T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI

GÖZETİMSİZ ÖZNİTELİK SEÇİM ALGORİTMALARININ KARŞILAŞTIRILMASI VE ENTROPİYE DAYALI YENİ BİR YÖNTEMİN ÖNERİLMESİ

SAMET DEMİREL

YÜKSEK LİSANS TEZİ

Jüri Üyeleri : **Dr. Öğr. Üyesi Fatih AYDIN (Tez Danışmanı)**
Prof. Dr. Ayhan İSTANBULLU
Dr. Öğr. Üyesi Emrah DÖNMEZ

BALIKESİR, NİSAN - 2024

ETİK BEYAN

Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak tarafımca hazırlanan "**Gözetimsiz Öznitelik Seçim Algoritmalarının Karşılaştırılması ve Entropiye Dayalı Yeni Bir Yöntemin Önerilmesi**" başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

Samet DEMİREL

ÖZET

GÖZETİMSİZ ÖZNİTELİK SEÇİM ALGORİTMALARININ KARŞILAŞTIRILMASI VE ENTROPİYE DAYALI YENİ BİR YÖNTEMİN ÖNERİLMESİ
YÜKSEK LİSANS TEZİ
SAMET DEMİREL
BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ FATİH AYDIN)

BALIKESİR, NİSAN - 2024

Özellik seçim işlemi, Makine Öğrenimi algoritmalarının çok boyutluluğun lanetinden (curse of dimensionality) etkilenmemesi için çok önemlidir. Özellik seçim algoritmaları bu sorunu çözmeye çalışmaktadır. Ancak, özellik seçim algoritmalarının bazı yetersizlikleri vardır: (i) Her bir makine öğrenme algoritmasının performansı seçilen özellikler üzerinde önemli ölçüde farklı olabilir. (ii) Sınıflandırıcıların performansında, alt kümedeki varyasyona bağlı olarak önemli dalgalanmalar da gözlemlenebilir. (iii) Seçilen özellikler büyük veri kümeleri üzerinde uzun zaman harcayabilmektedir.

Bu tezde, yukarıda bahsedilen sorunlarla başa çıkmak için, tek değişkenli ve filtre yaklaşımına dayanan, hızlı bir gözetimsiz özellik seçim algoritması önerilmektedir. Önerilen algoritma hem dağılımın kümülatif entropisini hem de dağılımın simetrisi ile hesaplanan Shannon entropisini her bir boyut için birlikte ele almaktadır. Son teknoloji algoritmalarla yapılan karşılaştırmalar sonucunda deneysel sonuçlar, önerilen yöntemin diğer yöntemlere kıyasla bu sorunlarla daha iyi başa çıkabildiğini göstermektedir.

ANAHTAR KELİMELER: Makine öğrenimi, gözetimsiz özellik seçimi, tek değişkenli filtre yaklaşımı, kümülatif entropi, Shannon entropisi

Bilim Kod / Kodları: 92431, 92432 Sayfa Sayısı : 63

ABSTRACT

A COMPARİSON OF UNSUPERVISED FEATURE SELECTION ALGORITHMS AND A NEW ENTROPY-BASED METHOD PROPOSAL
MSC THESIS
SAMET DEMİREL
BALIKESIR UNIVERSITY INSTITUTE OF SCIENCE
COMPUTER AND INFORMATION ENGINEERING

(SUPERVISOR: ASSIST. PROF. DR. FATİH AYDIN)

BALIKESİR, APRIL - 2024

Feature selection task is essential for Machine Learning algorithms not to be influenced by the curse of dimensionality. In this regard, feature selection methods try to address this trouble. However, feature selection methods have some deficiencies: (i) the performance of each machine learning method can be remarkably different on the selected features (ii) significant changes can also be followed in the performance of the classifiers by depending on differences in the subset of selected feature (iii) they spend a long time on huge data sets.

In this thesis, to cope with the aforementioned problems, we propose a fast unsupervised feature selection algorithm, which is based on a univariate and filter approach. The proposed method jointly regards both the cumulative entropy of the distribution and the Shannon entropy calculated by the symmetry of the distribution for each feature. As a result of comparisons with cutting-edge works, the experimental results demonstrate that the presented algorithm better overcomes these problems compared to other methods.

KEYWORDS: Machine learning, unsupervised feature selection, univariate-filter approach, cumulative entropy, Shannon entropy

Science Code / Codes: 92431, 92432 Page Number : 63

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

KISALTMALAR LİSTESİ

ACC : Accuracy
ADMM : An Adaptive Alternating Direction Method of Multipliers
CART : Classification and Regression Trees
DISR : Diversity-Induced Self-Representation
DUFS : Pairwise Dependence-based Unsupervised Feature Selection
EFS : Entropy-based Feature Selection (Önerilen Algoritma)
IUFS : Information-theoretic Unsupervised Feature Selection
KNN : k-Nearest Neighbors
LS : Laplacian score for unsupervised feature selection
MCFS : Multi-Cluster Feature Selection
NB : Naive Bayes
RF : Random Forest
RNE : Robust Neighborhood Embedding
RSR : Regularized Self-Representation
SPEC : the SPECtrum decomposition of the Laplacian matrix
SRCFS : unsupervised Feature Selection approach based on multi-Subspace Randomization and Collaboration
SVM : Support Vector Machine
UFS : Unsupervised Feature Selection
USFS : Unsupervised Soft-label Feature Selection

ÖNSÖZ

Lisansüstü eğitimimdeki danışmanlık sürecinde gösterdiği ilgi, sabır ve bilgi birikimi ile verdiği değerli geri bildirimler ve yönlendirmeler ışığında desteğini benden hiç esirgemeyen değerli danışmanım Dr. Öğr. Üyesi Fatih AYDIN'a en içten teşekkürlerimi sunmak isterim. Sizinle çalışmak, benim için büyük bir şeref ve ayrıcalıktı.

Bilgisayar ve Bilişim Mühendisliği Anabilim Dalının ilk mezun olan yüksek lisans öğrencisi olmanın gururunu yaşarken, bu süreçte bize her zaman sabırla destek olan ve tüm sorularımıza içtenlikle cevap veren Fen Bilimleri Enstitüsü Müdürü Prof. Dr. Dilek TÜRKER'e, Müdür Yardımcıları Doç. Dr. Sümeyye AYDOĞAN TÜRKOĞLU ve Doç. Dr. Alaaddin TOKTAŞ hocalarımıza, Enstitü Sekreteri Meltem YAMAN BOZKURT'a ve diğer Enstitü personellerine teşekkürlerimi bildirmek isterim.

Ayrıca çalışmalarım boyunca her daim yanımda olan ve beni destekleyen sevgili eşim Hanife DEMİREL ile çocuklarım Berat Oğuz DEMİREL ve Ayşe Asya DEMİREL'e teşekkürü borç bilirim. İyi ki varsınız.

Balıkesir, 2024 **Samet DEMİREL**

1. GİRİŞ

Günümüzde dijitalleşmenin hızlı bir şekilde artmasının birçok alanda geniş kapsamlı etkisi olmuştur. İnternet kullanımının yaygınlaşması, sosyal medya, elektronik ticaret, mobil uygulamalar, otomasyon sistemleri, cihazların ve makinelerin internete bağlanabilmesi her alanda çok büyük miktarda verinin oluşmasına neden olmaktadır [1]. Bu verilerin analiz edilerek verilerden anlamlı sonuçlar çıkartmak büyük önem arz etmektedir. İşletmelerin ve kuruluşların müşteri tercihlerini ve davranışlarını anlaması, pazarlama çalışmalarına yön vermesi, stratejik kararlar alma aşamasında destek sağlaması, yönetimsel süreçlerin iyileştirilmesi, bilimsel keşifler, sahtekarlık tespiti gibi birçok alanda bu veriler kullanılmaktadır.

Büyük veri kümeleri yapay zekâ teknolojileri kullanılarak hızlı bir şekilde değerlendirilebilir. Bunun için yapay zekanın bir alt dalı olan makine öğrenimi algoritmaları sıklıkla kullanılmaktadır. Makine öğrenimi yöntemleri geleneksel programlamadan farklı olarak önceden programlanmış adımlar yerine algoritmalar vasıtasıyla veriler arasındaki ilişkiyi ve deseni öğrenen bir model oluşturarak bu model sayesinde yeni veriler üzerinde tahminlerde bulunabilmektedir [2]. Ancak makine öğrenimi algoritmaları ile yüksek boyutlu veri kümeleriyle çalışırken problemlerle karşılaşılabilmektedir. Çok fazla özellik barından veri setleri öğrenme algoritmalarının karmaşık modeller oluşturmasına ve modelin verilere aşırı uyum sağlamasına neden olabilmektedir. Böyle durumlarda modelin öğrenme sürecinde yüksek başarı sağladığı görülse de test sürecinde performansı düşebilmektedir. Bunun yanında veri setinde bulunan tüm özelliklerin kullanılması hesaplama maliyetini ve bellek ihtiyacını önemli ölçüde artırabilmektedir [3]. Ayrıca veri kümelerindeki tüm özellikler makine öğrenimi algoritmalarını eğitmek için gerekli olmayabilir. Bazı özelliklerin tahmin sonucu ile ilgisi olmazken, bazı özellikler ise sonucu olumsuz yönde etkileyebilmektedir. Bu özellikleri belirleyerek daha alakalı özellikleri tespit etmek makine öğrenimi algoritmasının yükünü hafifletirken performansını da artıracaktır [4]. Çok fazla özellik bulunan veri setlerinin görselleştirilmesi de zordur. Yüksek boyutlu veri setlerinde makine öğrenimi algoritmalarının karşılaşabildiği bu olumsuz durumlar genel anlamda çok boyutluluğun laneti (curse of dimensionality) olarak bilinir [5].

Bu problemlerden kurtulmak ve makine öğrenimi algoritmalarını daha verimli bir şekilde kullanabilmek için verilere ön işleme teknikleri uygulanmaktadır. Yani veriyi makine

öğreniminin kullanımına daha uygun hale getirme işlemleri yapılır. En sık kullanılan veri önişleme tekniklerinden biri de boyut azaltma yöntemleridir. Boyut azaltma orijinal veri setinin taşıdığı bilgiyi en az kayıpla temsil edecek özellikleri belirleme görevidir. Boyut azaltma özellik seçimi ve özellik çıkarımı şeklinde iki türlü yapılabilir. Özellik seçiminde veri setindeki en bilgilendirici özellikler bulunmaya çalışılır. Böylece bir özellik alt kümesi elde edilir. Özellik çıkarımında ise veri setindeki tüm özellikler cebirsel işlemler kullanılarak daha az sayıda daha çok bilgi içeren temsili özelliklere dönüştürülür [5]. Özellik seçiminde orijinal veri setindeki özellikler üzerinde herhangi bir işlem yapılmadan doğrudan özellikler seçilir. Yani özelliklerin değerleri özellik çıkarımındaki gibi değişime uğramaz. Bundan dolayı özellik seçiminde özellik çıkarımına göre daha anlaşılabilir ve yorumlanabilir özellikler üretilir [6].

Özellik seçim algoritmaları, makine öğrenimi modelinin hızlı bir şekilde yapılandırılması ve performansının artırılması için destekleyici bir unsur olmaktadır. Özellik (öznitelik) seçimi, tüm orijinal özelliklerin kullanılmasına gerek kalmadan öğrenme modelinin performansının korunmasını veya bazı veri kümelerinde performansın artırılmasını sağlayan özellikleri belirleme görevidir [7]. Özellik seçimi, sınıflandırma, regresyon ve kümeleme gibi öğrenme görevlerinde oldukça faydalı bir görev üstlenmektedir. Çünkü depolama ve hesaplama gereksinimlerini azaltmanın yanı sıra çok boyutluluğun lanetini de ortadan kaldırmayı sağlamaktadır [8]. Ayrıca daha iyi genelleme yeteneğine sahip modellerin oluşturulmasına da katkıda bulunmaktadır [9].

