veri içindeki etkin özellikleri tespit ederek verinin boyutunu azaltır (Abdi & Williams). Oldukça popüler ortogonal linear dönüşümdür. Genel olarak PCA yöntemi verideki maksimum varyansa sahip verinin az boyuta sahip uzayda gösterimi şeklinde ifade edilebilir.

PCA dönüşümü denklem (1) de gösterildiği gibi yapılır.

$$\mu^T = X^T W \quad (1)$$

Burada,

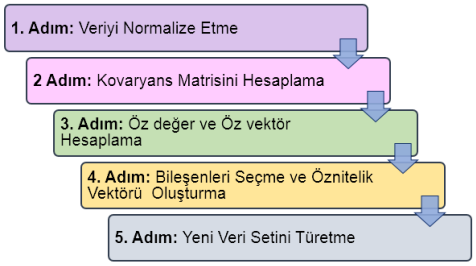
X ortogonal matrisi, μ^T linear dönüşümünü, W ise özvektörleri ifade eder.

Verinin PCA yöntemi ile ayrılmış şekli denklem (2) de verildiği gibidir (Maglaveras, Stamkopoulos, Diamantaras, Pappas, & Strintzis, 1998) .

$$X_i = \sum_{j=1}^{p} w_{ij} Q_j \quad (2)$$

Q_j, j=1, .., p, faktör veya özellik adı verilen gizli gösterim parametreleridir.

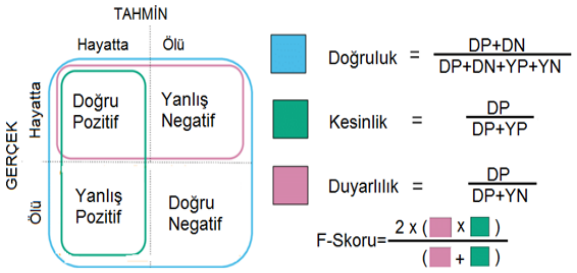
PCA yönteminin uygulanması için izlenmesi gereken adımlar Şekil 2' de verilmiştir.



Şekil 1. PCA yönteminin işlem adımları (Process steps of the PCA method)

2.4. Performans metrikleri (Performance metrics)

Algoritmaların sınıflandırma performansı hakkında en fazla bilgi içeren metrik karışıklık matrisi olduğundan performans karşılaştırmada en sık kullanılan metriktir. Karışıklık matrisinde 4 farklı değer bulunmaktadır. Gerçek durum pozitifken test sonucunun pozitif olması durumunda DP (Doğru Pozitif) değeri, gerçek durum negatifken test sonucunun pozitif olması durumunda YP (Yanlış Pozitif) değeri, gerçek durum negatifken test sonucunun negatif olması durumunda DN (Doğru Negatif) değeri ve gerçek durum pozitifken test sonucunun negatif olması durumunda ise YN (Yanlış Negatif) hesaplanır. Bu değerlerin karışıklık matrisinde gösterimi ve bu matristen hesaplanan diğer metrikler Şekil 2' de verilmiştir.

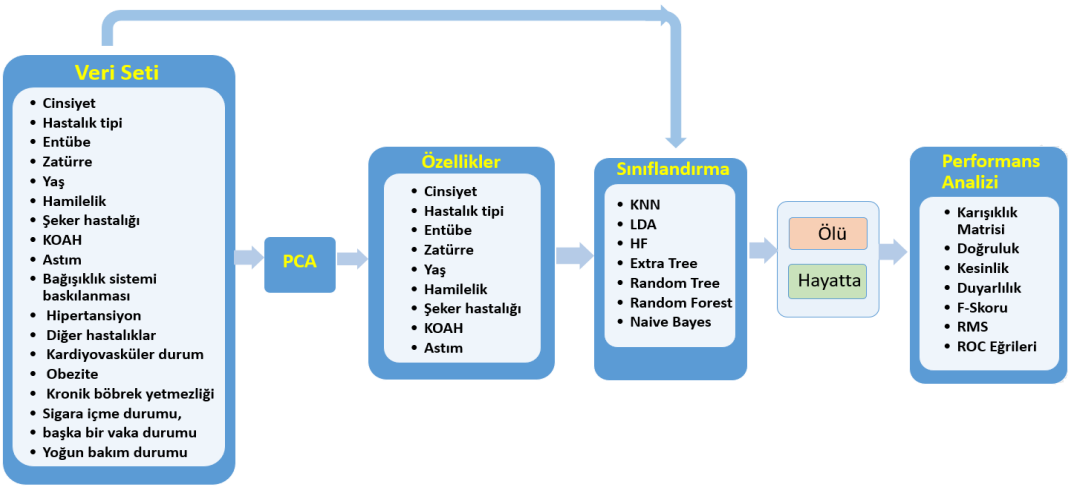


Şekil 2. Karışıklık matrisi ve diğer performans metrikleri (Confusion matrix and other performance metrics)

Covid-19 hastalarından ölüm oranlarının tespiti için yapılan çalışmayı özetleyen akış diyagramı Şekil 3'de verilmiştir. Akış diyagramından da anlaşılacağı gibi ölüm oranı tahmini için önce veri setine PCA yöntemi kullanılmadan bir sınıflandırma işlemi uygulanmış daha sonra veri setine PCA yöntemi uygulanarak veri setinin boyutu azaltılmış ve özellik seçimi yapılmıştır. Seçilen özellikler kullanılarak bir sınıflandırma yapılmıştır. Daha sonra PCA kullanılmadan ve PCA kullanılarak yapılan sınıflandırma işlemlerine performans analizi uygulanmıştır. Yapılan performans analizinde kullanılan performans metrikleri akış diyagramında belirtilmiştir. Son olarak en iyi performansa sahip teknik belirlenmiştir.

PCA yöntemi kullanılmadan ve PCA yöntemi kullanıldıktan sonra yapılan sınıflandırmalardan elde edilen karışıklık matrisi Şekil 4'te verilmiştir. Şekilde mavi ile gösterilen yerler algoritmaların doğru tahmin ettiği örnek sayısını ifade etmektedir. Görüldüğü gibi PCA yönteminin kullanılması ile algoritmaların doğru tahmin ettiği örnek sayılarının arttırmıştır. Örneğin KNN algoritması ile yapılan sınıflandırmada algoritma 91381 örneği doğru tahmin etmiştir. Bu sayı PCA yöntemi kullanılması ile 92202'e yükselmiştir.

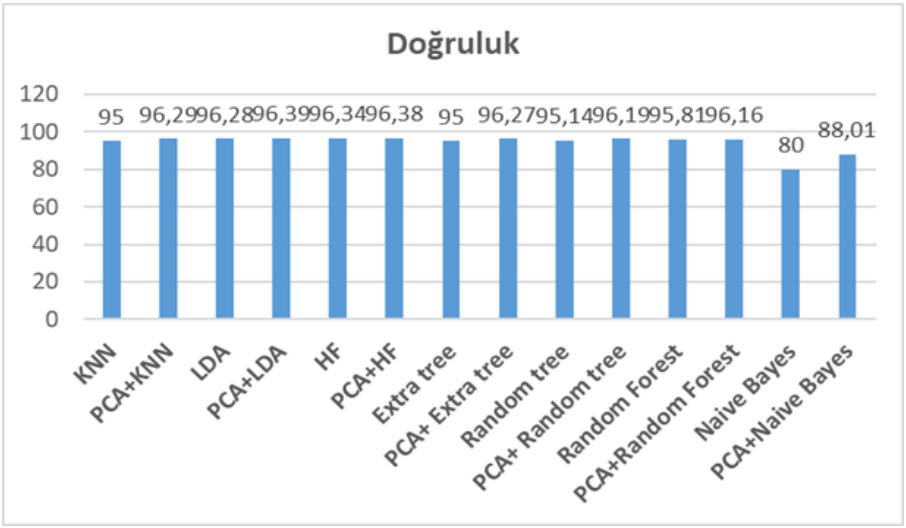
Algoritmaların yüzde cinsinden doğruluk değerlerini gösteren grafik Şekil 5' de diğer performans metrikleri de Tablo 2'de verilmiştir. PCA yönteminin algoritmaların veri seti üzerinde model oluşturulma süreleri ve toplam işlem süreleri karşılaştırmalı olarak gösterilmektedir. Tablodan PCA yönteminin işlem sürelerini azalttığı görülmektedir. Tüm işlemler Intel(R) Core(TM) i7-4600U CPU@ 2.10 Ghz işlemci, 8 GB Ram özelliklerine sahip Windows 10 işletim sistemi kurulu bir bilgisayar ile gerçekleştirilmiştir.



Şekil 3. Çalışmanın akış diyagramı (Flow chart of the study)

		Tahmin edilen Sınıf					*Tahmin edilen Sınıf*	
K-EYK	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ KNN**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	*90854*	*1519*		*Hayatta*	91991	382
		Ölü	*2905*	*527*		*Ölü*	3221	211
LDA	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ LDA**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	92030	343		*Hayatta*	92338	35
		Ölü	3215	217		*Ölü*	3426	6
HF	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ HF**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	92142	231		*Hayatta*	92291	82
		Ölü	3275	157		*Ölü*	3424	8
Extra tree	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ Extra Tree**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	90325	2048		*Hayatta*	91927	446
		Ölü	2703	729		*Ölü*	3200	232
Random Tree	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ Random Tree**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	90441	1932		*Hayatta*	91941	432
		Ölü	2724	708		*Ölü*	3210	222
Random Forest	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ Random Forest**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	91225	1148		*Hayatta*	91879	494
		Ölü	2858	574		*Ölü*	3176	256
Naive Bayes	*Gerçek Sınıf*		*Hayatta*	*Ölü*	**PCA+ Naive Bayes**	*Gerçek Sınıf*	*Hayatta*	*Ölü*
		Hayatta	74460	17913		*Hayatta*	82279	10094
		Ölü	444	2988		*Ölü*	1387	2045

Şekil 4. Karışıklık matrisi (Confusion matrix)

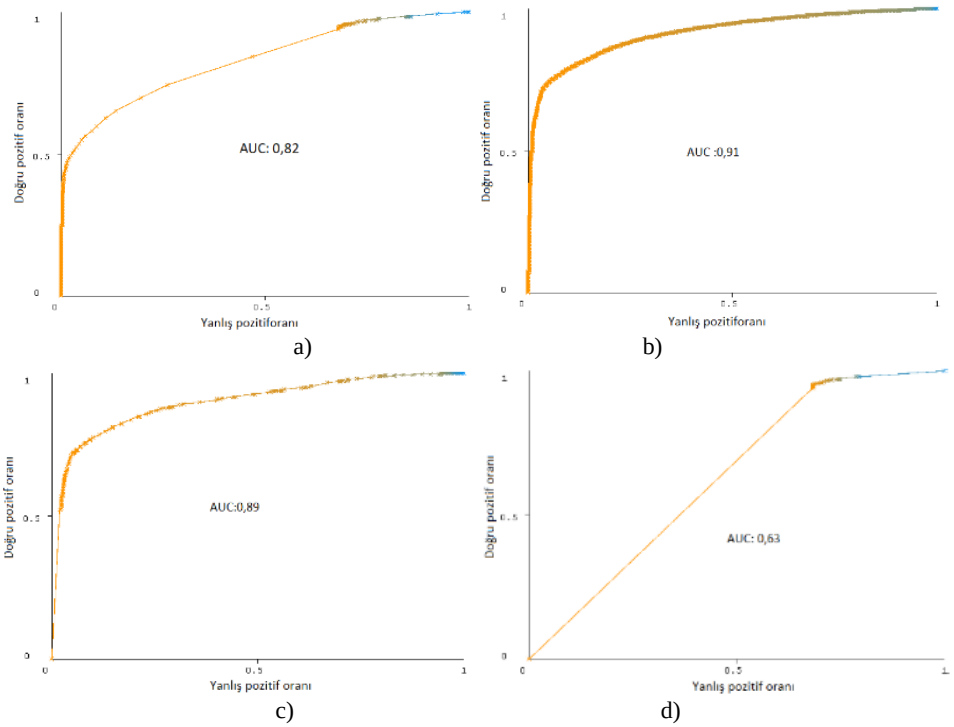


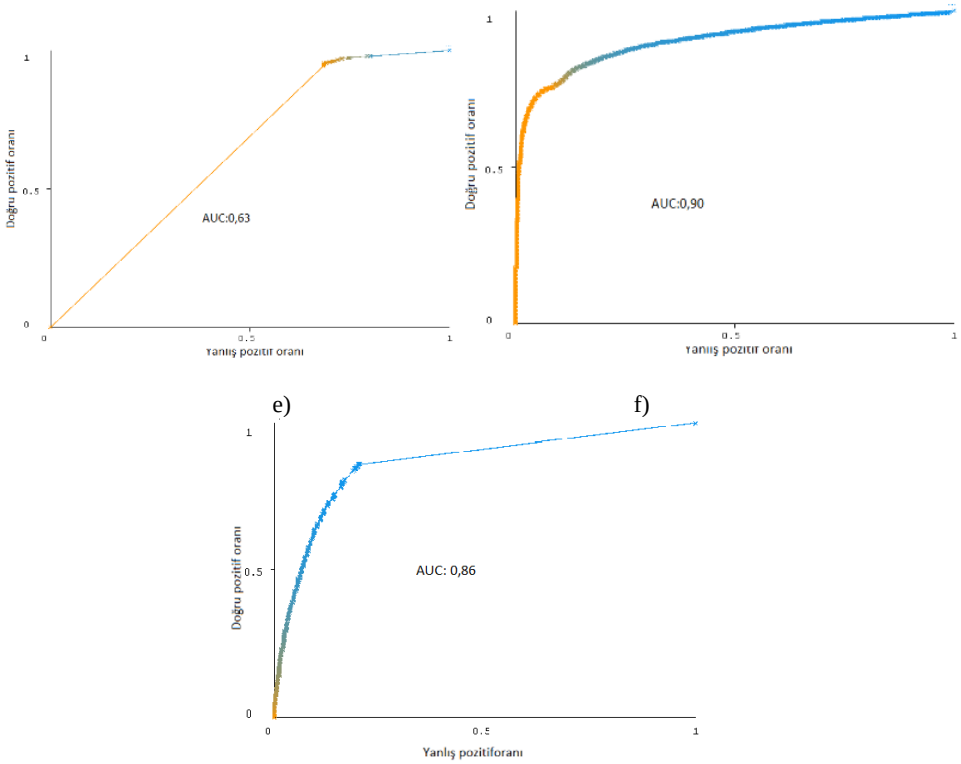
Şekil 5. Doğruluk değerleri (Accuracy values)

Tablo 2. Performans metrikleri (Performance metrics)

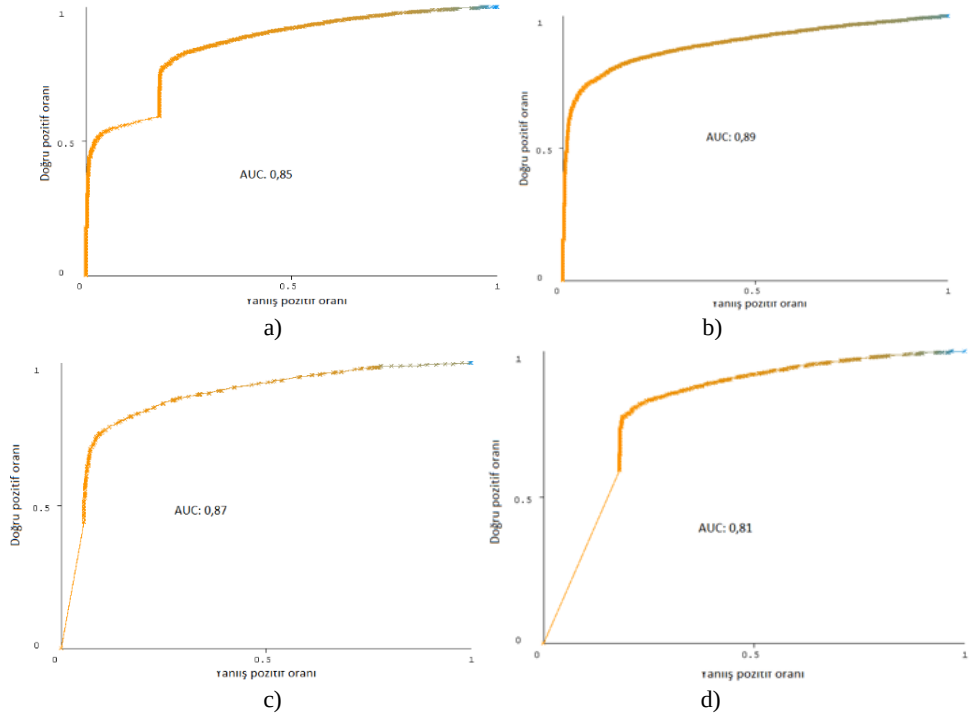
Yöntem	Kesinlik	Duyarlılık	F-Skoru	Rms	Özellik sayısı	Doğru sınıflandırılan örnek sayısı	Modelin Oluşturulma Süresi (sn)	Toplam İşlem Süresi (sn)
KNN	0,94	0,95	0,94	0,21	19	91381	0,11	1440
PCA+ KNN	0,94	0,96	0,94	0,18	9	92202	0,02	1320
LDA	0,94	0,96	0,95	0,17	19	92247	0,22	5
PCA+LDA	0,93	0,96	0,94	0,17	9	92344	0,12	3
HF	0,94	0,96	0,94	0,17	19	92299	0,94	10
PCA+HF	0.93	0.96	0.94	0,17	9	92299	0,71	6
Extra tree	0,94	0,95	0,94	0,22	19	91054	0,39	9
PCA+ Extra tree	0,94	0,96	0,94	0,18	9	92159	0,33	8
Random tree	0,94	0,95	0,94	0,22	19	91149	1,86	18
PCA+ Random tree	0,94	0,96	0,94	0,18	9	92163	1,16	13
Random Forest	0,94	0,95	0,95	0,18	19	91799	89,7	1200
PCA+Random Forest	0,94	0,96	0,95	0,17	9	92135	69,96	840
Naive Bayes	0,96	0,80	0,86	0,37	19	77448	0,3	5
PCA+Naive Bayes	0,95	0,88	0,91	0,27	9	86369	0,23	4

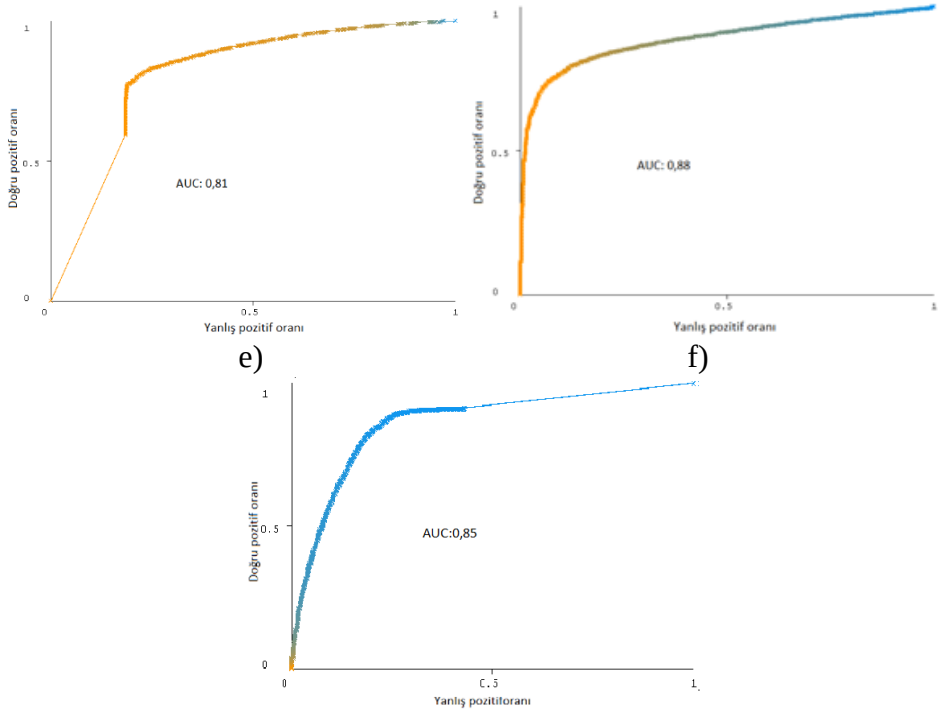
Temel bileşen analizinin uygulanması ile bütün algoritmaların işlem süreleri azalmış ve doğruluk oranları artmıştır. En düşük doğruluk oranı Naive bayes algoritmasına aittir. Naive bayes algoritmasının doğruluk oranı %80 dir. PCA uygulaması ile bu oran %88,01'e çıkmıştır. En yüksek doğruluk oranı ise PCA ve LDA algoritması uygulanması ile elde edilmiştir. Bu metrikler dışında kullanılan diğer metrik ROC eğrileridir. Yatay eksen yanlış pozitif oranını düşey eksen ise doğru pozitif oranını gösterecek şekilde çizilen eğrilere ROC eğrileri denir. Bu eğrilerin altında kalan alan da algoritmanın performansının değerlendirilmesinde kullanılır. Bu alan kısaca AUC ile gösterilir. Algoritmaların PCA yöntemi kullanılmadan ve PCA yöntemi kullanılması ile ölüm oranı tahminlerinden elde edilen ROC eğrileri Şekil 6 ve Şekil 7'te verilmiştir.





Şekil 6. ROC eğrileri a) KNN b) LDA c) HF d) Extra tree e) Random tree f) Naivebayes g)Random Forest





Şekil 7. ROC eğrileri a) PCA+KNN b) PCA+LDA c) PCA+HF d)PCA+Extra tree e) PCA+Random tree f) PCA+Naive Bayes g) PCA+Random Forest

İyi bir sınıflandırma için performans metriklerinin 1'e yakın değerler almaları gereklidir. Duyarlılık, Pozitif durumların ne kadar başarılı tahmin edildiğini gösteren bir metriktir. Algoritmaların genel olarak duyarlılık değerleri 0,95 ve üstü olması nedeniyle ölüm oranı tahminlerinde başarılı oldukları söylenebilir. Kesinlik, algoritmanın pozitif olarak tahmin ettiği değerlerin gerçekte kaç tanesinin pozitif olduğunu gösteren bir metriktir. F-skoru daha çok dengesiz veri setlerinin sınıflandırılmasında kullanılır. Yapılan tahminlerdeki hata oranını belirten RMS değeri de PCA yöntemi kullanılması ile azalmıştır.

4. Sonuçlar (Conclusions)

Bütün dünyayı etkileyen ve milyonlarca insanın ölümüne sebep olan Covid 19 virüsünden kaynaklanan ölüm oranlarının tahmin edilmesi için çalışmada PCA yöntemi ve sınıflandırma algoritmalarından yararlanılmıştır. Başarı oranını yükseltmek ve hangi özelliklere sahip olan hastanın ölüm riskinin yüksek olduğunun tespitinin yapılabilmesi için PCA yöntemi kullanılmıştır. Veri setinden PCA yöntemi kullanılmadan ve PCA yöntemi kullanıldıktan sonra yapılan sınıflandırma olmak üzere 2 farklı sınıflandırma uygulaması yapılmıştır. Sonuçların karşılaştırılmasında ise algoritmaların performans değerlendirmesinin yapılmasında kullanılan performans metrikleri incelenmiştir. Bu metrikler göz önüne alınarak yapılan değerlendirmede PCA yöntemi uygulandıktan sonra sınıflandırma işleminin yapılması durumunda bütün algoritmalarda performans metriklerinde bir iyileşme olduğu sınıflandırma başarısının arttığı görülmüştür. Bununla birlikte PCA uygulamasının yapılması ile hastanın cinsiyeti, Entübe, Zatürre olma durumu, yaşı, hamile olması ayrıca Şeker, KOAH ve Astım hastalığının bulunması durumu ölüm riskinin artmasında etkin rol oynadığı anlaşılmıştır. Bu hastalıklara sahip yaşlı insanların bu hastalıktan korunmak için daha dikkatli olmaları gerektiği, yakalanan insanların ise tedavisinde daha özen gösterilmesi gerekmektedir.

Kaynaklar (References)

Abdi, H., & Williams, L. J. 2010. Principal component analysis. Computational Statistics.

Akhtar, A., Akhtar, S., Bakhtawar, B., Kashif, A. A., Aziz, N., & Javeid, M. S. 2021. COVID-19 Detection from CBC using Machine Learning Techniques. International Journal of Technology, Innovation and Management (IJTIM), 1(2), 65-78.

Albahri, A. S., Hamid, R. A., Alwan, J. K., Al-Qays, Z., Zaidan, A., Zaidan, B., . . . Almahdi, E. 2020. Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (COVID-19): a systematic review. Journal of medical systems, 44, 1-11.

Amasyali, M. F., & Ersoy, O. 2009. Evaluation of regression ensembles on drug design datasets.

Bello-Chavolla, O. Y., Bahena-López, J. P., Antonio-Villa, N. E., Vargas-Vázquez, A., González-Díaz, A., Márquez-Salinas, A., . . . Aguilar-Salinas, C. A. (2020). Predicting mortality due to SARS-CoV-2: a mechanistic score relating obesity and diabetes to COVID-19 outcomes in

Mexico. *The Journal of Clinical Endocrinology & Metabolism, 105*(8), 2752-2761.

Bermejo, P., Gámez, J. A., & Puerta, J. M. 2011. Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. Expert Systems with Applications, 38(3), 2072-2080.

Breiman L., 2001, Random forests,machine learning, 2001 Kluwer Academic Publishers, 45(1), 5-32.

COVID-19 Mexico Patient Health Dataset. (2020, 05 19). Retrieved from Kaggle.com: https://www.kaggle.com/datasets/riteshahlawat/covid19-mexico-patient-health-dataset

Chinazzi, M., Davis, J. T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., . . . Sun, K. (2020). The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science, 368*(6489), 395-400.

de León, U. A.-P., Pérez, Á. G., & Avila-Vales, E. (2020). An SEIARD epidemic model for COVID-19 in Mexico: mathematical analysis and state-level forecast. *Chaos, Solitons & Fractals, 140*, 110165.

Drew, D. A., Nguyen, L. H., Steves, C. J., Menni, C., Freydin, M., Varsavsky, T., . . . Wolf, J. (2020). Rapid implementation of mobile technology for real-time epidemiology of COVID-19. *Science, 368*(6497), 1362-1367.

Escobedo-de la Peña, J., Rascón-Pacheco, R. A., de Jesús Ascencio-Montiel, I., González-Figueroa, E., Fernández-Gárate, J. E., Medina-Gómez, O. S., . . . Borja-Aburto, V. H. (2021). Hypertension, diabetes and obesity, major risk factors for death in patients with COVID-19 in Mexico. *Archives of medical research, 52*(4), 443-449.

Freund, Y., & Mason, L. 1999. The alternating decision tree learning algorithm. Paper presented at the icml.

Gansevoort, **R.** T., & Hilbrands, L. B. (2020). CKD is a key risk factor for COVID-19 mortality. *Nature Reviews Nephrology, 16*(12), 705-706.

Guan, X., Zhang, B., Fu, M., Li, M., Yuan, X., Zhu, Y., . . . Lu, Y. 2021. Clinical and inflammatory features-based machine learning model for fatal risk prediction of hospitalized COVID-19 patients: results from a retrospective cohort study. Annals of Medicine, 53(1), 257-266.

Jagodnik, K. M., Ray, F., Giorgi, F. M., & Lachmann, A. 2020. Correcting under-reported COVID-19 case numbers: estimating the true scale of the pandemic. medRxiv.

Kassania, S. H., Kassanib, P. H., Wesolowskic, M. J., Schneidera, K. A., & Detersa, R. 2021. Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: a machine learning based approach. Biocybernetics and Biomedical Engineering, 41(3), 867-879.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In *Machine learning proceedings 1992* (pp. 249-256): Elsevier.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence, 97*(1-2), 273-324.

Levin, A. T., Cochran, K., & Walsh, S. 2020. Assessing the age specificity of infection fatality rates for COVID-19: Meta-analysis & public policy implications. NBER Working Paper(w27597).

Li, B., Yu, S., & Lu, Q. 2003. An improved k-nearest neighbor algorithm for text categorization. arXiv preprint cs/0306099.

Li, J., Zhang, S., Lu, Y., & Yan, J. 2008. Real-time P2P traffic identification. Paper presented at the IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference.

Li, M., Zhang, Z., Cao, W., Liu, Y., Du, B., Chen, C., . . . Chen, C. 2021. Identifying novel factors associated with COVID-19 transmission and fatality using the machine learning approach. Science of the Total Environment, 764, 142810.

Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C., & Strintzis, M. 1998. ECG pattern recognition and classification using non-linear transformations and neural networks: A review. International journal of medical informatics, 52(1-3), 191-208.

Manski, C. F., & Molinari, F. 2021. Estimating the COVID-19 infection rate: Anatomy of an inference problem. Journal of Econometrics, 220(1), 181-192.

Muhammad, L., Algehyne, E. A., Usman, S. S., Ahmad, A., Chakraborty, C., & Mohammed, I. A. (2021). Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset. *SN computer science, 2*(1), 1-13.

Nemati**,** M., Ansary, J., & Nemati, N. (2020). Machine-learning approaches in COVID-19 survival analysis and discharge-time likelihood prediction using clinical data. *Patterns, 1*(5), 100074.

Novaković, J., Strbac, P., & Bulatović, D. (2011). Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of operations research, 21*(1), 119-135.

Parra-Bracamonte**,** G. M., Lopez-Villalobos, N., & Parra-Bracamonte, F. E. (2020). Clinical characteristics and risk factors for mortality of patients with COVID-19 in a large data set from Mexico. *Annals of epidemiology, 52*, 93-98. e92.

Quiroz-Juárez, M. A., Torres-Gómez, A., Hoyo-Ulloa, I., León-Montiel, R. d. J., & U'Ren, A. B. (2021). Identification of high-risk COVID-19 patients using machine learning. *Plos one, 16*(9), e0257234.

Quinlan, J. R. 1999. Simplifying decision trees. International Journal of Human-Computer Studies, 51(2), 497-510.

Rahmandad, H., Lim, T. Y., & Sterman, J. 2020. Estimating COVID-19 under-reporting across 86 nations: implications for projections and control. medRxiv.

Ríos-Silva, M., Murillo-Zamora, E., Mendoza-Cano, O., Trujillo, X., & Huerta, M. (2020). COVID-19 mortality among pregnant women in Mexico: a retrospective cohort study. *Journal of Global Health, 10*(2).

Singh, J., Green, M. B., Lindblom, S., Reif, M. S., Thakkar, N. P., & Papali, A. (2021). Telecritical care clinical and operational strategies in response to COVID-19. *Telemedicine and e-Health, 27*(3), 261-268.

Stella, L., Martínez, A. P., Bauso, D., & Colaneri, P. 2020. The role of asymptomatic individuals in the Covid-19 pandemic via complex networks. arXiv preprint arXiv:2009.03649.

Ünsal, A., Bileşenler, Ö., _Faktür, M., & Mali, D. A. Y. I. Ş. 1996. Başarılarının Analizi. In: Ankara.

Velavan, T. P., & Meyer, C. G. 2020. The COVID-19 epidemic. Tropical medicine & international health, 25(3), 278.

Xia, S., Xiong, Z., Luo, Y., Dong, L., & Zhang, G. 2015. Location difference of multiple distances-based k-nearest neighbors' algorithm. Knowledge-Based Systems, 90, 99-110.

Yang, S., Cao, P., Du, P., Wu, Z., Zhuang, Z., Yang, L., . . . Wang, X. 2020. Early estimation of the case fatality rate

of COVID-19 in mainland China: a data-driven analysis. Annals of translational medicine, 8(4).

Yavuz, Ü., & Dudak, M. N. 2020. Classification of covid-19 dataset with some machine learning methods. journal of amasya university the institute of sciences and technology, 1(1), 30-37.

Zawiah, M., Al-Ashwal, F. Y., Saeed, R. M., Kubas, M., Saeed, S., Khan, A. H., . . . Abduljabbar, R. (2020). Assessment of healthcare system capabilities and preparedness in Yemen to Confront the novel coronavirus 2019 (COVID-19) outbreak: a perspective of healthcare workers. *Frontiers in public health*, 419.

Zens, M., Brammertz, A., Herpich, J., Südkamp, N., & Hinterseer, M. (2020). App-based tracking of self-reported COVID-19 symptoms: analysis of questionnaire data. *Journal of medical Internet research, 22*(9), e21956.

Zhang, Y., Ding, L., & Wang, Y. 2011. Research and design of ID3 algorithm rules-based anti-spam email filtering. Paper presented at the 2011 IEEE 2nd International Conference on Software Engineering and Service Science.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 137-144
DOI: 10.38016/jista.1036047

Makine Öğrenmesi Yöntemleri ile Banka Müşterilerinin Kredi Alma Eğiliminin Karşılaştırmalı Analizi

Ali Tezcan Sarızeybek[1*], Onur Sevli[1*]

[1] Burdur Mehmet Akif Ersoy Üniversitesi, Yazılım Mühendisliği Bölümü, Bucak Teknoloji Fakültesi, Burdur, Türkiye

[2] Burdur Mehmet Akif Ersoy Üniversitesi, Bilgisayar Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Burdur, Türkiye

atsarizeybek@mehmetakif.edu.tr, onursevli@mehmetakif.edu.tr

Öz

Bankacılık, müşterilerle sık sık iletişime girilmesi gereken bir sektördür. Bankalar müşterilerine, onların durumlarına uygun bir kredi vermek istediğinde müşteriyi telefonla ararlar. Çoğu zaman müşteri, teklif edilen krediyi reddeder, bu da müşteriyle iletişime geçen personelin zamanından büyük bir kayıptır. Bu çalışmada, banka müşterilerinin verilerinin bulunduğu veri seti ele alınarak ve çeşitli makine öğrenmesi sınıflama modelleri kullanılarak müşterinin kredi alıp almayacağı tahmin edilmiştir. Elde edilen çalışma sonuçlarına göre, makine öğrenmesi yöntemleri ile müşterinin kredi alma eğilim tahmini başarılı bir şekilde gerçekleşmiştir. Çalışma sonucunda K-Best uygulanan modellerin arasında doğruluk değeri en yüksek olan sınıflandırıcı modelinin %98,86 ile Rastgele Orman algoritması olduğu, özellik seçimi yapılmadan eğitilen modellerin arasında en yüksek olan modelin %93,66 ile Rastgele Orman algoritması olduğu, cross-validation ve grid search ile eğitilen modellerin arasında ise en yüksek değerin %98,6 ile Rastgele Orman algoritmasında olduğu görülmüştür.

Anahtar kelimeler: Makine Öğrenmesi, Bankacılık, Kredi Tahminleme, Sınıflandırma Algoritmaları.

A Comparative Analysis of Bank Customers' Loan Propensity Using Machine Learning Methods

Abstract

Banking is a sector that requires frequent communication with customers. When banks want to give their customers a loan that suits their situation, they call the customer by phone. The often time customer rejects the loan offer, which is a huge waste of time from the staff contacting the customer. In this study, the customer's tendency to take credit was estimated using the data set of bank customers' data and various machine learning classification models. According to the results of the study, the prediction of the customer's tendency to take credit with machine learning methods has been successfully realized. As a result of the study, among the models applied K-Best, the classifier model with the highest accuracy value was found to be the Random Forest algorithm with 98.86%. Among the models trained without feature selection, the highest model was found to be the Random Forest algorithm with 93.66%. Among the models trained with cross-validation and grid search, the highest value was found in the Random Forest algorithm with 98.6%.

Keywords: Machine Learning, Banking, Credit Estimation, Classification Algorithms.

1. Giriş (Introduction)

Bu çalışma, Kranti Walke'nin hazırlamış olduğu "Bank Personal Loan Modelling" veri setinden yola çıkarak banka müşterilerinin teklif edilen krediyi alma eğilimlerinin farklı makine öğrenme yöntemleriyle tahmin edilmesini ve performans analizlerini konu almaktadır. Bankacılık sektöründe bankalar kendi müşterilerine onları telefonla arayarak ilgili müşteriye kredi teklifi yaparlar. Müşteri teklifi değerlendirir ve onay verip vermediğini belirtir; fakat bu belirtilen teklifi müşteriye ilettikten sonra alınan olumsuz cevap hem zaman hem de iş gücü almaktadır. Bunun önüne geçebilmek için müşterilerin verileri analiz yapılmalı elde edilen bilgilerden yola çıkarak kredi alma

* Sorumlu yazar.
E-posta adresi: atsarizeybek@mehmetakif.edu.tr

Alındı : 13 Aralık 2021
Revizyon : 8 Şubat 2022
Kabul : 22 Şubat 2022

eğilimlerini tahmin etmek zaman ve iş gücü tasarrufu açısından büyük yarar sağlayacaktır. Müşteri sayısının az olduğu durumlarda insan eli ile analiz yapmak basittir ancak müşteri sayısının çok fazla olduğu bir bankada analiz yapmak zor, hatta imkânsız hale gelir. Bu yüzden bilgisayarda incelemelerin yapılması, müşterinin kredi alma profilinin çıkarılması gerekir, bunun için de devreye makine öğrenmesi girmelidir. Sonuç sağlayabilmek için veri setinden müşterilerin maaş, aylık kredi kartı kullanımı, limit bilgileri gibi verilerin incelenerek bir makine öğrenmesi modeli oluşturulmalıdır. Müşteri verileri analiz edilerek müşterinin krediyi alıp almayacağı tahmin edildiği için diğer özellikler hedef alınarak farklı çıkarımlarda da bulunulabilir, örnek olarak kredi kart harcamalarından aylık gelir tahmini yapılabilir.

Bu çalışmada yapılan işlem sonucunda müşterinin kredi alma eğilimi doğru tahmin edilirse müşteriye yapılacak teklif tekrardan değerlendirilebilir, teklif paketi kapsamı bu çıkarıma göre genişletilebilir ya da daraltılabilir. Bu sayede banka müşteriye yüksek veya düşük bir teklifte bulunmaz. Müşterinin ihtiyacı doğrultusunda teklif yapılacağı için de müşteri servisten memnun kalır ve bankanın sadakat puanı yükselir. Teknolojinin hızla ilerlediği bu yıllarda bankacılık sektörünün de ilerlemesi gerekmektedir. Yapay zekanın bankacılık sistemlerinde uygulanması müşteri-banka arasındaki köprüyü sağlamlaştırmakla kalmayıp yapacağı çıkarımlarla müşteri ve bankanın detaylı analizlerinin yapılmasını sağlar. Bu sayede veriler ve müşteri ilişkileri sağlamlaşır. Bu çalışmada, bir bankanın müşteri verilerinden kredi alma eğilimi tahmin edilmeye çalışılmış, bunun için makine öğrenme modeli eğitilmiş, test verilerinin sonuçları ve gerçek verilerin kredi alma eğilimlerinin sonuçları karşılaştırılarak performans analizi yapılmıştır. Bu çalışmada elde edilmesi amaçlanan başarım makine öğrenmesi yöntemlerinde çapraz doğrulama ve ızgara arama yöntemlerinin sonuçlara nasıl katkı sağladığını gözlemlemek ve benzer veri seti kullanılmış çalışmalarla karşılaştırmaktır. Literatürde bu veri setini kullanan başka bir çalışma bulunmamış, bu yüzden müşterilerin davranışlarının tahminlemesi üzerine yapılan bu çalışmada diğer modellerin performans sonuçları ile karşılaştırması yapılmamıştır, literatürde bu veri seti ilk kez kullanıldığı için literatürde ilk olacaktır. Çalışmada sınıflandırma yöntemleri olarak k-En Yakın Komşu, Lojistik Regresyon, Karar Ağaçları, Rastgele Orman ve Destek Vektör Makinesi yöntemleri kullanılmış, sonuç olarak müşterinin kredi alıp almayacağının tahmin edilmesi amaçlanmıştır.

1.1. Benzer çalışmalar (Similar studies)

Yapılan literatür taramasında, çalışmalarda genel olarak kredi risk tahminleri yapılmış, bu çalışmadaki gibi müşterinin krediyi kabul etme tahminlemesi değil, bankanın müşterinin başvurusunu kabul etme tahminlemesi yapılmıştır.

Arun, Ishaan ve Sanmeet (2016) tarafından yapılan çalışmalarda müşterilerin başvuruları değerlendirilerek krediye en uygun müşterinin tahminlemesi amaçlanmış ve çalışmalar yapılmıştır. Cinsiyet, medeni durum, vasilik, eğitim durumu, gelir, ek gelir, başvurulan kredi miktarı, kredi süresi, sahip olunan taşınmaz mallar gibi veriler üzerinden kredinin kabul edilip edilmeyeceğinin tahmini amaçlanmıştır. Metot olarak Karar Ağaçları, Rastgele Orman, Lineer Modeller, Neural Net ve AdaBoost kullanılmış, sınıflandırma gerçekleştirilmiştir. Performans test sonuçları makaleye verilmemiştir.