Bu tez çalışmasında makine öğrenimi algoritmalarının yüksek boyutlu veri kümelerinde karşılaştığı problemleri azaltmak için kümülatif entropi [10] ve shannon entropisini [11] birlikte kullanarak özellik seçimi gerçekleştiren bir algoritma tanıtılmıştır.

Bu çerçevede yazılmış olan tezin ilerleyen bölümleri şu şekilde düzenlenmiştir: İkinci bölümde makine öğrenimi, özellik seçimi ve literatürde sıkça bahsedilen çözüm yaklaşımları ile ilgili genel bilgiler verilmiştir. Üçüncü bölümde önerilen algoritma anlatılmıştır. Dördüncü bölümde algoritmanın performansını ölçmek için yapılan deneysel ortam hakkında detaylı bilgiler verilmiştir. Beşinci bölümde çalışmanın bulguları detaylı bir biçimde paylaşılmıştır. Son olarak altıncı bölümde ise yapılan çalışmanın değerlendirmesi ve öneriler açıklanmıştır.

2. BOYUT AZALTMA

Makine öğrenimi algoritmalarıyla çalışırken yüksek boyutlu veri setleri model performansını olumsuz yönde etkileyebilir. Bunun önüne geçmek için veri setindeki ilgisiz ve tahmin sonucuna etkisi olmayan ya da çok daha az etkisi olan özelliklerin belirlenerek elenmesi işlemi yapılır. Bu işleme boyut azaltma denir. Bu bölümde makine öğrenimi, boyut azaltma ve literatürde sıklıkla kullanılan özellik seçim algoritmaları hakkında genel bilgiler verilmiştir.

2.1 Yapay Zekâ

Günümüzde yapay zekâ, birçok alanda önemli bir etkiye sahip olmaktadır. İnsan zekasının bilgi işleme alanında bilgisayar ve robot sistemlerine modellenmesi yapay zekâ disiplininin temelini oluşturmaktadır. Bu alandaki çalışmalar bilgisayarların karmaşık problemleri daha hızlı çözebilmesini, öğrenme becerilerini geliştirebilmesini ve kendi kendine karar verebilmesini sağlamayı amaçlamaktadır. Diğer bir deyişle, yapay zekâ ile bilgisayar sistemlerinin insan gibi düşünebilme ve muhakeme edebilme yeteneği kazanması amaçlanır. Böylece, bilgisayarların ve makinelerin daha akıllı, daha öğrenme kabiliyetine sahip ve daha özerk hale gelmesi sağlanabilmektedir. Yapay zekâ hem akademik dünyada hem de endüstride büyük ilgi görmektedir. Birçok teknoloji şirketi, yapay zekâ ve makine öğrenimi üzerine Ar-Ge çalışmaları yapmakta ve yapay zekâ temelli ürün ve hizmetler geliştirmektedir [12].

Bilgisayar sistemleri, büyük verilerle rahatça analizler yapılabilmemize ve analitik hesaplamaları hızlıca gerçekleştirmemize olanak vermektedir. Bunu gerçekleştirmek için yapay zekanın alt dalları olan veri madenciliği ve makine öğrenimi yöntemleri sıklıkla kullanılmaktadır [13].

2.2 Makine Öğrenimi

Klasik programlamada bir problemin çözümü için bilgisayarlara belirli talimatlar dizisini tanımlayarak işlemler yaptırılır. Bu talimatlar bütünü algoritma olarak tanımlanır ve belirli bir görevi gerçekleştirmek üzere tasarlanırlar. Ancak, günümüzde karmaşık ve büyük veri setleriyle çalışırken klasik algoritmalarda olduğu gibi her aşamanın programlanması ve her olası senaryoya göre işlemlerin belirlenmesi pek mümkün olmamaktadır. İşte bu noktada yapay zekanın bir alt dalı olan makine öğrenimi devreye girmektedir. Makine öğrenimi

algoritmaları belirli bir görevi gerçekleştirmek için programlanmak yerine veri setlerinden örüntüler çıkararak ve deneyimlerden öğrenerek çalışmaktadır. Bu algoritmalar kullanılarak veri setlerindeki örüntüler tespit edilebilir, veriler arasındaki ilişkiler belirlenebilir ve gelecekteki olaylar hakkında tahminlerde bulunulabilir. Makine öğrenimi algoritmalarının kullanılmasının temel amacı, algoritmalara veriye dayalı deneyimlerden öğrenme yeteneği kazandırarak birçok alanda insanlardan daha etkili sonuçlar elde edebilmeyi sağlamaktır [14].

Makine öğreniminin temel süreci eğitim verileriyle başlar. Eğitim verileri genellikle girdi-çıktı ilişkilerini temsil eder ve algoritmanın öğrenmesi için kullanılır. Eğitim verileri öğrenme algoritmasına gönderilir. Öğrenme algoritması ile verilerden çıkarımlara dayanan yeni kurallar dizisi oluşturularak bir makine öğrenimi modeli oluşturulur. Makine öğrenimi modeli eğitim verilerindeki örüntüleri ve ilişkileri tanımlamaktadır. Böylece gelecekte algoritmaya verilen giriş verileri analiz edilerek çıkarımlar yapılabilir.

Modeller temel olarak sınıflandırma, regresyon ve kümeleme gibi öğrenme görevlerinde kullanılabilmektedir [8]. Sınıflandırma, etiketli veriler ile bir öğrenme modeli oluşturularak bu modele daha sonra verilecek etiketsiz verilerin sınıflarını tahmin etme problemidir. Sınıflandırma gözetimli bir öğrenme görevidir. Sınıflandırmada kategorik bir etiketleme gerçekleşir. En bilindik sınıflandırma algoritmaları Rastgele Orman, Karar Ağaçları, Lojistik Regresyon ve Destek Vektör Makineleridir [14]. Regresyon, sınıflandırma problemine benzer bir görevdir. Etiketli verilerden oluşturulan bir model ile öğrenme gerçekleştirilip yeni gelen veriye bir tahminleme yapılır. Fakat regresyonda tahmin sonucu nümeriktir. Hava durumu ve piyasa eğilimleri gibi sürekli değişkenleri tahmin etmek için kullanılır [14]. Kümeleme görevinde ise etiketsiz verilerin birbiriyle olan ilişkilerine göre model oluşturulur. Kümeleme de gözetimsiz bir öğrenme yaklaşımı benimsenir. Öğrenme algoritmaları farklı eğitim verileri kullanılarak farklı modeller oluşturmak için de kullanılabilmektedir.

Günümüzde teknolojinin ve internetin etkisiyle veri boyutluluğu ve çeşidi önemli ölçüde artmıştır. Bu nedenle verileri analiz etmek ve anlamlı sonuçlar çıkartmak için kullanılan öğrenme algoritmaları yüksek boyutlu veri kümelerine maruz kalmaktadır. Makine öğrenimi algoritmalarının performansı, yüksek boyutlu, alakasız ve gereksiz veriler barındıran veri kümelerinde düşmektedir. Bu durum modelin eğitim süresini arttırmakta ve tahmin

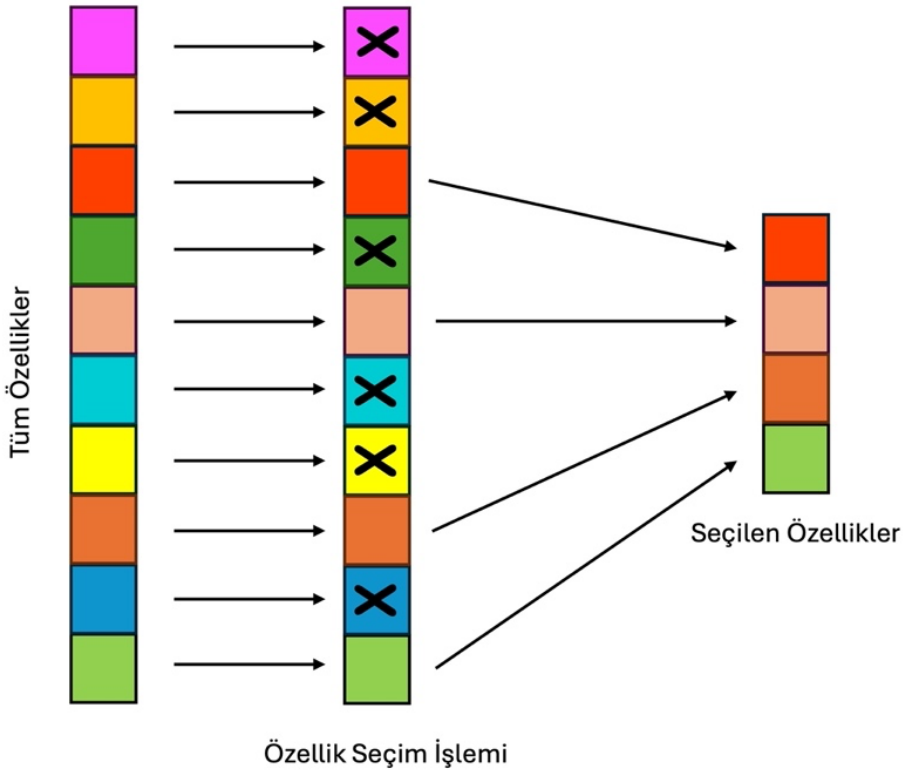
başarısını düşürmektedir. Bu gibi durumlardan etkilenmemek, model performansını yükseltmek ve gereksiz verileri elemek için boyut azaltma teknikleri kullanılmaktadır [15].

Boyut azaltma teknikleri veri setlerindeki ilgisiz ve gereksiz verileri tespit ederek ya da orijinal özellikleri birleştirerek daha fazla ayırt etme etkisine sahip özelliklerden oluşan bir özellik alt kümesi oluşturmayı sağlamaktadır. Boyut azaltma teknikleri özellik çıkarımı ve özellik seçimi şeklinde gruplandırılabilir.

Özellik seçiminde, orijinal veri setini filtreleyerek sadece ilgili özellikleri içeren azaltılmış bir veri seti oluşturulur. Özellik çıkarımında ise orijinal özellikler bir ağırlık matrisi ile birleştirilip daha küçük bir özellik seti oluşturularak boyutluluğun azaltılması sağlanır. Birleştirme sürecinde ilgisiz ve gereksiz özelliklere genellikle sıfır veya çok küçük katsayılar atanarak yeni oluşturulan özellikler üzerinde daha az etkiye sahip olması sağlanır. Özellik seçimi ile özellik çıkarımı arasındaki temel fark özellik seçimiyle elde edilen veri setinin orijinal özellikleri içermesiyken özellik çıkarımı ile oluşturulan veri setinin tamamen yeni oluşturulmuş daha az sayıda özelliği içermesidir [16].

2.3 Özellik Seçimi

Özellik seçimi, nitelik seçimi ya da değişken seçimi olarak da bilinir. Veri analizinde veriyi hazırlamak için kullanılan ön işlemlerden biridir. Özellik seçimi tüm orijinal özelliklerin kullanılmasına gerek kalmadan, orijinal veriyi en iyi temsil edecek özellikleri belirleme görevine denir [17]. Sınıflandırma, regresyon veya kümeleme gibi öğrenme görevlerinde büyük fayda sağlamaktadır. Çok büyük boyutlu veri setlerinde tüm özellikler göz önüne alındığında bunlardan birçoğu gereksiz veya ilgisiz olabilmektedir. Bunların makine öğrenimi performansını olumsuz etkilemesini önlemek için en ilgili özelliklerin belirlenmesi işlemi yapılır. Böylece orijinal veri setini en iyi temsil edecek özellikler alt kümesi seçilir. İlgili özellikler çıktı üzerinde etkisi olan ve diğer özellikler tarafından sağlanamayan bilgiyi ifade eder. Şekil 2.1'de özellik seçim işleminin gerçekleştirilme süreci şematik olarak gösterilmektedir.



Şekil 2.1: Özellik seçim işleminin şematik gösterimi.

Özellik seçimi işleminin avantajlarını aşağıdaki şekilde listelemek mümkündür [17].

- Veri kümesinin özellik boyutu azaltılırken öğrenme algoritmasının hızı artar.
- İlgisiz ve gürültülü özellikler belirlenerek veri seti temizlenir.
- Veri seti daha basit hale getirilir. Bu sayede veri seti daha anlaşılır hale gelirken veri setini görselleştirmek mümkün olur.
- Ortaya çıkan makine öğrenimi modelinin tahmin başarısı artar.

Özellik seçimi veri madenciliği ve makine öğreniminde sıklıkla kullanılmaktadır.

2.4 Özellik Seçim Yöntemleri

Özellik seçim yöntemleri farklı bakış açılarına göre iki kategoride sınıflandırılabilir. Özellik seçim yöntemlerini veri setlerindeki sınıf bilgilerinin kullanımı açısından ve uyguladıkları teknikler açısından sınıflandırmak mümkündür [18]. Şekil 2.2'de özellik seçim yöntemlerinin kategorilere ayrılması gösterilmektedir.

Şekil 2.2: Özellik seçim yöntemlerinin sınıflandırılması [18].

Literatürde özellik seçim algoritmalarının geliştirilmesinde kullanılan çeşitli teknikler mevcuttur. Bu teknikler: uyarlanabilir graf tabanlı yaklaşım (adaptive graph-based approach) [19], uyarlanabilir benzerlik öğrenimi (adaptive similarity learning) [20], otomatik kodlayıcı (autoencoder) [21], biyo-ilham yaklaşımı (bio-inspired approach) [22], kümeleme (clustering) [23], diferansiyel evrim (differential evolution) [24], Dirichlet süreci (Dirichlet process) [25], ayırt edici diskriminant analiz (discriminative discriminant analysis) [26], aşırı öğrenme makinesi (extreme learning machine) [27], graf gösterimi (graph representation) [28], yerçekimi arama algoritması (gravitational search algorithm) [29], gizli Markov modeli (hidden Markov model) [30], Hilbert-Schmidt bağımsızlık kriteri (Hilbert-Schmidt independence criterion) [31], tamsayı programlama (integer programming) [32], Kolmogorov-Smirnov testi (Kolmogorov-Smirnov test) [33], k-en yakın komşular (k-nearest neighbors) [34], Laplace puanı (Laplace score) [35], gizli gösterim (latent representation) [36], yerel duyarlı ikili kavram öğrenme (local sensitive dual concept learning) [37], yerel yapı öğrenme (local structure learning) [38], yerellik korumalı projeksiyon (locality preserving projection) [39], manifold öğrenme (manifold learning) [40], matris çarpanlara ayırma (matrix factorization) [41], maksimal bilgi sıkıştırma indeksi (maximal information compression index) [42], metasezgisel algoritmalar (metaheuristic algorithms) [43], metrik öğrenme (metric learning) [44], karşılıklı bilgi (mutual information) [45], parametrik olmayan bayes karışım modeli (nonparametric bayesian mixture model) [46], parçacık sürü optimizasyonu (particle swarm optimization) [47], temel bileşen analizi (principal component analysis) [48], regresyona dayalı yaklaşım (regression-based approach) [49], kendi kendini temsille öğrenme (self-representation learning) [50], seyrek öğrenme (sparse learning) [51], spektral öğrenme (spectral learning) [52], istatistiksel

öğrenme (statistical learning) [53], altuzay öğrenme (subspace learning) [54] ve simetrik belirsizlik (symmetrical uncertainty) [55] şeklinde sayılabilir.

2.4.1 Sınıf Bilgilerinin Kullanımına Göre Özellik Seçim Algoritmaları

Özellik seçim algoritmaları veri setlerinde bulunan sınıf bilgilerinin kullanımı açısından gözetimli, yarı gözetimli ve gözetimsiz olmak üzere üçe ayrılır. Sınıf bilgisi veri setindeki örneklerin hangi kategoriye veya sınıfa ait olduğunu belirten etiketlerdir. Bir hastalık teşhisi probleminde veri setindeki her hasta için bulunan özellikler (yaş, cinsiyet, kan basıncı, vb.) ile bu hastaların "hasta" ya da "sağlıklı" olduğu bilgisinin olması durumu etiket bilgisine örnek olarak gösterilebilir.

2.4.1.1 Gözetimli Özellik Seçimi (Supervised Feature Selection)

Gözetimli özellik seçiminde verilerin sınıf etiketlerinin olmasına ihtiyaç duyulur. Gözetimli bir özellik seçim algoritmasında özelliklerin veri setindeki etiket bilgisi ile olan korelasyonu veya doğru modeller oluşturmadaki faydaları değerlendirerek özelliğin uygunluğu belirlenmektedir [56]. Veri setindeki her örneğe atanan etiketler bir kategori, sıralı bir değer ya da sayısal bir değer olabilmektedir [7].

2.4.1.2 Yarı Gözetimli Özellik Seçimi (Semi-Supervised Feature Selection)

Gözetimli özellik seçimi yöntemlerinde büyük miktarda etiketli veriye ihtiyaç duyulurken gözetimsiz özellik seçimi yöntemlerinde ise etiket bilgisine ihtiyaç duyulmaz ve sınıf etiketleri göz ardı edilir. Bu durum ayırt edici özellikleri tanımlamada bazen eksikliklere yol açabilir. Aynı zamanda gerçek dünyada az miktarda etiketli ve büyük boyutlu veri daha yaygındır [6]. Tüm verilerin etiketlenmesi ise maliyetli bir işlemdir. Bu etkiyi azaltmak ve hem gözetimli hem de gözetimsiz özellik seçiminin faydalarından yararlanmak için yarı gözetimli özellik seçim yöntemleri kullanılmaktadır. Yarı gözetimli özellik seçiminde yalnızca bazı örneklerin etiketli olması yeterli olabilmektedir. Buradaki amaç gözetimsiz özellik seçiminin performansını iyileştirmek için ek bilgi olarak biraz da etiketli verinin kullanılmasıdır [57].

2.4.1.3 Gözetimsiz Özellik Seçimi (Unsupervised Feature Selection)

Gözetimsiz özellik seçiminde verilerin etiketli olmasına ihtiyaç duyulmadan özellik seçim işlemi gerçekleştirilir. Gözetimsiz özellik seçiminde verilerin kendilerine odaklanılır. Gözetimsiz özellik seçim algoritmalarının üç önemli üstünlüğü vardır: (i) sınıf bilgileri

olmadığından tarafsızdırlar, (ii) ön bilgi mevcut olmadığında bile verileri işleyebilirler ve (iii) gözetimli algoritmaların aksine aşırı uyumu azaltabilirler [7].

Son yıllarda yüzlerce gözetimsiz özellik seçim algoritması tanıtılmıştır. Bu gözetimsiz özellik seçim algoritmaları ile büyük veri, heterojen nitelikler, yüksek boyutlu veri kümeleri, görüntü işleme, veri kümeleme, kategorik veri kümeleri, kural çıkarma, metin madenciliği ve biyobelirteç keşfi gibi alt alanlardaki birçok sorun ele alınmaktadır [58].

2.4.2 Uyguladıkları Tekniklere Göre Özellik Seçim Algoritmaların Sınıflandırılması

Özellik seçim algoritmaları, özellikleri seçim stratejisine göre filtre, sarmal, hibrit ve gömülü olmak üzere dört temel yaklaşıma ayrılır [59].

2.4.2.1 Filtre (Filter) Yaklaşımı

Filtre yaklaşımında öğrenme algoritmasından bağımsız olarak, verilerin içsel özelliklerini değerlendirmek için, istatistiksel yöntemler kullanılmaktadır. Bu yöntemlerde veri setindeki her bir özelliğin hedef değişkenle veya diğer özelliklerle olan ilişkisi ölçülmektedir. Bu ölçümler genellikle istatistiksel testler veya bilgi kazancı gibi ölçümler kullanılarak yapılır [60].

Filtre yöntemlerinde genellikle her özellik kullanılan yaklaşımlara göre puanlanır. Daha sonra özellikler bu puanların anlamına göre azalan veya artan şekilde sıralanır. Buna göre ilgisiz (irrelevant) veya fazlalık (redundant) olanlar elenerek bir özellik alt kümesi çıkarılır [60].

Özellik değerlendirmesi tek değişkenli veya çok değişkenli olabilmektedir. Tek değişkenli özellik seçiminde her özellik tek başına ele alınırken, çok değişkenli özellik seçiminde özellikler birlikte değerlendirir. Bu nedenle çok değişkenli özellik seçiminde doğal olarak fazlalık özellikler de değerlendirmeye alınabilmektedir [6]. Bu teknikte kullanılan bazı yaygın istatistiksel ölçümler bilgi kazancı (information gain), Pearson korelasyonu, Ki kare, karşılıklı bilgi (mutual information) ve simetrik belirsizliktir (symmetric uncertainty) [60].

Filtreleme yaklaşımları özellik seçimi için genellikle hızlıdır ve hesaplama maliyeti düşüktür. Çünkü eğitim öncesinde öğrenme algoritmasından bağımsız olarak gerçekleştirilir.

Bu nedenle filtre yaklaşımları büyük boyutlu veri setlerinde birçok özellikle çalışırken tercih edilen yöntemlerdendir.

2.4.2.2 Sarmal (Wrapper) Yöntemler

Sarmal yaklaşımda seçilen bir makine öğrenimi algoritması ile en iyi tahmin yapmada etkili olan özellikler bulunarak özellik seçimi gerçekleştirilmektedir [6]. Bu nedenle, filtre yaklaşımından daha iyi sonuçlar verebilmesine karşın bu yöntemler daha yavaştır ve hesaplama maliyetleri yüksektir. Sarmal yöntemlerde özellik seçimi iki şekilde yapılabilmektedir. Bunlar ileri yönlü arama ve geri yönlü arama şeklinde gerçekleştirilmektedir. İleri yönlü aramada boş bir özellik alt kümesi ile özellik seçimine başlanır. Her aşamada en iyi özellikler bulundukça, bir durdurma kriteri sağlanana kadar bu kümeye eklenerek devam edilir. Geri yönlü arama da tüm özellikler ile alt küme seçimine başlanır. Her aşamada en kötü özellikler alt kümeden çıkarılarak bir durdurma kriteri sağlanana kadar devam edilir [61].

Genel olarak sarmal bir özellik seçimi modelinde işlemler aşağıdaki adımlarla gerçekleştirilmektedir [62]:

1. Özelliklerin bir alt kümesinin aranması,
2. Seçilen özellik alt kümesinin önceden belirlenen sınıflandırıcı performansına göre değerlendirilmesi,
3. İstenilen kaliteye ulaşılana kadar 1. ve 2. adımların tekrarlanması.

2.4.2.3 Hibrit (Hybrid) Yöntemler

Hibrit yaklaşımda, filtre ve sarmal yaklaşımların avantajları kullanılarak özellik seçim işlemi gerçekleştirilir. Bu yaklaşım türü filtre ve sarmal yaklaşımın bir varyantı olarak ortaya çıkmıştır. Bu yaklaşımda tasarım şemasına göre çoklu özellik seçiciler, tümevarım algoritmaları ve farklı alt kümeler kullanılabilmektedir [60].

2.4.2.4 Gömülü (Embedded) Yöntemler

Diğer yöntemlerdeki bazı eksikliklerden (filtre yaklaşımında sınıflandırıcıdan bağımsız çalışma, sarmal yaklaşımdaki maliyet) dolayı, filtre ve sarmal yöntemler arasındaki boşluğu doldurmak amacıyla, gömülü yöntemler önerilmiştir. İlk olarak filtre yönteminde olduğu gibi belirli bir önem derecesine sahip çeşitli aday özellik alt kümelerini seçmek için

istatistiksel kriterler kullanılır. İkinci olarak en yüksek sınıflandırma doğruluğuna sahip alt küme seçilir [63]. Böylece gömülü yöntem ile hem sarmal yöntem ile karşılaştırılabilir bir doğruluk, hem de filtre yöntemiyle karşılaştırılabilir bir verimlilik elde edilebilmektedir. Gömülü yöntemlerde özellik seçimi öğrenme süresinde gerçekleştirilmektedir [6].