Alan (2020) tarafından yapılan çalışmada, en iyi sınıflandırma modelinin belirlenmesi için çapraz doğrulama yöntemlerinden hold-out ve k-katlı çapraz doğrulama kullanılmış, bu işlemler 32 ayrı veri setine uygulanmıştır. Veri setlerinin arasında Portekiz Bankası Pazarlama veri setinde çapraz doğrulama ile en iyi modelin %88,75 doğruluk oranına sahip olan DVM sınıflandırıcısı olduğu tespit edilmiş, AUC oranı %53,32 ve F1 skor değeri ise %61,07 olarak elde edilmiştir. Veri seti üzerindeki çalışma sonucunda doğruluk değeri %86,35, AUC değeri %60,23 ve F1 skoru ise %61,83 olarak elde edilmiştir.

Serengil, İmece, Tosun, Büyükbaş ve Köroğlu (2021) tarafından yapılan çalışmada tahsili geciken kredinin farklı sınıflandırma algoritmaları ile tahmin edilmesi ve karşılaştırılması amaçlanmıştır. Özel bir bankanın 181.276 adet müşteri verisinden oluşan bir veri seti kullanılmıştır. Veri setinde müşteri ödeme geçmişi, bilançolar, önceki kredi kartı ödemeleri, diğer bankalardan gelen risk ve limit verileri gibi özellikler bulunmaktadır. Sonuç olarak elde edilecek değerler sağlıklı, gecikmeli ve tahsili gecikmiş kredi değerleridir. Veri seti farklı modellerde test edilmiş ve karşılaştırmalar sonucunda problem için en uygun modelin 0.90 özgüllük, 0.87 kesinlik, 0.77 duyarlılık ve 0.82 F1 skoru olarak elde edilmiştir.

Yapılan literatür taramasına göre bu probleme en uygun modeli bulmak için farklı sınıflandırıcılar kullanmalı ve aşırı uyma problemini ortadan kaldırmak için grid search ve modelin en iyi durumunu görmek için çapraz doğrulama yapılmalıdır.

1.2. Sınıflandırma yöntemleri (Classification methods)

Yapılan bu çalışmada makine öğrenmesi uygulamaları için çeşitli sınıflandırma algoritmaları kullanılmıştır. Çalışma kapsamında kullanılan veri setleri üzerinde k-En Yakın Komşu (kNN), Lojistik Regresyon, Karar Ağacı ve Rastgele Orman sınıflandırıcı algoritmaları uygulanmış ve performans değerlendirme analizleri yapılmıştır. Alt başlıklarda çalışmada kullanılan algoritmalar basitçe kısaca açıklanmıştır.

1.2.1. k_En Yakın Komşu Algoritması (k-Nearest Neighbor Algorithm)

k-En Yakın Komşu Algoritması, regresyon ve sınıflandırma için kullanılan popüler algoritmalardan biridir. Adından da anlaşılacağı üzere algoritmaya verilen değerler üzerinde, verilen k sayısı kadar en yakın noktalara bakılır, verilen örneğe en yakın olan k kadar değer seçilir ve çoğunluğa sahip olan değer örneğe atanır. K sayısında çakışma olmaması için genel olarak tek sayı verilmelidir (Rasjid, Setiawan,2017). Bu mesafelerin ölçülmesinde genellikle 1. eşitlikte verilen Öklid mesafe formülünü kullanılır. (Kılınç v.d.,2016)

$$d\,(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \quad (1)$$

1.2.2. Lojistik Regresyon Analizi (Logistic Regression Analysis)

Kesikli olarak iki değerin içinde, yani 1 veya 0 gibi, bir sonuç veren, kesikli veya sürekli verilerin incelenebildiği bir algoritmadır. Anlamlılık tespiti ile modelin ve parametrelerin olabilirlikleri tespit edilir ve tahminleme yapılır (Field,2009). 2. eşitlikte değerlerin gerçekleşme olasılığının formülü verilmiştir. (Coşkun v.d.,2004)

$$\Pi\,(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \quad (2)$$

1.2.3. Karar Ağaçları (Decision Trees)

Düğüm, dal ve yaprak olmak üzere 3 adet kısımdan oluşur. Her özellik bir düğüme atanmıştır, veriler kök düğümde saklanır. Kök tarafından itibaren yukarı doğru dalı olmayan düğümlere ve yapraklara gelene kadar bu sorular sorulur. Performans iyileştirmesi yapılması için birbirini bağlayan düğümlerin bağlantısını kesip yerine yaprak koyarak budama yapılmalı, veri ile alakasız dalların gereksiz yere işlenmesinden kaçınılmalıdır. Sınıflara ait entropi 3. eşitlikte verildiği gibi hesaplanır.

$$Entropi\,(T) = \sum_{1}^{n} p_i log_2\,(p_i) \quad (3)$$

Entropi hesaplandıktan sonra ise bölünme sonucunda elde edilen kazanç da 4. eşitlik gibi hesaplanır. (Akar, Güngör, 2012)

$$Kazanç\,(B,T) = Entropi\,(T) - \sum_{i=1}^{n}\frac{|T_i|}{|T|}Entropi\,(T) \quad (4)$$

1.2.4. Rastgele Orman (Random Forest)

Regresyon ve sınıflandırma için de kullanılabilen Rastgele Orman algoritması karar ağaçlarındaki overfitting problemini giderir (Ali, vd., 2012). Rastgele karar ağaçlarından bir orman oluşturur ve bunları birleştirir. İlk aşamada oluşan sınıflandırıcıya göre tahminleme yapılır. N adet özellik içinden K kadar özellik seçilir ve seçilen özellikler arasından en iyi noktaya göre sonraki düğüm hesaplanır. Düğüm, çocuk düğümlere ayrılır ve hedeflenen düğüm sayısına ulaşana kadar özellik seçme ve çocuk düğümlere ayırma tekrarlanır. Tahminleme aşamasına gelindiğinde ise rastgele seçilen bir karar ağacının oyları hesaplanır ve en yüksek oy alan tahmin seçilir. (Biau, Scornet, 2016)

1.2.5. Destek Vektör Makinesi (Support Vector Machine)

Destek Vektör Makinesi(DVM), veri setinde değişkenler arasındaki ilişkilerin bilinmediği, sınıflara ait verileri ayırabilmek için sınıflandırma problemlerinde kullanılır. Lineer verilerde doğrular üzerinde marjini en yüksek olan doğrunun seçilmesi amaçlanmış, lineer olmayan verilerde ise veriyi daha yüksek boyutta bir uzaya taşınır ve en iyi hiper düzlem bulunur. (Akşehirli, v.d., 2013) (Burges, 1998)

1.3. Algoritmaların karşılaştırılmasında kullanılan kriterler (Criteria used in comparing algorithms)

Algoritma sınıflandırma işlemi sonucunda elde edilen hata matrisinde 4 adet değer vardır. True Positive tahminin doğru, özelliğin de doğru olması, True Negative tahminin yanlış fakat özelliğin doğru yerde olması, False Positive tahminin doğru fakat özelliğin yanlış olması, False Negative ise tahminin ve özelliğin yanlış olması durumlarıdır. (Gök, 2017) Tablo 1'de karmaşıklık matrisinin değerleri vardır ve TP, TN, FP ve FN değerleri burada yerlerine yazıldıkları değerleri alırlar.

Tablo 1. Karmaşıklık Matrisi (Confusion Matrix)

		Tahmin Edilen	
		Pozitif	Negatif
Asıl Değer	Pozitif	TP	FN
	Negatif	FP	TN

1.3.1. Doğruluk Oranı (Accuracy)

Doğruluk oranı, modelde doğru tahmin edilen verilerin tüm verilere oranı ile bulunur. Örnek olarak 100 adet veri setinde doğru tahmin edilen veri sayısı 60 ise doğruluk oranı %60 olacaktır. Doğruluk oranı hesaplamak için 5. eşitlikteki formül kullanılır.

$$Doğruluk = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

1.3.2. Kesinlik (Precision)

Kesinlik oranı, modelde pozitif olarak tahmin edilen verilerin kaç adetinin gerçekte kaç tanesinin pozitif olduğunun oranıdır. Kesinlik oranı hesaplamak için 6. eşitlikteki formül kullanılır.

$$Kesinlik = \frac{TP}{TP+FP} \quad (6)$$

1.3.3. Duyarlılık (Recall)

Duyarlılık oranı, pozitif olması gereken değerlerin ne kadarının pozitif olarak tahmin edildiğini gösteren bir orandır. Duyarlılık hesaplaması için 7. eşitlikteki formül kullanılır.

$$Duyarlılık = \frac{TP}{TP + FN} \quad (7)$$

1.3.4. F-Değeri (F-Score)

F-Skor oranı, kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. Uç durumların da hesaba dahil edilmesi için harmonik ortalama alınır. F-Skor hesabı için 8. eşitlikteki formül kullanılır.

$$F1 = 2 * (\frac{Duyarlılık * Kesinlik}{Duyarlılık + Kesinlik}) \quad (8)$$

1.3.5. Alıcı Çalıştırma Karakteristik Eğrisi (Receiver Operating Characteristic Curve)

Alıcı Çalıştırma Karakteristik Eğrisi (ROC), tüm sınıflandırma eşiklerindeki sınıflandırıcının performansını gösterir. Sınıflandırma eşiğini düşürmek daha fazla değerin sınıflandırılması demektir. ROC eğrinin True Positive Rate (TPR) ve False Positive Rate (FPR) olmak üzere iki adet parametresi vardır. TPR, duyarlılığa benzerdir ve 9. eşitlik gibi hesaplanır.

$$TPR = \frac{TP}{TP + FN} \quad (9)$$

FPR ise 10. eşitlik gibi hesaplanır.

$$FPR = \frac{FP}{FP + TN} \quad (10)$$

1.3.6. AUC Oranı (Area Under the Curve Rate)

Area Under the Curve (AUC), ROC eğrisinin altında kalan alanı hesaplar, tüm sınıflandırma eşikleri arasında performans hesaplarının toplamlarını gösterir. AUC oranı 0-1 arasındadır, %100 hatalı tahmin eden bir modelin AUC oranı 0.0 iken %100 doğru tahmin eden bir modelin AUC oranı 1.0'dır. (Zhang, 2016)

2. YÖNTEM (Method)

2.1. Veri seti (Data set)

Bu yazıda Kranti Walke'nin hazırlamış olduğu "Bank Personal Loan Modelling" adlı, Thera Bank adında bir bankanın müşterilerinin verilerini içeren bir veri seti kullanılmıştır. Veri setinde yaş, deneyim, gelir, posta kodu, ailedeki kişi sayısı, aylık kredi kart harcama ortalaması, eğitim durumu, eğer varsa evinin mortgage değeri, menkul kıymet hesap varlığı, mevduat hesap varlığı, online işlem kullanıcısı olup olmadığı, kredi kart varlığı ve son teklif edilen kredi teklifini kabul edip etmediği özelliklerini içerir. Veri seti 14 öznitelik ve 5000 adet örnekten oluşmaktadır. Veri setine ait öznitelikler ve açıklamaları Tablo 2'de verilmiştir.

Veri seti içindeki eşsiz veri sayısı, ortalama, medyan ve standart sapma değerleri Tablo 3'te verilmiştir

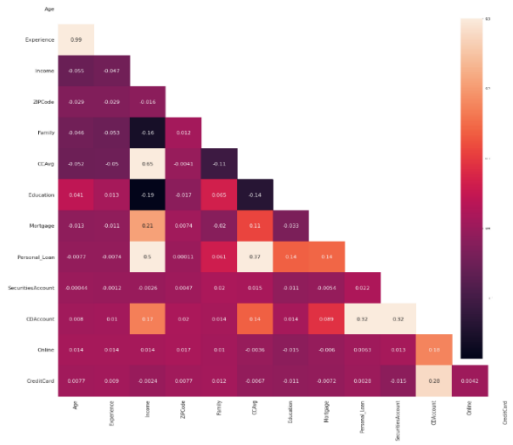
Tablo 2. Banka müşterileri veri seti öznitelikleri ve açıklamaları (Bank customers dataset attributes and descriptions)

Öznitelik	Açıklama
Age	Tamamlanan yıl olarak yaşı
Experience	Yıl olarak iş deneyimi
Income	Kişinin yıllık geliri (Gelir = Veri * 1000 $)
ZIPCode	Bulunduğu yerin posta kodu
Family	Yaşadığı yerdeki veya ailesinde bulunan kişi sayısı
CCAvg	Aylık ortalama kredi kartı harcama (Harcama = Veri * 1000 $)
Education	Eğitim Durumu 1 = Lisans 2 = Lisansüstü 3 = Doktora
Mortgage	Eğer varsa evinin mortgage değeri (Değer = Veri * 1000 $)
Personal_Loan	Son kampanyada müşterinin kredi teklifini kabul edip etmemesi 1 = Evet 0 = Hayır
SecuritiesAccount	Müşterinin bankada mevduat hesabının varlığı 1 = Var 0 = Yok
CDAccount	Müşterinin bankada menkul kıymet hesabının varlığı 1 = Var 0 = Yok
Online	Müşteri bankanın internet bankacılığını kullanıyor mu? 1 = Evet 0 = Hayır
CreditCard	Müşteri kredi kartı kullanıyor mu? 1 = Evet 0 = Hayır

Tablo 3. Veri setindeki eşsiz veri, standart sapma, ortalama ve medyan değerleri (Unique data, standard deviation, mean and median values in the data set)

Özellik	Eşsiz Veri Sayısı	Standart Sapma	Ortalama	Medyan	Min	Max
Age	45	11.463166	45.3384	45	23	67
Experience	47	11.467954	20.1046	20	0	43
Income	162	46.033729	73.7742	64	8	224
ZIPCode	467	2121.852197	93152.503	93437	9307	96651
Family	4	1.147663	2.3964	2	1	4
CCAvg	108	1.747659	1.937938	1.5	0	10
Education	3	0.839869	1.881	2	1	3
Mortgage	347	101.713802	56.498	0	0	635
Personal_Loan	2	0.294621	0.096	0	0	1
SecuritiesAccount	2	0.305809	0.1044	0	0	1
CDAccount	2	0.238250	0.0604	0	0	1
Online	2	0.490589	0.5968	1	0	1
CreditCard	2	0.455637	0.294	0	0	1

Veri setinin korelasyon matrisi ve sıcaklık haritası Şekil 1'de verilmiştir.



Şekil 1. Veri setinin sıcaklık haritası (The temperature map of the dataset)

2.2. Nitelik seçme (Feature selection)

Bu veri setinde özellikler, tek değişkenli analiz yöntemi olan K-Best yöntemi ile çıkarılmıştır. K-Best yöntemi, veri seti üzerinde hedefe yönelik tek değişkenli istatistik testler yapar ve en yüksek skor yapan özellikleri belirtilen k sayısı kadar alır. Bu çalışma için k sayısı, çalışmalara göre doğruluk ve kesinlik oranlarının en yüksek olduğu, 10 olarak belirlenmiştir, 9 olduğunda en yüksek doğruluk değeri %94, 8 olduğunda ise %88 çıkmaktadır. 11 Olduğunda da aynı şekilde doğruluk değeri düşmektedir. Özellik seçme aşamasının sonucunda özellik olarak kullanılarak sütunlar: Yaş (Age), gelir (Income), ailedeki kişi sayısı (Family), kredi kartı aylık ortalama harcama (CCAvg), eğitim durumu (Education), mortgage durumu (Mortgage), menkul kıymet hesap varlığı (SecuritiesAccount), mevduat hesap varlığı (CDAccount), online işlem kullanıcısı olup olmadığı (Online) ve kredi kart kullanıcısı olup olmadığı (CreditCard) sütunlarıdır, hedef sütun ise kredi alıp almadığı (Personal Loan) sütunudur, yani tahmin edilmek istenen veri bu sütundadır.

3. Çalışma Sonuçları (Study Results)

Bu çalışmada makine öğrenmesi sınıflandırma algoritmalarından kNN, Lojistik Regresyon, Karar Ağacı ve Rastgele Orman algoritmaları kullanılmıştır. Veri setinde model oluşturmak için veriler %70 eğitim, %30 test olarak ayrılmıştır. Bütün algoritmalarda rastgele durum 0 olarak belirlenmiştir. Rastgele Orman algoritmasında ağaç sayısı 10 olarak belirlenmiştir. kNN algoritmasında komşu sayısı 3 olarak belirlenmiştir. Rastgele Orman algoritmasında ağaç sayısı ve kNN algoritmasında komşu sayısı, sayılar denenip sonuç olarak doğruluk oranı en yüksek elde edilen sayılar alınmıştır. Veri seti üzerindeki null değerler silinmiş, bazı verilerdeki deneyim değeri 0'dan küçük olan veriler saptanmış, bu veriler silinmiştir.

Müşterilerden kredi teklifini reddedenlerin kredi kart harcama ortalamalarının orta değeri 1400 olarak elde edilmiş, kredi teklifini reddedenlerin kredi kart harcama ortalamaları ise 3800 olarak elde edilmiştir.

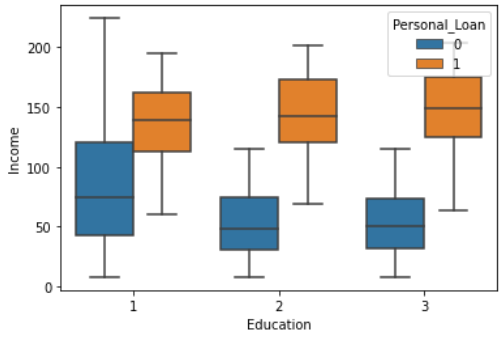
Tüm özelliklerin ortalaması Tablo 4'te verilmiştir.

Ortalama değerlere göre kredi teklifini kabul edenler müşterilerin %9,6'lık bir kısmını oluşturmaktadır. Bunun yanında internet bankacılığı kullananların oranı %59,68, kredi karta sahip olan müşteri oranı %29,4, aylık ortalama kredi kartı harcaması ise 1937,938$'dır. Yıllık gelir ortalaması ise 73.774,20$'dır.

Eğitim durumu ve yıllık gelir açısından kredi teklifini kabul etme durumu Şekil 2.'de verilmiştir.

Tablo 4. Özelliklerin ortalama değerleri (Average values of features)

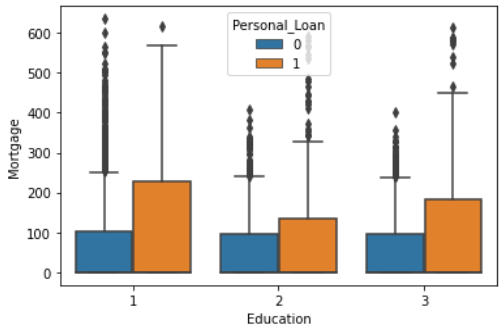
Özellik	Ortalama
Age	45.338400
Experience	20.104600
Income	73.774200
ZIPCode	93152.503000
CCAvg	1.937938
Education	1.881000
Mortgage	56.498800
Personal_Loan	0.096000
SecuritiesAccount	0.1044000
CDAccount	0.060400
Online	0.596800
CreditCard	0.294000



Şekil 2. Eğitim durumu ve yıllık gelire göre kredi teklifini kabul etme durumu kutu grafiği (Box chart of accepting loan offer by education and annual income)

Şekil 2'deki grafiğe göre geliri yüksek olan müşteriler daha çok kredi alma eğilimine sahip iken, lisans mezunu olan müşterilerde teklifi reddedenlerin gelir aralığı biraz daha fazladır.

Eğitim durumu ve mortgage değerine göre kredi teklifini kabul etme durumu Şekil 3'te verilmiştir.



Şekil 3. Eğitim durumu ve mortgage değerine göre kredi teklifini kabul etme durumu kuru grafiği (Graph of accepting a loan offer by education level and mortgage value)

Şekil 3'teki sıcaklık haritasına göre kredi teklifini kabul etme durumunun en çok yıllık gelir ve kredi kartı harcaması ile ilişkisi vardır.

Veri setinden ID, ZIPCode ve Experience özellikleri çıkarılarak %70 eğitim ve %30 test olarak ayrılmışlardır. Eğitim ve test için ayrılan veriler ölçeklenmiştir. ID ve ZIPCode özelliklerinin çıkarılma sebebi sonuca ulaşmak için gerekli olan özellikler olmadıkları içindir. Experience özelliğinin çıkarılma sebebi ise Age özelliği ile neredeyse aynı olduğu ve bu yüzden tekrarlı bir veri gibi görüleceği için çıkarılmıştır.

Tüm sınıflandırma algoritmalarına göre veriler modellenmiş ve doğruluk, kesinlik, duyarlılık ve F-Skor değerleri Tablo 5'te verilmiş ve doğruluk oranlarına göre bir sıraya konulmuştur. Elde edilen sonuçlara göre bu çalışma için uygulanabilecek en kötü algoritmanın Lojistik Regresyon yöntemi, en iyi algoritmanın ise Rastgele Orman algoritması olduğu gösterilmektedir.

Tablo 5. K-Best ile Model performans değerleri (Model performance values with K-Best)

İsim	Kesinlik	Duyarlılık	F-Skor	Doğruluk	Sıra
Lojistik Regresyon	0.93	0.95	0.98	0.95	5
Karar Ağacı	0.93	0.95	0.94	0.9793	3
Rastgele orman	0.99	0.94	0.96	0.9886	1
DVM	0.96	0.92	0.93	0.9832	2
kNN	0.96	0.83	0.88	0.966	4

Tablo 6. Özellik seçimi yapılmadan ölçülen model performans değerleri (Model performance values measured without feature selection)

İsim	Kesinlik	Duyarlılık	F-Skor	Doğruluk	Sıra
Lojistik Regresyon	0.88	0.94	0.91	0.93666	2
Karar Ağacı	0.90	0.89	0.89	0.88733	5
Rastgele orman	0.91	0.94	0.91	0.93666	1
DVM	0.88	0.91	0.90	0.91544	4
kNN	0.90	0.93	0.91	0.93133	3

Tablo 6'daki özellik seçimi yapılmadan uygulandığında elde edilen performans sonuçlarına göre 0.91 kesinlik ve 0.93666 doğruluk oranı ile Rastgele Orman algoritmasının en iyi sonuç verdiği elde edilmiştir. Tablo 6'ya ve Tablo 5'e göre özellik seçimi yapıldıktan sonra performans iyileşmesi gözlemlenmiştir.

Çalışma sonrasında aşırı uymayı (overfitting) engellemek için Grid Search (ızgara arama) ve bağımsız veri setinde en iyi modeli seçmek için çapraz doğrulama (cross-validation) uygulanıp testler uygulanmış, testin performans sonuçları Tablo 5'te verilmiştir. ROC eğrisi ise Şekil 4'te verilmiştir.

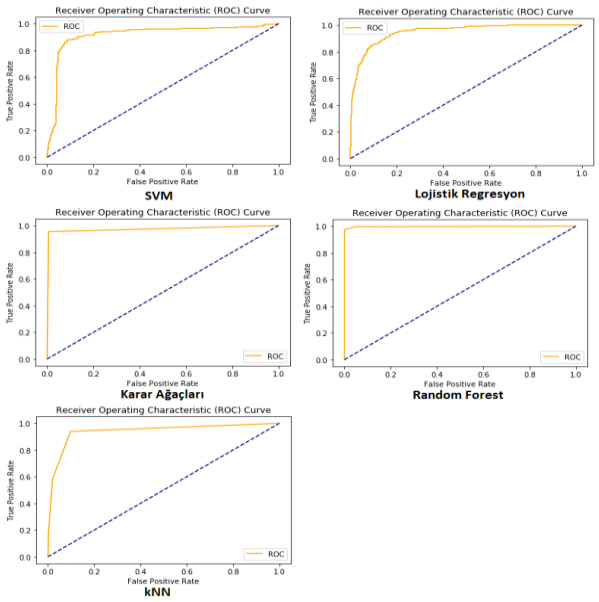
Çapraz doğrulama Stratified K-Fold yöntemi ile gerçekleştirilmiş, katlama(fold) oranı 5, diğer parametreler varsayılan olarak belirlenmiştir. Izgara aramada ise karar ağaçları algoritmasında en fazla derinlik(max depth) 2, 3, 4 olarak belirlenmiş; en az

örnek sayısı (min samples leaf) 0.12, 0.14, 0.16 ve 0.18 olarak belirlenmiştir, diğer algoritmalarda ise varsayılan değerler atanmış ve ızgara arama işlemi gerçekleştirilmiştir.

Tablo 7'ye göre bu problem için en uygun olan modelin çapraz doğrulama oranı 0.997, doğruluk oranı 0.986, AUC oranı 0.998 olan Rastgele Orman algoritması olduğu elde edilmiştir.

Tablo 7. Çapraz Doğrulama ve Grid Search uygulanan modellerin performans sonuçları (Performance results of models with Cross Validation and Grid Search applied)

Model	Çapraz Doğrulama Doğruluk Oranı	Model Doğruluk Oranı	AUC Oranı
Karar Ağaçları	0.99634	0.97933	0.97721
Lojistik Regresyon	0.88495	0.92933	0.95029
Rastgele orman	0.99734	0.986	0.99882
DVM	0.9279	0.942	0.95356
kNN	0.95840	0.91333	0.92927



Şekil 4. ROC Eğrisi (Roc Curve)

4. Sonuçlar (Conclusions)

Bankalarda yapay zekâ ve makine öğrenmesi kullanımı her geçen gün artmakta, örnek olarak müşterilerin ilgi alanları ve para çekme eğilimleri gibi işlemlerin tahminlemesi yapılmaktadır. Bu tahminlemelerin yapılabilmesinin bir insan tarafından yapılması imkansıza yakındır, çünkü bankaların binlerce, hatta milyonlarca müşterisi bulunmaktadır. Bu yüzden yapay zekâ ve makine öğrenmesinin kullanılması gerekmektedir.

Bu çalışmada makine öğrenmesi sınıflandırma algoritmalarından kNN, Rastgele Orman, Lojistik Regresyon, Karar Ağacı kullanılmış, K-Best ile yapılan özellik seçiminde yapılan çalışma sonucunda elde edilen verilere göre bu çalışma için en uygun olan sınıflandırma algoritması doğruluk değeri 0,9886 ile Rastgele Orman algoritmasıdır. İkinci en iyi algoritma ise 0,9832 ile Destek Vektör Makineleri algoritmasıdır. En kötü algoritma ise 0,95 ile Lojistik Regresyon algoritmasıdır. Özelliklerin fazla olması ve korelasyon matrislerinde çoğu özelliklerin değerlerinin birbirlerine yakın olmaması yüzünden kötü bir performans beklenirken, iki özellik arasında değil, çok sayıdaki özelliğin birbirlerine bağlantılı olduklarını gösterir. Özellik seçimi yapmadan yapılan çalışma sonucunda ise 0,91 Kesinlik, 0,94 Duyarlılık, 0,91 F-Skor ve 0,93666 Doğruluk ile Rastgele Orman en yüksek doğruluğa sahip model olmuş, özellik seçimi yapmanın önemi tüm sonuçların düşmesinden anlaşılmıştır. Çapraz doğrulama ve grid search uygulanan modelde en iyi sonucu çapraz doğrulama oranı 0,997, doğruluk oranı 0,986 ve AUC oranı 0,998 ile Rastgele Orman algoritması elde etmiştir, bu da veri setinde aşırı uyma problemini ve en iyi sonuç elde edecek olan modeli elde etmeye yardımcı olmuştur.

Bu çalışmanın sonucunda müşterilere teklif gönderirken müşterilerin kabul edebilme durumlarını tahmin edebilen bir uygulamanın yapılabileceği, bu sayede müşteri temsilcilerinin iş gücünden ve zamandan tasarruf edilebileceği düşünülmektedir. Daha önce bu veri seti ile bir çalışma yapılmadığı için gelecekte bu veri seti ile yapılacak olan çalışmalara katkı sağlaması düşünülmektedir.

Kaynaklar (References)

Akar, Ö., & Güngör, O., 2012. Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması. Jeodezi ve Jeoinformasyon Dergisi, 1 (2), 139-146.

Akşehirli, Ö. Y., Ankaralı, H., Aydın, D., Saraçlı, Ö., 2013. Tıbbi Tahminde Alternatif Bir Yaklaşım: Destek Vektör Makineleri. Turkiye Klinikleri Journal of Biostatistics, 5(1).

Alan, A., 2020. Makine öğrenmesi sınıflandırma yöntemlerinde performans metrikleri ile test tekniklerinin farklı veri setleri üzerinde değerlendirilmesi (Master's thesis, Fen Bilimleri Enstitüsü)

Ali, J., Khan, R., Ahmad, N., & Maqsood, I., 2012. Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9(5), 272.

Arun, K., Ishan, G., & Sanmeet, K., 2016. Loan approval prediction based on machine learning approach. IOSR J. Comput. Eng, 18 (3), 18-21.

Biau, G., & Scornet, E., 2016. A random forest guided tour. Test, 25 (2), 197-227.

Burges, C. J., 1998. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

Coşkun, S., Kartal, M., Coşkun, A., & Bircan, H., 2004. Lojistik regresyon analizinin incelenmesi ve diş hekimliğinde bir uygulaması. Cumhuriyet Üniversitesi Diş Hekimliği Fakültesi Dergisi, 7 (1), 42-50.

Field, A., 2013. Discovering statistics using IBM SPSS statistics. sage.

Gök, M., 2017. Makine öğrenmesi yöntemleri ile akademik başarının tahmin edilmesi. Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 5 (3), 139-148.

Kılınç, D., Borandağ, E., Yücalar, F., Tunalı, V., Şimşek, M. & Özçift, A., 2016. KNN Algoritması ve R Dili ile Metin Madenciliği Kullanılarak Bilimsel Makale Tasnifi. Marmara Fen Bilimleri Dergisi, 28 (3), 89-94. DOI: 10.7240/mufbed.69674

Rasjid, Z. E., & Setiawan, R., 2017. Performance comparison and optimization of text document classification using k-NN and naïve bayes classification techniques. Procedia computer science, 116, 107-112.

Serengil, S. I., Imece, S., Tosun, U. G., Buyukbas, E. B., & Koroglu, B., 2021. A Comparative Study of Machine Learning Approaches for Non Performing Loan Prediction. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 326-331). IEEE.

Zhang, Z., 2016. Introduction to machine learning: k-nearest neighbors. Annals of translational medicine, 4 (11) .

Research Article
Journal of Intelligent Systems: Theory and Applications 5(2) 2022: 145-152
DOI: 10.38016/jista.1078474

Using Machine Learning Algorithms for Jumping Distance Prediction of Male Long Jumpers

Mürsel Ozan İncetaş[1*], Murat Uçar[2], Işık Bayraktar[3], Murat Çilli[4]

[1] Alanya Alaaddin Keykubat University, Computer Programming, Antalya, Türkiye

[2] İskenderun Technical University, Management Information Systems, İskenderun, Türkiye

[3] Alanya Alaaddin Keykubat University, Sports Coaching Education, Antalya, Türkiye

[4] Sakarya University of Applied Sciences, Sports Coaching Education, Sakarya, Türkiye

ozan.incetas@alanya.edu.tr, murat.ucar@iste.edu.tr, drisikb@icloud.com, mcilli@subu.edu.tr

Abstract
The long jump is defined as an athletic event, and it has also been a standard event in modern Olympic Games. The purpose of the athletes is to make the distance as far as possible from a jumping point. The main purpose of this study was to determine the most successful machine learning algorithm in the prediction of the long jump distance of male athletes. In this paper, we used age and velocity variables for predicting the long jump performance of athletes. During the research, 328 valid jumps belonging to 73 Turkish male athletes were used as data. In determining the most successful algorithm, mean absolute error (MAE), root mean square error (RMSE), Mean Squared Error (MSE), R^2 score, Explained Variance Score (EVS), and Mean Squared Logarithmic Error (MSLE) values were taken into consideration. The outcomes of the analysis showed that long jump performance can be determined by chosen independent variables. The 5-fold cross-validation technique was used for the performance evaluation of the models. As a result of the experimental tests, the Gradient Boosting Regression Trees (GBRT) algorithm reached the best result with an MSE value of 0.0865. In this study, it was concluded that the machine learning approach suggested can be used by trainers to determine the long jump performance of male athletes.

Keywords: Long jump performance, machine learning, run-up velocity

1. Introduction

In the long jump, the goal is to gain speed on the running track and to jump as far from the board as possible. Besides the many parameters, the horizontal velocity, which has the highest biomechanical effect on flight distance, is very essential in the long jump (Hay, Miller, & Canterna, 1986; Linthorne, 2008). Some high-level long jumpers, such as Carl Lewis and Marion Jones, are also known to be high-level sprinters (Derse, Hansen, Tim, & Stolley, 2012). The fastest sprinters are not the best long jump athletes; however, it can be said the best long jumpers are the fastest ones. The long jump biomechanical analysis report of the 2009 IAAF World Athletics Championships confirms this; The athletes who ranked first had higher run-up velocities than others (Hommel, 2009). It was seen that the athletes who have the top three of the world rankings had 11 m/s of horizontal velocity (Fukasiro and Wakavama, 1992). As seen, run-up velocity is the most significant determinant of long jump performance (Açıkada, Arıtan, & Yazıcıoğlu, 1993; Bridgett, Galloway, & Linthorne, 2002; Bridgett and Linthorne, 2006; Hay, 1993; Hay, et al., 1986; Lees, Graham-Smith, & Fowler, 1994) It has been determined that there is a powerful relationship of 0.96 between the horizontal velocity and jump distance (Bridgett and Linthorne, 2006). Similarly, there are some studies indicating the relationship between velocity and jump distance (Bridgett, et al., 2002; Hay, 1993; Hay, et al., 1986; Lees, et al., 1994; Mishra and Rathore, 2016; Moura, Moura, & Borin, 2005, Rahim, et al., 2020; Takahashi & Wakahara, 2019). When the run-up velocity is artificially increased, a high increase in the jumping distance is observed (Schulek, 2002). According to the calculations, an increase of 0.1 m/s in velocity provides a rise in the jump distance by 6 to 12.8 cm (Bridgett and Linthorne, 2006; Hay, 1986).

* Corresponding Author.
E-mail: ozan.incetas@alanya.edu.tr

Recieved : 24 Feb 2022
Revision : 25 Apr 2022
Accepted : 13 Jun 2022

Studies focused on building a model to predict jumping distance related to run up velocity often used linear or nonlinear equations (Fukasiro and Wakavama, 1992; Hay and Miller, 1985; Lees, et al., 1994; Mikhailov, Yakunin, & Aleshinsky, 1981; Tiupa, Aleshinsky, Primakov, & Pereverzev, 1982). Most of these models were developed on a limited number of top athletes so models could predict non-acceptable jumping distance for extrapolated data. For instance, the nonlinear model of Mikhailov et al. (Mikhailov, et al., 1981) predicted a jumping distance of 44.25m for 10m/s run-up velocity. Some of these models had high accuracy estimations for low run-up velocities while others had better accuracy for high run-up velocities. Bayraktar and Çilli investigated a linear model, using 328 valid trials of 73 athletes during official competitions, which had better estimations for both lower and higher values (Bayraktar and Çilli, 2018).

The results of the studies showed that more sensitive and reliable models were needed. Linear or non-linear models did not have sufficient estimations for the wide range of velocity values. Recently, however, more advanced non-linear systems based on artificial intelligence have been used for modeling processes instead of linear approaches. Ofoghi et al. used machine learning techniques to develop approaches that predict performance models at the Track Cycling Omnium championships (Ofoghi, Zeleznikow, MacMahon, & Dwyer, 2010). In 2017, machine learning techniques were used to measure the hitting loads in tennis (Whiteside, Cant, Connolly, & Reid, 2017). As of 2018, there have been studies to estimate the performances of athletes. In a study to estimate biathlon shooting performances with the help of machine learning techniques, the results of the 5th season were tried to be accurately determined using the data of the previous 4 seasons (Maier, Meister, Trösch, & Wehrlin, 2018). The predicted accuracy rate of the study remained at 62%. In 2019, a classification approach was presented to predict the future success of potential young archers (Musa et al., 2019). As the studies indicated, it was clear that computers and especially machine learning (artificial intelligence) techniques could be used at many points that require experience from the choice of the athletes to the training load and the estimation of their degrees. Today, it is seen that ML techniques are used in many areas of sports, from predicting results in team sports (Bunker and Susnjak, 2022), to athlete health and injury prevention (Eetvelde et al., 2021). Despite the popularity of ML techniques in sports sciences recently, any study has been found in which ML techniques are used for prediction and modeling in the field of the long jump. In this study, we used different machine learning algorithms for estimating the jumping distance of male long jumpers. Thus, besides introducing ML techniques to the field of long jump, it has been tried to show that successful results can be obtained as an alternative to the techniques used in the past and based only on run-up velocity. In addition, detailed analyses were made on which ML approach could yield more successful results, and information on the parameters used was given. After determining the most successful model, we developed a web application that trainers could use.

This paper is organized as follows. Section 2 presents the research methodology, in which the data, analysis methods, and evaluation techniques are explained. Section 3 provides a comparative analysis of models. In Section 4, the results are discussed and explained. The paper is concluded in Section 5.

2. Material and Methods

2.1. Participants

Data used in this study consisted of 328 valid trials of 73 Turkish male athletes and were also used in the study presented by Bayraktar and Çilli (Bayraktar and Çilli, 2018). The average age of these long jumpers was 18.7 (±2.8) years old. All data were gathered from 11 competitions which were in the Turkish Athletic Federation's official calendar. Data collection was begun after the permission of the Turkish Athletic Federation and the approval of the Sakarya University Ethics Committee.

2.2. Research Design

The photocells were placed at 1, 6, and 11 meters behind the takeoff board to determine athletes' running times. For each jump, velocities V1, V2 and Vloss were calculated for the sections 11m-6m, the 6m-1m, and the difference between V2 and V1, respectively. In addition, official jump distances were recorded.

2.3. Dataset

The information about the obtained data from 328 valid trials is given in Table 1. The average age of the athletes, whose youngest is 14 years old and the oldest 28 years old, is 18.7. The average jumping distance of all athletes is 6.30 meters. The V1 and V2 averages of the athletes are 8.88 and 8.92 m/s, respectively.

Table 1. Mean and standard deviation values of the variables for the samples.

Variables	n	Mean (SD)	Min	Max
Age(year)		18.7 (2.80)	14.4	28.5
Jump Distance		6.30 (0.71)	4.53	7.74
V1 (m/s)	328	8.88 (0.71)	7.08	10.89
V2 (m/s)		8.92 (0.54)	7.52	10.20
Vloss (%)		0.71 (5.61)	-11.66	17.82

Figure 1. Calculated correlation values between jumping distance and variables.

As shown in Figure 1, correlation statistics were calculated between jumping distance and variables. It was found that the run-up velocity variables V1 and V2 had positive and strong relationships with jumping distance. The correlation between age and jumping distance was a positive and moderate relationship ($r=0.41$, $p>0.05$). The correlation between velocity losses and jumping distance was a negative and weak relationship ($r=-0.27$, $p>0.05$).

2.4. Machine Learning Methods

In this paper, five popular machine learning techniques were used: Artificial Neural Networks, Ridge Regression, Decision Trees, K-Nearest Neighbors Regression, Random Forest, and Gradient Boosting Regression Trees. These modeling techniques were briefly discussed below. In addition, for hyperparameter optimization, the Grid Search technique was used to find the most suitable one by trying different parameters. The parameters evaluated during the training phase was given in Table 2.

Table 2. Evaluated parameters of each machine learning methods

Model	Parameter	Start	Finish	Increment
ANN	Hidden Layer 1 Neuron	5	50	5
	Hidden Layer 2 Neuron	5	50	5
Ridge	Alpha	0.1	3	0.1
KNN	Neighbors	1	20	1
Gradient Boosting	Estimator	100	1000	100
Random Forest	Estimator	100	1000	100

Artificial neural networks (ANN) are biologically inspired mathematical techniques that can model complex nonlinear functions (Haykin, 2009). We used multilayer perceptron (MLP) Neural Network architecture with a backpropagation type supervised-learning algorithm. MLP was used to generate regression-type estimation models for numerical variables (Hornik, Stinchcombe, & White, 1990).

ANN architecture used in the study was given in Figure 2. The ANN had one input layer, two hidden layers, and one output layer. The input layer was used to receive the input data and the amount of the input layer neurons was adjusted by the type and number of input variables in the dataset. An output layer was used for giving a probability vector for predictions. The hidden layers were used for representing the input vector in a more abstract form. To find the optimum number of neurons for each hidden layer we tested different numbers of neurons between 5 and 50 through an iterative experimentation process. According to the test results seen in Table 3, the most successful RMSE score was obtained when 45 neurons were used in hidden layers 1 and 2. Rectified Linear Unit (ReLU) activation function was used for the hidden layers and the linear activation function was used for the output layer. Mean square error, which is the most commonly used regression loss function, was selected as the loss function. Adam optimizer was used in backpropagation and the learning rate was selected as 0.001.