Genel olarak tüm özellik seçim yöntemlerinde temel amaç, aşağıdaki kriterleri karşılayacak minimum boyutlu bir özellik alt kümesini seçmeye çalışmaktır [6].

- Seçilen özellikler ile sınıflandırma doğruluğu istatistiksel açıdan büyük ölçüde azalmamalıdır,
- Yalnızca seçilen özellikler verildiğinde ortaya çıkan kümeleme dağılımı, tüm özellikler dikkate alındığında orijinal kümeleme dağılımına olabildiğince yakın olmalıdır.

2.5 Gözetimsiz Özellik Seçim Yöntemleri

Literatürde bulunan gözetimsiz özellik seçim algoritmaları uyguladıkları tekniklere göre gruplandırılmış ve bu yöntemlerden başlıcaları kısaca tanıtılmıştır [64].

Spektral ve benzerlik tabanlı yöntemler

Laplacian score for unsupervised feature selection (LS), her bir özelliğin en yakın komşularını bularak yerelliği koruma yeteneğini kullanır ve böylece özellikleri seçer [65]. Robust Neighborhood Embedding (RNE), ağırlık matrisini elde etmek için her noktayı k-en yakın komşular aracılığıyla yeniden oluşturan lineer katsayılarla verilerin yerel geometrisini karakterize eder ve An Adaptive Alternating Direction Method of Multipliers (ADMM) yöntemiyle modeli çözemktedir [66]. The SPECtrum decomposition of the Laplacian Matrix (SPEC), hem gözetimli hem de gözetimsiz özellik seçimi için spektral graf teorisine dayalı birleşik bir çerçeve sunmaktadır [67]. General Spectral Sparse Regression (GSSR), spektral öğrenme ve seyrek öğrenmeyi birleştirerek özellik seçim sürecini yerine getirmektedir [52].

Kümeleme tabanlı yöntemler

Multi-Cluster Feature Selection (MCFS) seyrek özproblemi (eigenproblem) ve en küçük kareler problemini çözerek verilerin çoklu küme yapısını korur ve böylece ilgili özellikleri seçmektedir [51]. Influence Space and Graph-based Feature Selection (ISGFS), küme yoğunluğu özelliklerine göre özellikler arasındaki bağlantıyı öğrenen, alt uzay öğrenme ve

graf analizine dayalı gözetimsiz bir özellik seçim yöntemidir [68]. Parsa ve ark. küme benzerliği ve seyrek öğrenmeyi birleştirerek özellik seçim görevlerini gerçekleştirmek için yeni bir algoritma önermişlerdir [20].

Kendi kendini temsil tabanlı yöntemler

Regularized Self-Representation (RSR), herhangi bir özelliğin diğer uygun özelliklerin doğrusal kombinasyonu olarak yeniden üretilebildiği alt uzay kümelemesinde düşük sıralı gösterimi uyararak özellikleri seçmektedir [50]. Diversity-Induced Self-Representation (DISR), çeşitliliğe ve özelliklerin dahili kendi kendini temsil etme özelliğine dayalı olarak gereksiz özellikler azaltılarak özellikler seçilmektedir [58]. NOn-conVex Regularized Self-Representation (NOVRSR), sözde etiketler yerine bir veri setinin kendine benzerliğini kullanarak özellik seçimi gerçekleştirilmektedir [69].

Bilgi teorisi tabanlı yöntemler

Information-theoretic Unsupervised Feature Selection (IUFS), açgözlü bir yaklaşımla yerel optimumları arayarak seçilen özellikler arasındaki iş birliği bilgisini maksimize etmeyi amaçlar [70]. Pairwise Dependence-based Unsupervised Feature Selection (DUFS), özellikler arasındaki karşılıklı bilgiyi ortak bir entropi yoluyla ölçerek ve bir optimizasyon problemini çözerek bağımlı özellikleri seçer [71].

Rastgele alt uzay tabanlı yöntemler

Unsupervised Feature Selection approach based on multi-Subspace Randomization and Collaboration (SRCFS), her bir rastgele alt uzayda çok sayıda özellik üretilerek özellik değerlendirmesini gerçekleştirir ve ardından özellik sıralama vektörünün tamamını elde etmek için birden çok alt uzaydan gelen bilgileri birleştirir [72].

Özellik benzerliğinden yararlanma açısından yöntemler

Efficient Unsupervised Feature Selection method through Feature Clustering (EUFSFC), Fitness Orantılı Paylaşım kümelemesini, Maksimum Bilgi Sıkıştırma İndeksi ve Simetrik Belirsizlik gibi iki özellik benzerlik kriteri ile genişleterek özellik seçimini gerçekleştirir [73]. Unsupervised Feature Selection method via joint Dictionary and Graph Learning (DGL-UFS), yerel veri uzayını korumak için bir benzerlik matrisini kullanarak orijinal veri seti üzerinde oluşturulan sözlük temel uzayından özellikleri seçer [74]. Subspace Learning for unsupervised feature selection via Adaptive Structure learning and Rank approximation

(SLASR), uyarlanabilir bir benzerlik matrisi kullanılarak yinelemeli olarak öğrenilen manifold yapısıyla özellik seçimi için bir alt uzayı öğrenmeyi amaçlar [54]. Huang ve ark. benzerlik kaynaklı graph matrisi aracılığıyla uyarlanabilir olarak öğrenilen ve bir optimizasyon algoritması ile çözülen, alt uzay tarafından aykırı değerlerin olumsuz etkisinin azaltıldığı bir gözetimsiz özellik seçim yöntemi önermiştir [75]. Structured Optimal Graph Feature Selection (SOGFS), benzerlik matrisi aracılığıyla verilerin yerel yapısını öğrenerek ve gürültü özelliklerinin etkisini azaltarak özellikleri seçen bir yöntem olarak önerilmiştir [76].

Olasılıksal yaklaşım tabanlı yöntemler

Infinite VM Mixture Model Feature selection (InVMM- Fs), küresel veri vektörlerinin, von Mises dağılımları ile parametrik olmayan Bayesian karışım modelleri aracılığıyla kümelenmesine dayanan bir yöntemdir [77]. Generalized Inverted Dirichlet through continuous Hidden Markov Models with unsupervised localized Feature Selection (GID-HMM-FS), değişken Bayes tabanlı yaklaşımla öğrenilen bir dizi GID'nin sürekli HMM'sini geliştirerek özellikleri seçer [78]. Multi-step Markov Probability Relationship for Feature Selection (NMFS), verilerin içsel bilgilerini korumayı ve çok adımlı Markov geçiş olasılığı yoluyla bir nokta ile en uzak komşuları arasındaki ilişkiyi dikkate almayı amaçlar. Bir optimizasyon problemini yinelemeli olarak çözmeye çalışır [79]. Joint Adaptive Manifold and Embedding Learning (JAMEL), veri dağıtımı altında doğrusal olmayan manifold yapısını uyarlanabilir bir şekilde öğrenerek gereksiz ve gürültülü verileri filtreler ve ayırt edici özellikleri seçer [80].

Evrimsel yaklaşım tabanlı yöntemler

Martarelli ve Nagano üç özellik seçim algoritması önerdiler: Unsupervised Feature Selection by Biased Random-Key Genetic Algorithm I and II (UFSBRKGA-I, UFSBRKGA-II) ve Unsupervised Feature Selection by Particle Swarm Optimization (UFSPSO) [81]. UFSBRKGA-II ve UFSPSO popülasyonu oluşturmak için Laplacian skorunu, gözetimsiz ayırt edici özellik seçimini ve özellik seçimi için varyans eşiklemeyi kullanırken, UFSBRKGA-I rastgele bir popülasyonla başlar.

Sözde etiketlerin kullanımına göre yöntemler

Jointly Local Geographic Structure Preservation and Redundancy Minimization (JLSPRM), verilerin etiket bilgilerini öğrenmek için negatif olmayan spektral analizden ve etiketleri

daha doğru hale getirmek için yerel geometrik yapı korumadan yararlanır [82]. Ding ve ark. veri örnekleri arasındaki ilişkiyi kullanarak gizli bir gösterim oluşturmuş ve bunu sözde etiketler olarak kullanarak özellik seçim sürecini gerçekleştirmişlerdir [83]. Denetlenmeyen özellik seçimi için Dual Space Latent Representation Learning (DSLRL), gereksiz özelliklerin olumsuz etkisini azaltmak için verilerin içsel yapısını kullanır ve ayırt edici bilgiler elde etmek için sözde etiketler olarak verilerin gizli gösterim matrisini kullanır [84]. Unsupervised Soft-label Feature Selection (USFS), gürültülü verilerin ve aykırı değerlerin etkisini hafifletmeye ve açık olmayan veri dağıtımıyla tutarlı olması için soft etiketlerin kullanımına odaklanır. Optimizasyon problemlerini çözmek için yinelemeli bir yaklaşım kullanır [85].

Graf gömme tabanlı yöntemler

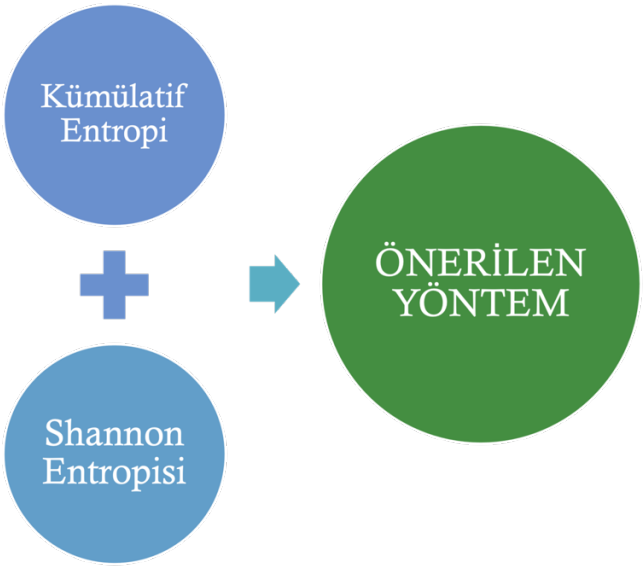
Robust Unsupervised Feature Selection (RUFS), graph matris öğrenmesi ve düşük boyutlu uzay öğrenmesi yoluyla özellik seçim işlemi gerçekleştirilir [86]. Zhou ve ark. verilerin içsel yapısını korumak için, uyarlanabilir çoklu graf (graph) öğrenimi yoluyla bir özellik seçim algoritması oluşturmuşlardır [87]. Unsupervised Feature Selection via Data Reconstruction and Side Information (DRSI-FS), özellik seçim işlemini ikili kısıtlamalar, veri yeniden yapılandırma hatasının azaltılması ve graf (graph) gömme ile ilgili konuları dikkate alarak gerçekleştirir [88].

Local Sensitive Dual Concept Learning (LSDCL), yerel duyarlı ikili kavram öğrenmeye dayalı gözetimsiz bir özellik seçim yöntemidir [89]. Unsupervised Feature Selection based Extreme Learning Machine (UFSELM), k-ortalama kümelemeyi kullanmadan bir ekstrem öğrenme makinesine dayalı olarak gözetimsiz bir özellik seçimi gerçekleştirir [90]. Reduction based algorithm on High Dimensional feature Selection with Interactions (RHDSI), üç aşamadan oluşan ve özellik torbalama ve diğer istatistiksel teknikleri kullanan bir özellik seçim yöntemidir [91]. Zhang ve ark. Alternating Direction Method of Multipliers (ADMM) ile optimize edilmiş 0-1 tamsayı kısıtlaması ile farklı öğrenme görevleri için en uygun özellik alt kümesini seçen bir çerçeve sunmuşlardır [92]. Chaudri ve ark. heterojen veri kümeleri için hem özellik sıralamasını hem de özellik seçimini gerçekleştiren iki aşamalı bir algoritma tanıtmışlardır [93]. Özellik sıralama süreci, bilgi kazancından yararlanır. Özellik seçim süreci ise gelişmiş bir Callinski-Harasbaz değerlendirme tabanlı seçim stratejisi aracılığıyla bir k-prototip kümeleme algoritmasını kullanır. Unsupervised Feature Selection via Transformed Auto-Encoder (UFS-TAE), derin bir otomatik kodlayıcı

aracılığıyla negatif olmama ve diklik ile sınırlandırılmış bir amaç fonksiyonunu çözmeyi amaçlar ve gradyan iniş yöntemini (Gradient Descent Method) kullanarak optimizasyon problemini çözer [94].

3. ÖNERİLEN YÖNTEM

Bu bölümde kümülatif entropi ve Shannon entropisini temel alan gözetimsiz bir özellik seçim yöntemi tanıtılmıştır. Şekil 3.1'de önerilen yöntemin (EFS) şematik gösterimi verilmiştir.



Şekil 3.1: Önerilen Yöntemin (EFS) şematik gösterimi.

3.1 Ön Hazırlık

Entropi bir özelliğin veri kümesindeki belirsizlik veya düzensizlik derecesini ifade etmektedir [61]. Entropinin bilgi teorisinde bilgi iletimi ve depolama alanlarında önemli bir yeri vardır. Bunun yanında veri madenciliği, makine öğrenimi ve yapay tahmin alanların da sıklıkla kullanılmaktadır [95].

3.2 Shannon Entropi (Shannon Entropy)

Shannon entropi, bilgi teorisinde temel bir kavramdır. 1948 yılında Claude Elwood Shannon tarafından "A Mathematical Theory of Communication" adlı makalede tanıtılmıştır [11]. Shannon entropi, bir olasılık dağılımının düzensizliğini veya belirsizliğini ölçmektedir [96]. Bir olasılık dağılımının entropisi o dağılımdaki olayların olasılıklarına dayanarak tahmin edilecek bilgi miktarını ifade eder.

Shannon entropi, bir olasılık dağılımının $P(x_i)$ olasılıklarıyla ağırlıklı olarak ortalama bilgi miktarını hesaplayarak bulunur:

$$H(X) = -\sum_i P(x_i)\ log\ P(x_i) \quad (1)$$

Burada $H(X)$ bir rastgele değişkenin entropisini, $P(x_i)$ ise x_i olayının olasılığını temsil eder.

Shannon entropi, iletişim sistemleri, bilgi kodlama, veri sıkıştırma ve bilgisayar bilimleri gibi birçok alanda önemli uygulamalara sahiptir [97].

Karaca ve Moonis; Multiple Sclerosis (MS) hastalarının ve sağlıklı bireylerin farklı alt gruplarını doğru bir şekilde sınıflandırmak için shannon entropi tabanlı özellik seçimi ve doğrusal dönüşüm tekniklerini birleştiren bir yöntem önermektedir [98].

3.3 Kümülatif Dağılım Fonksiyonu

Kümülatif Dağıtım Fonksiyonu, rastgele bir değişkenin olasılık dağılımını tanımlamak için kullanılan istatistiksel bir ölçüdür. Değişkenin belirli bir değerden daha küçük veya ona eşit olma olasılığını gösteren bir fonksiyondur. Kümülatif dağılım fonksiyonu; azalmama, değişken sonsuzluğa giderken 1'e yaklaşma, değişken negatif sonsuzluğa giderken 0'a yaklaşma ve sol süreklilik dahil olmak üzere belirli özellikleri karşılar [99]. Kümülatif dağılım fonksiyonu genellikle olasılıkları hesaplamak ve bir örneğe dayalı bir popülasyon hakkında çıkarımlar yapmak için istatistiksel analizlerde kullanılır. Hem sürekli hem de ayrık rastgele değişkenler için tanımlanabilmektedir [100]. Kümülatif dağılım fonksiyonu Denklem (2)'de gösterilen şekilde ifade edilir.

$$F_x(\mathrm{x}) = \mathrm{P}\ (\ \mathrm{X} \leq \mathrm{x}\ ) \quad (2)$$

3.4 Kümülatif Entropi

Kümülatif entropi, bir rastgele değişkenin dağılım fonksiyonunun kullanıldığı bir ölçümdür [10]. Diğer bir deyişle, rastgele bir değişken X'in kümülatif entropisi, X'te değerlendirilen ortalama hareketsizlik süresinin beklentisi olarak ifade edilmektedir [10]. Denklem (3)'te gösterilen şekilde hesaplanır.

$$CE(X) = -\int_{0}^{+\infty} F(x) log F(x) dx \quad (3)$$

Burada F(x) rassal değişkenin dağılım fonksiyonunu ifade eder ve log doğal logaritmayı temsil eder. Integral dağılım fonksiyonunun tüm olası değerleri üzerinden hesaplanır ve kümülatif entropinin değerini verir. Bu hesaplama, rassal değişkenin belirsizliğini veya bilgi içeriğini ölçmede kullanılır.

3.5 Önerilen Algoritma

Önerilen yöntem üç aşamadan oluşmaktadır. Birinci aşamada veri setindeki özelliklerin kümülatif entropi değerleri bulunur. Bu değere göre özellikler artan sırada sıralanır. İkinci aşamada veri setindeki özelliklerin shannon entropi değerleri hesaplanır ve bulunan entropi değerlerine göre özellikler azalan sırada sıralanır. Üçüncü aşamada ise ilk iki aşamada bulunan ve sıralanmış özellikler birlikte ele alınarak özelliklerin nihai sırası elde edilir.

1.Aşama:

Önerilen algoritmaya $X = \{x_i\}_{i=1}^{m} \Rightarrow x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right) \in \mathbb{R}^d, i = 1, \dots, m$ şeklinde bir eğitim seti verildiğinde; $f_{x^{(k)}}(x)$ normal olasılık yoğunluk fonksiyonlu sürekli bir rastgele değişken $x^{(k)}$ için, Denklem (4) ile verilen normal kümülatif dağılım fonksiyon değerleri hesaplanır. Daha sonra her özelliğin kümülatif entropisi Denklem (5) ile hesaplanır.

$$F(x; \mu, \sigma) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \quad (4)$$

$$CE\left(x^{(k)}\right) = -\sum_{i} F\left(x_i^{(k)}\right) \log_2 F\left(x_i^{(k)}\right) \quad (5)$$

Burada $erf(\cdot)$ normal dağılımın hata fonksiyonunu ifade eder ve şu şekilde gösterilir:

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{z} e^{-t^2} dt \quad (6)$$

Daha sonra bulunan entropi değerlerine göre özellikler artan entropi düzeninde sıralanır.

$$\mathrm{A} = \left\{a_j | a_j \in \{1, \dots, d\}, w_{a_j} \in CE\left(x^{(k)}\right), k \in \{1, \dots, d\}, j \in \{1, \dots, d\}, w_{a_j} \leq w_{a_{j+1}}\right\}$$

Kümülatif dağılım fonksiyonunun entropisi, olasılık dağılımından çekilen rastgele değişkenleri temsil etmek için gereken bit sayısını belirtir. Başka bir açıdan bakıldığında, X'in sık rastlanan değerleri en az bit ile temsil edilirken seyrek olanlar daha fazla bit ile ifade edilir. Bu nedenle önerilen algoritmanın ilk aşaması ihtiyaç duyulan en az bit sayısına sahip bir özelliğin en büyük öneme sahip olduğu varsayımına dayanır.

2.Aşama:

İkinci aşamada dağılımdaki simetriye göre özelliklerin Shannon entropi değerleri hesaplanır. Bu amaçla, üç merkezi eğilim ölçüsünün (ortalama, medyan ve mod) maksimumuna göre bir sınır belirlenir. Burada ortalama ($\overline{x^{(k)}}$), medyan ($\widetilde{x^{(k)}}$) ve mod ($\widehat{x^{(k)}}$) şeklinde belirtilmiştir.

Üç merkezi eğilim ölçüsünün maksimumu şu şekilde gösterilir:

$$\rho^{(k)} = \arg\max\left(\overline{x^{(k)}}, \widetilde{x^{(k)}}, \widehat{x^{(k)}}\right) \tag{7}$$

Daha sonra, orijinal veri seti (X) Denklem (8)'deki fonksiyon kullanılarak seyrek bir matrise dönüştürülür.

$$u\left(x_i^{(k)}\right) = \begin{cases} 0, & x_i^{(k)} < \rho^{(k)} \\ 1, & x_i^{(k)} \geq \rho^{(k)} \end{cases} \tag{8}$$

İkinci aşamada, dağılımın çarpıklığının entropisi (Denklem 9) ile hesaplanmaktadır.

$$H\left(x^{(k)}\right) = -\sum_{v\in\{0,1\}} \frac{\left|u\left(x^{(k)}\right) = v\right|}{|u(x^{(k)})|} \log_2 \frac{\left|u\left(x^{(k)}\right) = v\right|}{|u(x^{(k)})|} \tag{9}$$

Dönüştürülen veri setindeki her özelliğin entropisi hesaplanır ve bulunan entropi değerleri azalan entropi düzeninde sıralanır. Böylece entropisi en yüksek olan özellikler seçilir. İkinci aşamadaki varsayıma göre, en yüksek entropiye sahip özellikler en büyük öneme sahiptir.

$$\mathrm{B} = \left\{ b_j | b_j \in \{1, \dots, d\}, w_{b_j} \in H\left(x^{(k)}\right), k \in \{1, \dots, d\}, j \in \{1, \dots, d\}, w_{b_j} \geq w_{b_{j+1}} \right\}$$

3.Aşama:

Son aşamada, ilk iki aşamada özelliklerin önem sırasına göre sıralanarak elde edilen çıktılar (yani A ve B setleri) kaynaştırılmaktadır. Her özelliğin nihai sırası, ilk aşamada bulunan konumu (sırası) ile ikinci aşamada bulunan konumunun (sırasının) geometrik ortalaması Denklem (10)'da gösterildiği şekilde alınarak elde edilir. Böylece özelliklerin önem sırası algoritmanın çıktısı olarak verilir.

$$\mathrm{w}_{j=1,\dots,d} = \sqrt{\sum_{j} j\mathbf{1}_{\mathrm{A}_j}(j) \sum_{j} j\mathbf{1}_{\mathrm{B}_j}(j)} \tag{10}$$

$$\mathbf{1}_{\mathrm{A}_j}(j) = \begin{cases} 0, & \mathrm{A}_j \neq j \\ 1, & \mathrm{A}_j = j \end{cases} \tag{11}$$

3.6 Önerilen Algoritmanın Sözde Kodu

Algoritma: Entropi Tabanlı Özellik Seçimi (EFS)

Girdi:

$\boldsymbol{X} = \{\boldsymbol{x_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{m}} \Longrightarrow \boldsymbol{x_i} = \left(\boldsymbol{x_i^{(1)}}, \boldsymbol{x_i^{(2)}}, \dots, \boldsymbol{x_i^{(d)}}\right)$: Girdi verileri

1: Giriş verilerinin kümülatif dağılım fonksiyon değerlerini $\boldsymbol{P} \in \mathbb{R}^{\boldsymbol{m\times d}}$ Denklem (4) ile hesapla.

2: $\boldsymbol{P}$ **'nin** kümülatif entropisini ($\boldsymbol{c} \in \mathbb{R}^{\boldsymbol{d}}$) Denklem (5) ile hesapla.

3: $\boldsymbol{c}$**'yi** artan düzende sırala ve ilk aşama için özellik gösterge vektörünü ($\boldsymbol{k} \in \mathbb{R}^{\boldsymbol{d}}$) oluştur.

4: Üç merkezi eğilim ölçüsünün ($\rho \in \mathrm{R}^{\mathrm{d}}$) maksimumunu bul.

5: Orijinal girdi verilerini (X'i), Denklem (8) ile yönsüz ikili grafa dönüştür ($\boldsymbol{T} \in \mathbb{R}^{\boldsymbol{m\times d}}$).

6: $\boldsymbol{T'nin}$ Shannon entropisini ($\boldsymbol{h} \in \mathbb{R}^{\boldsymbol{d}}$) hesapla.

7: $\boldsymbol{h}$**'**yi azalan düzende sırala ve ikinci aşama için özellik gösterge vektörünü ($\boldsymbol{t} \in \mathbb{R}^{\boldsymbol{d}}$) oluştur.