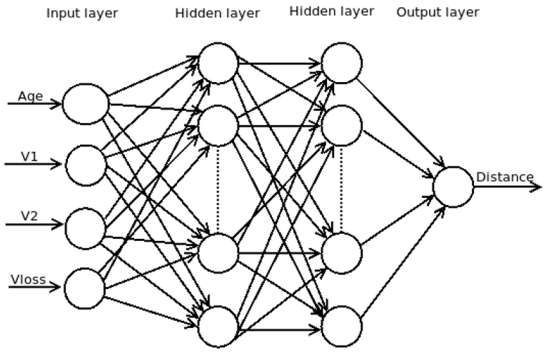


Figure 2. Graphical representation of the ANN architecture developed for the research.

Ridge regression (RR), is a technique used to calculate the approximate result of equations without a unique solution. RR adds a bias to the conventional regression calculation and reduces standard errors. In ridge regression, the alpha value is used for the regularization and it is selected as 1.9 in our model (Figure 3).

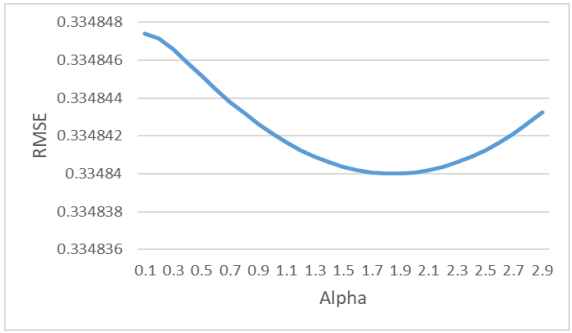


Figure 3. Alpha value vs. root mean squared error for ridge regression.

The *k-nearest neighbors (KNN)* algorithm is another machine learning method that can be easily used to calculate regression problems. In order to increase the efficiency of the KNN model, we determined the optimal value of the neighbor parameter used in the model. In this research k value was selected as 12 and Minkowski distance was selected as the similarity measure (Figure 4).

Table 3. RMSE values for different neuron numbers of first and second hidden layers

Layer1/Layer2	5	10	15	20	25	30	35	40	45	50
5	0.3296	0.3350	0.3289	0.3431	0.3466	0.3412	0.3390	0.3338	0.3427	0.3314
10	0.3264	0.3286	0.3346	0.3235	0.3278	0.3360	0.3328	0.3374	0.3429	0.3406
15	0.4064	0.3166	0.3288	0.3216	0.3219	0.3304	0.3249	0.3359	0.3406	0.3279
20	0.7172	0.3372	0.3272	0.3449	0.3249	0.3297	0.3185	0.3233	0.3287	0.3245
25	0.3317	0.3222	0.3197	0.3410	0.3208	0.3441	0.3354	0.3277	0.3300	0.3385
30	0.3327	0.3149	0.3330	0.3181	0.3216	0.3304	0.3332	0.3319	0.3252	0.3361
35	0.3329	0.3332	0.3217	0.3259	0.3261	0.3255	0.3252	0.3232	0.3292	0.3312
40	0.3188	0.3385	0.3357	0.3271	0.3359	0.3331	0.3426	0.3284	0.3305	0.3255
45	0.3148	0.3333	0.3373	0.3201	0.3336	0.3272	0.3211	0.3239	**0.3096**	0.3251
50	0.3225	0.3161	0.3241	0.3128	0.3195	0.3194	0.3230	0.3174	0.3268	0.3142

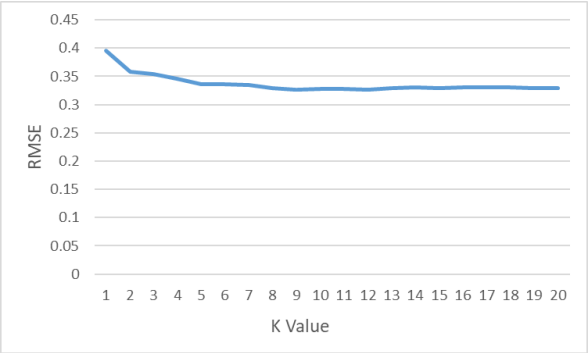


Figure 4. Number of neighbors vs. root mean squared error for KNN.

Decision Trees (DT) can identify different patterns by using dependent and various independent variables as an alternative to regression models (Cox, 2002). The decision tree approach generally establishes heuristic models that make more accurate predictions. The first and last nodes of the decision tree are called root and end nodes, while intermediate nodes are called leaf nodes. The variables in the nodes are checked with the training data set. Starting from the root (the top node), the decision tree algorithm creates the tree from the first node to the end nodes by determining which variable to be tested.

Random Forest algorithm is a very popular and highly sensitive learning algorithm for classification and regression tasks based on decision trees. A random forest consists of a combination of trees created using a random vector that is sampled independently from each input vector (Breiman, 2001). The Random Forest algorithm solves the over-fitting problems of decision trees. Figure 5 shows the test results for parameters of Random Forest algorithm.

Gradient Boosting Regression Trees (GBRT) enables the optimization of arbitrary differentiable loss functions by creating an additive forward stage-wise model. A regression tree fits on the adverse gradient of the specified loss function at each level. It is an accurate and effective model that can be used for the problems of classification and regression. The number of the boosting stages is determined by an iterative experimentation process. Gradient boosting is relatively robust to over-fit, so a big amount generally leads to better performance (in this study we used the 100 boosting stage). Figure 6 shows the test results for parameter selection.

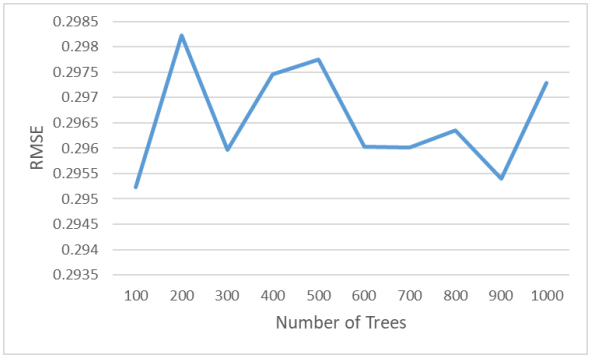


Figure 5. Number of trees vs. root mean squared error for random forest.

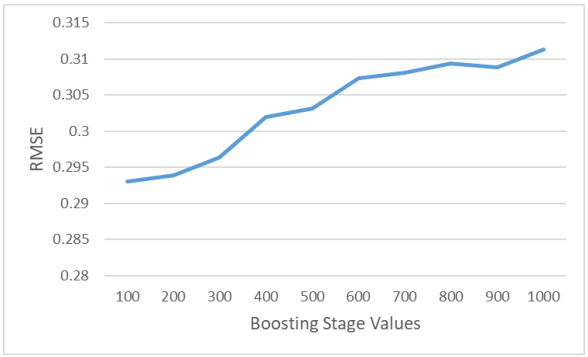


Figure 6. Number of the boosting stages vs. root mean squared error for GBRT.

3. Results

In the experiments, the performance of the methods was evaluated using the 5-cross validation approach. 80% of the data was used for training and 20% for testing in each fold and the experiments were repeated for each test group. To evaluate the prediction successes of algorithms, six error measurement techniques were used. They are the most popular metrics for the accurate evaluation of continuous variables.

Mean Absolute Error (MAE), without considering direction, evaluates the mean errors in the predictions set. It is the mean of the absolute difference between the predicted values and observed values. Mathematically, it is calculated using Eq. 1.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \qquad (1)$$

Root mean squared error (RMSE) measures the average error as quadratic. RMSE is the square root of the average of squared differences between predicted values and observed values. It is calculated using Eq. 2.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \qquad (2)$$

In addition, the results of the Mean Squared Error (MSE), R^2 score, Explained Variance Score (EVS), and Mean Squared Logarithmic Error (MSLE) metrics used in different ML studies are also included.

In this part, the performances of the proposed machine learning algorithms are evaluated. We compared previously developed linear and nonlinear models as well as machine learning techniques such as artificial neural networks, ridge regression, decision trees, K-nearest neighbors regression, random forest, and gradient boosting regression trees. The prediction results of the Nonlinear-1 model (Mikhailov, et al., 1981) are quite different from the real values as mentioned earlier. The estimations of the Nonlinear-2 model (Tiupa, et al., 1982) are slightly behind the results of other estimation algorithms. Although the prediction results of the Linear (Bayraktar and Çilli, 2018) model are better than the previous non-linear models, its success is below the machine learning techniques. The results showed that the GBRT had the lowest error for all used metrics (Table 4).

Table 4. Performance comparisons of machine learning algorithms and other models for distance prediction of all 5-folds average.

Model	MSE	RMSE	MAE	EVS	MSLE	R^2	Time (sec.)
Linear	0.1225	0.3489	0.2732	0.7606	0.0025	0.7546	-
Non-Linear-1	902.08	30.03	29.82	-24.54	2.6422	-1818	-
Non-Linear-2	0.1512	0.3861	0.2934	0.7570	0.0030	0.6975	-
ANN	0.0964	0.3096	0.2365	0.8110	0.0020	0.8061	2.885
Ridge	0.1127	0.3348	0.2643	0.7787	0.0023	0.7726	**0.035**
KNN	0.1082	0.3270	0.2520	0.7905	0.0023	0.7837	0.070
Decision Tree	0.1561	0.3907	0.2963	0.6823	0.0032	0.6736	0.049
Random Forest	0.0874	0.2952	0.2236	0.8287	0.0018	0.8235	0.789
Graident Boosting	**0.0865**	**0.2930**	**0.2198**	**0.8323**	**0.0018**	**0.8238**	0.280

The results of the GBRT algorithm for each fold were shown in detail in Table 5.

Table 5. Results of GBRT algorithm for each fold.

FOLD	MSE	RMSE	MAE	EVS	MSLE	R^2
1	0.09297	0.30491	0.23667	0.84040	0.00189	0.82985
2	0.06300	0.25099	0.19039	0.89252	0.00128	0.88677
3	0.10551	0.32482	0.22731	0.81141	0.00222	0.79088
4	0.09153	0.30254	0.24186	0.76458	0.00193	0.76055
5	0.07937	0.28172	0.20268	0.85237	0.00170	0.85101

The closest and farthest predictions of each method to the real data were given in Table 6. While the closest estimate was made with GBRT, the farthest estimate was made with the Nonlinear-1 method (Mikhailov, et al., 1981).

Table 6. Maximum and minimum difference between actual and predicted data

Algorithm	Max.	Min.
ANN	1.0550336	0.0020013
Ridge	0.9385549	0.0016124
KNN	1.4708333	0.0025000
Decision Tree	1.3000000	0.0000000
Random Forest	0.9080000	0.0009000
Graident Boosting	1.1547455	0.0000923
Linear	1.0223000	0.0001300
Nonlinear-1	39.3241981	20.4664035
Nonlinear-2	1.1759100	0.0028119

The results of the proposed models were compared to the real data in Figure 7. Measured jumping distances and predicted values were shown separately for each of the linear and machine learning methods. The red color indicated the measured distance, while the gray color (dashed line) indicated the results of the prediction methods. As can be seen in the graphs of machine learning methods, the similarity of red and gray lines was higher than the linear method. In addition, it was seen that the similarity of the predictions made with the GBRT algorithm to the real values was significantly more.

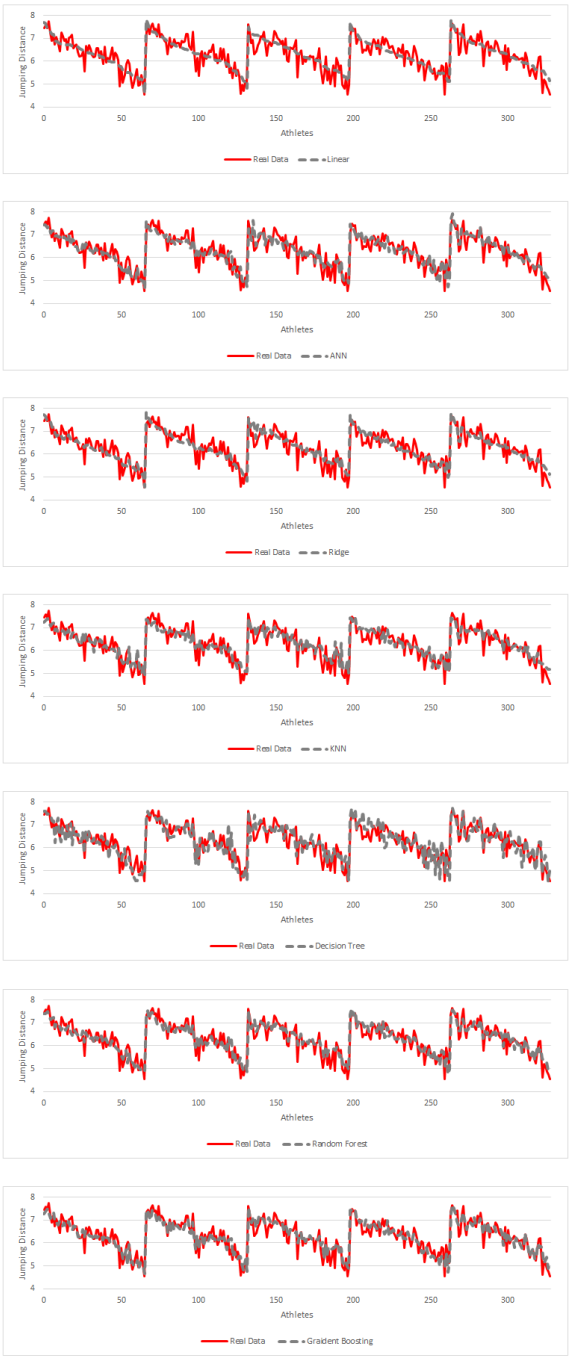


Figure 7. Comparing the predictions to the observed data.

Figure 8 shows the differences between the observed distances and predicted values for the GBRT method (purple color) and the Linear model (dashed line). While plotting the graph, the absolute differences between the estimated results and the actual values for both methods were ordered from largest to smallest. Therefore, predictions which were close to zero were more successful. It is obvious that the predictions yielded by the GBRT method are more successful than the linear method.

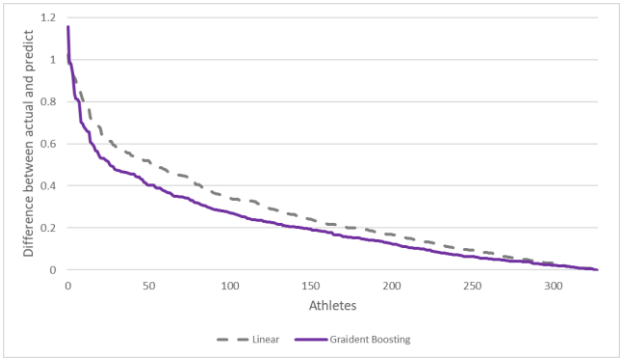


Figure 8. Difference between predicted and observed jump distances for Linear model and GBRT algorithm.

4. Discussion

The results of the study showed that besides the run-up velocity values such as V1 (0.81) and V2 (0.82), the age (0.41) parameter also had an effect on long jump distance. In the literature, the velocity values in the last 10 meters were utilized by most researchers, and models that could be expressed with a first-order or quadratic equation were developed. (Bayraktar and Çilli, 2018; Mikhailov, et al., 1981; Tiupa, et al., 1982).

In previous studies, a model was developed based on the data and the accuracy of the model was tried to be tested with the same data. However, in this study, 5-fold cross validation technique was applied and 80% of data were used for training and 20% of data were used for testing in each fold.

In many studies, it was stated that increasing the average speed would have a direct positive effect on jump distance (Bridgett, et al., 2002; Hay, 1993; Hay, et al., 1986; Lees, et al., 1994; Rahim, et al., 2020; Takahashi & Wakahara, 2019). There were also studies expressing the threshold velocity required for an athlete to jump 8 meters (Linthorne, 2008; Moura, et al., 2005). They argued that the horizontal velocity should be 10.5 m/s or 10.6 m/s. The common feature of all these studies was that the distance increases as the run-up velocity increases. Furthermore, these models did not contain any statement that the age of the athletes also had an effect on distance.

The fact that an action performed by a very complex organism, such as a human being, cannot be explained by a single phenomenon, is valid even in sports branches that are completely individual. For this reason, many physical and psychological researches are carried out in

the literature even in individual branches. It is also a well-known fact that physical and psychological conditions are important factors that may affect the outcome of the competition.

Nowadays, it is not realistic to explain these complex cause-effect relationships with a simple mathematical model. Instead, it is clear that the structure of artificial intelligence, which can easily establish complex relationships and solve complex models, should be utilized in the estimation of results.

It was seen that the machine learning method proposed in this study produced consistent results when compared with the accurate results of linear models. And also proposed model generated much lower errors than the error rates of the linear model.

In addition a web application were developed with the obtained results of this study using the Gradient Boosting Regression Trees algorithm (Figure 9). Trainers may use this application for athletes. When they use the velocities for the 11m-6m section (V1), the 6m-1m section (V2), and age as the input parameters, the program will produce an output (predicted jumping distance) for them. The web link of the used models and dataset is: https://github.com/mrtucar/LongJumpEstimation

5. Conclusions

In this study, a new method for the jumping distance prediction of male long jumpers was proposed based on a machine learning algorithm. To achieve the highest efficiency, various regression algorithms were applied. After the most successful model was determined, a web application was developed that trainers can use.

Figure 9. Developed web application.

It will be easier to calculate the "technical efficiency index" (TEI) (Bayraktar and Çilli, 2018) using the proposed method. With the help of the score calculated as TEI = 100 x Measured Distance / Estimated Distance, trainers will be able to evaluate the status of their athletes according to their jumps. Thus, athletes with a score of less than 100 points, will need to increase their technical skills. The trainers will be able to obtain the TEI values with the help of the proposed method for developing the exercises that will emphasize the technical skills of the athletes.

Considering that records and grades are developed with only a few centimeters today, it is clear that every step taken to improve the athlete's technical skills is valuable. Artificial intelligence applications, which have started to enter all areas of life with developing technology, will help coaches in many athletic events in the near future.

Fuller et al. stated that using ML methods in studies requiring physical activity such as sports branches has not reached a sufficient level yet (Fuller et al., 2022). They also stated that the increase in the number of studies on the use of ML methods in sports fields by using large and open datasets can contribute to the field. Therefore, we consider our study to be a valuable contribution in terms of utilizing and comparing machine learning algorithms that are used for the first time to estimate long jump distance. Furthermore, live predictions of the jumping distance of individual jumps could be attractive for broadcasting purposes.

It is considered that the estimation results of the artificial intelligence model will increase with the addition of body structure information and detailed information of training.

References

Açıkada, C., Arıtan, S., & Yazıcıoğlu, M. V. (1993). Balkan Gençler Şampiyonası Uzun Atlama Yaklaşma Koşusunun Analizi. [Analysis of the 1992 Balkan Junior Championship Long Jump Approach Run.]. *Atlet Bilim ve Teknoloji Dergisi, 9*, pp. 34-40.

Bayraktar, I., & Çilli, M. (2018). Estimation of jumping distance using run-up velocity for male long jumpers. *Pedagogics, psychology, medical-biological problems of physical training, 22*(3), pp. 124-129. https://doi.org/10.15561/18189172.2018.0302

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Bridgett, L. A., Galloway, M., & Linthorne, N. P. (2002). *The effect of run-up speed on long jump performance.* ISBS-Conference Proceedings Archive.

Bridgett, L. A., & Linthorne, N. P. J. J. o. s. s. (2006). Changes in long jump take-off technique with increasingrun-up speed. *24*(8), pp. 889-897. https://doi.org/10.1080/02640410500298040

Bunker, R., & Susnjak, T. (2022). The Application of Machine Learning Techniques for Predicting Match Results in Team Sport: A Review. *Journal of Artificial Intelligence Research, 73*, 1285-1322. https://doi.org/10.1613/jair.1.13509

Cox, L. A. (2002). Data mining and causal modeling of customer behaviors. *Telecommunication Systems, 21*(2-4), pp. 349-381. https://doi.org/10.1023/A:1020911018130

Derse, E., Hansen, J., Tim, O., & Stolley, S. (2012). *Track and Field Coaching Manual*: LA84 Foundation.

Eetvelde, H., Mendonça, L. D., Ley, C., Seil, R., & Tischer, T. (2021). Machine Learning Methods In Sport Injury

Prediction And Prevention: A Systematic Review. *Journal of Experimental Orthopaedics, 8*(1), 1-15. https://doi.org/10.1186/s40634-021-00346-x

Fukasiro, S., & Wakavama, A. (1992). The men's long jump. *New Studies in Athletics, 7*(1), pp. 53-56.

Fuller, D., Ferber, R., & Stanley, K. (2022). Why Machine Learning (ML) Has Failed Physical Activity Research and How We Can Improve. *BMJ Open Sport & Exercise Medicine, 8*(1), e001259. http://dx.doi.org/10.1136/bmjsem-2021-001259

Hay, J. G. (1986). The Biomechanics of the Long Jump. *Exercise and Sport Sciences Reviews/Series, 14*, pp. 401-446. Retrieved from <Go to ISI>://WOS:A1986E165200014

Hay, J. G. (1993). Citius, Altius, Longius (Faster, Higher, Longer) - the Biomechanics of Jumping for Distance. *Journal of Biomechanics, 26*, pp. 7-21. https://doi.org/10.1016/0021-9290(93)90076-Q

Hay, J. G., & Miller, J. A. (1985). Techniques Used in the Transition from Approach to Takeoff in the Long Jump. *International Journal of Sport Biomechanics, 1*(2), pp. 174-184. doi:10.1123/ijsb.1.2.174

Hay, J. G., Miller, J. A., & Canterna, R. W. (1986). The Techniques of Elite Male Long Jumpers. *Journal of Biomechanics, 19*(10), pp. 855-866. https://doi.org/10.1016/0021-9290(86)90136-3

Haykin, S. S. (2009). *Neural networks and learning machines*: Pearson education Upper Saddle River, NJ.

Hommel, H. (2009). *Long Jump (Final Report) - Scientific Research Project Biomechanical Analyses at the IAAF World CH in Athletics Berlin 2009*. https://www.iaaf.org/development/research

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. *Neural Networks, 3*(5), pp. 551-560. https://doi.org/10.1016/0893-6080(90)90005-6

Lees, A., Graham-Smith, P., & Fowler, N. J. J. o. a. B. (1994). A biomechanical analysis of the last stride, touchdown, and takeoff characteristics of the men's long jump. *10*(1), pp. 61-78. https://doi.org/10.1123/jab.10.1.61

Linthorne, N. P. (2008). *Routledge Handbook of Biomechanics and Human Movement Science*: Taylor & Francis. https://doi.org/10.4324/9780203889688

Maier, T., Meister, D., Trösch, S., & Wehrlin, J. P. (2018). Predicting biathlon shooting performance using machine learning. *Journal of sports sciences, 36*(20), pp. 2333-2339. https://doi.org/10.1080/02640414.2018.1455261

Mikhailov, N. G., Yakunin, N. A., & Aleshinsky, S. Y. (1981). Biomechanical assesment of take-off in long jump. *Teoria i Praktika Fizicheskoi Kultury, 5*, pp. 13-15.

Mishra, M. K., & Rathore, V. S. (2016). Speed and agility as predictors of long jump performance of male athletes. *Turkish Journal of Sport and Exercise, 18*(2), pp. 27-33.

Moura, N. A., Moura, T. F., & Borin, J. P. (2005). Approach speed and performance in the horizontal jumps: What do Brazilian athletes do? *IAF NEW STUDIES IN ATHLETICS, 20*(3), pp. 43-48.

Musa, R. M., Anwar, P. P. A. M., Taha, Z., Chang, S. W., Fakhri, A. N. A., & Abdullah, M. R. (2019). A machine learning approach of predicting high potential archers by means of physical fitness indicators. *PLOS ONE, 14*(1), pp. 1-12. https://doi.org/10.1371/journal.pone.0209638

Ofoghi, B., Zeleznikow, J., MacMahon, C., & Dwyer, D. (2010). A Machine Learning Approach to Predicting Winning Patterns in Track Cycling Omnium. *IFIP Advances in Information and Communication Technology presented at the meeting of Third IFIP TC12 International Conference on Artificial Intelligence (AI) / Held as Part of World Computer Congress (WCC), Brisbane, Australia*. https://doi.org/10.1007/978-3-642-15286-3_7

Rahim, M. A., Lee, E. L. Y., Malek, N. F., Suwankhong, D., & Nadzalan, A. M. (2020). Relationship Between Physical Fitness and Long Jump Performance. *International Journal of Scientific & Technology Research, 9*(4):1795-1797.

Schulek, A. (2002). Long jump with supramaximal and normal speed. *IAF NEW STUDIES IN ATHLETICS, 17*(2), pp. 37-46.

Takahashi, K., & Wakahara, T. (2019). Association Between Trunk And Gluteus Muscle Size And Long Jump Performance. *PloS one, 14*(11), e0225413. https://doi.org/10.1371/journal.pone.0225413

Tiupa, V., Aleshinsky, S., Primakov, I., & Pereverzev, A. (1982). The biomechanics of the movement of the body's general centre of mass during the long jump. *Teoria i Praktika Fizicheskoi Kultury, 5*, pp. 21-32.

Whiteside, D., Cant, O., Connolly, M., & Reid, M. (2017). Monitoring Hitting Load in Tennis Using Inertial Sensors and Machine Learning. *International Journal of Sports Physiology and Performance, 12*(9), pp. 1212-1217. https://doi.org/10.1123/ijspp.2016-0683

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 153-160
DOI: 10.38016/jista.1144535

Bireylerin Koroner Arter Hastalığı Risk Seviyesinin Bulanık Uzman Sistem Yaklaşımı İle Belirlenmesi

Çağatay Teke[1*]

[1] Bayburt Üniversitesi, Endüstri Mühendisliği Bölümü, Bayburt, Türkiye

cagatayteke@bayburt.edu.tr

Öz

Koroner Arter Hastalığı (KAH) dünya genelinde insanların hayatını kaybetmesine sebep olan en önemli hastalıklardan biridir. Tıp alanında yaşanan gelişmeler bu hastalığın tedavisini kolaylaştırsa da risk faktörlerinin belirlenmesi ve değerlendirilmesinde hala birtakım yetersizlikler söz konusudur. Bu çalışmada, KAH ile ilgili yaygın belirti ve şikayetleri olan bireyler göz önüne alınarak tanıda kullanılan çeşitli risk faktörleri belirlenmiştir. Ayrıca bulanık uzman sistem yöntemi kullanılarak bireylerin KAH risk düzeylerini tespit etmek amacıyla bir yapay zeka sistemi geliştirilmiştir. Tasarlanan sistem kural tabanlı olup, bu kural tabanı yapısı tıp uzmanlarından edinilen bilgilerle oluşturulmuştur. Sistem, bireylerin hastalık riskini azaltmak için kendi kendine risk değerlendirmesi ve özelleştirilmiş öneriler sunmaktadır. Bu sayede koroner arter hastalığından muzdarip kişilerin sayısındaki artış önlenebilir veya geciktirilebilir.

Anahtar kelimeler: Bulanık uzman sistem, yapay zekâ, risk değerlendirmesi, koroner arter hastalığı, bulanık mantık, kronik hastalıklar.

Determination of Coronary Artery Disease Risk Level of Individuals by Fuzzy Expert System Approach

Abstract

Coronary Artery Disease (CAD) is one of the most important diseases that cause people to die worldwide. Although developments in medicine facilitate the treatment of this disease, there are still some inadequacies in identifying and evaluating risk factors. In this study, various risk factors used in the diagnosis were determined by considering individuals with typical symptoms and complaints related to CAD. In addition, an artificial intelligence system has been developed to determine the CAD risk levels of individuals by using the fuzzy expert system method. The designed system is rule-based, and this rule-based structure was created with the knowledge obtained from medical experts. The system provides self-risk assessment and customized recommendations to reduce individuals' disease risk. In this way, the increase in the number of people who have coronary artery disease can be prevented or delayed.

Keywords: Fuzzy expert system, artificial intelligence, risk assessment, coronary artery disease, fuzzy logic, chronic diseases.

1. Giriş (Introduction)

Dünya çapında yapılan araştırmalara göre, kalp ve damar hastalıkları sebebiyle 2018 ile 2030 yılları arasında yaklaşık 23,6 milyon kişinin hayatını kaybetmesi beklenmektedir. Koroner Arter Hastalığının (KAH), ilgili hastalık grubu içerisinde en yüksek ölüm oranına sahip olduğu belirtilmektedir (Şahan ve Gezer, 2021). Buna bağlı olarak, günümüzde, sağlık sorunları ile ilgili bilimsel çalışmalar insanların daha sağlıklı bir yaşam sürmelerini sağlamak için artmaktadır. Tıbbi gelişmelerin de etkisiyle KAH tedavisi daha erişilebilir hale gelse de, hastaları etkileyen birçok risk faktörü ve bu faktörlerin birbiriyle ilişkisi nedeniyle bu hastalığın teşhisinde güçlükler yaşanmaktadır. Bir bireyin herhangi bir risk faktörü kötü olabilirken bir diğeri çok iyi olabilir. Bu nedenle, çeşitli risk faktörlerinin kombinasyonu ve etkileşimi, risk faktörlerinin insanlar üzerindeki etkisinin belirlenmesinde önemli bir rol oynamaktadır.

* Sorumlu yazar.
E-posta adresi: cagatayteke@bayburt.edu.tr

Alındı : 17 Temmuz 2022
Revizyon : 2 Ağustos 2022
Kabul : 4 Ağustos 2022

Bilgisayar ve yazılım teknolojilerindeki gelişmeler hesaplamalı analizler için kullanılabilse de risk faktörlerinin değerlendirilmesinde ve yorumlanmasında yeterince kullanılamamaktadır. Bunun temel nedeni, bu teknolojilerdeki gelişmelerin temelini klasik mantık kurallarının oluşturmasıdır. Geleneksel bilgisayar teknolojisinin ortaya koyduğu bu sınırlamalar nedeniyle insanlar için risk faktörlerini değerlendirmek ve yorumlamak için yoğun insan emeği ve beyin gücüne ihtiyaç duyulmaktadır. İnsanlar aldıkları eğitim, tecrübe ve edindikleri bilgi ile bunun üstesinden gelirler (Yıldız, 2008).

Bulanık mantık ilk olarak 1960'larda Zadeh tarafından matematiksel bir modelleme yaklaşımı olarak geliştirilmiştir. 1970'lerde Mamdani ve Assilian, bir buhar motorunu bulanık bir sistem modeli aracılığıyla kontrol etmiştir. İlerleyen yıllarda başarılı uygulamaların ardından bulanık mantığa olan ilgi artmış ve ardından 1989 yılında uluslararası bir çalışma ortamı olarak bulanık mantık mühendislik laboratuvarları kurulmuştur (Abduljabar, 2011). Bulanık mantık ile ilgili çalışmalar sonraki yıllarda da devam etmiştir. Örneğin, Allahverdi vd. (2007) koroner kalp hastalığı riskini belirlemek için bulanık bir uzman sistem tasarlamıştır. Sistem kullanıcılara yaşamları, beslenmeleri ve ilaç tedavisi alma durumu hakkında tavsiyelerde bulunmaktadır. Schuster vd. (2002) bir karar destek sistemi aracılığıyla bulanık mantık kullanarak koroner kalp hastalığı riskini değerlendirmiştir. Pal vd. (2012) klinik parametreleri kullanarak bulanık uzman sistem yaklaşımıyla KAH'ı incelemiştir. Çalışma kapsamında, risk değerlendirmesini desteklemek için bulanık bir uzman sistem geliştirilmiştir. Khatibi ve Montazer (2010) koroner kalp hastalığı riskini değerlendirmek için bulanık bir kanıtsal hibrit çıkarım motoru kullanmıştır. Bilgi birleştirme işlemini gerçekleştirmek için kanıtsal birleştirme kuralları kullanılmıştır. Muthukaruppan ve Er (2012) KAH'ı teşhis etmek için bulanık uzman sistem ve parçacık optimizasyonu yaklaşımlarını içeren hibrit bir sistem tasarlamıştır. Bulanık kural tabanı oluşturulurken karar ağaçları kullanılmıştır. Üyelik fonksiyonlarının belirlenmesinde ise parçacık optimizasyonu yaklaşımından faydalanılmıştır. Duarte vd. (2006) miyokard perfüzyon sintigrafisi hastalarını seçmek için bulanık küme teorisi üzerine uygulanan klinik-epidemiyolojik veriler ve koşu bandı testi sonuçlarını kullanmıştır. Adeli ve Neshat (2010) kalp hastalığının teşhisi için Matlab yazılımı aracılığıyla bulanık bir uzman sistem tasarlamıştır. Sikchi vd. (2013) kalp hastalıkları için genel bir bulanık uzman sistem tasarlamıştır. Bu sistem kalp hastalıklarının teşhisinde destekleyici bir araç olarak kullanılmıştır. Parvin ve Abhari (2012) kalp hastalığının teşhisi için bulanık bir veri tabanı oluşturmuştur. Teşhis için kullanılan verilerde belirsizlik olduğunda, bulanık mantık yaklaşımına dayalı veri tabanı karar vericilere doğru veriler sağlamıştır. Maranate vd. (2015) normalleştirilmiş bir ağırlık vektörüne dayalı bulanık bir analitik hiyerarşi süreci kullanarak obstrüktif uyku apnesi risk faktörlerine öncelik vermek için bir çalışma yapmıştır. Dominguez Hernández vd. (2013) atipik glandüler hücrelerde servikal kanseri teşhis etmek için bir uzman sistem geliştirmiştir. Sistem bulanık mantık ve görüntü işlemeye dayanmaktadır. Üç aşaması vardır. Bunlar risk teşhisi, sitolojik bir görüntünün yorumlanması ve kanser öncül yaralanmalarının belirlenmesidir. Sistem, daha doğru tanı için destekleyici bir araç olarak kabul edilmiştir. Abdualimov ve Obrezan (2021) koroner arter hastalığını tahmin etmede yapay zeka tekniklerinden yapay sinir ağlarını kullanmıştır. Bu kapsamda, hastaların elektrokardiyografi ve koroner anjiografi sonuçlarını kullanan bir yapay sinir ağı tasarlanmıştır. Bu yapay sinir ağının çıktısı ise koroner arter lezyonu varlığı olarak tanımlanmıştır. Faieq ve Mijwil (2022) kalp hastalığı teşhisi için destek vektör makineleri ve yapay sinir ağı yöntemini kullanarak tahmin uygulaması gerçekleştirmiştir. Her iki yöntemin tahmin performansları incelendiğinde, koroner arter hastalığı için destek vektör makineleri yöntemi ile yapılan tahminin doğruluğunun daha yüksek olduğu görülmüştür. Atomsa vd. (2022) koroner arter hastalığının teşhisi için bulanık mantık tabanlı bir uzman sistem geliştirmiştir. Mamdani çıkarım mekanizmasına sahip olan bu uzman sistem 174 kuralı bünyesinde barındırmaktadır. Geliştirilen bulanık uzman sistemin performansını ölçmek amacıyla Nijeryada toplanan veriler kullanılmıştır.

Bulanık uzman sistem sağlık, finans, imalat alanlarında sıklıkla kullanılmaktadır (Thani ve Kasbe, 2022; Matinfar ve Golpaygani, 2022; Abdulrahman vd., 2014; Masoumeh vd., 2021; Hernández-Vera vd., 2017; Amelia vd., 2009). Sağlık alanındaki çalışmalar incelendiğinde, bu çalışmaların hastalık teşhisi üzerine yoğunlaştığı görülmektedir (Thani ve Kasbe, 2022; Matinfar ve Golpaygani, 2022; Atomsa vd., 2022; Singla vd., 2020; Sikchi vd., 2013; Parvin ve Abhari, 2012; Muthukaruppan ve Er, 2012; Domínguez Hernández vd., 2013; Arab vd., 2021; Adeli ve Neshat, 2010). Bu çalışmada ise sınırlı sayıda çalışmanın yer aldığı koroner arter hastalığı risk belirlenmesi üzerine yoğunlaşılmıştır.

2. Materyal ve Metot **(Material and Method)**

2.1. Koroner arter hastalığı (Coronary artery disease)

KAH dünyadaki önemli kronik hastalıklardan biridir. Koroner arterler olarak bilinen kalbi besleyen atardamarların daralması veya tıkanması sonucunda kan akışının kısmen veya tamamen durmasıyla oluşur. Bu hastalık önceden fark edilmezse ve gerekli önlemler alınmaz ise kalpte emboli ve ritim bozuklukları nedeniyle kan akışının durmasına, kalp krizine ve ölüme neden olabilir (Anonim, 2021). Dünyadaki ölümlerin çoğu KAH'dan kaynaklanmaktadır. 2018 yılı verilerine

göre, dünyadaki toplam ölümlerin %16,6'sı KAH sebebiyle gerçekleşmiştir. Türkiye'nin de dâhil olduğu Avrupa bölgesinde ise bu oran %25,4'tür (Ahcıoğlu ve Yılmazel, 2021).

Çalışma kapsamında danışılan doktorlardan edinilen bilgilere göre KAH risk faktörleri LDL, HDL, hipertansiyon, sigara kullanımı, diyabet (tip 2), obezite yani vücut kitle indeksi (BMI), fiziksel aktivite, yaş ve genetik yatkınlık olarak sınıflandırılmıştır. Her bir risk faktörü aşağıda açıklanmıştır (Babacan Abanonu vd., 2009):

- Düşük yoğunluklu lipoprotein kolesterol (LDL): Çalışmalar, LDL'nin koroner kalp hastalıkları için en kritik faktör olduğunu göstermiştir. LDL değerinin düşmesinin koroner kalp hastalığı riskini azaltacağı ileri sürülmüştür. LDL kolesterol düzeyi düşük olan kişiler sigara, diyabet, obezite veya hipertansiyon gibi diğer yüksek risk faktörlerine sahip olsalar bile bu hastalık için risk düzeylerinin düşük olduğu gözlemlenmiştir. Amerikan Kalp Derneği tarafından yapılan çalışmalar sonucunda elde edilen verilere göre, LDL kolesterol düzeyi 130 mg/dL'nin üzerinde olan kişilerin KAH açısından yüksek risk taşıdığı gözlemlenmiştir.
- Yüksek yoğunluklu lipoprotein kolesterol (HDL): Sağlık üzerine yapılan araştırmalara göre bireylerin HDL kolesterol düzeyleri ile KAH olma riskleri arasında ters bir ilişki olduğu ortaya konmuştur. Düşük HDL kolesterol seviyeleri (40 mg/dL'den düşük) KAH için riskli iken, daha yüksek seviyelerin (60 mg/dL'den yüksek) kalbi koruyan faktörler arasında olduğu ortaya konmuştur.
- Hipertansiyon: KAH'ın en kritik risk faktörleri arasındadır. Tüm koroner arter vakalarının %35'i hipertansiyondan kaynaklanmaktadır.
- Sigara kullanımı: Sigara içmek KAH riskini 2-3 kat artırır ve diğer risk faktörleri ile birleştiğinde bu risk düzeyinin çok daha fazla artmasına neden olur. Sigara içenlerde kalp krizine bağlı ölüm erkeklerde 2,7 kat, kadınlarda ise 4,7 kat artmaktadır.
- Diyabet: Diyabet, KAH için bir diğer önemli risk faktörüdür ve riski erkeklerde iki kat, kadınlarda dört kat artırır. Düşük HDL ve yüksek LDL, diyabetli kişilerde KAH'a yol açan mekanizmalar arasındadır. Bu nedenle diyabet hastalarının LDL değerinin daha düşük bir seviyede (<100 mg/dL) tutulması gerektiği yapılan çalışmalar ile ortaya konmuştur.
- Obezite (BMI): Amerikan Kalp Derneği tarafından yapılan bir çalışmada obezitenin KAH için en kritik risk faktörlerinden biri olduğu görülmüştür. Yapılan çalışmalar, koroner kalp hastalıkları riskini azaltmak için hafif bir kilo kaybının bile önemli olduğunu göstermektedir.
- Fiziksel aktivite: Yetersiz fiziksel aktivite, riski ortalama olarak iki katına çıkarmaktadır. Düzenli fiziksel aktivitenin LDL kolesterol düzeyini azalttığı, HDL kolesterol düzeyini yükselttiği ve kan basıncını düşürdüğü görülmektedir.
- Yaş: Koroner kalp hastalığı riski yaşla birlikte artar. Kadınlarda 55, erkeklerde 45 yaşından sonra önemli ölçüde bir risk artışı söz konusudur.
- Aile öyküsü: Bireyin birinci derece akrabalarında koroner kalp hastalığı olayının varlığı riski artırmaktadır. Kişi diğer risk faktörlerini ortadan kaldırsa bile bu risk her zaman mevcuttur.