8: Birinci özellik gösterge vektörü $\boldsymbol{k}$ ve ikinci özellik gösterge vektörü $\boldsymbol{t}$ 'yi bağımsız değişken alarak sıralama vektörünü ($\mathbf{r} \in \mathbb{R}^{\boldsymbol{d}}$) Denklem (10) ile hesapla.

r'yi artan düzende sıralayarak sıralanmış özellik göstergesi vektörü **I** değerini döndür.

Çıktı:

$\boldsymbol{I} \in \mathbb{R}^{\boldsymbol{d}}$: Sıralanmış özellik göstergesi vektörü

3.7 Önerilen Algoritmanın Zaman Karmaşıklığı

Önerilen gözetimsiz özellik seçim algoritmasının zaman karmaşıklığı şu şekilde hesaplanabilir. İlk aşamada zaman karmaşıklığı ortalama veya en iyi durumda $O(2md + dlog_2 d)$, en kötü durumda $O(2md + d^2)$ olacaktır. İkinci aşamada zaman karmaşıklığı ortalama veya en iyi durumda $O(3md + dlog_2 d)$, en kötü durumda $O(3md + d^2)$ ile hesaplanır. Son aşamada ise zaman karmaşıklığı, en iyi durumda $O(1 + dlog_2 d)$, ortalama durumda $O(d^2 + dlog_2 d)$ ve en kötü durumda ise $O(2d^2)$ ile hesaplanır. Bu nedenle, algoritmanın genel zaman karmaşıklığı en iyi durumda $O(5md + 3dlog_2 d + 1)$, ortalama durumda $O(5md + 3dlog_2 d + d^2)$ ve en kötü durumda $O(5md + 4d^2)$ ile bulunur.

Özetlemek gerekirse en iyi durum için algoritmanın zaman karmaşıklığı m≫d olduğunda doğrusal, d≫m olduğunda doğrusal ve m≈d olduğunda ikinci derecedendir. Ortalama durum için algoritmanın zaman karmaşıklığı m≫d olduğunda doğrusal, d≫m veya m≈d olduğunda ise ikinci derecedendir. En kötü durum için algoritmanın zaman karmaşıklığı m≫d olduğunda doğrusal, d≫m veya m≈d olduğunda ise ikinci derecedendir. Sonuç olarak, algoritmanın zaman karmaşıklığı, girdi verilerine bağlı olarak doğrusal zamandan ikinci dereceden zamana kadar değişiklik gösterir.

3.8 Seçilen Özelliklerin Sayısının Belirlenmesi

Algoritmaları değerlendirirken seçilen özelliklerin sayısına karar vermek için bir alt sınır elde edilmiştir. ϵ bir sınıflandırma algoritmasının tüm girdi verileri üzerindeki hata oranını ifade etsin. Öğrenebilen hiçbir sınıflandırıcı, rastgele bir tahmin ediciden daha az doğruluk oranına sahip olamaz. Rastgele tahmin edicinin doğruluk oranı ile sınıflandırma algoritmasının doğruluk oranı sınırlandırılır. Çoğunluk belirleyicisinin doğruluk oranı n/m'ye eşittir. Burada n çoğunluk sınıfının sayısıdır. Çoğunluk sınıfı, bir sınıflandırma probleminde tahmin edilen sonuçların çoğunluğunu oluşturan sınıftır. Çoğunluk tahmincisinin hata oranı $1 - \frac{\mathrm{n}}{\mathrm{m}}$'dir. Ayrıca her bir özellikteki çoğunluk tahmincisinin hata oranı $1 - \frac{\mathrm{n}}{\mathrm{m}}$'dir. Özelliklerin birbirinden bağımsız olduğu varsayıldığında ilk d' özellik için hata oranı $\left(1 - \frac{\mathrm{n}}{\mathrm{m}}\right)^{\mathrm{d}'}$ olur. Buna göre, Eşitsizlik (12) tarafından verilen eşitsizliği sağlayan d' bulunur.

$$\left(1-\frac{n}{m}\right)^{d'} \leq \epsilon \tag{12}$$

$\left(1-\frac{n}{m}\right) \leq e^{-n/m}$ olduğu için

$$-\frac{m}{n}\ln\epsilon \leq d' \tag{13}$$

Seçilen en az d' özniteliğe sahip bir girdi verisi üzerindeki hata oranı yaklaşık olarak ϵ'dir. Ayrıca deneysel sonuçlar da bu sonucu doğrulamaktadır. Bu açıdan gözetimsiz özellik seçim algoritmalarını değerlendirirken küçük bir özellik alt kümesinin kullanılması yeterli olacaktır.

Yukarıdaki duruma ek olarak, herhangi iki gözetimsiz özellik seçim algoritmasının azalan önem sırasına göre sıraladığı özellikleri ele alındığında bu iki kümenin ilk k özelliğinin benzerlik olasılığı aşağıdaki şekilde hesaplanır. Buna göre, d özelliklerden k'nin sıralı düzenlemelerinin sayısı Denklem (14) ile verilir.

$$P_k^d = \frac{d!}{(d-k)!} \tag{14}$$

k adet özelliğin sıralı düzenlemelerinin sayısı k!'dir. Böylece, bu iki kümenin ilk k özelliklerinin benzerlik olasılığı Denklem (15) ile verilmiştir.

$$P_{benzerlik} = \frac{k!\,(d-k)!}{d!} \tag{15}$$

Sonuçlar, özellik sayısı arttıkça benzerlik olasılığının azaldığını göstermektedir. Bu nedenle, gözetimsiz özellik seçim algoritmalarını değerlendirmek için en fazla $d-1$ özellik seçilebilir. Sonuç olarak, $-\frac{m}{n}\ln\epsilon$ ile $d-1$ aralığındaki özelliklerin sayısı seçilebilir.

Bu çalışmada alt sınırdaki değişimi belirtmek için 1 ile 15 aralığında özellik sayısı seçilmiştir.

4. DENEYSEL PROSEDÜR

Bu bölümde, önerilen yöntemin performansını ölçmek için yapılan deneysel çalışmada kullanılan gözetimsiz özellik seçim yöntemleri ve veri setleri hakkında bilgiler verilmiştir.

4.1 Çalışmada Kullanılan Veri Setleri

Bu çalışmada farklı alanlardan on iki veri seti kullanılmıştır. Veri setleri makine öğrenimi algoritmalarının deneysel analizi için kullanılan açık veri havuzlarından temin edilmiştir. Veri setleri seçilirken özellik sayıları ve örnek sayılarının birbirinden farklı olmasına ve değişik alanlardan verilerin kullanılmasına dikkat edilmiştir. Bazı veri setlerinde örnek sayısı özellik sayısından fazla iken bazı veri setlerinde özellik sayısı örnek sayısından fazladır. Tablo 4.1'de veri setlerinin tanımlayıcı bilgileri gösterilmektedir. Daha sonra da veri setleri hakkında kısa bilgiler verilmiştir.

Tablo 4.1: Deneylerde kullanılan veri setlerinin özellikleri (*m* örnek sayısı, *d* özellik sayısı, *c* sınıf sayısı ve *r* dengesizlik oranıdır).

#	Veri Seti	*m*	*d*	*c*	*r*	Alan
1	cardiotocography	2126	21	3	9,40	Medikal
2	climate model	540	18	2	10,73	İklim
3	colon	62	2000	2	1,82	Biyolojik
4	connectionist bench	208	60	2	1,14	Fizik ve kimya
5	diabetic retinopathy	1151	19	2	1,13	Sağlık ve Tıp
6	dna	3186	180	3	2,16	Biyolojik
7	ecoli-uni	336	343	8	71,50	Biyolojik
8	flowmeterA	87	36	2	1,49	Arıza Tespiti
9	Madelon	2000	500	2	1,00	Yapay
10	qsar biodegradation	1055	41	2	1,96	Kimyasal
11	vehicle	846	18	4	1,10	Görüntü
12	wall following robot	5456	24	4	6,72	Bilgisayar Bilimi

Cardiotocography Veri Seti

Cardiotocography veri seti doğum uzmanları tarafından sınıflandırılan kardiyotokogramlardaki fetal kalp hızı ve uterus kasılması özelliklerinin ölçümlerinden oluşan değerleri içerir [101]. Ölçüm değerlerine göre fetal durum sınıf etiketleri (N=normal; S=şüpheli; P=patolojik) olarak işaretlenmiştir.

Climate Model Veri Seti

Climate Model veri seti, iklim modeli belirsizlik ölçümü (Uncertainty Quantification [UQ]) toplulukları sırasında karşılaşılan simülasyon kazalarının kayıtlarını içermektedir. The Community Climate System Model (CCSM), dünyanın iklim sistemini simüle etmeye yönelik birleşik bir iklim modelidir. Aynı anda dünyanın atmosferini, okyanusunu, kara yüzeyini ve deniz buzunu simüle eden dört ayrı modelden ve bir merkezi bağlayıcı bileşenden oluşan CCSM, araştırmacıların dünyanın geçmiş, mevcut ve gelecekteki iklim durumlarına ilişkin temel araştırmalar yürütmesine olanak tanır. CCSM, the Parallel Ocean Program bileşeni içindeki 18 model parametresinin belirsizliklerini örneklemek için the Lawrence Livermore National Laboratory (LLNL)'nin UQ Pipeline yazılım sisteminde bir Latin hiperküp yöntemi kullanılarak oluşturulmuştur. Her biri 180 topluluk üyesi içeren üç ayrı Latin hiperküp topluluğu gerçekleştirilmiştir. 540 simülasyondan 46'sı parametre değerlerinin kombinasyonunda sayısal nedenlerden dolayı başarısız olmuştur. Veri setinde bu örnekler “başarılı ve” başarısız” olarak etiketlenmiştir [102].

Colon Veri Seti

Colon veri seti Arizona Eyalet Üniversitesi'nde geliştirilen Python'da açık kaynaklı bir özellik seçim deposu olan scikit-feature projesinden alınmıştır [103].

Connectionist Bench Veri Seti

Connectionist Bench veri seti bir mayın tarlasında bulunan 208 metal ve kaya parçalarına çeşitli açılarda ve çeşitli koşullar altında gönderilen sonar sinyallerinin yansıtılmasıyla elde edilen frekans değerlerini içermektedir. Mayın tarlasına gönderilen sinyallerden alınan frekans değerlerine göre o nesnenin metal mi yoksa kaya mı olduğunun bilgisini barındırır. Veri seti her örnekte 0,0 ile 1,0 aralığında 60 özellikten oluşan değerler içerir. Her özellik, belirli bir frekans bandındaki, belirli bir süre boyunca entegre edilen enerjiyi temsil eder. Eğer nesne bir kaya ise "R" harfi ile, bir metal ise "M" harfi ile etiketlenmiştir [104]. Örnekler metal bir silindirden yansıyan sonar sinyalleri ile kabaca silindirik bir kayadan yansıyan sonar sinyallerini ayırt edecek şekilde bir ağı eğitmek için kullanılabilir.

Diabetic Retinopathy Veri Seti

Diabetic Retinopathy veri seti, bir görüntünün diyabetik retinopati belirtileri içerip içermediğini tahmin etmek için Messidor görüntü setinden çıkarılan özellikleri içerir. Diyabetik retinopati, şeker hastalarında görülen bir göz rahatsızlığıdır. Tüm özellikler tespit

edilen bir lezyonu, bir anatomik parçanın tanımlayıcı bir özelliğini ya da görüntü düzeyinde bir tanımlayıcıyı temsil eder. Veriler tam sayı ve reel sayılardan oluşmaktadır [105]. Sınıf etiketleri: 1= diyabetik retinopati belirtileri var, 0= diyabetik retinopati belirtisi yok anlamındadır.

DNA Veri Seti

DNA veri seti Irvine veri tabanının işlenmiş bir versiyonudur. Veri tabanında DNA'da bulunan Nükleotidleri temsil eden sembolik değişkenler (A,G,T,C) 3 ikili gösterge değişkeniyle değiştirilmiştir. Böylece orijinal 60 sembolik nitelik, 180 ikili niteliğe dönüştürülmüştür [106].