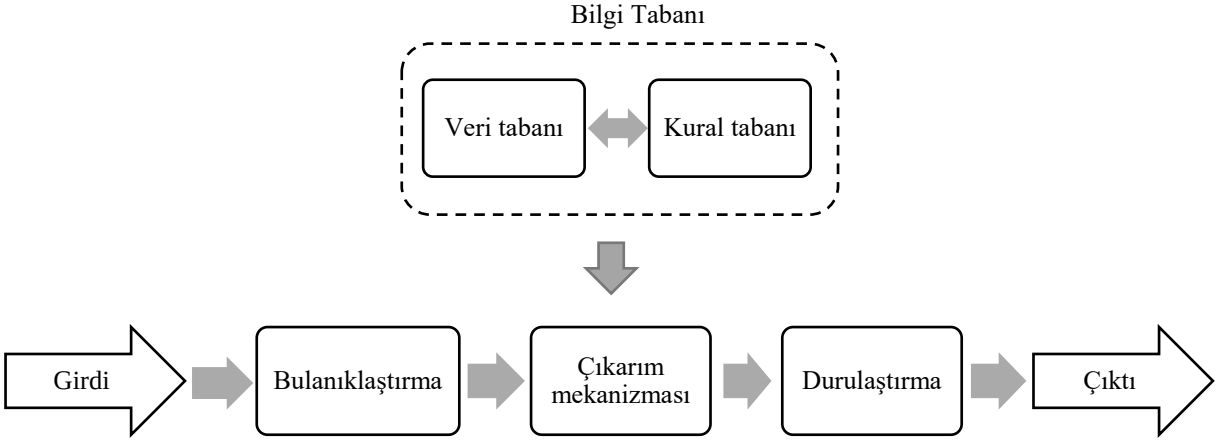
2.2. Bulanık uzman sistem yapısı (Fuzzy expert system structure)

KAH tanısı için karar verme aşamasında kurulacak bir sistemde bulanık mantık ilkelerinin uygulanması, klasik mantık ilkelerinin olumsuz yönlerini ortadan kaldıracaktır. Klasik mantığın temeli olan küme teorisinde bir nesne ya bir kümeye aittir ya da değildir. Böylece klasik kümeler için üyelik derecesinin ya 1 ya da 0 olduğu söylenebilir. Başka bir üyelik derecesi düşünülemez. Bu durum hastaların risk düzeylerinin belirlenmesinde hatalara yol açmaktadır. Öte yandan, bulanık küme teorisinde üyelik derecesi 0 ile 1 arasında değerler alabilir. Diğer bir deyişle üyelik derecesi [0, 1] aralığı arasında değerler alır ve $\mu(x)$ olarak gösterilir. Bu, bireylerin risk düzeylerini belirlemede daha doğru sonuçlar verir (Dobric ve Zarkovic, 2021; Karimi vd., 2022).

Bulanık uzman sistem, belirsizliklerin, çelişkilerin ve dilsel ifadelerin bilgisayar ortamında işlenmesini sağlayan bir yapay zeka teknolojisidir. Bir bulanık sistem dört unsurdan oluşur; bulanıklaştırma, bilgi tabanı, çıkarım mekanizması ve durulaştırma. Bulanık uzman sistemin ana yapısı Şekil 1'de gösterilmektedir. Bu sistemin ilk adımı girdi değişkenlerini belirlemek ve girdi değerlerini bulanıklaştırmaktır. İkinci adım, uzman bilgisi aracılığıyla bir bilgi tabanı oluşturmaktır. Giriş ve çıkış değerleri arasındaki ilişkiler bu adımda belirlenir. Uzman bilgisi kullanılarak oluşturulan kurallara dayalı olarak elde edilen değerler, üçüncü adım olarak çıkarım mekanizmasında işlenir. Son olarak, net değerlerin elde edilmesi için bulanık çıktı değerleri, durulaştırma birimine gönderilir (Singla vd., 2020; Arab vd., 2021).

Girdi değişkenlerinin üyelik fonksiyonları, uzmanlar tarafından edinilen bilgiler kullanılarak oluşturulmuştur. Bu çalışmada, bu bilgileri temsil etme kabiliyeti en yüksek olan üçgensel ve yamuksal üyelik fonksiyonları kullanılmıştır. Üçgenin köşeleri olarak üçgen üyelik fonksiyonu için a_1, a_2 ve a_3 olmak üzere üç parametre vardır. a, b, c ve d, yamuk üyelik fonksiyonunun köşeleri olarak dört parametreye sahiptir. Bu üyelik fonksiyonlarına uygun üyelik derecesi belirleyebilmek için (1) denklemindeki formüllerden yararlanılır.

Bilgi tabanı, bulanık mantık kontrolü ile dilsel ifadelerin kullanılmasını sağlayan bir ara yüzdür. Veri tabanı ve kural tabanı olmak üzere iki kısımdan oluşmaktadır. Veri tabanı, giriş ve çıkış değerlerinin dilsel tanımlarını, üyelik fonksiyonlarını, değişkenlerle ilgili bilgileri ve bulanık mantık kontrolünde kullanılan bulanık fonksiyonların tanımlarını kapsar. Kural tabanı ise KAH için uzmanlar tarafından belirlenen denetim kurallarını içerir. Bu kurallar, KAH'ın girdi ve çıktı



Şekil 1. Bulanık uzman sistemin yapısı (Structure of the fuzzy expert system)

$$\mu(X_i) = \begin{cases} 0 & ,x \le a_1 \; ve \; x \ge a_3 \\ \frac{x-a_1}{a_2-a_1} & ,a_1 < x \le a_2 \\ \frac{a_3-x}{a_3-a_2} & ,a_2 < x < a_3 \end{cases} \qquad \mu(X_i) = \begin{cases} 0 & ,\mathrm{x} \le a \; ve \; x \ge d \\ \frac{x-a}{b-a} & ,\mathrm{a} < x < b \\ 1 & ,\mathrm{b} \le x \le c \\ \frac{d-x}{d-c} & ,\mathrm{c} < x < d \end{cases} \tag{1}$$

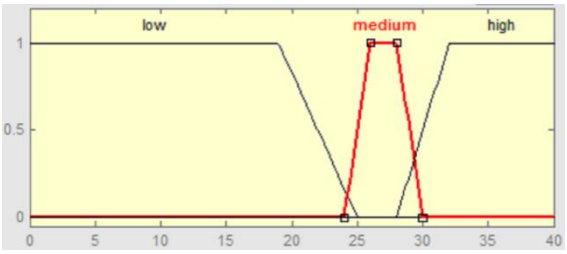
parametreleri arasındaki mantıksal ilişkileri açıklar. "if-then" komutları kuralları oluşturur.

Çıkarım birimi, kontrol fonksiyonunun yürütüldüğü ve karar verme sürecinin gerçekleştiği yapıdır. Bilgi tabanından elde edilen kurallar ve bulanıklaştırma arayüzünden elde edilen bulanık girdiler işlenerek karar verilir. Seçilen mantıksal çıkarım mekanizması ile karar verildikten sonra net olmayan sonuçlar elde edilir (Singla vd., 2020; Arab vd., 2021). Bilginin modelleme türüne göre kural tabanında farklı yöntemler kullanılmaktadır. Bunlar Mamdani, Sugeno, Tsukamoto, Larsen, Şen, Zadeh, Dines-Rescher ve Gödel yöntemleridir. Bu yöntemlerden bazıları spesifik bir alana yönelmişken bazıları ise daha geniş bir kullanım alanına sahiptir (Özkan, 2018). Özellikle Mamdani ve Sugeno yöntemleri yaygın kullanım alanına sahiptir. Bu iki yöntemde girdi değişkenlerinin bulanıklaştırılması ve bulanık mantıkla ilgili işlemler benzerdir. İkisi arasındaki fark üyelik fonksiyonudur. Mamdani yönteminde, kurallar min operatöründen geçirildikten sonra her kuralın çıktı üzerinde ne kadar etkili olduğu sonucuna varılır. Bu çıktılarda max operatörü kullanıldıktan sonra bulanık sonuç elde edilir. Sonuç olarak bir bulanık küme elde edilir. Sugeno yöntemi ise girdi değişkenleri bulanık bir küme olmasına rağmen kesin bir çıktı verir (Vukadinovic, 2013).

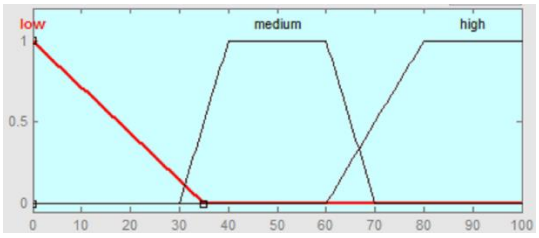
Çıkarım mekanizmasından elde edilen bulanık değerler, durulaştırma ara yüzü vasıtasıyla net değerlere dönüştürülür. Durulaştırmanın farklı yöntemleri vardır. Bunlardan en sık kullanılanı ağırlık merkezi yöntemidir. Burada, üyelik fonksiyonu ile sınırlandırılan alanın ağırlık merkezi, en belirgin parça değeri olarak tanımlanır. Son olarak elde edilen değer sistemin çıktısıdır ve yüzde olarak KAH risk seviyesini verir.

2.3. Uygulama (Implementation)

Bulanık uzman sistem, MATLAB programı bulanık mantık araç kutusu kullanılarak geliştirilmiştir. Bulanık uzman sistemin ilk adımı girdi ve çıktı değerlerinin belirlenmesidir. Sistemde dokuz girdi değişkeni ve bir çıktı değişkeni vardır. Girdi parametrelerinin ve çıktı parametresinin değerleri doktorlardan edinilen bilgilere göre oluşturulmuştur. Girdi değişkenleri ve bunlara ait örnek bir üyelik fonksiyonu Tablo 1 ve Şekil 2'de, çıktı değişkeni ve üyelik fonksiyonu ise Tablo 2 ve Şekil 3'de gösterilmiştir.



Şekil 2. Obezite (BMI) faktörünün üyelik fonksiyonları
(Membership functions of BMI factor)



Şekil 3. KAH risk seviyesinin üyelik fonksiyonları
(Membership functions of CAD risk level)

Tablo 1. Girdi değişkenleri (Input parameters)

Girdi değişkeni	Aralık değeri	Bulanık küme adı
LDL	0-100	Düşük
	90-140	Orta
	130-500	Yüksek
HDL	0-45	Düşük
	40-65	Orta
	60-300	Yüksek
Hipertansiyon	0	Hayır
	1	Evet
Sigara kullanımı (günlük adet)	0-7	Düşük
	4-18	Orta
	12-100	Yüksek
Diyabet (Tip 2)	0	Hayır
	1	Evet
(BMI)	0-25	Düşük
	24-30	Orta
	28-40	Yüksek
Fiziksel aktivite (haftalık, dakika)	0-90	Düşük
	80-170	Orta
	160-250	Yüksek
	240-400	Çok Yüksek
Yaş	0-40	Düşük
	35-55	Orta
	45-100	Yüksek
Aile öyküsü	0	Hayır
	1	Evet

Tablo 2. Çıktı değişkeni (Output parameter)

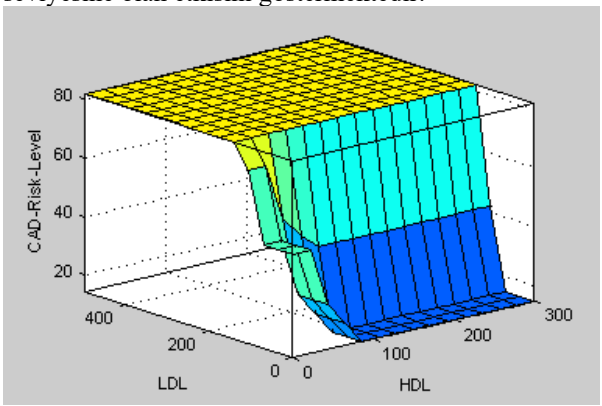
Çıktı değişkeni	Aralık değeri	Bulanık küme adı
KAH risk seviyesi (%)	0-35	Düşük
	30-70	Orta
	60-100	Yüksek

Değişkenler arasındaki ilişkileri anlamak için üyelik fonksiyonları belirlendikten sonra bulanık kural tabanı oluşturulmuştur. Uygulama için oluşturulan bulanık modelin kural tabanı tamamen uzman bilgi ve deneyimlerine dayalı olarak oluşturulmuştur. Mevcut girdi değişkenlerinden beşi üç bulanık kümeye, biri dört bulanık kümeye, diğer üçü ise iki bulanık kümeye ayrılarak aralarındaki etkileşimden 7776 kural elde edilmiştir. Giriş ve çıkış değerlerini kullanan örnek kurallar Şekil 4'de verilmiştir.

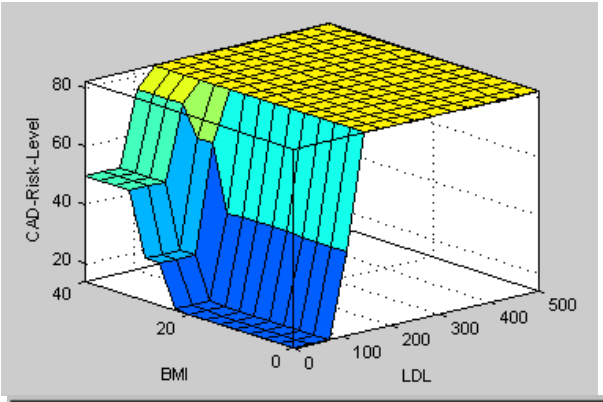
Tasarlanan sistemde çıkarım mekanizması olarak Mamdani yaklaşımı kullanılmıştır. Bir operatörlü tüm kurallar için sistemde girişlerin mantıksal kombinasyonları oluşturulmuştur. Ayrıca kuralların toplama işlemi için max yöntemi kullanılmıştır. Çıkarım sonucu elde edilen bulanık değerler durulaştırma ünitesine gönderilerek gerçek sayılara dönüştürülmüştür. Durulaştırma alt sisteminde ağırlık merkezi yöntemi kullanılmıştır.

3. Bulgular ve Tartışma (Findings and Discussion

Sistemdeki değişkenlerin yüzey görüntüleyici örnekleri Şekil 5 ve Şekil 6'da verilmiştir. Şekil 5, LDL ve HDL değişkenlerinin koroner arter hastalığı risk seviyesine olan etkisini ifade ederken Şekil 6. ise BMI ve LDL değişkenlerinin koroner arter hastalığı risk seviyesine olan etkisini göstermektedir.

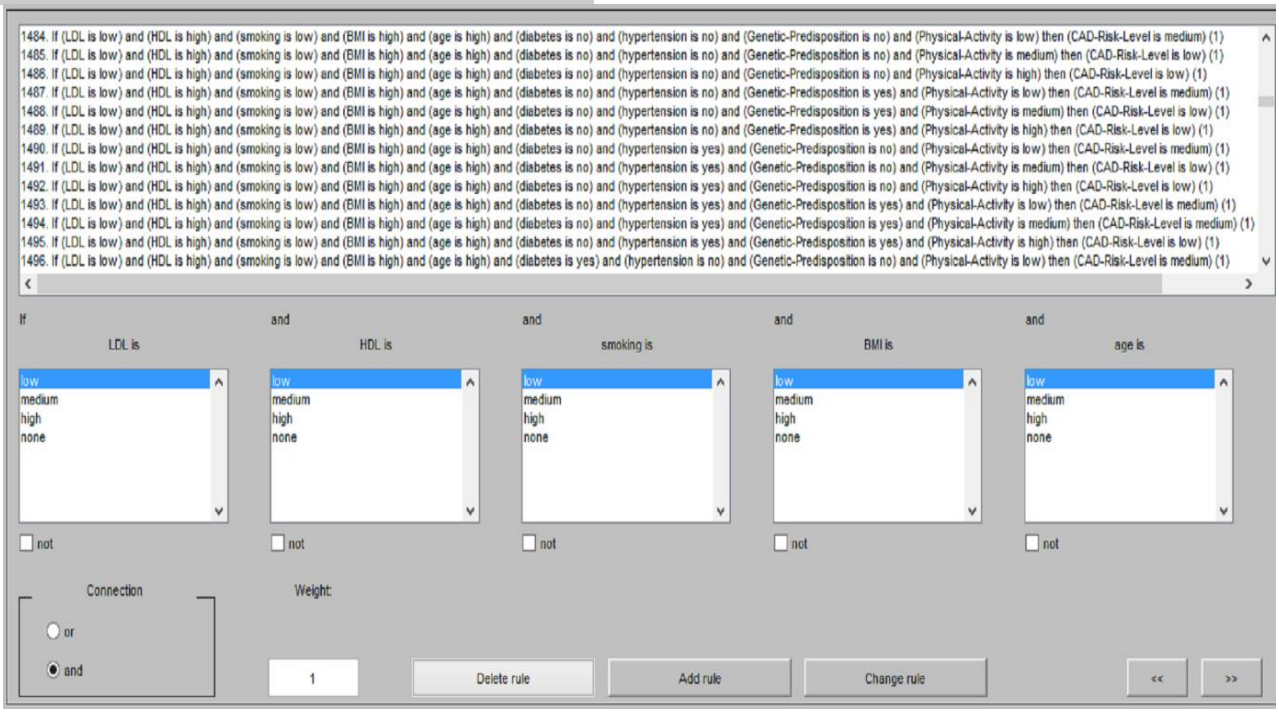


Şekil 5. LDL ve HDL faktörlerine ait yüzey görüntüsü
(Surface viewer of LDL and HDL factor)



Şekil 6. BMI ve LDL faktörlerine ait yüzey görüntüsü
(Surface viewer of BMI and LDL factor)

Tasarlanan sistem ayrıca bireylerdeki KAH risk düzeyini azaltmak için kişiselleştirilmiş ipuçları ve öneriler de sağlar. Örnek girdi verilerine karşılık gelen %83 KAH risk seviyesi için kişiselleştirilmiş ipuçları ve öneriler örneği Tablo 3'te sunulmuştur.



Şekil 4. Tasarlanan sistemin kural tabanından bir kesit (A section from the rule base of the designed system)

Tablo 3. Kişiselleştirilmiş öneriler örneği (Example of customized recommendations)

Bireyin koroner arter hastalığı risk düzeyi %83'tür. Bu nedenle, birey yüksek bir risk düzeyine sahiptir. Kardialji (kalp ağrısı) ve taşikardi (kalp çarpıntısı) koroner arter hastalığının belirtileri olabilir. Bireyin koroner arter hastalığı risk değerinin yüksek olması, bireyin koroner arter hastalığı hastası olma olasılığının yüksek olduğunu gösterir. Kolesterol seviyelerini dengeler ve fiziksel aktiviteler gerçekleştirirseniz koroner arter hastalığı riskini azaltabilirsiniz. Sonuçları doktorunuzla paylaşmalı ve riskinizi tartışmalısınız. Doktorunuz size yardımcı olacak ve bazı tıbbi muayeneler isteyecektir. Vakit kaybetmeden önce elektrokardiyografi (EKG) yaptırmalısınız. Riskinizi azaltmak için acele etmelisiniz.

4. Sonuçlar (Conclusions)

Bu çalışmada, bireylerin KAH risk düzeyini belirlemek için bulanık uzman sistem geliştirilmiştir. KAH'a neden olan faktörlerin etki düzeyleri ve birbirleri üzerindeki etkileri çeşitli belirsizliklere sahiptir. Bu nedenle uzman sistem tasarlanırken bulanık mantık yaklaşımı kullanılmıştır. Ayrıca bulanık kümeler ve kural tabanı oluşturulurken uzman doktorlardan edinilen bilgi ve deneyimlerden faydalanılmıştır.

Geliştirilen sistem, bireyin ilgili tıbbi verileri aracılığıyla bireyin KAH risk değerini yüzde olarak vermektedir. Ayrıca, riski azaltmak için kişiselleştirilmiş öneriler de sunmaktadır. Tasarlanan sistem sayesinde bireyler KAH risk düzeylerini değerlendirebilir ve bu riski azaltabilir. Sistem kolay, anlaşılır ve hızlı olduğu için rahatlıkla kullanılabilir. Ancak bu sistem hastane testleri veya doktor muayenelerinin yerini almayı amaçlamamaktadır.

Bulanık uzman sistem yaklaşımı, gelecekte farklı kronik hastalıklarla ilgili çalışmalarda kullanılabilir. Tüm kronik hastalıklar dikkate alınarak bireylerin toplam kronik hastalık risk düzeyinin belirlenmesi ileri bir çalışma olarak düşünülebilir.

Teşekkür (Acknowledgment)

Koroner arter hastalığı hakkında değerli görüşlerini paylaşan Sakarya Üniversitesi Eğitim ve Araştırma Hastanesi doktorlarına teşekkür ederim.

Kaynaklar (References)

Abdualimov, T.P., Obrezan, A.G., 2021. Prediction of the fact and degree of coronary artery disease using the processing of clinical and instrumental data by artificial intelligence. Vestnik of Saint Petersburg University. Medicine, 16(3), 153–158.

Abduljabar, J.S., 2011. Bulanık mantık yöntemleri kullanılarak gazlı içeceklerde karbondioksit kontrolü. Ankara Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, Türkiye.

Abdulrahman, U.F.I., Panford, J.K., Hayfron-Acquah, J.B., 2014. Fuzzy logic approach to credit scoring for micro finances in Ghana. International Journal of Computer Applications, 94(8), 11-18.

Adeli, A., Neshat, M., 2010. A fuzzy expert system for heart disease diagnosis. International MultiConference of Engineeers and Computer Scientists, 17-19 March 2010, Hong Kong, pp. 1-6.

Ahcıoğlu, A., Yılmazel, G., 2021. Halk sağlığı gözüyle koroner arter hastalığı ve sağlık okuryazarlığı. Türkiye Sağlık Okuryazarlığı Dergisi, 2(2), 81-88.

Allahverdi, N., Torun, S., Saritas, I., 2007. Design of a fuzzy expert system for determination of coronary heart disease risk. International Conference on Computer Systems and Technologies, 14-15 June 2007, Bulgaria, pp. 1-8.

Amelia, L., Wahab, D.A., Hassan, A., 2009. Modelling of palm oil production using fuzzy expert system. Expert Systems with Applications, 36(5), 8735-8749.

Anonim, 2021. Angiography to diagnose the coronary artery disease [Internet]. Baskent University Ankara Hospital. http://www.baskent-ank.edu.tr/saglik-rehberi/oku.php?konu=koroner-arter-hastaliginin-tanisinde-anjiyografi. Erişim Tarihi: 5 Haziran 2021.

Arab, S., Rezaee, K., Moghaddam, G., 2021. A novel fuzzy expert system design to assist with peptic ulcer disease diagnosis. Cogent Engineering, 8, 1-23.

Atomsa, Y., Muhammad, L.J., Ishaq, F.S., Abdullahi, Y., 2022. Feature selection based fuzzy expert system for efficient diagnosis of coronary artery disease. Journal of Clinical and Medical Images, Case Reports, 2(2), 1-9.

Babacan Abanonu, G., Türkyılmaz, E., Güzelbulut, F., Denizli, N., Dayan, A., Okuroğlu, N., Karatoprak, C., Aydın, N., Demirtunç, R., 2009. Koroner arter hastalığı majör risk faktörleri ve c-reaktif proteinin değerlendirilmesi. Haydarpaşa Numune Eğitim ve Araştırma Hastanesi Tıp Dergisi, 49(3), 159-167.

Dobrić, G., Žarković, M., 2021. Fuzzy expert system for metal-oxide surge arrester condition monitoring. Electrical Engineering, 103, 91-101.

Domínguez Hernández, K.R., Aguilar Lasserre, A.A., Posada Gómez, R., Palet Guzmán, J.A., González Sánchez, B.E., 2013. Development of an expert system as a diagnostic support of cervical cancer in atypical glandular cells, based on fuzzy logics and image interpretation. Computational and Mathematical Methods in Medicine, 1–17.

Duarte, P.S., Mastrocolla, L.E., Farsky, P.S., Sampaio, C.R.E.P.S., Tonelli, P.A., Barros, L.C., Ortega, N.R., Pereira, J.C.R., 2006. Selection of patients for myocardial perfusion scintigraphy based on fuzzy sets theory applied to clinical-epidemiological data and treadmill test results. Brazilian Journal of Medical and Biological Research, 39(1), 9–18.

Faieq, A.K., Mijwil, M.M., 2022. Prediction of heart diseases utilising support vector machine and artificial neural network. Indonesian Journal of Electrical Engineering and Computer Science, 26(1), 374-380.

Hernández-Vera, B., Lasserre, A.A.A., Cedillo-Campos, M.G., Herrera-Franco, L.E., Ochoa-Robles, J., 2017. Expert system based on fuzzy logic to define the production process in the coffee industry. Journal of Food Process Engineering, 40(2), e12389.

Karimi, H., Khamforoosh, K., Maihami, V., 2022. Improvement of DBR routing protocol in underwater wireless sensor networks using fuzzy logic and bloom filter. Plos One, 17(2), 1-20.

Khatibi, V., Montazer, G.A., 2010. A fuzzy-evidential hybrid inference engine for coronary heart disease risk assessment. Expert Systems with Applications, 37(12), 8536–8542.

Maranate, T., Pongpullponsak, A., Ruttanaumpawan, P., 2015. The prioritization of clinical risk factors of obstructive sleep apnea severity using fuzzy analytic hierarchy process. Computational and Mathematical Methods in Medicine, 1–13.

Masoumeh, Z., Mohamad, K., Hasan, J., 2021. Design of a new fuzzy expert system for project portfolio risk management. Innovation Management and Operational Strategies, 1(4), 403-421.

Matinfar, F., Golpaygani, A.T., 2022. A fuzzy expert system for early diagnosis of multiple sclerosis. Journal of Biomedical Physics & Engineering, 12(2), 181-188.

Muthukaruppan, S., Er, M.J., 2012. A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease. Expert Systems with Applications, 39(14), 11657–11665.

Özkan, M., 2018. Bulanık çıkarım sistemi ile bireysel personel performansının değerlendirilmesinde bir uygulama. Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi, 19(2), 372-388.

Pal, D., Mandana, K.M., Pal, S., Sarkar, D., Chakraborty, C., 2012. Fuzzy expert system approach for coronary artery disease screening using clinical parameters. Knowledge-Based Systems, 36, 162–174.

Parvin, R, Abhari, A., 2012. Fuzzy database for heart disease diagnosis. Medical Processes Modeling and Simulation of the 2012 Autumn Simulation Multi-Conference, 28-31 October 2012, USA.

Schuster, A., Adamson, K., Bell, D.A., 2002. Fuzzy Logic in a decision support system in the domain of coronary heart disease risk assessment. In: Barro S., Marin R. (ed.) Fuzzy logic in medicine, Physica, Heidelberg.

Sikchi, S.S., Sikchi, S., Ali, M.S., 2013. Design of fuzzy expert system for diagnosis of cardiac diseases. International Journal of Medical Science and Public Health, 2(1), 56–61.

Singla, N., Sadawarti, H., Singla, J., Kaur, B., 2020. Development of multilayer fuzzy inference system for diagnosis of renal cancer. Journal of Intelligent & Fuzzy Systems, 39, 885–898.

Şahan, D., Gezer, D., 2021. Koroner arter hastalarında çevrimiçi sağlık uygulamalarının kullanımı. Van Sağlık Bilimleri Dergisi, 14(1), 106-113.

Thani, I., Kasbe, T., 2022. Expert system based on fuzzy rules for diagnosing breast cancer. Health and Technology, 12, 473-489.

Vukadinovic, D., 2013. Fuzzy logic: applications, systems and Technologies, Nova Science Publishers, New York.

Yildiz, B., 2008. Ratio analysis with fuzzy logic: an emprical study. World Account Sci., 10(2), 183–205.

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 161-167
DOI: 10.38016/jista.1118740

Gemi Çeşitlerinin Derin Öğrenme Tabanlı Sınıflandırılmasında Farklı Ölçeklerdeki Görüntülerin Kullanımı

Emirhan Kıran[1],Bahadır Karasulu[2], Emin Borandağ[3*]

[1] Havelsan, Yazılım Geliştirme Bölümü, İstanbul, Türkiye

[2] Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği Bölümü, Çanakkale, Türkiye

[3] Manisa Celal Bayar Üniversitesi, Yazılım Mühendisliği Bölümü, Manisa, Türkiye

ekiran@havelsan.com.tr, bahadirkarasulu@comu.edu.tr, emin.borandag@cbu.edu.tr

Öz

Günümüzde lojistik ve deniz ulaşımına dayanan ticaret oldukça önem kazanmıştır. Buna dair oluşan trafik göz önüne alındığında gemi çeşitlerinin sınıflandırılarak ayrıştırılması, taşıma, depolama maliyetleri açısından ve güvenlik konusunda önem arz etmektedir. Deniz üzerinde farklı görevleri icra etmekte olan gemilerin sınıflandırılması bu çalışmada ele alınarak, derin öğrenme yöntemleri sayesinde yüksek doğrulukta bir gemi sınıflandırma yapılabilmesi için gemi görüntüleri veri kümesi oluşturulmuştur. Veri kümesinden elde edilen içeriği ifade ederken özniteliklerin daha yüksek seviyeden anlamsal olarak zengin olmasından dolayı, klasik makine öğrenmesi yöntemine kıyasla derin öğrenme çalışmamızda tercih edilmiştir. Derin öğrenme modellerinin eğitiminde ve test edilmesinde kullanılmak üzere bu veri kümesi açık kaynaklı Internet sitelerinden ağ kazıma (web scraping) yöntemi sayesinde çeşitli gemi görüntülerinin edinimi ile oluşturulmuştur. YOLOv5 ve Xception derin öğrenme modelleri eğitilerek en uygun sınıflandırma başarımı elde edilmiştir. Deneyler sonucunda her iki model ile yaklaşık olarak %96 ilâ %99 arası doğruluk oranında başarım değerine ulaşılmıştır. Varılan bilimsel bulgulara ve tartışmaya çalışmamızda yer verilmektedir.

Anahtar kelimeler: Derin Öğrenme, Gemi Sınıflandırma, Görüntü İşleme.

Deep Learning based Ship Variants Classification Using Different Scale Images

Abstract

Nowadays, trade based on logistics and sea transportation has gained importance. Considering the traffic related to this, the classification and discrimination of ship types are important in terms of transportation and storage costs, and safety. The classification of ships performing different tasks on the sea is handled in this study, and a ship image dataset has been created in order to make a high accuracy ship classification thanks to deep learning methods. It is preferred in our deep learning study in comparison to the classical machine learning method, as the features are semantically richer as the higher level, while expressing content from the dataset. It was created by the acquisition of various ship images thanks to the web scraping method. YOLOv5 and Xception deep learning models were trained to obtain the most appropriate classification performance. As a result of the experiments, an accuracy rate of approximately between 96% and 99% was achieved with both models. Scientific findings and discussion are included in our study as well.

Keywords: Deep Learning, Ship Classification, Image Processing.

* Sorumlu yazar.
E-posta adresi: emin.borandag@cbu.edu.tr

Alındı : 19 Mayıs 2022
Revizyon : 1 Ağustos 2022
Kabul : 17 Ağustos 2022

1. Giriş (Introduction)

Sayısal görüntüler kullanılarak yapılan gemi sınıflandırılması, denizlerdeki trafiğin yönetilmesi ve izlenmesi, güvenlik kontrol uygulamaları gibi birçok alanda kullanılmaktadır. Gemi sınıflandırma problemleri için zor olan nokta; çok çeşitli gemilerin olması ve bazı gemilerin birbirine çok benzemesidir. Bunların yanı sıra, iyi bir veri kümesi elde edebilmek de ayrı bir problemdir. Gemi tanıma ve sınıflandırma amacıyla şablon eşleştirme uygulamalarında anahtar noktaların elde edilmesi için betimleyici özniteliklerin oluşturulmasında Ölçekle Değişmez Öznitelik Dönüşümü (Scale-Invariant Feature Transform, SIFT) yöntemi (Lowe, 2004) elde ettiği öznitelik ve eşleştirmeler bakımından yüksek bir başarım sonucu verememektedir.

Dolapcı ve Özcan'ın 2021 yılındaki çalışmasında; gemi görüntülerinin bloklarının da melez öznitelik vektörleri ile temsil edilmesi gerçekleştirilerek, Apache Spark'daki makine öğrenmesi yöntemleri ile kullanılarak görüntülerin sınıflandırılması sağlanmıştır. Naif Bayes, Karar Ağaçları ve Rastgele Orman yöntemleri kullanılarak sınıflandırma gerçekleştirilmiş, kümeleme mimarisi ile çok daha hızlı bir sınıflandırma sonucunda geliştirdikleri melez yöntem ile %99,62 oranı elde etmişlerdir (Dolapcı ve Özcan, 2021).

Derin öğrenme için oluşturulan modeller, katmanlı yapıları sayesinde veriden elde edilen özniteliklerin temsili öğrenme (representation learning) yoluyla alt seviye özniteliklerden yola çıkarak daha üst seviyede özniteliklerin içerildiği haliyle anlamlı bilginin elde edilmesini otomatik bir biçimde sağlamayı hedeflemektedir. Derin öğrenme alanında temel taşlarından biri olarak evrişimli sinir ağları (convolutional neural networks, CNN) (LeCun vd., 1989) sınıflandırma problemlerinde sağlamış olduğu öznitelik soyutlaması tabanlı temsili öğrenme nedeniyle yüksek başarım elde edilmesini sağlamaktadır. Evrişimli sinir ağları (ESA) öznitelik elde edilmesini otomatikleştiren güçlü yeteneklere sahip olan bir yapıdır. Çalışmamızdaki gemi sınıflandırma ve tanıma için ESA tabanlı sinir ağı modelleri olan iki model kullanılmaktadır. Bu yapılar, çalışmadaki on üç farklı sınıfa ait gemilerin sınıflandırılmasında güçlü birer sınıflandırıcı ve gemi tipi ayrıştırıcısı olarak kullanılmıştır.

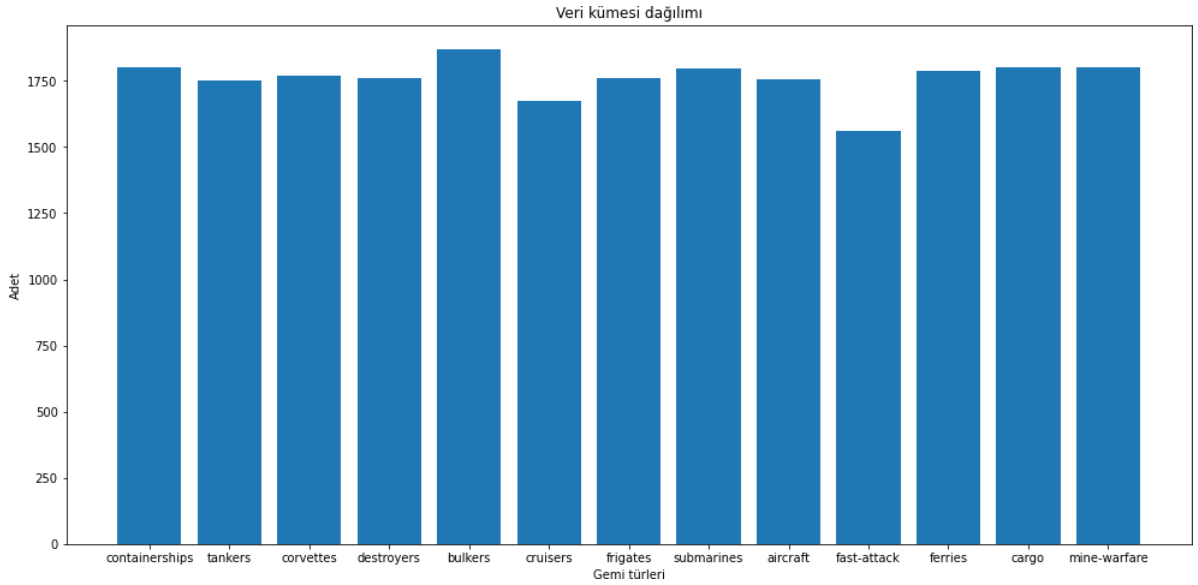
Literatürdeki benzer bir çalışma olarak Zhenzhen ve arkadaşlarının gerçekleştirdiği (Zhenzhen vd., 2019) çalışma incelendiğinde elde edilen başarım değerleri ve sınıflandırılmak istenen gemi çeşitlerinin sayısının yetersiz olduğu görülmüştür. Bu nedenle ilgili çalışmadan farklı olarak yeni bir veri kümesi oluşturulmuş, YOLOv5 (Jiang vd., 2022) derin öğrenme mimari modeli ve ön eğitimli Xception (Chollet, 2016) derin öğrenme mimari modeli öğrenme transferi (transfer learning) yoluyla eğitilerek sonuçları en uygun sınıflandırma sonuç kararının verilebilmesi için kullanılmıştır. Deneylerdeki gerçekleştirilen eğitim sonucunda derin öğrenme mimari modelleri olarak YOLOv5 ve Xception modelleriyle yaklaşık %96 ilâ %99 arasında doğruluk oranı olarak başarım elde edilmiştir. Çalışmamız beş bölümden oluşmaktadır. İkinci bölümde, çalışma kapsamında oluşturularak deneylerde kullanılan gemi görüntüleri veri kümesinin detayları verilmektedir. Üçüncü bölümde çalışmadaki materyal ve yöntemlerin hakkında bilgiler verilerek, deneylerde kullanılan derin öğrenme modellerinin teknik detaylarına yer verilmektedir. Dördüncü bölümde deneylerden elde edilen nicel sonuçlar, bu sonuçlara dair yorumlamalar ve derin sinir ağı mimari modellerinin kıyaslamaları verilmektedir. Beşinci bölümde sonuçlar üzerinden varılan bilimsel bulgular değerlendirilerek yapılan tartışmaya yer verilmiştir.

2. Veri Kümesi (Dataset)

Teknolojinin de gelişmesiyle birlikte denizcilik ve gemi üretiminde de gelişmeler hızla devam etmektedir. Hızla gelişen denizcilik faaliyetleri, farklı görevler için farklı gemilerin üretilmesini gerektirmektedir. Genel olarak bakıldığında sivil ve askeri olmak üzere iki farklı gemi çeşidi bulunmaktadır. Daha detaylı incelendiğinde sivil gemiler; yolcu gemisi, yük gemisi, konteyner gemisi ve kargo gemisi tipi bulunduğu görülmektedir. Bunların haricinde askeri gemiler ise; firkateyn, destroyer, uçak gemisi, deniz altı olmak üzere alt türlere ayrılmaktadır. Çalışmamızda sekiz farklı savaş gemisi ve beş farklı sivil gemi olmak üzere toplamda on üç farklı tipte gemi sınıflandırılması gerçekleştirilmiştir.

Derin öğrenme modelleri ön eğitimsiz olduğunda eğitimin sağlıklı olarak başarımını artırmak için oldukça fazla veriye ihtiyaç duyulmaktadır. Internet sitelerinde bulunan çeşitli çeşitli halka açık görüntü veri kümeleri göz önüne alındığında veri sayısının az, veri kalitesinin yetersiz ve dengeli olmayan veri dağılımına sahip veri kümelerinin bulunduğu görülmektedir. Bu sorunu çözmek adına Python programlama dili için geliştirilen HTTP/1.1 altyapısını kullanarak doğrudan sorgu cümleleri belirtmeye ve POST verisini kodlamaya gerek kalmadan kullanımı sağlayan "*requests*" (Reitz, 2022) ve HTML/XML dosyalarından veri çekmeye yarayan ve ayrıştırma ağaç yapısını düzenleme, arama ve yönetmede kullanılan "*BeautifulSoup*" (Richardson, 2021) kütüphaneleri sayesinde iStock (Livingstone, 2000), Unsplash (Cho, 2013), Pexels (Obrecht, 2014), GettyImages (Getty, 2007) Vessel Finder ve Marine Traffic (Lekkas, 2007) Internet sitelerinden ağ kazıma (web scrapling) yöntemi ile 13 farklı sınıf için 22.900 adet görüntü kullanılmıştır. Bu görüntülerden 20.300 âdeti eğitim geri kalan 2.600 adet ise test verisi için kullanılmıştır. Bu verilerin YOLOv5 modeli eğitimlerinde kullanılabilir hale gelebilmesi adına "*LabelImg*" (Rosenfeld vd., 2015) isimli bir programla nesneleri içeren referans bilgi halinde etiketlenmesi (annotation) gerekmektedir. Veri kümesindeki tüm görüntüler

içerdikleri nesnelere göre bu yöntem ile etiketlenmiştir. Şekil 1'de veri kümesindeki verilerin sınıflara göre dağılımı görülmektedir.