Ecoli-uni Veri Seti

Veri seti, E.coli proteinlerinin hücre lokalizasyon bölgelerindeki amino asit dizilerini kullanarak sınıflandırılması problemini açıklamaktadır . Yani proteinin katlanmadan önceki kimyasal bileşimine dayanarak bir proteinin hücreye nasıl bağlanacağını tahmin etmeyi amaçlar. 8 farklı sınıfa ayrılmış 336 E.coli proteini içerir [107].

FlowmeterA Veri Seti

Sıvı ultrasonik debimetrenin arıza teşhisi için oluşturulmuş veri setidir. Sayaçtan gelen veriler 8 yollu bir sıvı ultrasonik akış ölçerden alınmıştır. 87 teşhis parametresi örneği içermektedir [108].

Madelon Veri Seti

MADELON, NIPS 2003 özellik seçimi atölyesinin bir parçası olan yapay bir veri kümesidir. Veri seti sürekli girdi değişkenlerine sahip iki sınıflı bir sınıflandırma problemidir. MADELON beş boyutlu bir hiper küpün köşelerine yerleştirilen ve rastgele +1 veya -1 olarak etiketlenen 32 küme halinde gruplandırılmış veri noktalarını içeren yapay bir veri kümesidir. Beş boyut, 5 bilgilendirici özelliği oluşturur. Bu özelliklerin 15 doğrusal kombinasyonu, 20 bilgilendirici özellikten oluşan bir set oluşturmak üzere eklenmiştir. Bu 20 özelliğe dayanarak örnekleri 2 sınıfa ayırmak gerekir. Veri setine tahmin gücü olmayan bir takım dikkat dağıtıcı özellikler de eklenmiştir. Özelliklerin ve desenlerin sırası rastgele seçilmiştir [109].

Qsar Biodegradation Veri Seti

Bu veri seti 1055 kimyasal maddeyi biyolojik olarak parçalanabilen hazır veya hazır olmayan şeklinde sınıflandırmak için kullanılan 41 özelliğe (moleküler tanımlayıcılar) ilişkin değerleri içerir. Quantitative Structure Activity Relationships (QSAR) biyolojik bozunma veri seti Milano Kemometri ve QSAR Araştırma Grubunda (Université degli Studi Milano - Bicocca, Milano, İtalya) oluşturulmuştur. Veriler moleküllerin kimyasal yapısı ile biyolojik parçalanması arasındaki ilişkilerin incelenmesine yönelik QSAR (Kantitatif Yapı Aktivite İlişkileri) modellerinin geliştirilmesinde kullanılmıştır. 1055 kimyasalın biyolojik bozuma deneysel değerleri, Japonya Ulusal Teknoloji ve Değerlendirme Enstitüsü'nün web sayfasından toplanmıştır [110].

Vehicle Veri Seti

Vehicle veri seti dört farklı aracın farklı açılardan çekilmiş resimlerinden elde edilen özelliklerini içermektedir. Bu veri setindeki özellikler, Hiyerarşik Görüntü İşleme Sistemi uzantısı (Hierarchical Image Processing System) BINATTS tarafından siluetlerden çıkarılmıştır. BINATTS ölçek bağımsız özelliklerin bir kombinasyonunu çıkarmak için klasik momentler temelli ölçülerden (örneğin ölçeklenmiş varyans, eğrilik ve major/minor eksenler etrafındaki basıklık ve boşluklar, dairesellik, dikdörtgenlik ve yoğunluk gibi sezgisel ölçülerden) yararlanarak çalışır. Amaç siluetten çıkarılan bir dizi özelliği kullanarak belirli bir silueti dört araç türünden biri olarak sınıflandırmaktır [111].

Wall following robot Veri Seti

Veriler SCITOS G5 robotunun bel çevresinde bulunan 24 sensörden robotun duvarı takip ederek saat yönünde 4 tur boyunca odada gezinmesi esnasında toplanmıştır. Toplanan veriler ile öğrenme algoritmalarının robota komut vererek duvarı takip etmesi ve çarpışma olmadan odanın içinde gezinmesi amaçlanmaktadır [112].

4.2 Çalışmada Kullanılan Özellik Seçim Algoritmaları

Tablo 4.2'de deneylerde kullanılan gözetimsiz özellik seçim algoritmaları ve algoritmaların özellikleri hakkında bilgiler gösterilmektedir.

Tablo 4.2: Deneylerde kullanılan gözetimsiz özellik seçim algoritmaları.

#	Yöntem	Yaklaşım	Kategori	Teknik
1	DISR	Filtre	Çok Değişkenli	Diversity and the internal self-representation
2	DUFS	Filtre	Çok Değişkenli	İkili bağımlılık (Joint entropy)
3	IUFS	Filtre	Çok Değişkenli	The alternative conditional expectation and the generalized maximal correlation
4	LS	Filtre	Tek Değişkenli	Laplacian eigenmaps and LPP
5	MCFS	Filtre	Çok Değişkenli	Spectral embedding and sparse learning
6	RNE	Filtre	Çok Değişkenli	The locally linear embedding
7	RSR	Filtre	Çok Değişkenli	Regularized self-representation
8	SRCFS	Filtre	Çok Değişkenli	Balanced multi-subspace randomization
9	SPEC	Filtre	Tek Değişkenli	Spectral graph theory
10	USFS	Filtre	Çok Değişkenli	Soft-label learning

Diversity-Induced Self-Representation (DISR), çeşitliliğe ve özelliklerin dahili kendi kendini temsil etme özelliğine dayalı olarak gereksiz özellikleri azaltarak özellikleri seçer [58].

Pairwise Dependence-based Unsupervised Feature Selection (DUFS), özellikler arasındaki karşılıklı bilgiyi ortak bir entropi yoluyla ölçerek ve bir optimizasyon problemini çözerek bağımlı özellikleri seçer [71].

Information-theoretic Unsupervised Feature Selection (IUFS), açgözlü bir yaklaşımla yerel optimumları arayarak seçilen özellikler arasındaki iş birliği bilgisini maksimize etmeyi amaçlar [70].

Laplacian score for unsupervised feature selection (LS), her bir özelliğin en yakın komşularını bularak yerelliği koruma yeteneğini kullanır ve böylece özellikleri seçer [65].

Multi-Cluster Feature Selection (MCFS), seyrek özproblemi (eigenproblem) ve en küçük kareler problemini çözerek verilerin çoklu küme yapısını korur ve böylece ilgili özellikleri seçer [51].

Robust Neighborhood Embedding (RNE), ağırlık matrisini elde etmek için her noktayı k-en yakın komşular aracılığıyla yeniden oluşturan lineer katsayılarla verilerin yerel

geometrisini karakterize eder ve An Adaptive Alternating Direction Method of Multipliers (ADMM) yöntemiyle modeli çözer [66].

Regularized Self-Representation (RSR), herhangi bir özelliğin diğer uygun özelliklerin doğrusal kombinasyonu olarak yeniden üretilebildiği alt uzay kümelemesinde düşük sıralı gösterimi uyararak özellikleri seçer [50].

Unsupervised Feature Selection approach based on multi-Subspace Randomization and Collaboration (SRCFS), her bir rastgele alt uzayda çok sayıda özellik üreterek özellik değerlendirmesini gerçekleştirir ve ardından özellik sıralama vektörünün tamamını elde etmek için birden çok alt uzaydan gelen bilgileri birleştirir [72].

The SPECtrum decomposition of the Laplacian Matrix (SPEC), hem gözetimli hem de gözetimsiz özellik seçimi için spektral graf teorisine dayalı birleşik bir çerçeve sunar [67].

Unsupervised Soft-label Feature Selection (USFS), gürültülü verilerin ve aykırı değerlerin etkisini hafifletmeye ve açık olmayan veri dağıtımıyla tutarlı olması için soft etiketlerin kullanımına odaklanır. Optimizasyon problemlerini çözmek için yinelemeli bir yaklaşım kullanır [85].

4.3 Çalışmada Kullanılan Sınıflandırma Algoritmaları

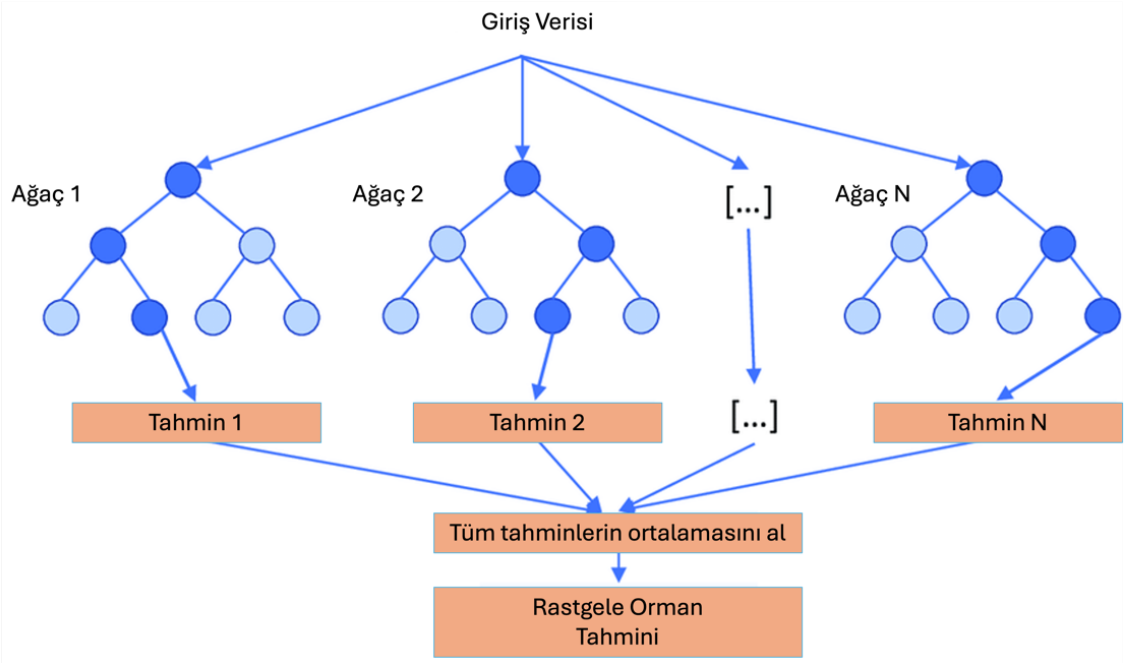
Önerilen yöntemi sınıflandırma performansı açısından değerlendirmek için iyi bilinen 5 sınıflandırıcı kullanılmıştır: Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), k-Nearest Neighbors (KNN) ve Naive Bayes (NB). Bu beş sınıflandırıcı, özellik seçim yöntemlerini doğrulamak için en çok kullanılan ve en bilindik sınıflandırıcılar arasında yer almaları nedeniyle seçilmiştir.

Rastgele Orman (Random Forest [RF])

Rastgele orman birçok karar ağacının bir araya getirilmesiyle oluşturulan gözetimli bir makine öğrenimi algoritmasıdır. Sınıflandırma ve regresyon problemlerinde kullanılmaktadır.

Rastgele orman algoritmasında eğitim veri setinden rastgele alt kümeler seçilir. Her alt küme üzerinde birçok karar ağacı oluşturulur. Bu ağaçlar ile veri örnekleri kullanılarak

sınıflandırma yapılmaktadır. Her ağaç farklı özelliklerle ve rastgele veri noktalarıyla eğitilir. Oluşturulan karar ağaçları bir araya getirilir ve tahmin yapmak için kullanılır. Sınıflandırma problemlerinde her karar ağacının tahmin ettiği sınıfların ortalaması alınarak veya oylama yapılarak (en çok oyu alan) sınıf tahmini belirlenir. Regresyon problemlerinde ise her karar ağacının tahmin ettiği değerlerin ortalaması alınarak tahmin belirlenir [113]. Şekil 4.1'de rastgele orman algoritmasının şematik gösterimi verilmiştir.