Şekil 1 Veri kümesi sınıfların dağılım grafiği (Bar chart plot for ship variant dataset classes)

3. Materyal ve Yöntemler (Material and Methods)

Sayısal görüntüler ile gemi sınıflandırma ve tanıma probleminin çözülebilmesi için ilk olarak veri kümesi üzerinde etiketleme ve ön işleme adımları tamamlanmıştır. Ön işleme aşamasında, ağ kazıma (web scraping) ile elde edilen 30.416 adet veri incelenmiştir. Elde edilen bazı görüntülerin boyut ve çözünürlük değerleri uygun olmadığı görülmüş ayrıca elde edilen veri kümesi üzerinde incelemeler yapılarak gemi içi, sahil ve mürettebat gibi geminin algılanmasını zorlaştıranlar tespit edilerek veri kümesinden silinmiştir. Bu ön işleme aşaması sonucunda 22.900 adet görüntü içeren bir veri kümesi elde edilmiştir. Bu sayede verilerin deneylerde kullanılan sinir ağı modellerinin girdi biçimine uygun hale gelmesi sağlanmıştır. Sonraki aşamadaysa ESA tabanlı YOLOv5 ve Xception modelleri için eğitim gerçekleştirilmiştir. Bu bölümde deneylerde kullanılan bu modellerin teknik detaylarından bahsedilmektedir.

3.1. Evrişimli Sinir Ağı Tabanlı Modeller (Models Based on Convolutional Neural Network)

Çalışmamızdaki deneylerde kullanılmak üzere Xception ve YOLOv5 modelleri programatik yoldan oluşturularak, girdi verisi çalışmamızdaki veri kümesindeki görüntüler olacak şekilde bu modellere verilmiştir. Literatürdeki kullanım amacıyla Xception modeli, "*ImageNet*" isimli 1.000 sınıflı büyük bir veri kümesiyle eğitildiğinden dolayı ön eğitimli olarak kullanılarak öznitelik elde etme, eğitim transferi (transfer learning) ve ince ayarlama (fine tuning) açısından deneylerimizdeki sınıflandırma görevlerinin başarımını artırmaktadır. ImageNet, temel aldığı hiyerarşinin her bir düğümünün yüzlerce ve binlerce görüntüyle gösterildiği *WordNet* hiyerarşisine göre düzenlenmiş bir görüntü veritabanıdır. YOLOv5 modeli danışmanlı eğitim (supervised learning) tabanlı çalıştığı için etiketli veriye ihtiyaç duymakta, eğer etiketli veri sınıfları yeterince iyi tanımlanmışsa başarımı artmaktadır. Eğitim ortamının oluşturulmasında Tensorflow v2.0 (Abadi vd., 2016), Keras v2.8.0 (Chollet, 2016), Numpy (Harris vd., 2020), Pandas (Daniel, 2018) PyTorch 1.10.0 (Lorica, 2017) kütüphaneleri kullanılarak veri işleme aşamaları gerçekleştirilmiştir.

Çalışmamızda eğitimi gerçekleştirilen ilk model Xception tabanlıdır. Bu model görüntülerden öznitelik elde etmek için evrişim katmanları, boyut indirgemek amacıyla azami biriktirme (max pooling) katmanını ve tüm bu özellikleri birleştirmek için ise tam bağlı (fully connected) katmanları kullanmaktadır. Bunların dışında normalizasyon ve düzenleme işlemleri için de çeşitli katmanlar tercih edilmiştir. Bunlara ek olarak her evrişimli ve tam bağlı katmanlarda doğrusal olmayan olan düzenlileştirilmiş doğrusal birim (Rectified Linear Unit, ReLU) aktivasyon fonksiyonu seçilmiştir. Geliştirilen bu model Xception tabanlıdır fakat bazı kısımları değiştirilmiştir. Çalışmaya özgü sınıflandırma işlemi için temel alınan modelin tam bağlı katmanları devre dışı bırakılmıştır. Bunun yerine iki boyutlu genel ortalama biriktirme

(GlobalAveragePooling2D) katmanı ve Yoğun (Dense) katmanları modelin sonuna eklenerek Xception tarafından elde edilen öznitelik haritası indirgenerek algılayıcı ağı biçimindeki tam bağlı katmana sınıflandırılmak üzere geçirilmiştir, bu yolla gemi sınıflandırma problemi için uygun bir model elde edilmiştir.

Çalışma kapsamında eğitimi gerçekleştirilen ikinci model YOLOv5 tabanlıdır. YOLO (You Only Look Once) görüntüleri bir ızgara sistemine bölen nesne algılama tabanlı bir yapıdır (Jiang vd., 2022). Izgaradaki her hücre, kendi içindeki nesneleri algılamaktan sorumludur. YOLO, hafif sıklet bir model büyüklüğüne sahip olduğundan ve veri iyileştirme sağladığı için hızı ve doğruluğu oldukça yüksek, en ünlü nesne algılama modellerinden biri olarak kabul edilmekte (Jiang vd., 2022), Xception'dan farklı olarak sınıflandırılan nesnelerin konumlarını da belirlemektedir. YOLOv5 modelinin mimarisini yakından bakılacak olursa görsel verilerden öznitelik elde etmek için evrişim katmanlarını, boyut indirgemek için de azami biriktirme katmanı kullanılmakta, fakat. Xception'dan farklı bir yapıda tasarlanmıştır (Chollet, 2016).

Çalışmamızdaki YOLOv5 modelinin farklı alt modelleri mevcuttur. Bunlar belirlenmek istenilen nesnelerin boyutlarına bağlı olarak tercih edilmektedir. Tespit edilen nesne olarak gemi haricinde kalan diğer nesneler arka plan nesneleri olarak sınıflandırılmıştır. Gemi tanıma ve sınıflandırma problemine uygun olarak rota ölçekli ve büyük ölçekli modeller ile eğitim işlemleri gerçekleştirilmiştir. Başarım ve sınıflandırma ölçütlerine bakıldığında büyük ölçekli modelin daha başarılı olduğu görülmektedir. Gerçekleştirilen çalışma kapsamında büyük ölçekli modelinin kullanılması uygun görülmüştür. Buna istinaden elde edilen deney sonuçlarında 13 adet gemi sınıfının yanı sıra arka plan sınıfına ait değerlerde elde edilmiştir.

4. Deneysel Sonuçlar **(Experimental Results)**

Deneylerde kullanılan görüntüler, eğitim ve test aşamalarında kullanılmak üzere ayrılmıştır. Deneylerde, 20.300 adet görüntü verisi ile eğitilen derin öğrenme modellerini test etmek için veri kümesinde ayırmış olduğumuz her bir sınıf için 200 adet görüntüden oluşan toplam 13 sınıf için 2.600 adet veri ile test işlemleri gerçekleştirilmiştir. Bu test işlemleri sonucunda elde edilen ortalama doğruluk oranı YOLOv5 için %96,6 iken Xception modeli için %99,44 olarak bulunmuştur. Test sonuçlarını derinlemesine incelediğimizde; bazı savaş gemilerinin birbirlerine benzerliğinden dolayı başarım oranı kategori bazında ortalamanın altında kalmıştır. Girdi görüntüleri farklı Internet sitelerinden edinildiği için farklı ölçeklerde ve çözünürlüklerde olmaktadır. Bunlar özellikle YOLOv5 ve Xception modelleri için uygun ölçek ve çözünürlüklere indirgenerek girdi görüntüleri kümesine eklenmişlerdir. YOLOv5 için deneylerde kullanılan renkli üç kanal (Kırmızı-Yeşil-Mavi, RGB) girdi görüntüsü çözünürlüğü *640x640* iken, Xception için *299x299* çözünürlüktedir.

Çalışmamızda bilgi elde etme kuramındaki nesnel ölçütler olarak, duyarlık (precision) ve anma (recall) ölçütleri deneydeki modellerin başarımlarının ölçülmesinde ve değerlendirilmesinde kullanılmıştır. Bu ölçütlere batığımızda; her gemi sınıfı için hesaplanarak sonuçlar incelenmiştir. Bu ölçütlerde sıklıkla kullanılan belirli nesnel ölçümler göz önüne alınır. Buna göre doğru pozitif (true positive, TP), doğru negatif (true negative, TN), yanlış pozitif (false positive, FP) ve yanlış negatif (false negative, FN) ölçümleri belirli kıyaslama yapılacak ölçütlerin hesaplanmasında da kullanılmaktadır. Duyarlık ölçütü pozitif olarak tahminlenen değerlerin kaç tanesinin gerçekten de pozitif değerli olduğunu ifade etmektedir ve Eşitlik (1) ile verilmektedir (Karasulu, 2018).

$$Duyarlık = |TP| / (|TP| + |FP|) \quad (1)$$

Anma ölçütü ise ilgili modelin pozitif olarak tahmin etmesi gerekirken ne kadar değeri pozitif tahmin ettiğini gösterir ve Eşitlik (2) ile verilmektedir (Karasulu, 2018).

$$Anma = |TP| / (|TP| + |FN|) \quad (2)$$

Doğruluk oranı ölçütü, bir modelin başarımını ölçmede kullanılan en genel ölçüttür, fakat tek başına yeterli kanı oluşmasına yardımcı olamayabilir. Bu nedenle diğer ölçütleri de göz önüne almak gerekir. Doğruluk oranı ölçütü, Eşitlik (3) ile ifade edilmektedir (Karasulu, 2018).

$$Doğruluk = (|TP| + |TN|) / (|TP| + |FP| + |TN| + |FN|) \quad (3)$$

F1-skoru ise duyarlık ve anma'nın harmonik ortalaması olarak ifade edilir. Yukarıda bahsi geçen ölçütler ondalık sayı olarak veya yüz ile genişletilerek yüzdelik oran olarak da literatürdeki çalışmalarda kullanılabilmektedir. Bu bölümde her iki modelin bağımsız deneysel sonuçlarına nesnel ölçüt değerleri üzerinden yer verilmektedir.

YOLOv5 modelinin eğitim işlemlerinden sonra yapılan test işlemlerinde yüksek bir doğruluk oranında başarıma sahip olduğu gözlemlenmektedir. Buna göre; test işleminde 50 eğitim yineleme (epoch) sonucunda ortalama olarak %96,6 doğruluk oranı, %93,6 duyarlık, %93,6 anma değerleri elde edilmiştir. Sınıflandırma problemi için yapılan eğitim esnasında 0,0092 kayıp değerine kadar inen bir iyileştirme söz konusudur. Tablo 1'de YOLOv5 için deneyler sonucunda elde edilen gemi türlerine ait nesnel ölçüt değerleri yüzdelik oran olarak gösterilmiştir. Özellikle bu tabloda Xception modeli ile YOLOv5 modelinin daha kolay karşılaştırılabilmesi için deneylerimizde kullanılan 13 gemi sınıfı üzerinden bir gösterim yapılmıştır. Tabloda

görüldüğü üzere en yüksek doğruluk oranına sahip olan "*Aircraft*" gemi çeşidi olmuştur. Bunun başlıca sebebi görüntü içerisinde oransal olarak daha fazla miktarda özniteliğin bu gemi çeşitleri için tespit edilmiş olmasıdır.

Tablo 1. YOLOv5 ile bulunan gemi çeşitleri için nesnel ölçüt değerleri (Objective metric values for ship variant detected by YOLOv5)

Gemi Çeşitleri	Duyarlık	Anma	Doğruluk oranı
Aircraft	%99,1	%97,0	%99,4
Destroyer	%88,9	%88,9	%94,7
Containership	%98,5	%97,9	%99
Bulker	%96,2	%90,4	%95
Cargo	%96,2	%97,5	%97,8
Corvette	%87,4	%87,3	%92,1
Frigate	%85,1	%87,7	%93
Submarine	%97,9	%94,1	%99,3
Cruiser	%88,6	%94,7	%93,8
Tanker	%93,8	%94,2	%97,6
Ferrie	%96,2	%97,2	%99,1
Mine-warfare	%94,1	%98,0	%99,2
Fast-attack	%93,0	%91,3	%95,6
Bütün Sınıflar için Başarım Ortalama Değerleri	**%93,6**	**%93,6**	**%96,6**

YOLOv5 modelinin deneyler sırasında ilgili gemi sınıflarını ayrıştırırken sınıf farklılıklarını ne kadar doğru bir biçimde ortaya koyarak sınıflandırma yaptığını yukarıda bahsi geçen *TP, FP, TN, FN* tipi ölçülere göre ilgili dağılımın nasıl olduğuna dair normalize edilmiş çapraz tahmin matrisi (confusion matrix) olarak Şekil 2'de verilmektedir. Aynı veri kümesiyle eğitimi gerçekleştirilen Xception modelinin, YOLOv5 modeline göre sınıflandırma sonucu daha başarılı olmuştur.

		Gerçek Etiket													
		Aircraft	Bulker	Cargo	Containership	Corvette	Cruiser	Destroyer	Fast-attack	Ferrie	Frigate	Mine-warfare	Submarine	Tanker	Arka Plan
Tahminlenen Etiket	Aircraft	0,97	0	0	0	0	0,01	0	0	0	0	0	0	0	0,02
	Bulker	0	0,90	0,01	0,01	0	0	0	0	0	0	0	0	0,01	0,07
	Cargo	0	0,01	0,97	0	0	0	0	0	0	0	0	0	0	0,02
	Containership	0	0	0	0,97	0	0	0	0	0	0	0	0	0	0,02
	Corvette	0	0	0	0	0,84	0	0,01	0,04	0	0,05	0,01	0	0	0,03
	Cruiser	0	0,01	0	0	0,01	0,94	0,03	0	0	0,01	0	0	0	0,05
	Destroyer	0	0	0	0	0,02	0,03	0,88	0	0	0,05	0	0	0	0
	Fast-attack	0	0	0	0	0,03	0	0	0,90	0	0	0	0	0	0,02
	Ferrie	0	0	0	0	0	0	0	0	0,87	0	0	0	0	0,07
	Frigate	0	0	0	0	0,06	0,01	0,06	0	0	0,87	0	0	0	0,13
	Mine-warfare	0	0	0	0	0	0	0	0,02	0	0	0,98	0	0	0
	Submarine	0	0	0	0	0	0	0	0	0	0,01	0	0,97	0	0
	Tanker	0	0,01	0,02	0	0	0	0	0	0	0	0	0	0,95	0,02
	Arka Plan	0,03	0,07	0,00	0,02	0,04	0,01	0,02	0,04	0,13	0,01	0,01	0,03	0,04	0,02

Şekil 2 YOLOv5 modeline ait normalize edilmiş çapraz tahmin matrisi (Normalized confusion matrix for YOLOv5 model)

Eğitim esnasında yapılan 50 eğitim yineleme (epoch) adımı sonunda ortalama olarak Xception modeliyle %99,44 doğruluk oranında başarım değeri elde etmiştir. Normalize edilmiş çapraz tahmin matrisi incelendiğinde, bazı gemi türlerinin birbirlerine benzemesinden dolayı başarım oranı düşük gözükmektedir. Xception modeline ait normalize edilmiş çapraz tahmin matrisi Şekil 3'te görülmektedir.

		Gerçek Etiket												
		Aircraft	Bulker	Cargo	Containership	Corvette	Cruiser	Destroyer	Fast-attack	Ferrie	Frigate	Mine-warfare	Submarine	Tanker
Tahminlenen Etiket	Aircraft	0,98	0	0	0	0	0,01	0	0	0	0	0	0,005	0
	Bulker	0,005	0,97	0	0	0	0	0	0	0	0	0,005	0	0,02
	Cargo	0,005	0,01	1	0	0	0	0	0	0	0	0	0	0,01
	Containership	0	0,01	0	1	0	0	0	0	0	0	0	0	0,005
	Corvette	0	0	0	0	0,87	0,005	0	0,015	0	0,03	0	0,005	0
	Cruiser	0	0	0	0	0,015	0,955	0,03	0,005	0	0,02	0	0	0
	Destroyer	0	0,005	0	0	0,015	0,03	0,935	0,005	0	0,02	0,005	0	0
	Fast-attack	0	0	0	0	0,035	0	0	0,96	0	0	0	0	0
	Ferrie	0,005	0,005	0	0	0	0	0	0	1	0	0	0,005	0
	Frigate	0,005	0	0	0	0,035	0	0,035	0	0	0,925	0	0	0
	Mine-warfare	0	0	0	0	0,025	0	0	0,015	0	0,005	0,99	0	0,005
	Submarine	0	0	0	0	0,005	0	0	0	0	0	0	0,985	0
	Tanker	0	0	0	0	0	0	0	0	0	0	0	0	0,96

Şekil 3 Xception modeline ait normalize edilmiş çapraz tahmin matrisi (Normalized confusion matrix for Xception model)

Yukarıdaki nesnel ölçütleri kullanılarak her bir gemi türü sınıfı için bu hesaplamaları gerçekleştirdiğimizde sonuçlar yüzdelik oran olarak aşağıda verilen Tablo 2' deki şekilde olmaktadır. Tabloda görüldüğü üzere en yüksek doğruluk oranına sahip olarak "*Containership*" ve "*Ferrie*" gemi çeşitleri olmuştur. Bunun başlıca sebebi görüntü içerisinde oransal olarak daha fazla miktarda özniteliğin bu gemi çeşitleri için tespit edilmiş olmasıdır. Deneylerde kullanılan her iki modele ait normalize edilmiş çapraz tahmin tabloları ve nesnel ölçüt tabloları incelendiğinde, tasarımsal olarak derin öğrenmenin gemi sınıflarının düzgün bir biçimde ayrıştırılarak doğru olarak sınıflandırılmasında uygun bir araç olduğunu göstermektedir. Başarımın veri büyüklüğüne doğru orantılı olarak arttığı görülmektedir. Modelden modele başarım oranlarının belirleyici bir unsur olduğu anlaşılmaktadır.

Tablo 2. Xception ile bulunan gemi çeşitleri için nesnel ölçüt değerleri (Objective metric values for ship variant detected by Xception)

Gemi Çeşitleri	Duyarlık	Anma	Doğruluk oranı
Aircraft	%98	%98	%99,73
Destroyer	%92	%94	%98,88
Containership	%99	%100	%99,88
Bulker	%97	%97	%99,54
Cargo	%98	%100	%99,81
Corvette	%94	%87	%98,58
Frigate	%93	%93	%98,85
Submarine	%99	%98	%99,85
Cruiser	%93	%95	%99,12
Tanker	%100	%96	%99,69
Ferrie	%99	%100	%99,88
Mine-warfare	%95	%99	%99,54
Fast-attack	%96	%96	%99,42
Bütün Sınıflar için Başarım Ortalama Değerleri	**%96,38**	**%96,38**	**%99,44**

Elde edilen sonuçlar, bu tarz bir gerçek zamanlı gemi sınıfı ayrıştırma ve sınıflandırma görevi için her iki model de kullanılabilir olduğu gösterse de Xception modeli daha yüksek başarım göstermesi nedeniyle tercih edilebilir konumdadır.

4. Sonuçlar (Conclusions)

Eğitilen her iki model için tablolarda verilen sonuçlar incelendiğinde her iki modelin de yaklaşık %96 ilâ %99 arası doğruluk oranında başarım değerleri elde ettiği görülmektedir. Elde edilen sonuçlar modellerin sınıflandırma adına oldukça kararlı ve başarımlarının yüksek olduğu anlaşılmaktadır. Her iki modelde de dikkat edilmesi gereken nokta; "*Corvette*", "*Destroyer*" ve "*Frigate*" tipindeki gemilere dair sınıfların çapraz tahmin matrisinde ve nesnel ölçüt değerlerinin diğer sınıflara göre ortalamanın altında değer aldığıdır. Bunun nedenine baktığımızda ilgili gemilerin birbirine oldukça çok benzemesidir. Bunun önüne geçebilmek için iki yöntem denenebilir; veri kümesindeki veri sayısını artırmak veya her iki modele de Uzun Kısa Süreli Bellek (Long Short Term Memory, LSTM) katmanları eklenerek işlenen veri içersindeki uzun süreli bağımlılıklara (veri blokları arası dolaylı ilişkilere) dair çıkarımların yapılabilmesi olarak bir çözüm ifade edilebilir. Gemilerin boyutlarına ve ortamın durumuna göre bir analiz yapılacak olursa; veri kümesinde bulunan görüntüler arasında gece çekilen, görüntü kalitesi düşük ve gemilerin küçük gözüktüğü görüntüler de bulunmaktadır. Nesnel ölçüt değerleri incelendiğinde, ortam veya gemi boyutunun belirli bir boyuta kadar küçülmesi tahmin işlemini aksatmamaktadır. Yapılan çalışmada klasik makine öğrenmesi yöntemine kıyasla, derin öğrenme modelleri kullanılması sayesinde, içeriğin anlamsal olarak

zenginleştirilmesi sağlanarak literatürdeki benzeri çalışmalara göre daha yüksek bir başarım oranı elde edilmiştir. İleriki çalışmalarımızda sistemin topluluk (ensemble) oluşturacak şekilde birden çok modelin melezlenmesiyle başarımın iyileştirilmesi hedeflenmektedir.

Kaynaklar (References)

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J. et al., 2016, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Distributed, Parallel, and Cluster Computing (cs.DC); Machine Learning (cs.LG) arXiv Preprint, arXiv: 1603.04467.

Cho M., 2013, Unsplash, https://unsplash.com/

Chollet, F., 2016, Xception: Deep Learning with Depth wise Separable Convolutions, arXiv Preprints. arXiv:1610.02357v3

Daniel Y.C., 2018, Pandas for Everyone: Python Data Analysis. Addison-Wesley, Boston: ISBN 978-0-13-454693-3.

Dolapcı, B., Özcan, C., 2021. Automatic Ship Detection and Classification using Machine Learning from Remote Sensing Images on Apache Spark, Journal of Intelligent Systems: Theory and Applications, 4(2), 94-102.

Getty, M., 2007, GettyImages, https://www.gettyimages.com/

Harris, C.R., Millman, K.J., van der Walt, S.J. et al., 2020, Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Jiang P.,, Ergu, D.,, Liu, F., Cai, Y.,, Bo, M., A , 2022, Review of Yolo Algorithm Developments, Procedia Computer Science, 10(1), 1066-1073, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.01.135.

Karasulu, B., 2018. Kısıtlanmış Boltzmann makinesi ve farklı sınıflandırıcılarla oluşturulan sınıflandırma iş hatlarının başarımının değerlendirilmesi, Bilişim Teknolojileri Dergisi, 11(3), 223-233.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4), 541-551.

Lekkas, D., 2007, Marine Traffic: https://www.marinetraffic.com/en/ais/home/centerx:2.8

Livingstone, B., 2000, IStockPhoto, https://istockphoto.com/

Lorica, B., 2017, Why AI and machine learning researchers are beginning to embrace PyTorch, O'Reilly Media.

Lowe, D.G., 2004, Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60 (1), 91–110.

Obrecht, C., 2014, Pexels, https://www.pexels.com/

Reitz, K., A., 2022, Requests Python HTTP, https://docs.python-requests.org/en/latest/

Richardson, L., 2021, Beautiful Soup Python Kütüphanesi. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Rosenfeld, L., Morville P., Arango, J., 2015, Information Architecture for the web and beyond. O'Reilly Media, Inc. ISBN 9781491911686.

Zhenzhen, L., Baojun, Z., Linbo, T., Zhen, L. and Fan, F., 2019, Ship classification based on convolutional neural networks. The Journal of Engineering, 7343-7346. https://doi.org/10.1049/joe.2019.0422

Research Article
Journal of Intelligent Systems: Theory and Applications 5(2) (2022) 168-177
DOI: 10.38016/jista.1153108

Auxiliary Learning of Non-Monotonic Hyperparameter Scheduling System Via Grid Search

Alaa Ali Hameed[1*]

[1] Istinye University, Computer Engineering, Istanbul, Türkiye

alaa.hameed@istinye.edu.tr

Abstract
Recent advancements in advanced neural networks have given rise to new adaptive learning strategies. Conventional learning strategies suffer from many issues, such as slow convergence and lack of robustness. To fully exploit its potential, these issues must be resolved. Both issues are related to the step-size, and momentum term, which is generally fixed and remains uniform for all weights associated with each network layer. In this study, the recently published Back-Propagation Algorithm with Variable Adaptive Momentum (BPVAM) algorithm has been proposed to overcome these issues and improve effectiveness for classification. The study was conducted on various hyperparameters based on the grid search approach, then the optimal values of hyperparameters have trained these algorithms. Six cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the training models. It is empirically proven that the convergence behavior of the model is improved in terms of the mean and standard deviation for accuracy and the sum of squared error (SSE). A comprehensive set of experiments indicated that the BPVAM is a robust and highly efficient algorithm.

Keywords: Adaptive neural networks; Hyperparameter; Steady-state error; Optimization.

Grid Arama Yoluyla Monotonik Olmayan Hiperparametre Planlama Sisteminin Yardımcı Öğrenimi

Öz
Gelişmiş sinir ağlarındaki son gelişmeler, yeni uyarlanabilir öğrenme stratejilerine yol açmıştır. Geleneksel öğrenme stratejileri, yavaş yakınsama ve sağlamlık eksikliği gibi birçok sorundan muzdariptir. Potansiyelinden tam olarak yararlanmak için bu sorunların çözülmesi gerekir. Her iki konu da adım boyutu ve genellikle sabit olan ve her ağ katmanıyla ilişkili tüm ağırlıklar için tek tip kalan momentum terimi ile ilgilidir. Bu çalışmada, bu sorunların üstesinden gelmek ve sınıflandırma etkinliğini artırmak için yakın zamanda yayınlanan Değişken Uyarlanabilir Momentumlu Geri Yayılım Algoritması (BPVAM) algoritması önerilmiştir. Çalışma grid arama yaklaşımına dayalı olarak çeşitli hiperparametreler üzerinde yürütülmüş, daha sonra hiperparametrelerin optimal değerleri bu algoritmaları eğitmiştir. Hiperparametrenin eğitim modelleri üzerindeki etkisini değerlendirmek için hiperparametrenin değişen değerlerine sahip altı durum ele alındı. Modelin yakınsama davranışının, doğruluk için ortalama ve standart sapma ve karesel hatanın toplamı (SSE) açısından iyileştirildiği deneysel olarak kanıtlanmıştır. Kapsamlı bir deney seti, BPVAM'nin sağlam ve yüksek verimli bir algoritma olduğunu gösterdi.

Anahtar Kelimeler: Uyarlanabilir sinir ağları; Hiperparametre; Kararlı durum hatası; Optimizasyon.

1. Introduction

Advanced Adaptive Neural Networks (AANNs) are the latest developments for classification that have shown their effectiveness in solving different problems in various domains. For instance, AANNs are employed for pattern recognition (Jain et al., 2018) (Jain et al., 2019), object detection (Erol et al., 2018) (Rahman et al., 2020), images classification (Sharma et al., 2018) (Patel et al., 2019), medical diagnosis (Sarvamangala and Kulkarni, 2022) (Yu et al., 2021) (Houssein et al., 2021), etc (Demircan Keskin et al., 2022) (Güney et al., 2022) (Gemirter and Goularas, 2021). Recently,

* Corresponding Author.
E-mail: alaa.hameed@istinye.edu.tr

Recieved : 03 Aug 2022
Revision : 09 Sep 2022
Accepted : 12 Sep 2022

AANNs have gained more attention due to their applicability to large datasets in an efficient manner.

Machine learning models required training samples to learn the patterns in the data. The performance of the machine learning models is evaluated using a cost function. It will determine how accurately a model learns patterns from data. In addition, the model has many hyper-parameters that should be selected to minimize the cost function. The learning process is repeated over several epochs to obtain an optimal set of these parameters, generally termed learning. Therefore, the choice of the cost function is subjective as it depends on the model and the training data (Hinton et al., 2012) (Mestres et al., 2017). There are various methods that can be employed for training the neural networks, however, gradient-based methods are most commonly used due to their simplicity and efficiency. It aims to reduce the gradient of the cost function to obtain optimal weights during training (Krizhevsky et al., 2012) (Park et al., 2020). Although neural networks are prevalent, several issues must be addressed to carry out the training process smoothly (Hertel et al., 2020) (Sandha et al., 2020) (Sun et al., 2022). The most common issues include vanishing and exploding gradients (Bengio et al., 1994) (Glorot and Bengio, 2010.) and overfitting (Liu et al., 2021).

Another problem that can affect the neural network's performance is the presence of local minima. This situation may occur when training the model on a large dataset using more complex models. The gradient descent algorithm may face a gradient vanishing problem if it gets stuck in local minima. In addition, selecting an optimal learning rate is crucial for obtaining good accuracy for the model. Research has shown that too small value for the learning rate results in slow convergence of the model. In contrast, if a large value of learning rate is selected, then it may cause the model to skip the global optima (Jagtap et al., 2020) (Jin et al., 2022).

Recent research has shown that instead of using a fixed learning rate, an adaptive learning rate offers faster convergence with good accuracy (Seong et al., 2018) (Yan et al., 2020). Moreover, a large learning rate should not be used, which can lead to super-convergence and have regularizing effects (Smith and Topin, 2019).

The literature review reveals that researchers have proposed different solutions to the gradient vanishing problem (Liu et al., 2021). For instance, adding a momentum term can accelerate the weight updating processing that may help the model to push out of the local optima. The momentum term will keep changing the weights continuously with an appropriate ratio. During the training, it is possible that the derivative of the cost function produces zero value. Even in such a situation the model continues to update weights using the previous iteration's values of the cost function (Sutskever et al., 2013). It is interesting to note that during learning it is not possible to determine whether the solution obtained is optimal or reached a local. In both cases, the model will stopped as there will be no change in the parameter values over consecutive iterations. The model depends on several parameters that affect its performance. Learning rate (LR) also known as step-size is one of the crucial parameters. Fine tuning LR plays crucial role in obtaining optimal solution. Selection of a small value may allow the model to reach the optimal solution very slowly. In contrast, a large value may allow the model to reach the optimal solution faster. However, there is a trade-off between selection of a large/small value with the optimal solution. Therefore, care must be taken in selection of this crucial parameter. This problem can be solved using a scheduled rate. The most commonly used technique is to multiply the gradient with a constant during training of the model. The main issue with such technique is that the LR may not scale well during training. There are various solutions proposed to overcome this problem, such as time-based techniques where the LR is altered as the training proceeds (Li and Arora, 2019). Some other techniques, such as Adagrad and RMSProp are also proposed to solve this problem. These techniques apply adaptive optimization on the LR to adapt its value during the training (Duchi JDUCHI and Singer, 2011) (Reddi et al., 2019) (Yi et al., 2020). Some research proposed to combine both adaptive optimization adaptive LR schedules to further improve the accuracy of the model. However, these methods only apply a function in such as way that it decreases the LR as the model training proceeds. The main drawback of such techniques is that it may stuck in local minimum due to small gradient changes (Rumelhart et al., 1986) (Sohl-Dickstein et al.,2014).

Other advanced techniques to solve these bottlenecks include different activation functions (Klambauer et al., 2017) (Nair and Hinton, 2010), batch normalization (Ioffe and Szegedy, 2015), novel initialization schemes (He et al., 2015), and dropout (Srivastava et al., 2014). The main drawback of these methods is the higher computational overhead, which limits the performance improvements in terms of CPU cost, convergence rate, and optimal error.

The most common techniques for optimizing the deep neural networks (DNN) include batch gradient (BGD) and stochastic gradient descent (SGD) algorithms. BGD is usually slower and is more suitable for a small size dataset. On the other hand, SGD is faster and is more suitable to process large size data. Typically, SGD produces less reliable results which may also lead to bad convergence. In (Yang, 2021), authors proposed a new method based on the Kalman filter for better optimization of the network using adaptive filtering. The method employed the historical state of the optimization, which helped reduce the estimation variance in the SGD algorithm. This led to faster convergence and resulted in better gradient direction estimation even in the presence of noise.

Other gradient-based methods, such as adaptive gradient methods (AGMs), can also be employed to

optimize nonconvex problems in machine learning, specifically deep learning. In (Tong et al., 2022), two improvements of AGMs are proposed to enhance the model's accuracy further. It was observed that the anisotropic scale of the adaptive learning rate (A-LR) has high variations across multiple dimensions of the nonconvex optimization problem. This variation may lead to slower convergence and the model may get stuck in the local minima. The literature shows that a number of research are dedicated to improving the AGMs using A-LR. Another main bottleneck that plays vital role in obtaining the optimal accuracy is finding optimal values for its hyperparameters used in the A-LR. In some works, authors proposed adding activation functions in A-LR such as softplus function for AGM's improvement. Two such methods, namely SADAM and SAMSGRAD are also proposed to improve the model accuracy. Results showed that SAMSGRAD exhibit faster convergence than the AMSGRAD under various conditions such as nonconvex, non-strongly convex, and Polyak-Łojasiewicz conditions.

Another adaptive gradient descent algorithm that is commonly used in backpropagation (BP) for training feed-forward neural networks (FFNNs) is called Adam. The Adam algorithm's main issue is that it might fail to reach global optima. Solutions based on metaheuristic methods exist, which help train FFNNs to overcome the local minima issue. However, the solutions also have compromise on the convergence efficiency of the model compared to the Adam optimizer. A solution was proposed in terms of an ensemble of differential evolution and Adam (EDEAdam), combining both Adam optimizer and differential evolution algorithm, which forms a robust and efficient search mechanism to achieve better results in both global and local search. The integration of these two methods not only helped improve results but also showed faster convergence speed (Xue et al., 2022).

Hameed et al. (Hameed et al., 2016), proposed a BP algorithm with variable adaptive momentum (BPVAM). The algorithm improves the convergence behavior by achieving faster convergence, optimal error, and lower mathematical complexity, reducing the overall CPU cost and processing time. The learning rate is a crucial parameter that controls the model. The learning rate parameter depends on the input data's eigenvalues of the autocorrelation matrix.

This study investigates the learning performance of BPVAM algorithm. An adaptive momentum scheduler is introduced to overcome the gradient vanishing problem. A detailed set of experiments are performed on various benchmark datasets to evaluate the performance of the proposed model. The main contributions of this study are highlighted as follows:

- Introduction of a variable adaptive momentum term in the weight update equation.
- Fine-tuning the hyperparameters for computing an optimal momentum in stochastic gradient descent in BPVAM algorithm
- Investigate the model's behavior with adaptive momentum term and compare it with models with a fixed learning rate.
- Diverse set of experiments on different benchmark datasets are performed to test the efficiency and robustness of the proposed model.

The paper is organized as follows. In Section 2, details about the adaptive learning rate algorithms are presented. Extensive set of experiments are presented in Section 3. Finally, the paper is completed with conclusion.

2. Backpropagation Algorithm with Variable Adaptive Momentum (BPVAM)

Hameed et al (Hameed et al., 2016), introduced the BPVAM algorithm see fig. 1, where α (the adaptive momentum) is controlled by the learning rate parameter η. In this case, if given initial weights Ψ^0 and Ψ^1, and a momentum factor $\alpha\ \epsilon(0,1)$, BPVAM updates the weight vector iteratively which means that equation (28) can now be represented as

$$\Delta\Psi_i = \eta\delta_w\Psi_i + \alpha_\Psi^i\Delta\Psi_{i-1},\ i = 1, 2, ..., \quad (1)$$

Where $\eta > 0$ is the learning rate which is assumed to be a constant in this work and $\alpha_\Psi^i = (\alpha_{\Psi_0}^i, \alpha_{\Psi_1}^i, \alpha_{\Psi_2}^i, \dots\dots\dots, \alpha_{\Psi_q}^i)$ is the momentum coefficient vector at the i^{th} training iteration which is constituted by the coefficient α_Ψ^i for every $\Delta\Psi_i^i$ (i= 0, 1, 2, ….,q) and for each α_Ψ^i, it is adjusted after each training epoch by

$$\alpha_{\Psi_i}^i = \begin{cases} \alpha \cdot \frac{-\eta\delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1}}{\left\|\Delta\Psi_i^{i-1}\right\|^2} & if\ \delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1} < 0 \\ 0 & otherwise \end{cases} \quad (2)$$

In the BPAM, α (the adaptive momentum) was controlled by the learning rate η, where η is dependent on the eigenvalues of the autocorrelation matrix of the input.

The work presented by (Hameed et al., 2016) estimates the autocorrelation matrix *R(i)* of the input recursively as

$$R_i = \beta R_i + Rxx \quad (3)$$

Where β is the forgotten factor (0<<β<1), and $Rxx = E\{X(i)X^T(i)\}$, E is the expectation operator. Tacking the expected value of both sides of equation (32) produces

$$\bar{R}_i = \frac{1-\beta^i}{1-\beta}Rxx \quad (4)$$

Where $\bar{R}_i = E\{R_i\}$. Solving equation (32) in the steady state ($i \to \infty$) yields

$$\bar{R}_i = \frac{1}{1-\beta} \tag{5}$$

In this case, equation (34) implies that the eigenvalues of the estimated autocorrelation matrix increase exponentially, and in the limit they become $\frac{1}{1-\beta}$ times the original value.

The work done by (Hameed et al., 2016) also proposed a variable momentum, which is expressed by

$$\alpha_i = \frac{\lambda}{1-\beta^i} \tag{6}$$

Where $\lambda < \frac{2-2\beta}{max\ eigen\ value\ of \mathbf{R_{xx}}}$ and this case β is the forgetting factor ($0 \ll \beta < 1$),

Assuming that β is large, this will force the term $1 - \beta^i$ to reach unity, and assuming that the initial $\alpha(i)$ is relatively large, to provide fast convergence of the weights. By updating equations (27) and (28), with time it becomes very close to λ (a small positive constant) hence it provides law error, equation (27) and (28) can then be represented as

$$\Delta\Psi_{ji}(i+1) = \eta\delta_y \mathbf{x}_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{ji}(i) \tag{7}$$

$$\Delta\Psi_{kj}(i+1) = \eta\delta_o y_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{kj}(i),\ i = 0,1,\ldots \tag{8}$$

Where i represents the number of iterations and $\Delta\Psi$ is defined as updating the weights.

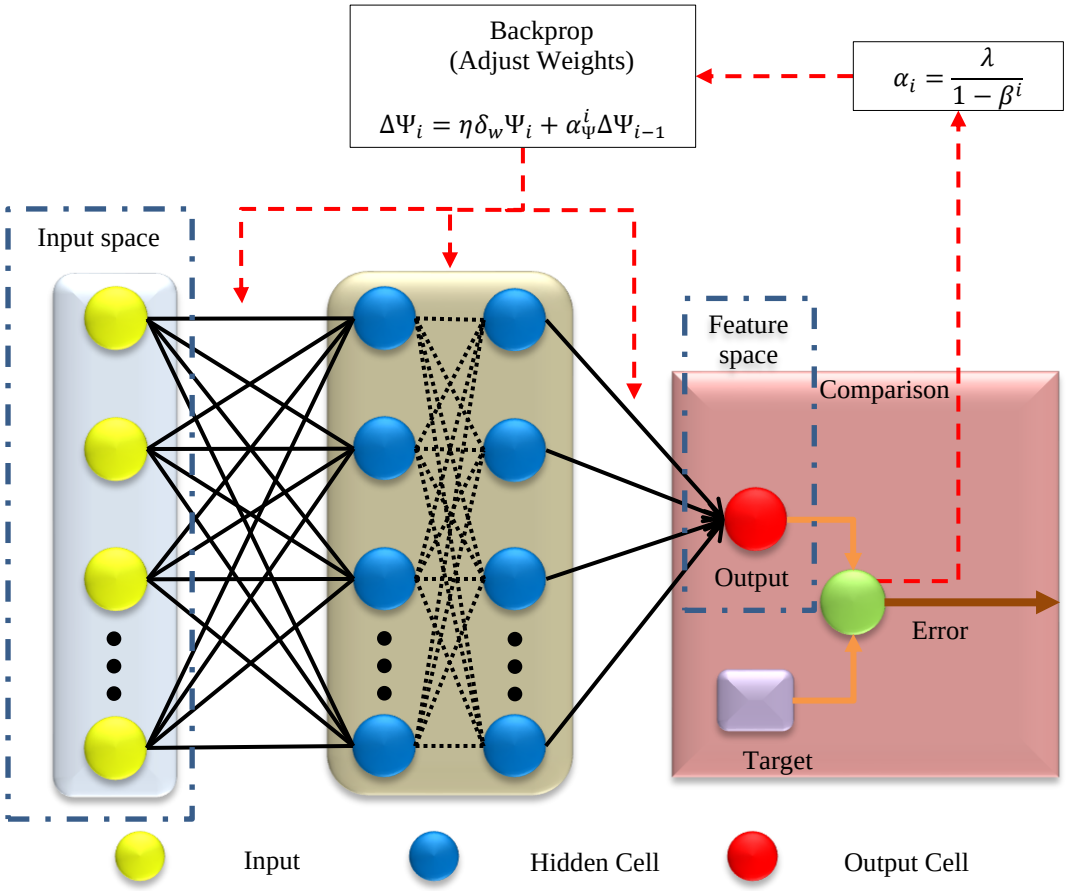


Figure 1. BPVAM architecture

3. Experimental Results

The experiments were performed on four different data sets obtained from various domains. A diverse set of datasets were considered for testing the application of the proposed method for different types of data. These datasets include Breast cancer, Heart Disease, Lung Cancer, and Iris. Each dataset has a varying number of samples, attributes, and classes (Asuncion and Newman,2007).

3.1 Preprocessing and Experimental Setup

All data in the dataset was normalized between 0 and 1 using the Min-Max normalization method. The main advantage of the normalization is maintaining stability in the network by allowing all the weights to converge almost simultaneously. Moreover, missing data were replaced with the mean value of the attribute.

All the experiments were performed in the Matlab™ environment. The models were executed on a Dell machine with Intel core i7, 2.10 GHz processor with 16 GB of RAM and NVIDIA™ GeForce TMGTX 1080. The dataset was divided into training (70%) and testing (30%) for each experiment. Since the dataset was balanced, therefore, no augmentation was performed.

3.2 Evaluation

The performance evaluation of BPVAM and its comparison with conventional BP was carried out in terms of accuracy and SSE on four benchmark datasets. Moreover, the models were also compared in terms of mean and standard deviation behaviors over the whole training process. Since the models depend on various hyperparameters, therefore, the optimal values of these hyperparameters were obtained using the Grid Search algorithm. The obtained optimal values of hyperparameters were then used to train the models. Six different cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the model accuracy. The experimental setup was similar to the one presented by the authors in (Hameed et al., 2016). The evaluation results for each dataset are described in detail as below.

Table 1 summarizes the results obtained on the **breast cancer** dataset. As it can be seen, the error convergence for the BPVAM (4.678) is better than the conventional BP (4.707) algorithm in terms of SSE for case 6. For other cases (1-5), the performance of BPVAM was also higher than conventional BP as it produced less error. Similarly, in terms of accuracy, the BPVAM algorithm produced higher accuracy compared BP in general overall six cases. It is observed that BPVAM obtained optimal results with $\eta = 0.9$, $\lambda = 0.0085$, and $\beta = 0.992$, whereas conventional BP produced best results with $\alpha = 0.01$, and $\eta = 0.9$.

Table 2 summarizes the evaluation results obtained **heart disease** dataset. These results showed that the BPVAM always produced better results than the conventional BP algorithm. Highest accuracy (61.96%) was obtained for BPVAM and lowest error (3.961) with parameter values of η = 0.03, λ = 0.022, β = 0.995. It is interesting to note that the accuracy of models tend to become close to each other as the parameter values were decreased from case-1 to case-6.

Table 3 shows the experimental results obtained by the models on the **Lung Cancer** dataset. The SSE and accuracy of BPVAM were BP 0.0054 and 60.00%, respectively. Similarly, for BP the SSE and accuracy remained 0.0063 and 60.00, respectively. The cases show that the convergence behavior of BP is very slow and very sensitive to the hyperparameter selection compared to BPVAM. The best results were obtained for BPVAM with parameters $\eta = 0.1$, $\lambda = 0.005$, and $\beta = 0.9980$. For BP BP optimal results were obtained with $\alpha = 0.05$, and $\eta = 0.1$.

Table 4 summarizes the results obtained on the **Iris** dataset. The results show that case 1 produced the optimal results for BPVAM with an accuracy of 84.44% and SSE of 0.853, while for BP the accuracy was 77.78% and SSE of 0.991 for BP. Following all cases from 1 to 6, it shows the BPVAM is more robust and can keep improving the network model steadily.

Further experiments were performed to compare the performance of the two models in terms of mean and standard deviation. Figure 2 and 3 shows the comparison of models in terms of the mean and standard deviation obtained for accuracy and SSE, respectively. It is evident that despite the improvement of the BP algorithm, the significant change indicates the sensitivity of the algorithm to its selection of parameters. On the other hand, the BPVAM algorithm shows its superiority from the first case until the sixth case. It increases the mean accuracy of the model while decreasing the standard deviation over all cases. We can deduce that the overall BPVAM model outperformed BP in terms of accuracy and SSE.

Table 1. Performance comparison metrics of the tested algorithms for Breast Cancer dataset

Case	Algorithm	α	η	λ	β	SSE	Accuracy (%)
1	BP	0.06	0.4	-	-	7.244	60.34
	BPVAM	-	0.4	0.0090	0.997	6.873	63.79
2	BP	0.05	0.5	-	-	5.691	67.24
	BPVAM	-	0.5	0.0089	0.996	5.685	67.24
3	BP	0.04	0.6	-	-	5.656	68.97
	BPVAM	-	0.6	0.0088	0.995	5.053	70.69
4	BP	0.03	0.7	-	-	4.924	72.41
	BPVAM	-	0.7	0.0087	0.994	4.787	75.86
5	BP	0.02	0.8	-	-	4.801	74.14
	BPVAM	-	0.8	0.0086	0.993	4.780	75.86
6	BP	0.01	0.9	-	-	4.707	77.59
	BPVAM	-	0.9	0.0085	0.992	4.678	77.59

Table 2. Performance comparison metrics of the tested algorithms for Heart Disease dataset

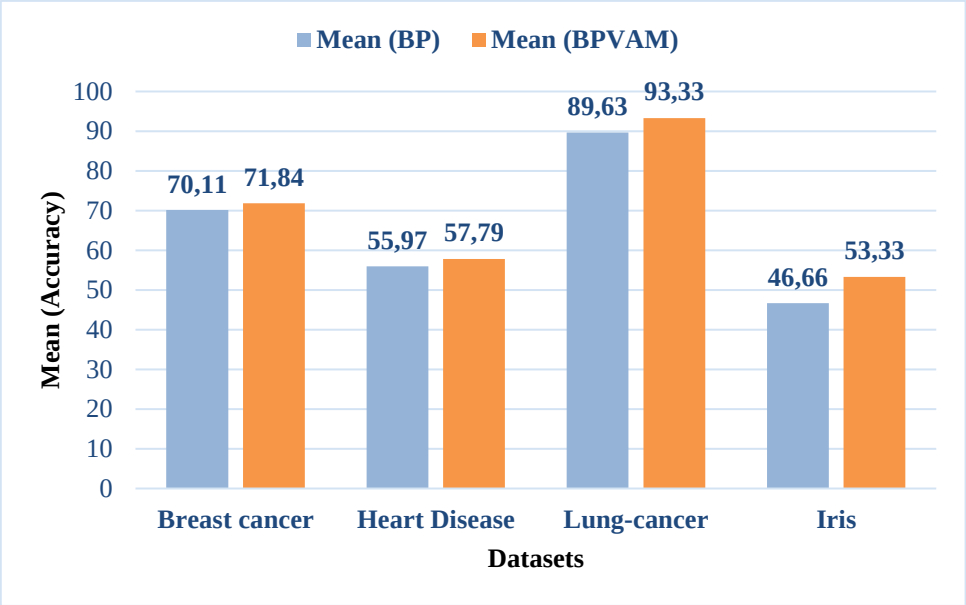
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.06	0.08	-	-	5.306	48.91
	BPVAM	-	0.08	0.027	0.999	4.900	51.09
2	BP	0.05	0.07	-	-	4.648	52.17
	BPVAM	-	0.07	0.026	0.999	4.615	54.35
3	BP	0.04	0.06	-	-	4.586	55.43
	BPVAM	-	0.06	0.025	0.998	4.022	58.70
4	BP	0.03	0.05	-	-	4.332	57.61
	BPVAM	-	0.05	0.024	0.997	4.017	59.78
5	BP	0.02	0.04	-	-	4.020	59.78
	BPVAM	-	0.04	0.023	0.996	4.001	60.87
6	BP	0.01	0.03	-	-	3.969	61.96
	BPVAM	-	0.03	0.022	0.995	3.961	61.96

Table 3. Performance comparison metrics of the tested algorithms for Lung-Cancer dataset

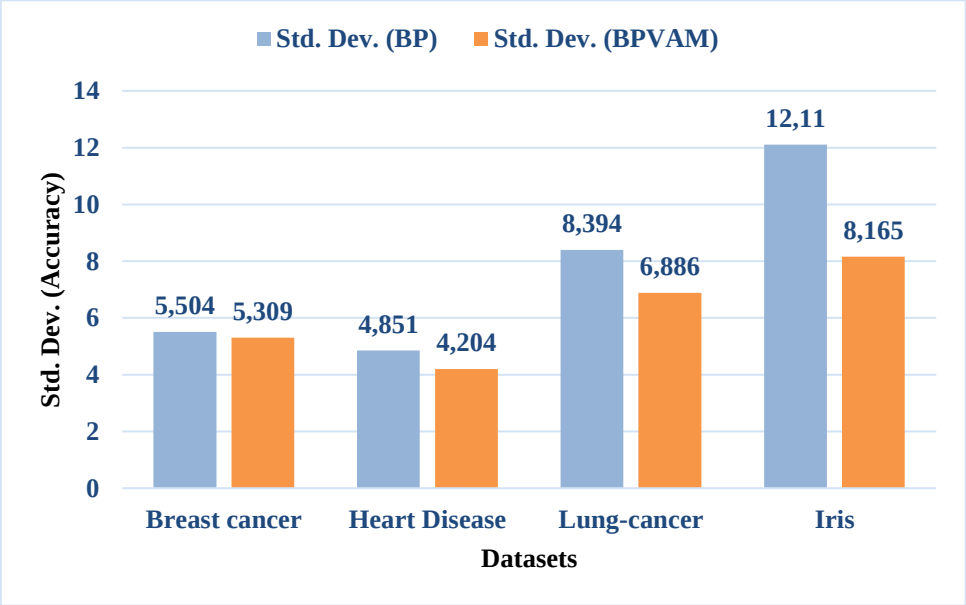
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.10	0.6	-	-	0.0275	30.00
	BPVAM	-	0.6	0.010	0.9994	0.0161	40.00
2	BP	0.09	0.5	-	-	0.0163	40.00
	BPVAM	-	0.5	0.009	0.9993	0.0081	50.00
3	BP	0.08	0.4	-	-	0.0120	40.00
	BPVAM	-	0.4	0.008	0.9992	0.0075	50.00
4	BP	0.07	0.3	-	-	0.0081	50.00
	BPVAM	-	0.3	0.007	0.9991	0.0066	60.00
5	BP	0.06	0.2	-	-	0.0072	60.00
	BPVAM	-	0.2	0.006	0.9990	0.0060	60.00
6	BP	0.05	0.1	-	-	0.0063	60.00
	BPVAM	-	0.1	0.005	0.9980	0.0054	60.00

Table 4. Performance comparison metrics of the tested algorithms for Iris dataset

Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.006	0.10	-	-	0.991	77.78
	BPVAM	-	0.10	0.07	0.9994	0.853	84.44
2	BP	0.005	0.09	-	-	0.926	82.22
	BPVAM	-	0.09	0.06	0.9993	0.812	86.67
3	BP	0.004	0.08	-	-	0.756	88.89
	BPVAM	-	0.08	0.05	0.9992	0.618	91.11
4	BP	0.003	0.07	-	-	0.516	93.33
	BPVAM	-	0.07	0.04	0.9991	0.201	97.78
5	BP	0.002	0.06	-	-	0.334	95.56
	BPVAM	-	0.06	0.03	0.9990	0.182	100.00
6	BP	0.001	0.05	-	-	0.184	100.00
	BPVAM	-	0.05	0.02	0.9980	0.180	100.00

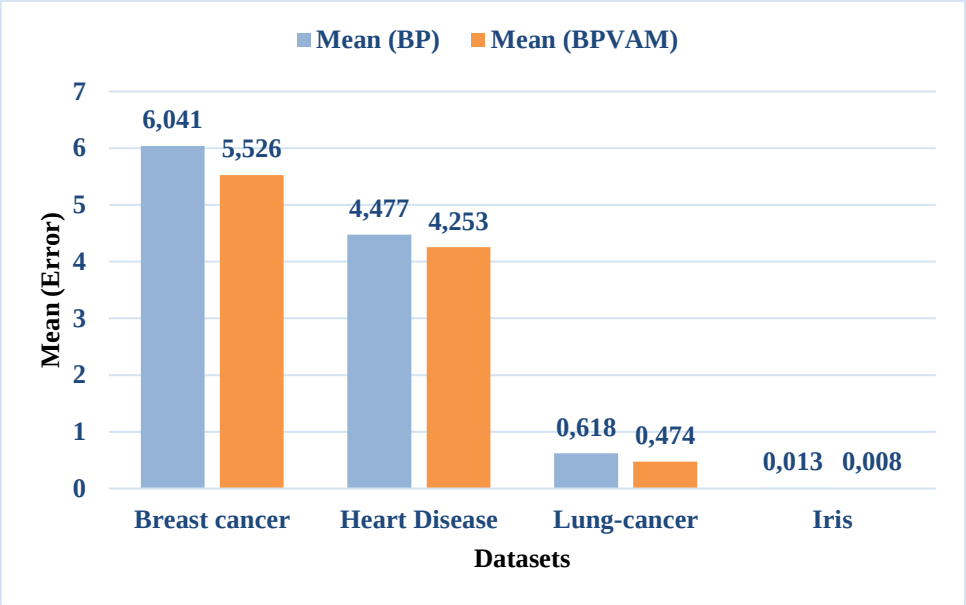


(a)

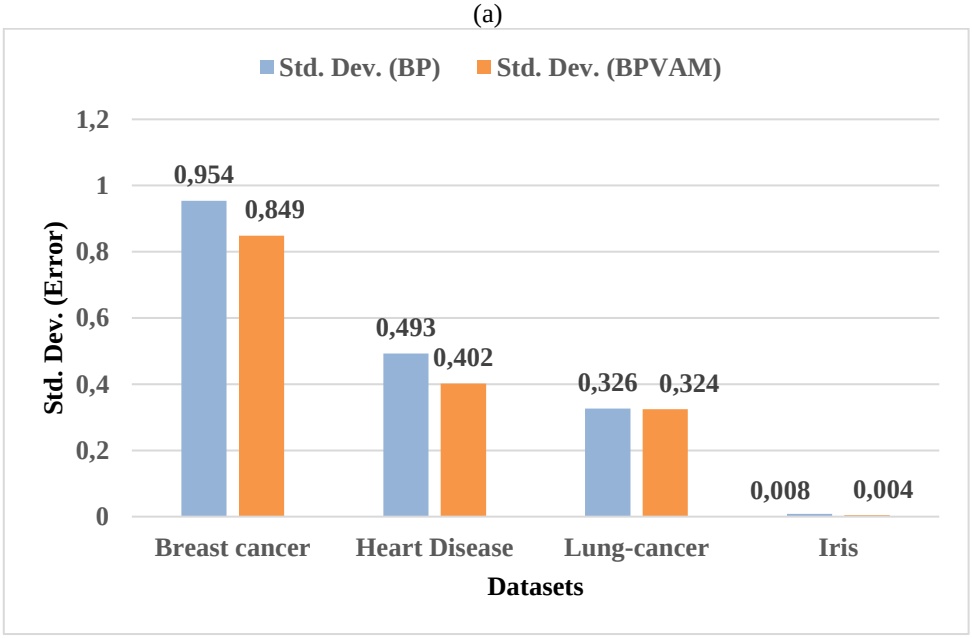


(b)

Figure 2. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean accuracy, and b) standard deviation accuracy



(a)



(b)

Figure 3. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean error, and b) standard deviation error

4. Conclusions

This study investigated the approach for obtaining an optimal set of hyperparameters for the machine learning model. Moreover, the model's weight matrix is updated using the adaptive momentum to help it overcome the local optima problem. The algorithm is controlled by different hyperparameters, which are fine-tuned using grid search. The results showed that the BPVAM algorithm obtains better convergence behavior than BP in the optimal steady-state models. The experiments investigated the compared methods from different aspects by considering the whole learning behavior in different training cases. The optimal results obtained on four benchmark datasets indicate that BPVAM improved the accuracy and robustness of the model. Moreover, this study suggests a significant improvement in accuracy, mean error, and standard deviation when the BPVAM is optimized with adaptive momentum. It can be observed that BPVAM exhibit features to guarantee its convergence and produce a much lower SSE against any valid data sets. In the future, we aim to apply this optimization algorithm to obtain an optimal set of parameters for a deep end-to-end neural network to overcome the issue of obtaining the optimal

hyperparameters, we also are plan to monitor the progress of the hyperparameter optimization in real-time. This will allow the extraction of highly discriminative features from input data that can improve the model's performance.

References

A. and Newman, D. J. (2007). UCI Machine Learning Repository, Department of Information and Computer Sciences, University of California, Irvine. Available at www.ics.uci.edu/~mlearn/MLRepository.html.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks 5, 157–166. https://doi.org/10.1109/72.279181

Demircan Keskin, F., Çiçekli, U., İçli, D., 2022. Prediction of Failure Categories in Plastic Extrusion Process with Deep Learning. Journal of Intelligent Systems: Theory and Applications, 5(1), 27–34. https://doi.org/10.38016/jista.878854

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

Erol, B.A., Majumdar, A., Lwowski, J., Benavidez, P., Rad, P., Jamshidi, M., 2018. Improved deep neural network object tracking system for applications in home robotics, in: Studies in Computational Intelligence. Springer Verlag, pp. 369–395. https://doi.org/10.1007/978-3-319-89629-8_14

Gemirter, C. B., Goularas, D., 2021. A Turkish Question Answering System Based on Deep Learning Neural Networks. Journal of Intelligent Systems: Theory and Applications, 4(2), 65–75. https://doi.org/10.38016/jista.815823

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256.

Güney, E., Çakmak, O., Kocaman, Ç., 2022. Classification of Stockwell Transform Based Power Quality Disturbance with Support Vector Machine and Artificial Neural Networks. Journal of Intelligent Systems: Theory and Applications, 5(1), 75–84. https://doi.org/10.38016/jista.996541

Hameed, A.A., Karlik, B., Salman, M.S., 2016. Back-propagation algorithm with variable adaptive momentum. Knowledge-Based Systems 114, 79–87. https://doi.org/10.1016/j.knosys.2016.10.001

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034.

Hertel, L., Collado, J., Sadowski, P., Ott, J., Baldi, P., 2020. Sherpa: Robust hyperparameter optimization for machine learning. SoftwareX, 12, 100591.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Houssein, E.H., Emam, M.M., Ali, A.A., Suganthan, P.N., 2021. Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review. Expert Systems with Applications, 167. https://doi.org/10.1016/j.eswa.2020.114161

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pp. 448-456.

Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E., 2020. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics 404. https://doi.org/10.1016/j.jcp.2019.109136

Jain, D.K., Shamsolmoali, P., Sehdev, P., 2019. Extended deep neural network for facial emotion recognition. Pattern Recognition Letters 120, 69–74. https://doi.org/10.1016/j.patrec.2019.01.008

Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., Zareapoor, M., 2018. Hybrid deep neural networks for face emotion recognition. Pattern Recognition Letters 115, 101–106. https://doi.org/10.1016/j.patrec.2018.04.010

Jin, J., Zhu, J., Gong, J., Chen, W., 2022. Novel activation functions-based ZNN models for fixed-time solving dynamirc Sylvester equation. Neural Computing and Applications, 1-19. https://doi.org/10.1007/s00521-022-06905-2

Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., 2017. Self-Normalizing Neural Networks, Advances in neural information processing systems, 30.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems, 25.

Li, Z., Arora, S., 2019. An Exponential Learning Rate Schedule for Deep Learning. arXiv preprint arXiv:1910.07454.

Liu, M., Chen, L., Du, X., Jin, L., Shang, M., 2021. Activated Gradients for Deep Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 1–13. https://doi.org/10.1109/tnnls.2021.3106044

Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Hibbett, M.J., Estrada, G., Ma'ruf, K., Coras, F., Ermagan, V., Latapie, H., Cassar, C., Evans, J., Maino, F., Walrand, J., Cabellos, A., 2017. Knowledge-defined networking. Computer Communication Review 47, 1–10. https://doi.org/10.1145/3138808.3138810

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In Appearing in Proceedings of the 27 th International Conference on Machine Learning (ICML).

Park, J., Yi, D., Ji, S., 2020. A novel learning rate schedule in optimization for neural networks and it's convergence. Symmetry (Basel) 12. https://doi.org/10.3390/SYM12040660

Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., Yang, B., 2019. Deep learning-based object classification on automotive radar spectra, in: 2019 IEEE Radar Conference, RadarConf 2019. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/RADAR.2019.8835775

Rahman, M.M., Tan, Y., Xue, J., Lu, K., 2020. Notice of Removal: Recent Advances in 3D Object Detection in the Era of Deep Neural Networks: A Survey. IEEE Transactions on Image Processing. https://doi.org/10.1109/TIP.2019.2955239

Reddi, S.J., Kale, S., Kumar, S., 2019. On the Convergence of Adam and Beyond. arXiv preprint arXiv:1904.09237.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Sandha, S. S., Aggarwal, M., Fedorov, I., Srivastava, M. 2020. Mango: A python library for parallel hyperparameter tuning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3987-3991.

Sarvamangala, D.R., Kulkarni, R. v., 2022. Convolutional neural networks in medical image understanding: a survey. Evolutionary Intelligence, 1-22. https://doi.org/10.1007/s12065-020-00540-3

Seong, S., Lee, Y., Kee, Y., Han, D., Kim, J., 2018. Towards Flatter Loss Surface via Nonmonotonic Learning Rate Scheduling, In UAI.

Sharma, N., Jain, V., Mishra, A., 2018. An Analysis of Convolutional Neural Networks for Image Classification, in: Procedia Computer Science. Elsevier B.V., pp. 377–384. https://doi.org/10.1016/j.procs.2018.05.198

Smith, L.N., Topin, N., 2019. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Artificial intelligence and machine learning for multi-domain operations applications, Vol. 11006, pp. 369-386.

Sohl-Dickstein, J., Poole, B., Ganguli, S., 2014. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In International Conference on Machine Learning, pp. 604-612.

Srivastava, N., Hinton, G., Krizhevsky, A., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research. 15(1), 1929-1958.

Sun, J., Yang, Y., Xun, G., Zhang, A., 2022. Scheduling Hyperparameters to Improve Generalization: From Centralized SGD to Asynchronous SGD. ACM Transactions on Knowledge Discovery from Data (TKDD). https://dl.acm.org/doi/pdf/10.1145/3544782.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pp. 1139-1147.

Tong, Q., Liang, G., Bi, J., 2022. Calibrating the adaptive learning rate to improve convergence of ADAM. Neurocomputing 481, 333–356. https://doi.org/10.1016/j.neucom.2022.01.014

Xue, Y., Tong, Y., Neri, F., 2022. An ensemble of differential evolution and Adam for training feed-forward neural networks. Information Sciences. Information Sciences, 608, 453-471. https://doi.org/10.1016/j.ins.2022.06.036

Yan, Z., Chen, J., Hu, R., Huang, T., Chen, Y., Wen, S., 2020. Training memristor-based multilayer neuromorphic networks with SGD, momentum and adaptive learning rates. Neural Networks 128, 142–149. https://doi.org/10.1016/j.neunet.2020.04.025

Yang, X., 2021. Kalman optimizer for consistent gradient descent, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Institute of Electrical and Electronics Engineers Inc., pp. 3900–3904. https://doi.org/10.1109/ICASSP39728.2021.9414588

YD., Ahn, J., Ji, S., 2020. An effective optimization method for machine learning based on ADAM. Applied Sciences (Switzerland) 10. https://doi.org/10.3390/app10031073

YH., Yang, L.T., Zhang, Q., Armstrong, D., Deen, M.J., 2021. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. Neurocomputing 444, 92–110. https://doi.org/10.1016/j.neucom.2020.04.157

Research Article
Journal of Intelligent Systems: Theory and Applications 5(2) (2022) 168-177
DOI: 10.38016/jista.1153108

Auxiliary Learning of Non-Monotonic Hyperparameter Scheduling System Via Grid Search

Alaa Ali Hameed[1*]

[1] Istinye University, Computer Engineering, Istanbul, Türkiye

alaa.hameed@istinye.edu.tr

Abstract
Recent advancements in advanced neural networks have given rise to new adaptive learning strategies. Conventional learning strategies suffer from many issues, such as slow convergence and lack of robustness. To fully exploit its potential, these issues must be resolved. Both issues are related to the step-size, and momentum term, which is generally fixed and remains uniform for all weights associated with each network layer. In this study, the recently published Back-Propagation Algorithm with Variable Adaptive Momentum (BPVAM) algorithm has been proposed to overcome these issues and improve effectiveness for classification. The study was conducted on various hyperparameters based on the grid search approach, then the optimal values of hyperparameters have trained these algorithms. Six cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the training models. It is empirically proven that the convergence behavior of the model is improved in terms of the mean and standard deviation for accuracy and the sum of squared error (SSE). A comprehensive set of experiments indicated that the BPVAM is a robust and highly efficient algorithm.

Keywords: Adaptive neural networks; Hyperparameter; Steady-state error; Optimization.

Grid Arama Yoluyla Monotonik Olmayan Hiperparametre Planlama Sisteminin Yardımcı Öğrenimi

Öz
Gelişmiş sinir ağlarındaki son gelişmeler, yeni uyarlanabilir öğrenme stratejilerine yol açmıştır. Geleneksel öğrenme stratejileri, yavaş yakınsama ve sağlamlık eksikliği gibi birçok sorundan muzdariptir. Potansiyelinden tam olarak yararlanmak için bu sorunların çözülmesi gerekir. Her iki konu da adım boyutu ve genellikle sabit olan ve her ağ katmanıyla ilişkili tüm ağırlıklar için tek tip kalan momentum terimi ile ilgilidir. Bu çalışmada, bu sorunların üstesinden gelmek ve sınıflandırma etkinliğini artırmak için yakın zamanda yayınlanan Değişken Uyarlanabilir Momentumlu Geri Yayılım Algoritması (BPVAM) algoritması önerilmiştir. Çalışma grid arama yaklaşımına dayalı olarak çeşitli hiperparametreler üzerinde yürütülmüş, daha sonra hiperparametrelerin optimal değerleri bu algoritmaları eğitmiştir. Hiperparametrenin eğitim modelleri üzerindeki etkisini değerlendirmek için hiperparametrenin değişen değerlerine sahip altı durum ele alındı. Modelin yakınsama davranışının, doğruluk için ortalama ve standart sapma ve karesel hatanın toplamı (SSE) açısından iyileştirildiği deneysel olarak kanıtlanmıştır. Kapsamlı bir deney seti, BPVAM'nin sağlam ve yüksek verimli bir algoritma olduğunu gösterdi.

Anahtar Kelimeler: Uyarlanabilir sinir ağları; Hiperparametre; Kararlı durum hatası; Optimizasyon.

1. Introduction

Advanced Adaptive Neural Networks (AANNs) are the latest developments for classification that have shown their effectiveness in solving different problems in various domains. For instance, AANNs are employed for pattern recognition (Jain et al., 2018) (Jain et al., 2019), object detection (Erol et al., 2018) (Rahman et al., 2020), images classification (Sharma et al., 2018) (Patel et al., 2019), medical diagnosis (Sarvamangala and Kulkarni, 2022) (Yu et al., 2021) (Houssein et al., 2021), etc (Demircan Keskin et al., 2022) (Güney et al., 2022) (Gemirter and Goularas, 2021). Recently,

* Corresponding Author.
E-mail: alaa.hameed@istinye.edu.tr

Recieved : 03 Aug 2022
Revision : 09 Sep 2022
Accepted : 12 Sep 2022

AANNs have gained more attention due to their applicability to large datasets in an efficient manner.

Machine learning models required training samples to learn the patterns in the data. The performance of the machine learning models is evaluated using a cost function. It will determine how accurately a model learns patterns from data. In addition, the model has many hyper-parameters that should be selected to minimize the cost function. The learning process is repeated over several epochs to obtain an optimal set of these parameters, generally termed learning. Therefore, the choice of the cost function is subjective as it depends on the model and the training data (Hinton et al., 2012) (Mestres et al., 2017). There are various methods that can be employed for training the neural networks, however, gradient-based methods are most commonly used due to their simplicity and efficiency. It aims to reduce the gradient of the cost function to obtain optimal weights during training (Krizhevsky et al., 2012) (Park et al., 2020). Although neural networks are prevalent, several issues must be addressed to carry out the training process smoothly (Hertel et al., 2020) (Sandha et al., 2020) (Sun et al., 2022). The most common issues include vanishing and exploding gradients (Bengio et al., 1994) (Glorot and Bengio, 2010.) and overfitting (Liu et al., 2021).

Another problem that can affect the neural network's performance is the presence of local minima. This situation may occur when training the model on a large dataset using more complex models. The gradient descent algorithm may face a gradient vanishing problem if it gets stuck in local minima. In addition, selecting an optimal learning rate is crucial for obtaining good accuracy for the model. Research has shown that too small value for the learning rate results in slow convergence of the model. In contrast, if a large value of learning rate is selected, then it may cause the model to skip the global optima (Jagtap et al., 2020) (Jin et al., 2022).

Recent research has shown that instead of using a fixed learning rate, an adaptive learning rate offers faster convergence with good accuracy (Seong et al., 2018) (Yan et al., 2020). Moreover, a large learning rate should not be used, which can lead to super-convergence and have regularizing effects (Smith and Topin, 2019).

The literature review reveals that researchers have proposed different solutions to the gradient vanishing problem (Liu et al., 2021). For instance, adding a momentum term can accelerate the weight updating processing that may help the model to push out of the local optima. The momentum term will keep changing the weights continuously with an appropriate ratio. During the training, it is possible that the derivative of the cost function produces zero value. Even in such a situation the model continues to update weights using the previous iteration's values of the cost function (Sutskever et al., 2013). It is interesting to note that during learning it is not possible to determine whether the solution obtained is optimal or reached a local. In both cases, the model will stopped as there will be no change in the parameter values over consecutive iterations. The model depends on several parameters that affect its performance. Learning rate (LR) also known as step-size is one of the crucial parameters. Fine tuning LR plays crucial role in obtaining optimal solution. Selection of a small value may allow the model to reach the optimal solution very slowly. In contrast, a large value may allow the model to reach the optimal solution faster. However, there is a trade-off between selection of a large/small value with the optimal solution. Therefore, care must be taken in selection of this crucial parameter. This problem can be solved using a scheduled rate. The most commonly used technique is to multiply the gradient with a constant during training of the model. The main issue with such technique is that the LR may not scale well during training. There are various solutions proposed to overcome this problem, such as time-based techniques where the LR is altered as the training proceeds (Li and Arora, 2019). Some other techniques, such as Adagrad and RMSProp are also proposed to solve this problem. These techniques apply adaptive optimization on the LR to adapt its value during the training (Duchi JDUCHI and Singer, 2011) (Reddi et al., 2019) (Yi et al., 2020). Some research proposed to combine both adaptive optimization adaptive LR schedules to further improve the accuracy of the model. However, these methods only apply a function in such as way that it decreases the LR as the model training proceeds. The main drawback of such techniques is that it may stuck in local minimum due to small gradient changes (Rumelhart et al., 1986) (Sohl-Dickstein et al.,2014).

Other advanced techniques to solve these bottlenecks include different activation functions (Klambauer et al., 2017) (Nair and Hinton, 2010), batch normalization (Ioffe and Szegedy, 2015), novel initialization schemes (He et al., 2015), and dropout (Srivastava et al., 2014). The main drawback of these methods is the higher computational overhead, which limits the performance improvements in terms of CPU cost, convergence rate, and optimal error.

The most common techniques for optimizing the deep neural networks (DNN) include batch gradient (BGD) and stochastic gradient descent (SGD) algorithms. BGD is usually slower and is more suitable for a small size dataset. On the other hand, SGD is faster and is more suitable to process large size data. Typically, SGD produces less reliable results which may also lead to bad convergence. In (Yang, 2021), authors proposed a new method based on the Kalman filter for better optimization of the network using adaptive filtering. The method employed the historical state of the optimization, which helped reduce the estimation variance in the SGD algorithm. This led to faster convergence and resulted in better gradient direction estimation even in the presence of noise.

Other gradient-based methods, such as adaptive gradient methods (AGMs), can also be employed to

optimize nonconvex problems in machine learning, specifically deep learning. In (Tong et al., 2022), two improvements of AGMs are proposed to enhance the model's accuracy further. It was observed that the anisotropic scale of the adaptive learning rate (A-LR) has high variations across multiple dimensions of the nonconvex optimization problem. This variation may lead to slower convergence and the model may get stuck in the local minima. The literature shows that a number of research are dedicated to improving the AGMs using A-LR. Another main bottleneck that plays vital role in obtaining the optimal accuracy is finding optimal values for its hyperparameters used in the A-LR. In some works, authors proposed adding activation functions in A-LR such as softplus function for AGM's improvement. Two such methods, namely SADAM and SAMSGRAD are also proposed to improve the model accuracy. Results showed that SAMSGRAD exhibit faster convergence than the AMSGRAD under various conditions such as nonconvex, non-strongly convex, and Polyak-Łojasiewicz conditions.

Another adaptive gradient descent algorithm that is commonly used in backpropagation (BP) for training feed-forward neural networks (FFNNs) is called Adam. The Adam algorithm's main issue is that it might fail to reach global optima. Solutions based on metaheuristic methods exist, which help train FFNNs to overcome the local minima issue. However, the solutions also have compromise on the convergence efficiency of the model compared to the Adam optimizer. A solution was proposed in terms of an ensemble of differential evolution and Adam (EDEAdam), combining both Adam optimizer and differential evolution algorithm, which forms a robust and efficient search mechanism to achieve better results in both global and local search. The integration of these two methods not only helped improve results but also showed faster convergence speed (Xue et al., 2022).

Hameed et al. (Hameed et al., 2016), proposed a BP algorithm with variable adaptive momentum (BPVAM). The algorithm improves the convergence behavior by achieving faster convergence, optimal error, and lower mathematical complexity, reducing the overall CPU cost and processing time. The learning rate is a crucial parameter that controls the model. The learning rate parameter depends on the input data's eigenvalues of the autocorrelation matrix.

This study investigates the learning performance of BPVAM algorithm. An adaptive momentum scheduler is introduced to overcome the gradient vanishing problem. A detailed set of experiments are performed on various benchmark datasets to evaluate the performance of the proposed model. The main contributions of this study are highlighted as follows:

- Introduction of a variable adaptive momentum term in the weight update equation.
- Fine-tuning the hyperparameters for computing an optimal momentum in stochastic gradient descent in BPVAM algorithm
- Investigate the model's behavior with adaptive momentum term and compare it with models with a fixed learning rate.
- Diverse set of experiments on different benchmark datasets are performed to test the efficiency and robustness of the proposed model.

The paper is organized as follows. In Section 2, details about the adaptive learning rate algorithms are presented. Extensive set of experiments are presented in Section 3. Finally, the paper is completed with conclusion.

2. Backpropagation Algorithm with Variable Adaptive Momentum (BPVAM)

Hameed et al (Hameed et al., 2016), introduced the BPVAM algorithm see fig. 1, where α (the adaptive momentum) is controlled by the learning rate parameter η. In this case, if given initial weights Ψ^0 and Ψ^1, and a momentum factor $\alpha\ \epsilon(0,1)$, BPVAM updates the weight vector iteratively which means that equation (28) can now be represented as

$$\Delta\Psi_i = \eta\delta_w\Psi_i + \alpha_\Psi^i\Delta\Psi_{i-1},\ i = 1, 2, ..., \quad (1)$$

Where $\eta > 0$ is the learning rate which is assumed to be a constant in this work and $\alpha_\Psi^i = (\alpha_{\Psi_0}^i, \alpha_{\Psi_1}^i, \alpha_{\Psi_2}^i, \dots\dots\dots, \alpha_{\Psi_q}^i)$ is the momentum coefficient vector at the i^{th} training iteration which is constituted by the coefficient α_Ψ^i for every $\Delta\Psi_i^i$ (i= 0, 1, 2, ….,q) and for each α_Ψ^i, it is adjusted after each training epoch by

$$\alpha_{\Psi_i}^i = \begin{cases} \alpha \cdot \frac{-\eta\delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1}}{\left\|\Delta\Psi_i^{i-1}\right\|^2} & if\ \delta_{\Psi_i}\Psi_i \cdot \Delta\Psi_i^{i-1} < 0 \\ 0 & otherwise \end{cases} \quad (2)$$

In the BPAM, α (the adaptive momentum) was controlled by the learning rate η, where η is dependent on the eigenvalues of the autocorrelation matrix of the input.

The work presented by (Hameed et al., 2016) estimates the autocorrelation matrix *R(i)* of the input recursively as

$$R_i = \beta R_i + Rxx \quad (3)$$

Where β is the forgotten factor (0<<β<1), and $Rxx = E\{X(i)X^T(i)\}$, E is the expectation operator. Tacking the expected value of both sides of equation (32) produces

$$\bar{R}_i = \frac{1-\beta^i}{1-\beta}Rxx \quad (4)$$

Where $\bar{R}_i = E\{R_i\}$. Solving equation (32) in the steady state ($i \to \infty$) yields

$$\bar{R}_i = \frac{1}{1-\beta} \tag{5}$$

In this case, equation (34) implies that the eigenvalues of the estimated autocorrelation matrix increase exponentially, and in the limit they become $\frac{1}{1-\beta}$ times the original value.

The work done by (Hameed et al., 2016) also proposed a variable momentum, which is expressed by

$$\alpha_i = \frac{\lambda}{1-\beta^i} \tag{6}$$

Where $\lambda < \frac{2-2\beta}{max\ eigen\ value\ of \mathbf{R_{xx}}}$ and this case β is the forgetting factor ($0 \ll \beta < 1$),

Assuming that β is large, this will force the term $1 - \beta^i$ to reach unity, and assuming that the initial $\alpha(i)$ is relatively large, to provide fast convergence of the weights. By updating equations (27) and (28), with time it becomes very close to λ (a small positive constant) hence it provides law error, equation (27) and (28) can then be represented as

$$\Delta\Psi_{ji}(i+1) = \eta\delta_y \mathbf{x}_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{ji}(i) \tag{7}$$

$$\Delta\Psi_{kj}(i+1) = \eta\delta_o y_i(i) + \left(\frac{\lambda}{1-\beta^i}\right)\Delta\Psi_{kj}(i),\ i = 0,1,\ldots \tag{8}$$

Where i represents the number of iterations and $\Delta\Psi$ is defined as updating the weights.

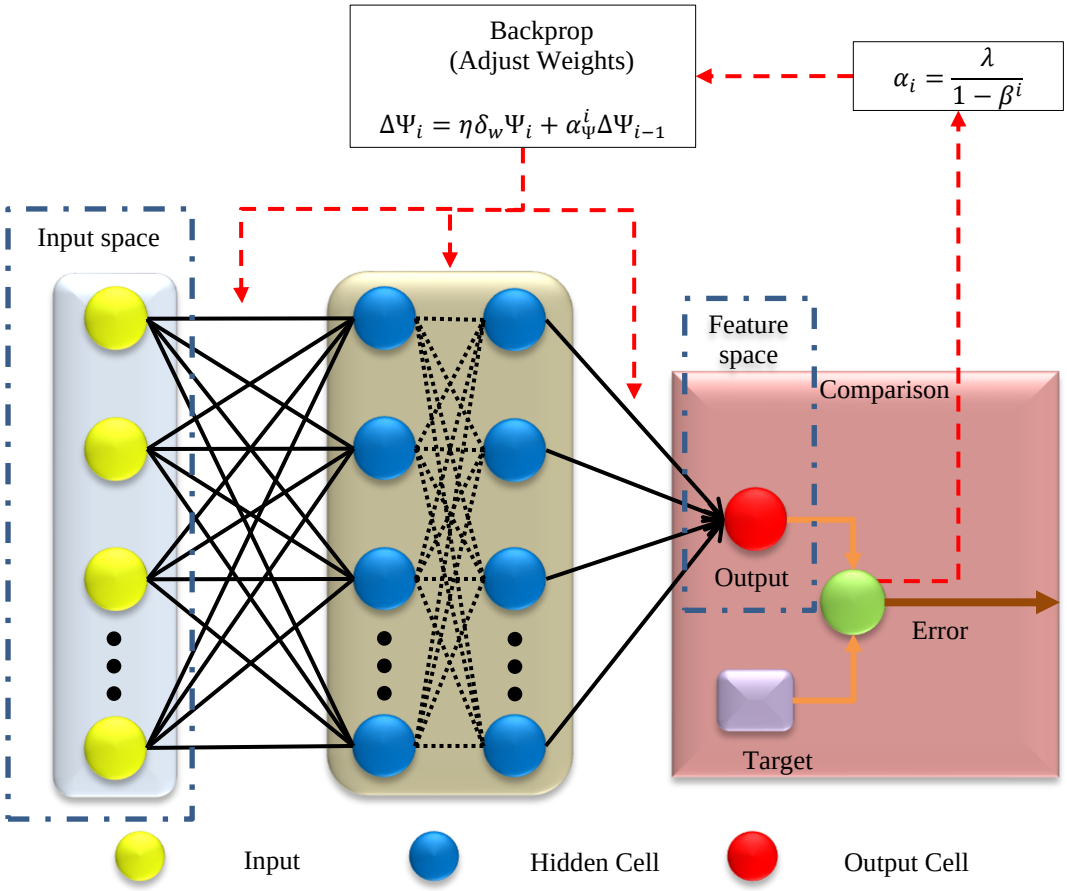


Figure 1. BPVAM architecture

3. Experimental Results

The experiments were performed on four different data sets obtained from various domains. A diverse set of datasets were considered for testing the application of the proposed method for different types of data. These datasets include Breast cancer, Heart Disease, Lung Cancer, and Iris. Each dataset has a varying number of samples, attributes, and classes (Asuncion and Newman,2007).

3.1 Preprocessing and Experimental Setup

All data in the dataset was normalized between 0 and 1 using the Min-Max normalization method. The main advantage of the normalization is maintaining stability in the network by allowing all the weights to converge almost simultaneously. Moreover, missing data were replaced with the mean value of the attribute.

All the experiments were performed in the Matlab™ environment. The models were executed on a Dell machine with Intel core i7, 2.10 GHz processor with 16 GB of RAM and NVIDIA™ GeForce TMGTX 1080. The dataset was divided into training (70%) and testing (30%) for each experiment. Since the dataset was balanced, therefore, no augmentation was performed.

3.2 Evaluation

The performance evaluation of BPVAM and its comparison with conventional BP was carried out in terms of accuracy and SSE on four benchmark datasets. Moreover, the models were also compared in terms of mean and standard deviation behaviors over the whole training process. Since the models depend on various hyperparameters, therefore, the optimal values of these hyperparameters were obtained using the Grid Search algorithm. The obtained optimal values of hyperparameters were then used to train the models. Six different cases were considered with varying values of the hyperparameter to evaluate the impact of the hyperparameter on the model accuracy. The experimental setup was similar to the one presented by the authors in (Hameed et al., 2016). The evaluation results for each dataset are described in detail as below.

Table 1 summarizes the results obtained on the **breast cancer** dataset. As it can be seen, the error convergence for the BPVAM (4.678) is better than the conventional BP (4.707) algorithm in terms of SSE for case 6. For other cases (1-5), the performance of BPVAM was also higher than conventional BP as it produced less error. Similarly, in terms of accuracy, the BPVAM algorithm produced higher accuracy compared BP in general overall six cases. It is observed that BPVAM obtained optimal results with $\eta = 0.9$, $\lambda = 0.0085$, and $\beta = 0.992$, whereas conventional BP produced best results with $\alpha = 0.01$, and $\eta = 0.9$.

Table 2 summarizes the evaluation results obtained **heart disease** dataset. These results showed that the BPVAM always produced better results than the conventional BP algorithm. Highest accuracy (61.96%) was obtained for BPVAM and lowest error (3.961) with parameter values of η = 0.03, λ = 0.022, β = 0.995. It is interesting to note that the accuracy of models tend to become close to each other as the parameter values were decreased from case-1 to case-6.

Table 3 shows the experimental results obtained by the models on the **Lung Cancer** dataset. The SSE and accuracy of BPVAM were BP 0.0054 and 60.00%, respectively. Similarly, for BP the SSE and accuracy remained 0.0063 and 60.00, respectively. The cases show that the convergence behavior of BP is very slow and very sensitive to the hyperparameter selection compared to BPVAM. The best results were obtained for BPVAM with parameters $\eta = 0.1$, $\lambda = 0.005$, and $\beta = 0.9980$. For BP BP optimal results were obtained with $\alpha = 0.05$, and $\eta = 0.1$.

Table 4 summarizes the results obtained on the **Iris** dataset. The results show that case 1 produced the optimal results for BPVAM with an accuracy of 84.44% and SSE of 0.853, while for BP the accuracy was 77.78% and SSE of 0.991 for BP. Following all cases from 1 to 6, it shows the BPVAM is more robust and can keep improving the network model steadily.

Further experiments were performed to compare the performance of the two models in terms of mean and standard deviation. Figure 2 and 3 shows the comparison of models in terms of the mean and standard deviation obtained for accuracy and SSE, respectively. It is evident that despite the improvement of the BP algorithm, the significant change indicates the sensitivity of the algorithm to its selection of parameters. On the other hand, the BPVAM algorithm shows its superiority from the first case until the sixth case. It increases the mean accuracy of the model while decreasing the standard deviation over all cases. We can deduce that the overall BPVAM model outperformed BP in terms of accuracy and SSE.

Table 1. Performance comparison metrics of the tested algorithms for Breast Cancer dataset

Case	Algorithm	α	η	λ	β	SSE	Accuracy (%)
1	BP	0.06	0.4	-	-	7.244	60.34
	BPVAM	-	0.4	0.0090	0.997	6.873	63.79
2	BP	0.05	0.5	-	-	5.691	67.24
	BPVAM	-	0.5	0.0089	0.996	5.685	67.24
3	BP	0.04	0.6	-	-	5.656	68.97
	BPVAM	-	0.6	0.0088	0.995	5.053	70.69
4	BP	0.03	0.7	-	-	4.924	72.41
	BPVAM	-	0.7	0.0087	0.994	4.787	75.86
5	BP	0.02	0.8	-	-	4.801	74.14
	BPVAM	-	0.8	0.0086	0.993	4.780	75.86
6	BP	0.01	0.9	-	-	4.707	77.59
	BPVAM	-	0.9	0.0085	0.992	4.678	77.59

Table 2. Performance comparison metrics of the tested algorithms for Heart Disease dataset

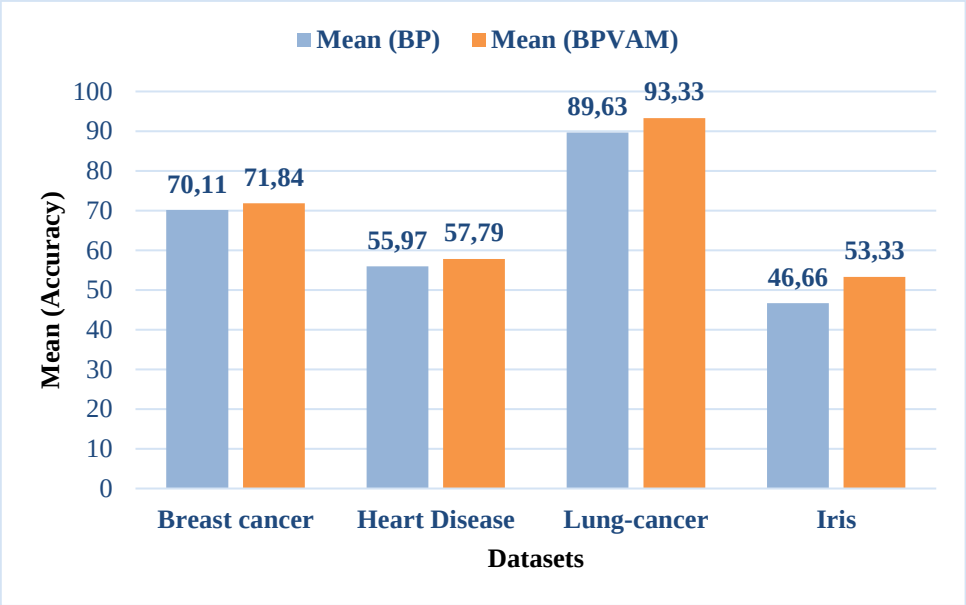
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.06	0.08	-	-	5.306	48.91
	BPVAM	-	0.08	0.027	0.999	4.900	51.09
2	BP	0.05	0.07	-	-	4.648	52.17
	BPVAM	-	0.07	0.026	0.999	4.615	54.35
3	BP	0.04	0.06	-	-	4.586	55.43
	BPVAM	-	0.06	0.025	0.998	4.022	58.70
4	BP	0.03	0.05	-	-	4.332	57.61
	BPVAM	-	0.05	0.024	0.997	4.017	59.78
5	BP	0.02	0.04	-	-	4.020	59.78
	BPVAM	-	0.04	0.023	0.996	4.001	60.87
6	BP	0.01	0.03	-	-	3.969	61.96
	BPVAM	-	0.03	0.022	0.995	3.961	61.96

Table 3. Performance comparison metrics of the tested algorithms for Lung-Cancer dataset

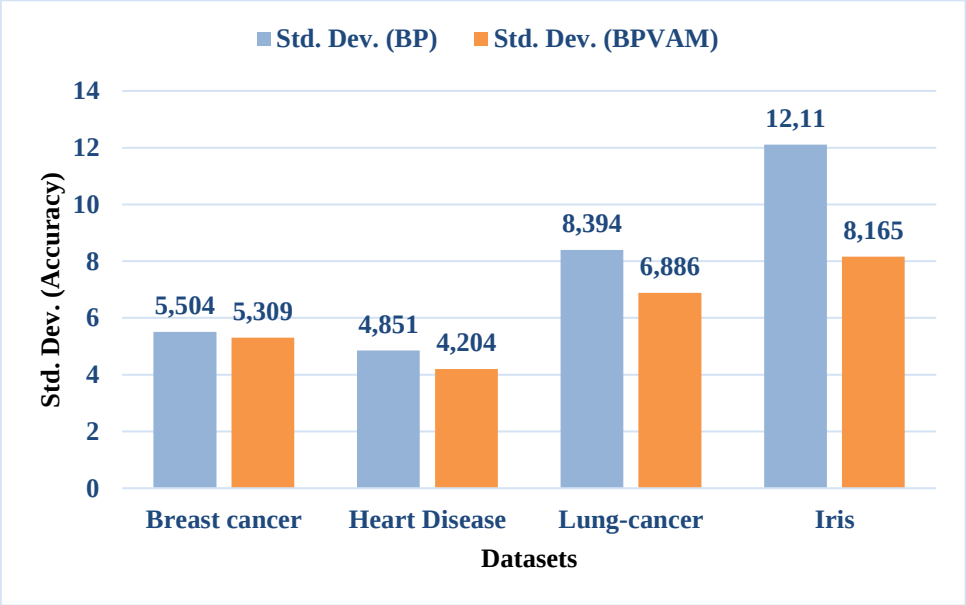
Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.10	0.6	-	-	0.0275	30.00
	BPVAM	-	0.6	0.010	0.9994	0.0161	40.00
2	BP	0.09	0.5	-	-	0.0163	40.00
	BPVAM	-	0.5	0.009	0.9993	0.0081	50.00
3	BP	0.08	0.4	-	-	0.0120	40.00
	BPVAM	-	0.4	0.008	0.9992	0.0075	50.00
4	BP	0.07	0.3	-	-	0.0081	50.00
	BPVAM	-	0.3	0.007	0.9991	0.0066	60.00
5	BP	0.06	0.2	-	-	0.0072	60.00
	BPVAM	-	0.2	0.006	0.9990	0.0060	60.00
6	BP	0.05	0.1	-	-	0.0063	60.00
	BPVAM	-	0.1	0.005	0.9980	0.0054	60.00

Table 4. Performance comparison metrics of the tested algorithms for Iris dataset

Case	Algorithm	α	η	λ	β	Training Cost	Accuracy Performance
1	BP	0.006	0.10	-	-	0.991	77.78
	BPVAM	-	0.10	0.07	0.9994	0.853	84.44
2	BP	0.005	0.09	-	-	0.926	82.22
	BPVAM	-	0.09	0.06	0.9993	0.812	86.67
3	BP	0.004	0.08	-	-	0.756	88.89
	BPVAM	-	0.08	0.05	0.9992	0.618	91.11
4	BP	0.003	0.07	-	-	0.516	93.33
	BPVAM	-	0.07	0.04	0.9991	0.201	97.78
5	BP	0.002	0.06	-	-	0.334	95.56
	BPVAM	-	0.06	0.03	0.9990	0.182	100.00
6	BP	0.001	0.05	-	-	0.184	100.00
	BPVAM	-	0.05	0.02	0.9980	0.180	100.00

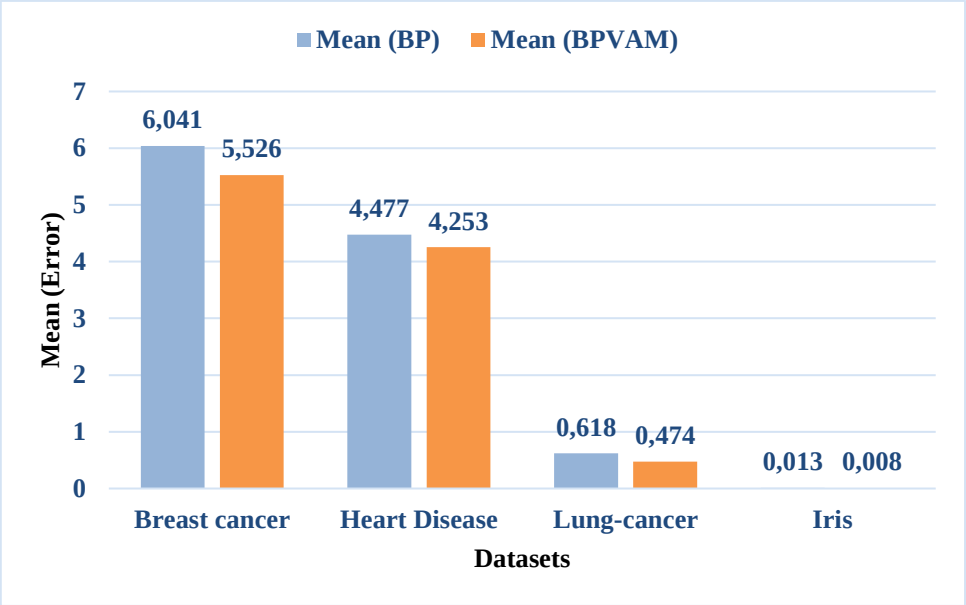


(a)

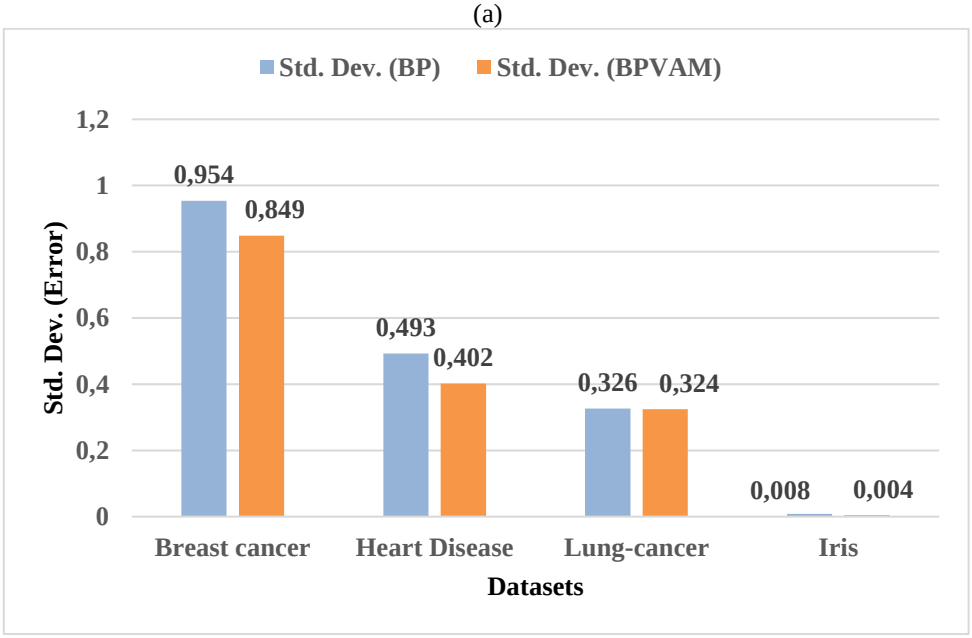


(b)

Figure 2. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean accuracy, and b) standard deviation accuracy



(a)



(b)

Figure 3. Performance evaluation metrics of BP and BPVAM for four benchmarks, a) mean error, and b) standard deviation error

4. Conclusions

This study investigated the approach for obtaining an optimal set of hyperparameters for the machine learning model. Moreover, the model's weight matrix is updated using the adaptive momentum to help it overcome the local optima problem. The algorithm is controlled by different hyperparameters, which are fine-tuned using grid search. The results showed that the BPVAM algorithm obtains better convergence behavior than BP in the optimal steady-state models. The experiments investigated the compared methods from different aspects by considering the whole learning behavior in different training cases. The optimal results obtained on four benchmark datasets indicate that BPVAM improved the accuracy and robustness of the model. Moreover, this study suggests a significant improvement in accuracy, mean error, and standard deviation when the BPVAM is optimized with adaptive momentum. It can be observed that BPVAM exhibit features to guarantee its convergence and produce a much lower SSE against any valid data sets. In the future, we aim to apply this optimization algorithm to obtain an optimal set of parameters for a deep end-to-end neural network to overcome the issue of obtaining the optimal

hyperparameters, we also are plan to monitor the progress of the hyperparameter optimization in real-time. This will allow the extraction of highly discriminative features from input data that can improve the model's performance.

References

A. and Newman, D. J. (2007). UCI Machine Learning Repository, Department of Information and Computer Sciences, University of California, Irvine. Available at www.ics.uci.edu/~mlearn/MLRepository.html.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks 5, 157–166. https://doi.org/10.1109/72.279181

Demircan Keskin, F., Çiçekli, U., İçli, D., 2022. Prediction of Failure Categories in Plastic Extrusion Process with Deep Learning. Journal of Intelligent Systems: Theory and Applications, 5(1), 27–34. https://doi.org/10.38016/jista.878854

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

Erol, B.A., Majumdar, A., Lwowski, J., Benavidez, P., Rad, P., Jamshidi, M., 2018. Improved deep neural network object tracking system for applications in home robotics, in: Studies in Computational Intelligence. Springer Verlag, pp. 369–395. https://doi.org/10.1007/978-3-319-89629-8_14

Gemirter, C. B., Goularas, D., 2021. A Turkish Question Answering System Based on Deep Learning Neural Networks. Journal of Intelligent Systems: Theory and Applications, 4(2), 65–75. https://doi.org/10.38016/jista.815823

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256.

Güney, E., Çakmak, O., Kocaman, Ç., 2022. Classification of Stockwell Transform Based Power Quality Disturbance with Support Vector Machine and Artificial Neural Networks. Journal of Intelligent Systems: Theory and Applications, 5(1), 75–84. https://doi.org/10.38016/jista.996541

Hameed, A.A., Karlik, B., Salman, M.S., 2016. Back-propagation algorithm with variable adaptive momentum. Knowledge-Based Systems 114, 79–87. https://doi.org/10.1016/j.knosys.2016.10.001

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034.

Hertel, L., Collado, J., Sadowski, P., Ott, J., Baldi, P., 2020. Sherpa: Robust hyperparameter optimization for machine learning. SoftwareX, 12, 100591.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Houssein, E.H., Emam, M.M., Ali, A.A., Suganthan, P.N., 2021. Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review. Expert Systems with Applications, 167. https://doi.org/10.1016/j.eswa.2020.114161

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pp. 448-456.

Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E., 2020. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics 404. https://doi.org/10.1016/j.jcp.2019.109136

Jain, D.K., Shamsolmoali, P., Sehdev, P., 2019. Extended deep neural network for facial emotion recognition. Pattern Recognition Letters 120, 69–74. https://doi.org/10.1016/j.patrec.2019.01.008

Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., Zareapoor, M., 2018. Hybrid deep neural networks for face emotion recognition. Pattern Recognition Letters 115, 101–106. https://doi.org/10.1016/j.patrec.2018.04.010

Jin, J., Zhu, J., Gong, J., Chen, W., 2022. Novel activation functions-based ZNN models for fixed-time solving dynamirc Sylvester equation. Neural Computing and Applications, 1-19. https://doi.org/10.1007/s00521-022-06905-2

Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., 2017. Self-Normalizing Neural Networks, Advances in neural information processing systems, 30.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems, 25.

Li, Z., Arora, S., 2019. An Exponential Learning Rate Schedule for Deep Learning. arXiv preprint arXiv:1910.07454.

Liu, M., Chen, L., Du, X., Jin, L., Shang, M., 2021. Activated Gradients for Deep Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 1–13. https://doi.org/10.1109/tnnls.2021.3106044

Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Hibbett, M.J., Estrada, G., Ma'ruf, K., Coras, F., Ermagan, V., Latapie, H., Cassar, C., Evans, J., Maino, F., Walrand, J., Cabellos, A., 2017. Knowledge-defined networking. Computer Communication Review 47, 1–10. https://doi.org/10.1145/3138808.3138810

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In Appearing in Proceedings of the 27 th International Conference on Machine Learning (ICML).

Park, J., Yi, D., Ji, S., 2020. A novel learning rate schedule in optimization for neural networks and it's convergence. Symmetry (Basel) 12. https://doi.org/10.3390/SYM12040660

Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., Yang, B., 2019. Deep learning-based object classification on automotive radar spectra, in: 2019 IEEE Radar Conference, RadarConf 2019. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/RADAR.2019.8835775

Rahman, M.M., Tan, Y., Xue, J., Lu, K., 2020. Notice of Removal: Recent Advances in 3D Object Detection in the Era of Deep Neural Networks: A Survey. IEEE Transactions on Image Processing. https://doi.org/10.1109/TIP.2019.2955239

Reddi, S.J., Kale, S., Kumar, S., 2019. On the Convergence of Adam and Beyond. arXiv preprint arXiv:1904.09237.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Sandha, S. S., Aggarwal, M., Fedorov, I., Srivastava, M. 2020. Mango: A python library for parallel hyperparameter tuning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3987-3991.

Sarvamangala, D.R., Kulkarni, R. v., 2022. Convolutional neural networks in medical image understanding: a survey. Evolutionary Intelligence, 1-22. https://doi.org/10.1007/s12065-020-00540-3

Seong, S., Lee, Y., Kee, Y., Han, D., Kim, J., 2018. Towards Flatter Loss Surface via Nonmonotonic Learning Rate Scheduling, In UAI.

Sharma, N., Jain, V., Mishra, A., 2018. An Analysis of Convolutional Neural Networks for Image Classification, in: Procedia Computer Science. Elsevier B.V., pp. 377–384. https://doi.org/10.1016/j.procs.2018.05.198

Smith, L.N., Topin, N., 2019. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Artificial intelligence and machine learning for multi-domain operations applications, Vol. 11006, pp. 369-386.

Sohl-Dickstein, J., Poole, B., Ganguli, S., 2014. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In International Conference on Machine Learning, pp. 604-612.

Srivastava, N., Hinton, G., Krizhevsky, A., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research. 15(1), 1929-1958.

Sun, J., Yang, Y., Xun, G., Zhang, A., 2022. Scheduling Hyperparameters to Improve Generalization: From Centralized SGD to Asynchronous SGD. ACM Transactions on Knowledge Discovery from Data (TKDD). https://dl.acm.org/doi/pdf/10.1145/3544782.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pp. 1139-1147.

Tong, Q., Liang, G., Bi, J., 2022. Calibrating the adaptive learning rate to improve convergence of ADAM. Neurocomputing 481, 333–356. https://doi.org/10.1016/j.neucom.2022.01.014

Xue, Y., Tong, Y., Neri, F., 2022. An ensemble of differential evolution and Adam for training feed-forward neural networks. Information Sciences. Information Sciences, 608, 453-471. https://doi.org/10.1016/j.ins.2022.06.036

Yan, Z., Chen, J., Hu, R., Huang, T., Chen, Y., Wen, S., 2020. Training memristor-based multilayer neuromorphic networks with SGD, momentum and adaptive learning rates. Neural Networks 128, 142–149. https://doi.org/10.1016/j.neunet.2020.04.025

Yang, X., 2021. Kalman optimizer for consistent gradient descent, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. Institute of Electrical and Electronics Engineers Inc., pp. 3900–3904. https://doi.org/10.1109/ICASSP39728.2021.9414588

YD., Ahn, J., Ji, S., 2020. An effective optimization method for machine learning based on ADAM. Applied Sciences (Switzerland) 10. https://doi.org/10.3390/app10031073

YH., Yang, L.T., Zhang, Q., Armstrong, D., Deen, M.J., 2021. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. Neurocomputing 444, 92–110. https://doi.org/10.1016/j.neucom.2020.04.157

Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 5(2) (2022) 178-188
DOI: 10.38016/jista.1120383

Tekrarlayan Sinir Ağları Temelli Rüzgâr Hızı Tahmin Modelleri: Yalova Bölgesinde Bir Uygulama

Fuat Kosanoğlu[1*], Zeliha Nur Kiriş[2], Ömer Faruk Beyca[3]

[1] Yalova Üniversitesi, Endüstri Mühendisliği Bölümü, Yalova, Türkiye

[2] İstanbul Teknik Üniversitesi, Endüstri Mühendisliği Bölümü, İstanbul, Türkiye

[3] İstanbul Teknik Üniversitesi, Endüstri Mühendisliği Bölümü, İstanbul, Türkiye

fuat.kosanoglu@yalova.edu.tr, kiris17@itu.edu.tr, beyca@itu.edu.tr

Öz

Küresel ısınma ve fosil yakıtların çevreye verdiği zararlardan dolayı son yıllarda yenilenebilir enerji kaynakları büyük önem kazanmıştır. Özellikle Paris İklim Anlaşmasıyla ülkeler çevreye zararlı gaz salınımını azaltmak konusunda taahhütlerde bulunmuşlardır. Günümüzde en önemli yenilenebilir enerji kaynaklarından biri de rüzgâr enerjisidir. Türkiye'nin sahip olduğu rüzgâr potansiyeli düşünüldüğünde enerji üretiminde rüzgâr enerjisi daha da önem kazanmaktadır. Rüzgâr enerjisi temiz bir enerji kaynağı olmasına rağmen rüzgârın değişken bir kaynak olması nedeniyle üretilen enerjinin verimli kullanılıp dağıtılabilmesi ve planlama yapılabilmesi sağlıklı rüzgâr enerjisi üretim tahminlerine dayanmaktadır. Bu çalışmada, dört farklı Tekrarlayan Sinir Ağları (TSA) modeli rüzgâr enerjisi üretim tahminlemesi için kullanılmıştır. Çalışmada, Türkiye'nin Yalova ilinde bulunan bir istasyondan elde edilen veriler kullanılarak kısa süreli rüzgâr hızı tahmini yapılmıştır. Analizde bir saat sonrasını tahmin ederek oluşacak ani arıza ve bakım planlamalarına müdahale edilmesi amaçlanmıştır. Öncelikle istasyondan alınan veriler incelenmiş, veri analizleri yapılmış, var olan verilerden yeni veriler üretilmiş ve veri setleri modeller için uygun hale getirilmiştir. Modellerden elde edilen performans sonuçları kabul edilebilir aralıkta olup TSA yöntemlerinin rüzgâr hızı tahmininde başarılı bir şekilde kullanılabileceğini, ve geleneksel zaman serisi yöntemlerine göre daha iyi sonuçlar verdiği sonucuna varılmıştır.

Anahtar kelimeler: rüzgâr hızı tahmini, rüzgâr enerjisi, ARIMA, tekrarlayan sinir ağları

Recurrent Neural Networks Based Wind Speed Forecasting Models: A Case Study of Yalova

Abstract

Global warming and other adversarial effects caused by fossil fuel sources, renewable energy sources have been attracted more than ever. Especially, parties of Paris Climate Agreement countries pledge to reduce greenhouse gas emissions. Among renewable energy sources, wind energy is one of the significant and eligible source to produce energy sustainably. Wind energy is also one of the most important renewable energy source due to Turkey's notable wind energy potential. Although wind energy is one of the most important clean energy sources, there are several challenges, such as intermittent and uncertain nature of wind places. Therefore, efficient and reliable energy planning and distribution mostly rely on prediction of wind energy with high accuracy. In this study, we propose four Reccurent Neural Network (RNN) methods to predict short-term wind energy production. We utilize data obtained from a station located in Yalova, Turkey to assess the performance of proposed algorithms. In our analysis, we plan to improve maintenance planning and intervene the sudden breakdowns by predicting 1 hour ahead energy production. First, we analyze the data received from the station, and the data sets were made suitable for the models. The performance results obtained from the models are plausible. Our results indicate that RNN methods can be successfully used to predict wind speed.

Keywords: wind speed forecasting, wind energy, ARIMA, recurrent neural networks

* Sorumlu yazar.
E-posta adresi: fuat.kosanoglu@yalova.edu.tr

Alındı : 24 Mayıs 2022
Revizyon : 26 Ağustos 2022
Kabul : 17 Eylül 2022

1. Giriş (Introduction)

Dünya nüfusunun artması ve teknolojik cihazların hayatımızda önemli yer alması, beraberinde enerjiye olan ihtiyacı da artırmaktadır. Bu ihtiyacı karşılamak için kullanılan kaynakların başında fosil yakıtlar olan petrol, doğalgaz ve kömür yer almaktadır. Bu kaynakların kısıtlı olması gerçeği, uzun vadede artan elektrik ihtiyacının karşılanmasında yeterli olmayacağı sonucunu ortaya koymaktadır. Bununla beraber, küresel ısınma ve fosil yakıtların çevreye verdiği zararlar fosil yakıtlara alternatif, temiz ve sürdürülebilir enerji kaynaklarına yönelimi zorunlu kılmıştır. Özellikle Paris İklim Anlaşmasıyla ülkeler çevreye zararlı gaz salınımını azaltmak konusunda taahhütlerde bulunmuşlardır. Bu nedenlerden dolayı son yıllarda yenilenebilir enerji kaynakları büyük önem kazanmış ve bu kaynaklardan enerji üretimi için yapılan yatırımlar hız kazanmıştır. Başlıca yenilenebilir enerji kaynakları, güneş enerjisi, rüzgâr enerjisi, jeotermal enerji olarak listelenebilir.

Türkiye'de toplam elektrik enerjisinin %7.4'ü rüzgar enerjisi ile sağlanmaktadır. Rüzgâr yoğunluğuna bağlı olarak üretim miktarları değişen santraller için, konum ve iklim koşulları rüzgâr çiftlikleri kurulumu için önemli parametrelerdir. Türkiye'de kurulu rüzgar enerjisi santrallerinin büyük çoğunluğu kıyı Ege ve doğu Marmara'da bulunmakta olup %19.2 oranıyla İzmir başta gelmektedir (*Türkiye Rüzgar Enerjisi İstatistik Raporu*, 2019).

Yük tahmini, devletin ve enerji sağlayan şirketlerin arz-talep dengesini kurmak için gereken enerjiyi tahmin ettikleri bir tekniktir. Güç sistemlerinin etkin kullanılması ve planlanmasında, yük tahminin doğru yapılması büyük önem arz etmektedir. Enerji tahmininin yüksek yapılması, işletmenin inşası sırasında önemli ölçüde kaynak israfıma sebep olmaktadır. Düşük tahmin gerçekleştiğinde ise enerjiden faydalanan kesimin enerji talebi karşılanamaz ve sistemin verimsiz bir şekilde çalışmasına neden olur (Wang, Guo ve Huang, 2011).

Dünyada, ülkelerin gelişmişlik düzeylerini ifade eden değişkenlerden biri enerji tüketimi olması sebebiyle Türkiye gibi gelişmekte olan ülkeler için enerji önemli bir ekonomik göstergedir. Bu gösterge neticesinde dışa bağımlılık, elektrik faturalarının artması, teminde aksaklık ve sıkıntı yaşanması gibi durumlar sebebiyle halkın refahı için bir sorun teşkil etmektedir. Bağımlı olunan enerjinin azaltılması için yerli kaynaklara yatırım yapılmalıdır. Bu nedenle rüzgâr enerjisi büyük önem arz etmektedir. Rüzgâr enerjisinden elektrik üretiminde ise doğru rüzgâr hızı tahminlemesi kritik öneme sahiptir.

Bu çalışmada farklı derin öğrenme metotları ve klasik zaman serisi metotları kullanılarak kısa dönemli rüzgâr hızı tahminlemesi yapılması amaçlanmıştır. Bu çalışma akademisyen ve uygulayıcılara rüzgâr hız tahminlemesi yaparken veri ön işlemesi, veri modellemesi ve tahminleme yapmak için kılavuz sunmaktadır. Bu çalışma kapsamında 3 farklı derin öğrenme metodu ve 1 klasik zaman serisi metodu (ARIMA) Yalova ili rüzgâr verileri kullanılarak uygulanmıştır. Modellerin performansı Mutlak Hataların Ortalama Yüzdesi (MAPE), mutlak hataların ortalaması (MAE) ve hataların karelerinin ortalaması (MSE) ölçekleri kullanılarak karşılaştırılmıştır. Modellerde girdi olarak geçmiş 24 saatlik veriler kullanılarak bir sonraki saatin rüzgâr hızı tahminlemesi hedeflenmiştir.

Çalışmanın ikinci bölümünde detaylı yük tahminleme ile ilgili literatür taramasına yer verilmiştir. Üçüncü bölümde yük tahmini ve yük tahminine etki yapan faktörler ele alınmıştır. Dördüncü bölümde çalışmada kullanılan yöntemler anlatılmıştır. Beşinci bölümde Yalova ili için elde edilen veriler kullanılarak önerilen modeller test edilmiştir. Son bölümde elde edilen sonuçlar ve öneriler sunulmuştur.

2. Literatür Araştırması (Literature review)

Enerji endüstrisinde yük tahmini, üretim planlama ve fiyat tahmininde en önemli bileşen olarak karşımıza çıkmaktadır. Bu tahminler, hem güç sistemleri, hem ticari kuruluşlar tarafından kullanılmaktadır. 2010-2019 yılları arasında yapılan çalışmalara bakıldığında, enerji tahmin çalışmalarının yarısı yük tahmin çalışmalarıdır. Diğer çalışmalar, fiyat, rüzgâr ve güneş tahminleridir (Hong vd., 2020).

Rüzgâr hızı tahmin modelleri ile ilgili çalışmalar kısa dönem ve uzun dönem olmak üzere iki ana grupta sınıflandırılabilir. Kısa dönem tahmin modelleri 1 saatten birkaç günlük tahminleri içermekte iken uzun dönem tahminler ise 1 yıl sonrasına kadar uzanmaktadır. Her iki tahminleme dönemleri için istatistiksel yöntemler, fiziksel yöntemler, yapay zekâ ve hibrit modeller geliştirilmiştir (Lei vd., 2009). Geleneksel istatistiksel yöntemler optimum çözümü garanti etmemektedir. Fiziksel yöntemler ise kısa vadeli tahmin sağlayamamakta ve sayesinde rüzgâr enerjisi tahminlerinde gittikçe daha fazla kullanılmaktadır. Hibrit yöntemler ise farklı tahmin yöntemlerinin özelliklerini bir araya getirerek daha yüksek performans elde etmeyi hedefleyen yöntemlerdir.

Kısa dönem rüzgâr enerjisi tahminlemede istatistiksel yöntemlerin kullanıldığı ilk çalışmalardan biri Brown ve diğerleri, (1984) tarafından yapılmıştır. Bu çalışmada tahminlemede otoregresif zaman serileri modelleri (AR) kullanılmıştır. Torres v.d, (2005) otoregresif-hareketli ortalama modeli (ARMA) modeli kullanarak saatlik rüzgar hızı tahminleri yapmışlardır. Rajagopalan & Santoso (2009)aynı yöntemi kullanarak rüzgar hızı tahminleri yapmış ve elde ettikleri sonuçlar yöntemin 1 saat sonrası için tahminlemede başarılı olduğunu göstermiştir. Sfetsos (2002) rüzgar hızı tahminlemede otoregresif entegre hareketli ortalama

(ARIMA) modelini kullanmış, 10 dakika ve 1 saatlik ortalamalar kullanarak 1 saat sonrası için tahminleme yapmıştır. Bu çalışma 10 dakikalık ortalamaların kullanılmasının çok daha etkili olduğunu göstermiştir. Cadenas vd., (2016) rüzgar enerjisi tahmini için tek değişkenli ARIMA ve çok değişkenli NARX modellerini kullanmışlardır. NARX modelinin ARIMA modeline göre daha iyi sonuçlar verdiği gözlemlenmiştir. Ozkan ve Karagöz (2015) dinamik kümeleme ve doğrusal regresyon tekniklerinin birleştirildiği hibrit bir model ortaya koymuşlardır. Bu modelin öne çıkan en önemli özelliği daha az geçmiş veriye ihtiyaç duyması olarak belirtilmiştir. Xie vd., (2019) parametrik olmayan zaman serisi yöntemlerini kısa dönem rüzgâr enerjisi tahminlemede kullanmıslardır. Aasim, Singh ve Mohapatra (2019) tekrarlayan wavelet temelli ARIMA modeli kullanarak kısa dönem rüzgâr enerjisi tahmini yapmışlardır. Önerilen model 1, 3, 5, 7 ve 10 dakikalık zaman aralıklarının tahmini için test edilmiş ve süreklilik modeli ile kıyaslandığında daha iyi sonuçlar verdiği ortaya konmuştur.

Uzun dönem rüzgâr enerjisi tahminlemede istatistiksel yöntemlerin kullanıldığı çalışmaların sayısı nispeten daha azdır. Uzun dönem rüzgâr enerjisi tahminlemede, verilerin yapısı nedeniyle istatistiksel yöntemler tek başlarına çok etkili değildirler vd., 2014). Eldali ve diğerleri (2016) ARIMA tekniğini kullanarak uzun dönem rüzgâr enerjisi tahminleme çalışması yapmışlarıdır. Kavasseri ve Seetharaman (2009) kesirli-ARIMA tekniğini 1 gün sonrası için rüzgâr hızı tahminlemesinde kullanmışlardır. Barbosa de Alencar vd., (2017) ARIMA, Yapay Sinir Ağları (YSA) ve hibrit modelleri kullanarak kısa ve uzun dönem rüzgâr hızı tahminleri yapmışlardır. Dokuz vd., (2018) kümeleme yöntemleri ve ARIMA yöntemlerini kullanarak geliştirdikleri hibrit algoritmayı kullanarak 1 yıl sonrası rüzgâr hızı tahminlemesi yapmışlardır.

Yapay zekâ yöntemleri hem kısa hem de uzun dönem rüzgâr hızı tahminine olanak tanımaktadır. Dumitri ve Gligor (2017) YSA temelli FANN (İleri beslemeli yapay sinir ağları) modeli ile rüzgâr enerjisi tahmini yapılmışlardır. Bu çalışmada Romanya verileri kullanılmış olup günlük rüzgâr hızı tahminleri yapılmıştır. Rüzgâr hızı tahmininde geçmiş yıllara ait günlük rüzgâr enerjisi üretimi ve meteorolojik veriler kullanılmıştır. Modelin tahminleme performansı RMSE ölçütü ile değerlendirilmiştir. Madhiarasan (2021) farklı YSA modelleri kullanarak uzun süreli rüzgar hızı tahmini yapmış ve modellerin tahminleme performanslarını kıyaslamıştır.

Yu vd., (2018) dalgacık dönüşümünü (wavelet transform) geçmiş yıllara ait rüzgâr hızı verilerini alt serilere sınıflandırmak için kullanmışlardır. Bu şekilde, rüzgâr hızı tahmininde alt seriler düşük frekansları, tekrarlayan sinir ağları (RNN) ise derin özelliklerin ortaya çıkarılmasında kullanılmıştır. Diğer bir çalışmada RNN ve hata düzeltme metotları kullanılarak kısa süreli rüzgâr hız tahmini yapılmıştır vd., 2021). LSTM ve GRU metotları daha derin özelliklerin ortaya çıkarılmasında kullanılmıştır. Higashiyama vd., (2017) evrişimli sinir ağları (CNN) mimarisini kullanarak rüzgâr enerjisi tahmininde sayısal hava tahmin verilerinin yüksek boyutlarından kaynaklanan problemlere çözüm bulmaya çalışmıştır. Liu vd., (2018) iki aşamalı bir model geliştirerek rüzgâr hızı tahmini yapmışlardır. İlk aşamada WPD (Wavelet Packet Decomposition) yöntemi kullanılarak rüzgâr hızı zaman serisi alt serilere ayrılmışlardır. İkinci aşamada, CNN yöntemi kullanılarak yüksek frekanslı alt serilerin tahmini yapılırken, CNNLSTM yöntemi ile daha düşük frekanslı olan alt katmanların tahmin yapılmıştır. Çalışmada önerilen model literatürdeki sekiz farklı modelle kıyaslanmış olup, rüzgâr hızı ani değişikliklerine karşılaştırılan modellere göre daha iyi performans gösterdiği belirtilmiştir. Neshat vd., (2021) çift yönlü LSTM yöntemini kullanarak kısa süreli rüzgar hızı tahmini yapmışlardır. Bu çalışmada parametrelerin tahmini için ayrıca bir optimizasyon algoritması kullanılmıştır.

Zhang vd., (2019) meteorolojik veriler ile rüzgar hızı arasındaki korelasyon ve nedenselliği incelenmiştir. Araştırmada kullanılan değişkenler rüzgâr hızı, hava basıncı, nem, sıcaklık ve diğerleri olarak belirtilmiştir. Karmaşık nedensellik ile modelin yorumlanabilirligini arttırmak için standart Uzun Kısa Süreli Bellek Ağları (LSTM) yerine NLSTM (Neighborhood Gates) yöntemi kullanılmıştır.

Yu vd., (2019) rüzgâr enerjisi ve rüzgâr hızı tahminlerinde zaman serileri çalışmalarının aslında rüzgârın zamansal ve mekânsal değişimini yansıtamadığını ifade etmişlerdir. Bu nedenle zaman ve mekânsal özelliklere (STF) bağlı derin CNN yöntemini uygulamışlardır. Fu vd., (2019) rüzgâr enerjisi tahmini yapılırken rüzgâr hızı, rüzgâr yönü ve sıcaklık değişkenlerini kullanılmışlardır. Bu çalışmada yerel optimumdan kaçınmak için normla tavuk sürüsü algoritması yerine geliştirilmiş tavuk sürüsü algoritması kullanılmıştır. Demolli vd., (2019) farklı konumlardaki rüzgar verilerini kullanılarak beş farklı makine öğrenmesi algoritmasını (LASSO, kNN, XGBoost, RF, ve SVR) karşılatılırmışlardır. Çalışmada, model Niğde bölgesine ait veriler ile eğitilmiş ve dört farklı bölge için (Bozcaada, Çeşme, Mamak ve Silivri) rüzgâr hızı tahmin çalışması yapılıp bu bölgelerin rüzgâr enerjisi için elverişliliği belirlenmiştir.

Bu çalışmada Yalova ilinde yer alan istasyondan elde edilen 01.06.2015 ve 31.05.2020 tarihleri arasındaki veriler kullanılarak rüzgâr hızı tahmini yapılması amaçlanmaktadır. Rüzgâr hızı tahmininde TSA temelli 3 farklı model kullanılmış olup, hangi modellerin daha etkin olduğu sorusuna cevap aranmaya çalışılmıştır. Bununla beraber klasik zaman serisi

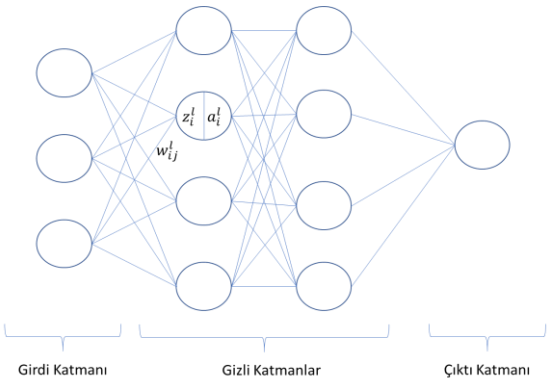
metodu ile kıyaslanarak TSA temelli modellerin performansı değerlendirilmiştir.

3. Metodoloji (Methodology)

Bu çalışmada meteorolojik veriler kullanılarak rüzgâr hızı tahmininin yapılması amaçlanmaktadır. Literatür araştırmasının da belirttiği üzere rüzgâr hızının tahmin edilmesinde yapay zekâ tekniklerinin geleneksel istatistiksel yöntemlere göre daha iyi sonuçlar vermektedir. Bu çalışmada uzun kısa süreli bellek (Long-Short Term Memory-LSTM) ve kapılı tekrarlayan hücre (Gated Recurrent Unit-GRU) temelli derin öğrenme algoritmaları kullanılmıştır.

3.1. Yapay sinir ağları (Artificial neural networks)

Yapay zekâ ve makine öğrenimi alt uygulaması olan derin öğrenme, yapay sinir ağları adı verilen beynin yapısı ve işlevinden esinlenen algoritmalarla ilgilidir. Veri işleme ve karar vermede kullanılabilecek kalıpları oluştururken bu algoritmalar kullanılır. Makine öğreniminin gerçekleşmesi için hiyerarşik yapıda yapay sinir ağları kullanılır. Geleneksel modellerin veriyi doğrusal bir şekilde analiz etmesinin aksine, yapay sinir ağları algoritmaları hiyerarşik yapıları sayesinde veriyi doğrusal olmayan bir yapı ile işleyebilirler. Yapay sinir ağları geleneksel olarak üç çeşit katmandan oluşurlar; girdi katmanı, gizli katman(lar) ve çıktı katmanı. Aşağıdaki figürde klasik bir yapay sinir ağları mimarisi gösterilmiştir.



Şekil 1. Yapay Sinir Ağları Mimarisi (Artificial Neural Networks Architecture)

Yapay sinir ağlarında l'ninci katmandaki i'ninci nöronun girdisi bir önceki katmandaki nöronların ağırlıklı etkisi ile aşağıdaki gibi hesaplanır.

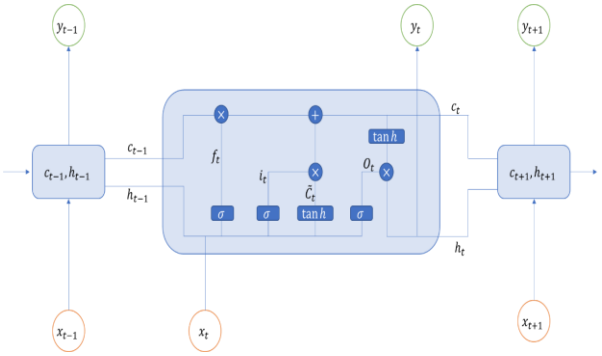
$$z_i^l = \sum_j \omega_{ij}^l a_j^{l-1} \tag{1}$$

Nöronlara gelen bilgi aktivasyon fonksiyonu sayesinde işlenerek diğer nöronlara aktarılır. Yapay sinir ağları literatüründe çeşitli fonksiyonları kullanılmaktadır. Bu çalışmada gizli katmanlarda RELU aktivasyonu fonksiyonu kullanılırken, çıktı katmanında lineer aktivasyon fonksiyonu kullanılmıştır.

$$relu(x) = \{0,\ x < 0\ x,\ x \geq 0 \tag{2}$$
$$linear(x) = x \tag{3}$$

3.1.1. Uzun kısa süreli bellek (Long-short term memory-LSTM)

Uzun Kısa Süreli Bellek (LSTM) mimarisi zaman serileri veya sıralı dizileri tahminleme için geliştirilmiş olan yapay sinir ağları mimarisidir. LSTM algoritmasında bir sonraki adımı tahminlemek için geçmiş durumların doğrudan veya dolaylı etkileri kullanılarak tahmin yapılır. LSTM mimarisi gradyan sönümlenmesini engellemek adına unutma kapılarına kullanarak işe yarayacak bilgileri kullanırken diğerlerini engeller. Tipik bir LSTM hücresi aşağıdaki gibidir.



Şekil 2. Uzun kısa süreli bellek mimarisi (LSTM architecture)

LSTM mimarisinde süreç unutma, yeni hafıza ve girdi, ve çıktı olmak üzere üç aşamadan oluşmaktadır. Birinci aşama girdi (X_t) ve önceki gizli katman (h_{t-1}) verilerini kullanarak unut kapısında (f_t) hangi bilgilerin kullanışlı olduğunun karar verildiği aşamadır. Bu aşamada aktivasyon fonksiyonu olarak Eşitlik 4'te verilen sigmoid fonksiyonu kullanılır.

$$f_t = \sigma\left(W_{f,x}X_t + W_{f,h}h_{t-1} + b_f\right) \tag{4}$$

İkinci aşamada, birinci aşamada olduğu gibi girdi (X_t) ve önceki gizli katman (h_{t-1}) verilerini kullanarak hangi bilgilerin uzun dönem hafızaya ekleneceğine karar verilir. Bu aşamada $tanh$ fonksiyonu kullanılarak önceki gizli katman ve girdi (X_t) verileri birleştirilir ve yeni bilgiyi oluşturacak aday bilgiler $(\tilde{C})$ elde edilir. Girdi kapısında i_t sigmoid fonksiyonu kullanılarak hangi bilgilerin saklanmasına karar verilir.

Son olarak Eşitlik 7'de verildiği gibi yeni bilgiler oluşturulur.

$$\tilde{C}_t = tanh(W_{c,x}X_t + W_{c,h}h_{t-1} + b_c) \quad (5)$$

$$i_t = \sigma(W_{i,x}X_t + W_{i,h}h_{t-1} + b_i) \quad (6)$$

$$C_t = C_{t-1}\, f_t + i_t\tilde{C}_t \quad (7)$$

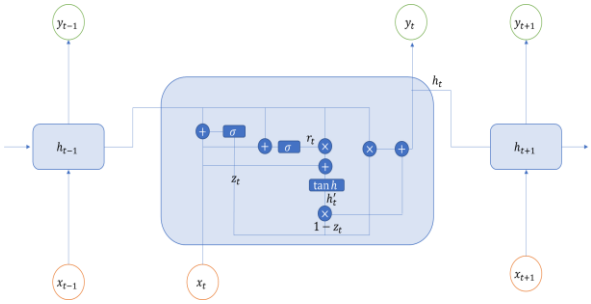
Son aşamada yeni bilgiler, önceki gizli katman ve girdi verileri kullanılarak çıktı kapısında (o_t) çıktı verileri (Eşitlik 8) ve yeni gizli katman (Eşitlik 9) elde edilir.

$$o_t = \sigma(W_{o,x}X_t + W_{o,h}h_{t-1} + b_o) \quad (8)$$

$$h_t = o_t tanh\,(C_t) \quad (9)$$

3.1.2. *Kapılı tekrarlayan hücre (Gated recurrent unit-GRU)*

Kapılı tekrarlayan hücre (GRU) mimarisi LSTM'de olduğu gibi zaman serileri veya sıralı dizileri tahminleme için geliştirilmiş olan yapay sinir ağları mimarisidir. GRU mimarisi gradyan sönümlenmesini engellemek adına LSTM mimarisine benzer bir şekilde unutma kapılarına kullanarak işe yarayacak bilgileri kullanırken diğerlerini engeller. Şekil 3 GRU hücresi mimarisi gösterilmiştir.



Şekil 3. Kapılı tekrarlayan hücre mimarisi (GTU architecture)

Güncelleme kapısında (z_t), girdi (X_t) ve önceki gizli katman (h_{t-1}) verilerini kullanarak geçmiş bilgilerin ne kadarını geleceğe aktarılacağına Eşitlik 10'da verilen sigmoid fonksiyonu ile karar verilir.

$$z_t = \sigma(W^{(z)}X_t + U^{(z)}h_{t-1}) \quad (10)$$

Daha sonra silme kapısında (r_t), geçmiş bilgilerin ne kadarının silineceğine karar verilir.

$$r_t = \sigma(W^{(r)}X_t + U^{(r)}h_{t-1}) \quad (11)$$

Eşitlik 12'de yeni aday bilgiler, silme kapısı kullanılarak elde edilir.

$$h'_t = tanh(W\,X_t + r_t U\,h_{t-1}) \quad (12)$$

Son aşamada yeni aday bilgiler arasından hangilerinin saklanmasına karar verilerek yeni gizli katman oluşturulur.

$$h_t = z_t h_{t-1} + (1 - z_t)h'_t) \quad (13)$$

3.2. *Performans ölçütleri (Evaluation metrics)*

Bu çalışmada önerilen TSA modellerinin rüzgâr hızı tahminleme doğruluklarını ölçmek için üç farklı ölçüt kullanılmıştır. Bu ölçütler 14-16 denklemlerinde verilmiştir. Hata terimi $e_j = y_j - \hat{y}_j$, y_j gözlem değeriy$_j$ iken $\hat{y}_j$ tahmin edilen değerdir.

$$Ortalama\ Mutlak\ Hata\ (MAE) = \frac{1}{n}\sum_{j=1}^{n}|e_j| \quad (14)$$

$$Ortalama\ Mutlak\ Yüzde\ Hata\ (MAPE) = \frac{100}{n}\sum_{j}^{n}\frac{|e_j|}{|y_j|} \quad (15)$$

$$Ortalama\ Kare\ Hata\ (MSE) = \frac{1}{n}\sum_{j=1}^{n}e_j^2 \quad (16)$$

4. Deneysel Çalışma (Experimental Study)

Bu çalışmada kullanılan modellerin uygulaması Python programlama dilinde yapılmıştır. Önerilen tekrarlayan sinir ağları yöntemlerinin kısa dönemli rüzgâr hızı tahminlemesi Yalova bölgesinde elde edilen veriler üzerinde uygulanmıştır.

Bu bölümde sırasıyla veri toplanması, veri ön işlemesi, modelin belirlenmesi, uygulanması, ve sonuçlar anlatılacaktır.

Teorik olarak bu veriler kullanılarak üretilebilecek rüzgâr enerjisi miktarları hesaplanabilmektedir. Bu nedenle çalışmada rüzgâr hızı tahminlemesi yapılmış ve istenildiği durumda üretilebilecek rüzgâr enerjisi değerleri elde edilebilmektedir. Akışkanlar mekaniği tanımına göre; rüzgâr türbininin güç göstergesi olan rüzgâr enerjisinin hesaplanması Eşitlik 17'de gösterilmiştir.

$$P = \frac{1}{2}C_p\rho A\vartheta^3 \quad (17)$$

C_p Türbin performans katsayı, P güç, ρ hava yoğunluğu, ϑ^3 türbin alanındaki rüzgar hızını, A türbin kanadı alanını ifade etmektedir (Che vd., 2016).

4.1. *Veri Toplanması (Data collection)*

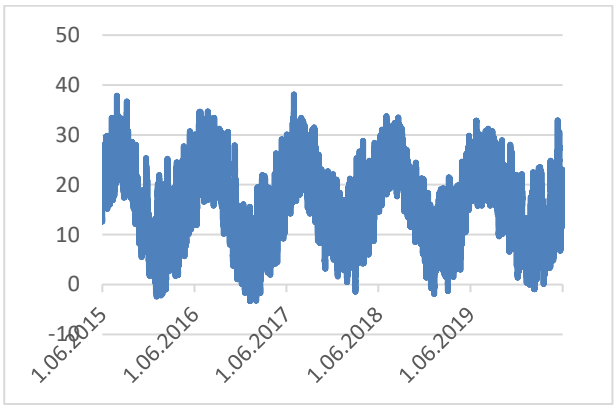
Rüzgar hızı, sıcaklık, nem, basınç ve gün özelliği gibi parametreler enerji üretimini etkileyen faktörlerdir.

Kullanılan veriler 01.06.2015 ve 31.05.2020 tarihleri arası saatlik verilerden oluşmaktadır. Veri kümesi

toplamda 43848 adet veriden oluşmaktadır, bu verilerin %65'i ağın eğitilmesi, %15'i doğrulanması ve %20'si de test için kullanılmıştır. Yapay sinir ağlarında, rüzgâr hızı, sıcaklık, nem, basınç, rüzgâr yönü, saat ve ay bilgileri bir saat sonraki rüzgâr hızını tahmini için girdi olarak kullanılmıştır.

4.1.1. *Sıcaklık verileri (Temperature data)*

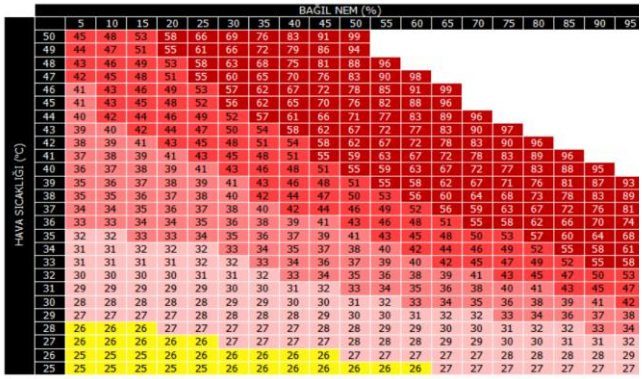
Rüzgar hızı sıcaklık değişimlerinden kaynaklandığı için rüzgar hızına direk etkisi vardır. Sıcaklık değerleri ℃ cinsinden ifade edilmektedir. Şekil 4'te gösterilen sıcaklık verilerine bakıldığında mevsimsellik görülmektedir.



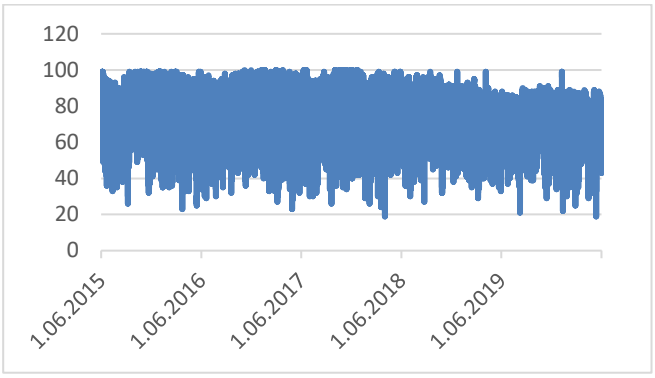
Şekil 4. Sıcaklık veri dağılımı (Temperature data distribution)

4.1.2. *Nem verileri (Humidity data)*

Nem miktarı, sıcaklığı etkilemekte olup dolaylı olarak rüzgâr hızına etki etmektedir. Sıcaklık ile nem arasındaki ilişkinin hissedilen sıcaklığa etkisi Şekil 5'te gösterilmiştir. Şekil 6'da ise zamana bağlı olarak bağıl nem grafiği gösterilmiştir.



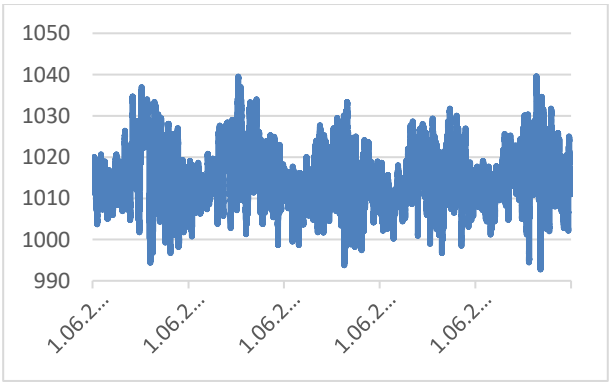
Şekil 5. Bağıl nem-sıcaklık ilişkisi (Relative humidity-temperature matrix)



Şekil 6. Saatlik bağıl nem dağılımı (Hourly relative humidity distribution)

4.1.3. *Basınç verileri (Pressure data)*

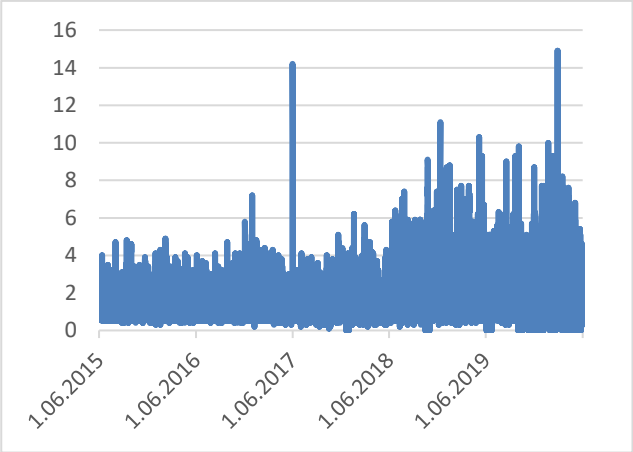
Rüzgâr hızını etkileyen bir diğer faktör ise basınçtır. Atmosferin basıncına göre rüzgâr hızı değişmektedir. Şekil 7'de basınç verileri hPa cinsinden gösterilmektedir.



Şekil 7. Saatlik aktüel basınç dağılımı (Hourly actual pressure distribution)

4.1.4. *Rüzgâr hızı verileri (Wind speed data)*

Rüzgâr hızı tahminlemesinin en önemli girdisi rüzgâr hızına ait geçmiş verilerdir. Yalova'nın 01.06.2015 ve 31.05.2020 tarihleri arasındaki saatlik rüzgar hızı verileri Şekil 8'de ifade edilmiştir.

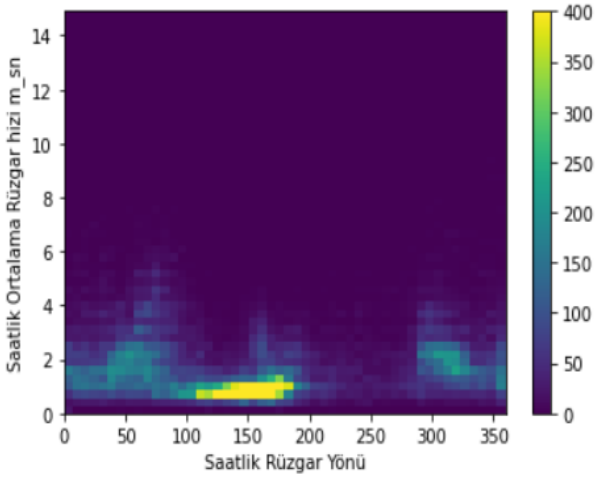


Şekil 8. Saatlik ortalama rüzgâr hızı dağılımı (Hourly average wind speed distribution)

4.1.5. *Rüzgâr yönü verileri (Wind direction data)*

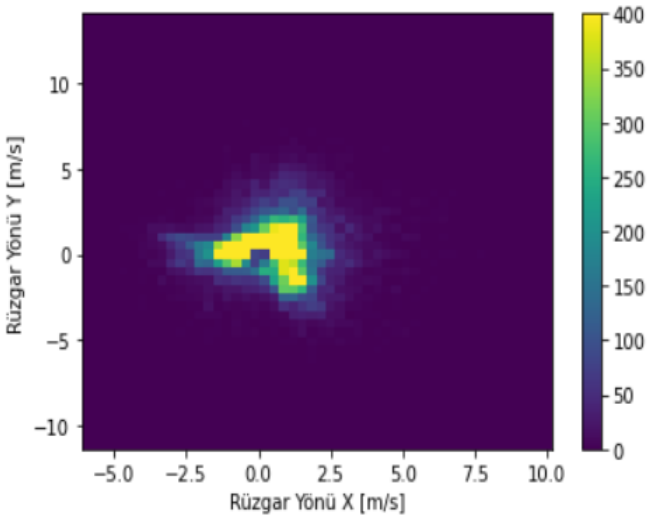
Rüzgâr yönü verisi açısal olarak verilmiştir. Açısal verileri direk olarak kullanmak bazı sorunlar oluşturmaktadır. Örnek olarak 360 ° ve 0 ° yön olarak birbirlerine yakın olmalarına rağmen sayısal olarak uzak gözükmektedir. Bu nedenle rüzgâr yönü bilgisi için bir önişlem yapılmıştır. Rüzgâr yönü bilgisi kullanılarak rüzgâr hızı X ve Y bileşenlerine ayrılmıştır.

Rüzgâr hızının rüzgâr yönüne göre dağılımı Şekil 9'da gösterilmiştir.



Şekil 9. Rüzgâr hızının rüzgâr yönüne göre dağılımı (Wind speed-wind direction distribution)

Rüzgar yönünü açısal olarak aldığımız vakit 0° ile 359° yön olarak birbirlerine çok yakın olmalarına rağmen sayısal olarak birbirlerine uzak gözükmektedirler. Bu yüzde rüzgar hızlarını X ve Y koordinatlarına ayrılmış olarak kullandık Şekil 10'da gösterildiği gibi.



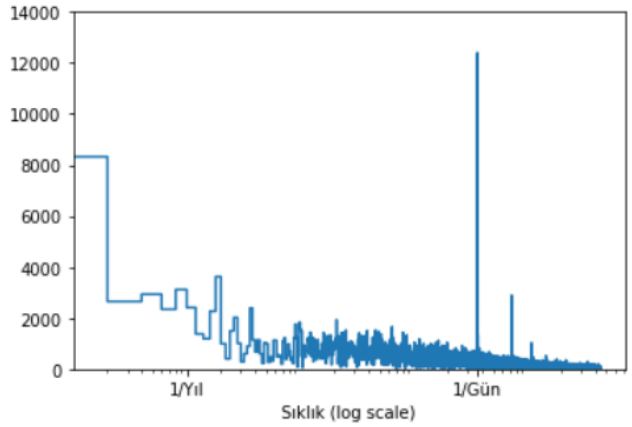
Şekil 10. Rüzgâr yönünün X ve Y koordinat dağılımı (X and Y coordinate distribution of wind direction)

4.1.6. *Zaman verileri (Time data)*

Rüzgâr verileri günlük ve yıllık mevsimsellik göstermektedir. Saat ve gün verilerini sinüs ve kosinüs değişimleri kullanarak Şekil 11'de gösterildiği gibi modele dahil ettik.

Şekil 11. Günlük saatin sinüs ve kosinüs dağılımları (Sine and cosine distribution of daily hour values)

Şekil 12'de rüzgâr hızının Fast Fourier Transfrom (FFT) analizi gösterilmiştir. FFT analizi göstermektedir ki rüzgâr hızında günlük bir periyotluk söz konusudur.



Şekil 12. Yıllık ve günlük sıklık dağılım (Annual and daily temperature distribution)

4.2. *TSA modelleri (RNN models)*

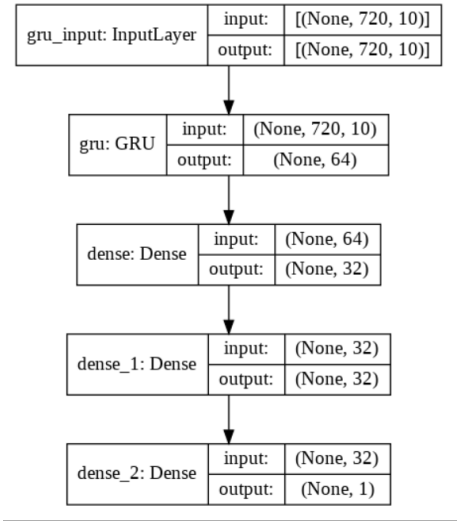
Bu çalışmada rüzgâr hızı tahmini için 3 model önerilmiş olup, modellerin mimari yapısı bu bölümde ifade edilecektir.

Meteorolojik veriler farklı birimlere ve aralıklara sahip olduklarından dolayı (örneğin sıcaklık verileri -4 ile +39 aralığında yer almakta iken basınç verileri 992 ile 1039 aralığında değerler almaktadır), bu verilerin öncelikle normalize edilmesi gerekmektedir. Değerler [0,1] arasında Eşitlik 18 deki gibi normalize edilmiştir.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{18}$$

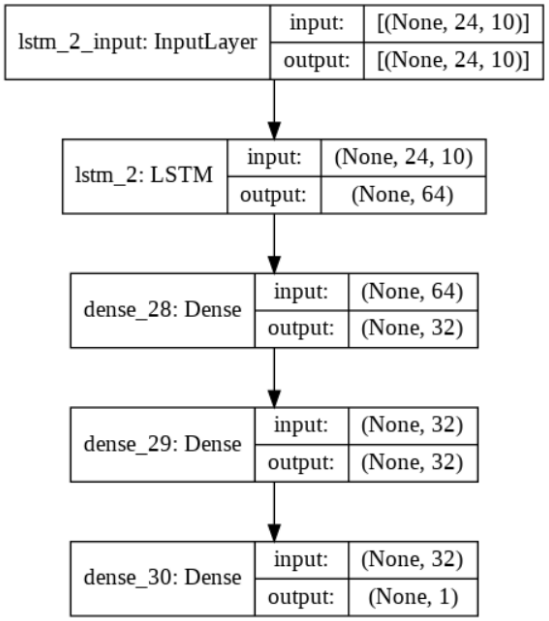
4.2.1. *TSA modellerinin oluşturulması ve eğitimi (RNN models creation and training)*

Bu çalışmada rüzgar hızı tahminlemesinde üç farklı TSA modeli kullanılmıştır. İlk modelde sıralı bir şekilde 1 GRU katmanı 3 tam bağlantılı katmandan oluşmaktadır. Gizli katmanlarda RELU aktivasyon fonksiyonu kullanılırken son katmanda doğrusal aktivasyon fonksiyonu kullanılmıştır. Şekil 13'te her bir katmanın hangi katman çeşidi olduğu kaç tane nöron sahip olduğu ve verinin akış yönü gösterilmiştir.



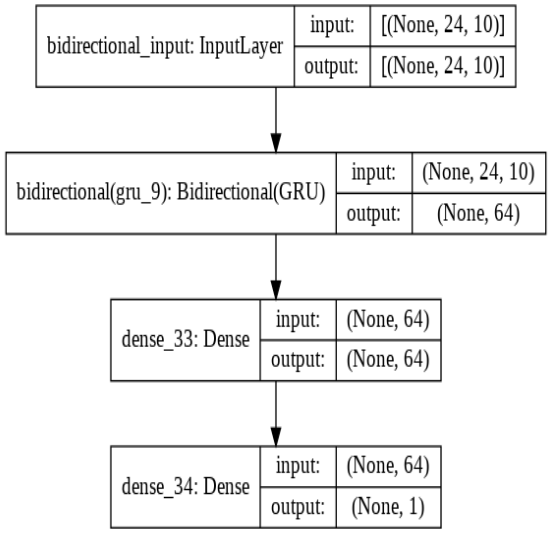
Şekil 13. Model 1 akış diyagramı (Model 1 flow diagram)

Kullanılan ikinci model, ilk modelden farklı olarak ilk katmanda GRU yerine LSTM katmanı tercih edilmiştir. Model 2 için akış diyagramı Şekil 14'te gösterilmiştir.



Şekil 24. Model 2 akış diyagramı (Model 2 flow diagram)

Kullanılan son TSA modelinde ise hem ileri doğru ilişkiyi hem de geriye doğru olan ilişkiyi kavrayabilmesi adına İki Yönlü LSTM katmanının 2 tam bağlantılı katman izlemiştir. Model 3 için akış Şekil 15'te gösterilmiştir.



Şekil 35. Model 3 akış diyagramı (Model 3 flow diagram)

Bu çalışmada kullanılan veri setinin %80'i eğitim, %20'si test olarak ayrılmıştır. Eğitim veri setinin de %10 u doğrulama veri seti olarak kullanılmıştır.

4.2.2. TSA modellerinin tahmin performansı (Performance of proposed TNN models)

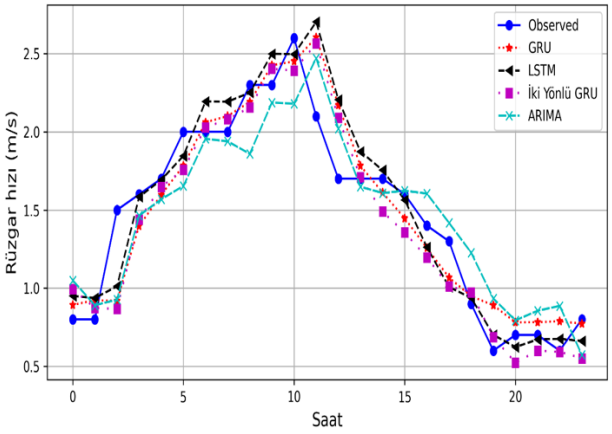
Kullanılan TSA modelleri ve geleneksek zaman serisi modeli (ARIMA) kullanılarak, 1 saat sonrası için rüzgar hızı tahmini yapılmaya çalışılmıştır. Model eğitilirken 24 saatlik veri kullanılmıştır. Tablo 1'de her bir model için tahmin doğruluğunu gösteren MSE, MAPE, MAE ve R^2 değerleri verilmiştir. Her bir kriter için LSTM (Model 2) modeli en iyi değerleri almıştır. Tablo 1'de gösterilen hata değerleri arasındaki farklar çok yüksek olmamakla beraber, yıllık rüzgâr enerjisi üretim miktarı (2021 yılı için 34250 GWh) ve üretici firmaların üretim tahmininde yaptıkları hataların ceza maliyetleri (Aksoy vd., 2013) göz önünde bulundurulduğunda yapılan iyileştirme büyük önem taşımaktadır.

Bu çalışmada zaman serileri için geliştirilmiş olan ve literatürde sıklıkla kullanılan geri beslemeli sinir ağları mimarilerinden LSTM ve GRU seçilmiştir. Bu ileri seviye metotlarla karşılaştırmak ve üstünlüğünü göstermek için klasik bir zaman serisi metodu olan ARIMA seçilmiştir. Derin öğrenme metotları için çeşitli katman sayısı ve nöron sayıları denenerek optimum parametreler bulunmaya çalışılmıştı. Model parametrelerini optimize etmek için eğitim kümesinin %10'luk kısmı doğrulama kümesi olarak kullanılmıştı. Bulunan optimum parametreler şekiller 13,14 ve 15'te gösterilmiştir. ARIMA metotlarından mevsimsel ARIMA (sARIMA) metodu seçilmiştir. Parametre optimizasyonu (grid arama) yaparak optimum ARIMA parametreleri (p=1, d = 1, q = 0) mevsimsel parametreler ise (P = 1, D= 0, Q = 0) olarak bulunmuştur. Mevsimsel periyot ise 24 saat (1 gün) olarak belirlenmiştir.

Tablo 1. Modellerin tahminleme performansları (Performance of proposed ANN models)

Model	R^2	MSE	MAPE	MAE
GRU	0.7530	0.1243	18.9448	0.2533
LSTM	**0.7586**	**0.1215**	**18.4373**	**0.2485**
İki Yönlü GRU	0.7579	0.1219	18.4987	0.2503
ARIMA	0.7435	0.1291	20.2623	0.2621

Bununla beraber her model için tahmin değerleri ve gerçekleşen rüzgâr hızı değerleri 24 saatlik grafik üzerinde Şekil 16'da gösterilmiştir.



Şekil 16. 24 saatlik gerçek değer ile model tahminlerinin kıyası. (Comparison of actual and forecasted wind data for 24 hours)

5. Sonuç ve Öneriler (Conclusion and Future Research)

Bu çalışmada kısa dönemli rüzgâr hızı tahmininde modern makine öğrenmesi algoritmalarının performanslarını karşılaştırmak adına 3 farklı TSA modeli sunulmuştur. Ayrıca TSA modelleri klasik zaman serisi metodu (ARIMA) ile karşılaştırılmıştır.

Oluşturulan modellerle yapılan tahmin sonuçları R^2, MSE, MAPE ve MAE değerleri hesaplanmıştır. Bu değerlere bakıldığında; her model ile yapılan tahminlere ait değerlendirme kriterleri oranları kabul edilebilir bir aralık içerisinde olduğu görülürken bazı modellerde diğer modellere kıyasla daha düşük değerler elde edilmiştir. Genel olarak en düşük hata oranına sahip olan modeller TSA modelleri olup bunların içinden de en her üç değerlendirme kriteri içinde LSTM modeli en iyi sonucu vermiştir.

Bu çalışmada elde edilen sonuçlar ile, TSA yöntemlerinin rüzgâr hızı tahmininde başarılı bir şekilde kullanılabileceğini, ve geleneksel zaman serisi yöntemlerine göre daha iyi sonuçlar verdiği sonucuna varılmıştır.

Önerilen TSA modellerini geliştirmek ve tahminleme performansını artırmak için farklı optimizasyon metotlarının kullanılması yeni bir çalışma konusu olabilir. Ağırlıkların daha doğru belirlenmesi ile daha doğru sonuçlar elde edilebilir. Bu kapsamda sıklıkla kullanılan tahmin yöntemlerinin birlikte kullanılmasıyla hibrit modeller oluşturulabilir. Bu şekilde her yöntemin avantajları kullanılarak daha doğru ağırlıklara ulaşılabilir ve tahminlerin doğruluğu artırılabilir. Bununla beraber, modelin eğitilmesinde farklı veri setlerinin kullanılması ile daha düşük hata oranları elde edilebilir.

Kaynaklar **(References)**

Aasim, S.N. vd., (2019) 'Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting', *Renewable Energy*, 136, pp. 758–768.

Aksoy vd., (2013) 'Rüzgâr gücü üretimi için tahmin ve teklif sistemi tasarımı', *Endüstri Mühendisli Dergisi*, 24(3), pp. 4–15.

Azad, H. B., Mekhilef, S., Ganapathy, V. G. (2014) 'Long-Term Wind Speed Forecasting and General Pattern Recognition Using Neural Networks', *IEEE Transactions on Sustainable Energy*, 5(2), pp. 546–553.

Barbosa de Alencar, D. vd., (2017) 'Different Models for Forecasting Wind Power Generation: Case Study', *Energies*

Brown, B. G., Katz, R. W., Murphy, A. H. (1984) 'Time Series Models to Simulate and Forecast Wind Speed and Wind Power', *Journal of Climate and Applied Meteorology*. American Meteorological Society, 23(8), pp. 1184–1195.

Cadenas, E. vd., (2016) 'Wind Speed Prediction Using a Univariate ARIMA Model and a Multivariate NARX Model', *Energies*

Che, Y. vd., (2016) 'A wind power forecasting system based on the weather research and forecasting model and Kalman filtering over a wind-farm in Japan', *Journal of Renewable and Sustainable Energy*, 8(1), p. 13302

Demolli, H. vd., (2019) 'Wind power forecasting based on daily wind speed data using machine learning algorithms', *Energy Conversion and Management*, 198, p. 111823.

Dokuz, A. S. vd., (2018) 'Year-ahead wind speed forecasting using a clustering-statistical hybrid method', in *CIEA'2018 International Conference on Innovative Engineering Applications*, pp. 971–975.

Duan, Jikai vd., (2021) 'Short-term wind speed forecasting using recurrent neural networks with error correction', *Energy*, 217, p. 119397.

Dumitru, C.-D., Gligor, A. (2017) 'Daily Average Wind Energy Forecasting Using Artificial Neural Networks', *Procedia Engineering*, 181, pp. 829–836.

Eldali, F. A. vd., (2016) 'Employing ARIMA models to improve wind power forecasts: A case study in ERCOT', in *2016 North American Power Symposium (NAPS)*, pp. 1–6.

Fu, C. vd., (2019) 'Short-Term Wind Power Prediction Based on Improved Chicken Algorithm Optimization Support Vector Machine', *Sustainability* .

Higashiyama, K., Fujimoto, Y., Hayashi, Y. (2017) 'Feature extraction of numerical weather prediction results toward reliable wind power prediction', in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6.

Hong, T. vd., (2020) 'Energy Forecasting: A Review and Outlook', *IEEE Open Access Journal of Power and Energy*, 7, pp. 376–388.

Kavasseri, R. G., Seetharaman, K. (2009) 'Day-ahead wind speed forecasting using f-ARIMA models', *Renewable Energy*, 34(5), pp. 1388–1393.

Lei, M. vd., (2009) 'A review on the forecasting of wind speed and generated power', *Renewable and Sustainable Energy Reviews*, 13(4), pp. 915–920.

Li, C. vd., (2018) 'Short-term wind power prediction based on data mining technology and improved support vector machine method: A case study in Northwest China', *Journal of Cleaner Production*, 205, pp. 909–922.

Madhiarasan, M. (2021) 'Long-term wind speed prediction using artificial neural network-based approaches', *AIMS Geosciences*. AIMS Press, 7(4), pp. 542–552.

Neshat, M. vd., (2021) 'A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm', *Energy Conversion and Management*, 236, p. 114002.

Ozkan, M. B., Karagoz, P. (2015) 'A Novel Wind Power Forecast Model: Statistical Hybrid Wind Power Forecast Technique (SHWIP)', *IEEE Transactions on Industrial Informatics*, 11(2), pp. 375–387.

Rajagopalan, S.,Santoso, S. (2009) 'Wind power forecasting and error analysis using the autoregressive moving average modeling', in *2009 IEEE Power & Energy Society General Meeting*, pp. 1–6.

Sfetsos, A. (2002) 'A novel approach for the forecasting of mean hourly wind speed time series', *Renewable Energy*, 27(2), pp. 163–174. doi:

Torres, J. L. vd., (2005) 'Forecast of hourly average wind speed with ARMA models in Navarre (Spain)', *Solar Energy*, 79(1), pp. 65–77.

Türkiye Rüzgar Enerjisi İstatistik Raporu (2019). Available at: https://tureb.com.tr//lib/uploads/4e77501b714739a9

.pdf.
Wang, X., Guo, P., Huang, X. (2011) 'A Review of Wind Power Forecasting Models', *Energy Procedia*, 12, pp. 770–778.
Yu, C. vd., (2018) 'A novel framework for wind speed prediction based on recurrent neural networks and support vector machine', *Energy Conversion and Management*, 178, pp. 137–145.
Yu, R. vd., (2019) 'Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space', *Applied Energy*, 238, pp. 249–257.
Zhang, Z. vd., (2019) 'Long Short-Term Memory Network based on Neighborhood Gates for processing complex causality in wind speed prediction', *Energy Conversion and Management*, 192, pp. 37–51.