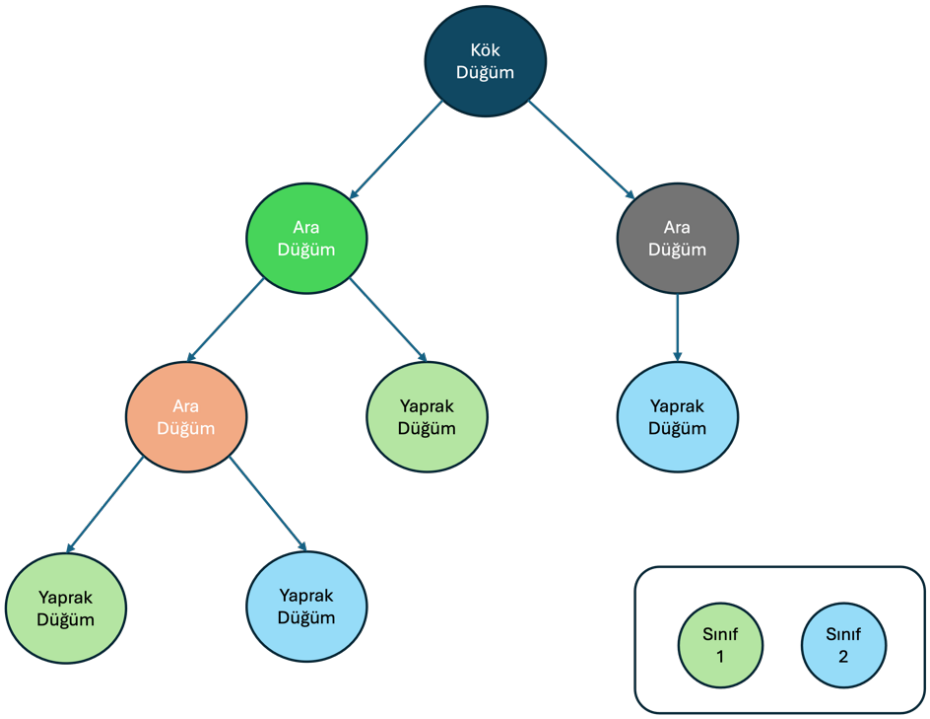


Şekil 4.1: Rastgele Orman (Random Forest) şematik gösterimi [114].

Sınıflandırma ve Regresyon Ağaçları (Classification and Regression Trees [CART])

Sınıflandırma ve regresyon ağaçları (CART) 1984 yılında Leo Breiman tarafından önerilmiş bir yöntemdir ve Random Forest (RF)'ın temelini oluşturur [115]. Kural tabanlı bir algoritmadır. Bu algoritmada veri seti, bağımlı değişkene göre homojen alt gruplara ayrılırken, bir ağaç şeklinde hiyerarşik bir düzende dallanmaktadır. Bu ağaç yapısında ara düğümler en iyi ayrımı yapan bağımlı değişkenleri gösterir. Bu düğümlerin dallarında ayırıcı bağımlı değişkenlerin kritik değerleri belirtilirken, yaprak düğümlerinde bağımlı değişkenin değerleri gösterilir. Ağacın kök düğümünden başlayarak yaprak düğümlere kadar uzanan hatlar boyunca, sınıflar arası ayrımı maksimize eden ve her sınıfın içindeki varyasyonu minimize eden ayırma kuralları bulunmaktadır. Bu yaklaşım hem kategorik hem de sürekli bağımlı değişkenlerin modellenmesine olanak tanır. Bağımlı değişken kategorik ise yöntem

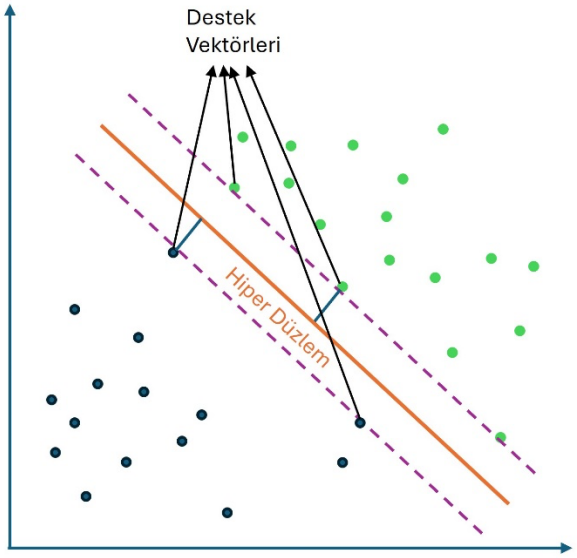
sınıflandırma ağacı olarak adlandırılırken, sürekli ise regresyon ağacı olarak adlandırılmaktadır [116]. Şekil 4.2'de sınıflandırma ve regresyon ağaçlarının şematik gösterimi verilmiştir.



Şekil 4.2: Sınıflandırma ve Regresyon Ağaçları Algoritmasının şematik gösterimi.

Destek Yöney Makinesi (Support Vector Machine [SVM])

Sınıflandırma problemlerinde en sık kullanılan algoritmalardan biri olan destek yöney makineleri, 1963 yılında Vapnik'in çalışmalarında ortaya çıkan istatistiksel öğrenme teorisini temel almaktadır. Çok boyutlu verilerde sınıfların birbirinden ayrılmasını sağlayan hiper düzlemlerin bulunmaya çalışıldığı istatistik tabanlı denetimli bir makine öğrenmesi yöntemidir [117]. Ayırıcı hiper düzlem sınıflar arasındaki genişliğin maksimum olduğu karar yüzeyi olarak tanımlanabilir. Bu hiper düzleme en uzak ve ait olduğu sınıfın sınırını çizen noktalar ise destek vektörleri olarak adlandırılmaktadır. Şekil 4.3'te destek yöney makinesinin şematik gösterimi verilmiştir.



Şekil 4.3: Destek Yöney Makinesi (Support Vektor Machine) şematik gösterimi.

K-En Yakın Komşular (k-Nearest Neighbors [KNN])

K-En yakın komşular algoritması sınıflandırma ve regresyon problemlerinde kullanılan basit bir makine öğrenmesi algoritmasıdır. Temel prensibi veri noktalarının komşularının etrafındaki etiketlere dayanarak yeni bir veri noktasını sınıflandırmaktır [118].

KNN algoritması eğitim aşamasında herhangi bir model oluşturmaz. Veri seti, basitçe algoritmanın belleğinde saklanır. Yeni bir veri noktasının sınıflandırılması veya tahmin edilmesi gerektiğinde KNN algoritmasında şu adımlar izlenir:

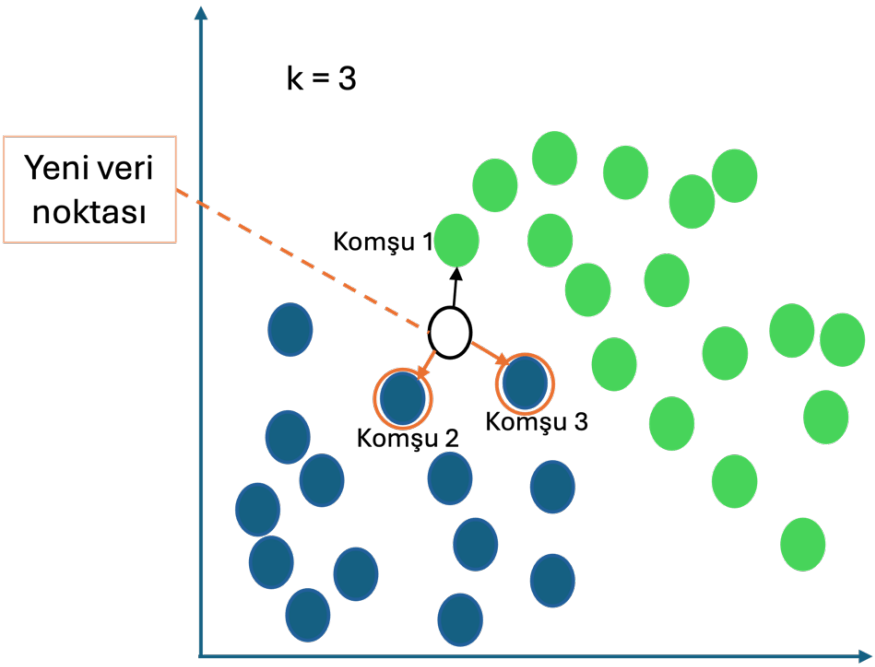
1. Kullanıcı tarafından daha önceden belirlenmiş "k" adet en yakın komşu veri noktası bulunur. Komşuluk, Öklid, Minkowski veya Manhattan gibi bir mesafe metriği kullanılarak belirlenir.

2. Bulunan komşu noktalarının etiketleri incelenir.

3. Sınıflandırma yapılıyorsa, bu komşu noktaların etiketleri arasında çoğunluk oyu alınarak yeni veri noktasının sınıfı belirlenir. Regresyon yapıyorsa, komşu noktaların etiketlerinin ortalaması alınarak tahmin edilen değer elde edilir.

K Değerinin Seçimi: k değeri komşu sayısını belirler. Kullanıcı tarafından belirlenen bu değer genellikle tek sayı olarak seçilmektedir. Küçük seçilen k değerleri aşırı uyuma (overfitting) yol açabilirken, büyük seçilen k değerleri de yanlılığa (bias) yol açabilmektedir.

Uygun bir k değerinin seçimi deneme yanılma yoluyla belirlenir. Şekil 4.4'te K-En yakın komşular algoritmasının şematik gösterimi verilmiştir.



Şekil 4.4: K-En Yakın Komşular (k-Nearest Neigbors) şematik gösterimi.

Saf Bayes (Naive Bayes [NB])

Thomas Bayes'in adını verdiği Bayes Teoremine dayanan olasılıksal bir sınıflandırma yöntemidir. Her özelliğin sınıflara bağımsız olarak katkıda bulunduğu varsayılır. Sınıf olasılıklarını hesaplamak için eğitim verilerinin frekansları kullanılır [119].

Naive Bayes algoritmasında verilen bir örneğin her bir sınıfa ait olma olasılığı hesaplanır ve en yüksek olasılığa sahip olan sınıf seçilir. Örneğin bir e-postanın spam olma olasılığı spam olmama olasılığından daha yüksekse e-posta spam olarak sınıflandırılır.

4.4 Performans Ölçütleri

Sınıflandırıcıların performansını ölçmek için kullanılan çeşitli yöntemler bulunmaktadır. Bu çalışmada kullanılan yöntemler kısaca açıklanmıştır.

Doğruluk (Accuracy [ACC]) oranı

Öğrenme algoritmasının doğru olarak sınıflandırdığı örneklerin oranıdır. Doğruluk oranı etiketleri doğru tahmin edilen örneklerin sayısının toplam örnek sayısına bölünmesi ile elde edilir. Doğruluk oranı Denklem (16) ile hesaplanır.

$$Doğruluk\ Oranı = \frac{Doğru\ sınıflandırılan\ örnek\ sayısı}{Tüm\ örneklerin\ sayısı} \quad (16)$$

Bu oran veri setleri ile eğitilen algoritmanın gerçek hayattaki kullanımı sırasında yapacağı tahminlerin ne kadar doğrulukta olabileceğini belirtmektedir.

Hata Oranı: Yanlış sınıflandırılmış örneklerin sayısının toplam örnek sayısına bölümü ile elde edilir. Hata oranı Denklem (17) ile hesaplanır.

$$Hata\ Oranı = \frac{Yanlış\ sınıflandırılan\ örnek\ sayısı}{Tüm\ örneklerin\ sayısı} \quad (17)$$

Çapraz Doğrulama (Cross Validation):

K Katlı Çapraz Doğrulama (K-Fold Cross-Validation) yönteminde veri seti k eşit parçaya bölünür. Her bir parça sırayla test verisi olarak kullanılırken k-1 tanesi eğitim için kullanılmaktadır. Bunun sonucunda, her bir örnek en az bir kez test örneği olarak kullanılmış olur. Böylece modelin genel performansı daha doğru değerlendirilebilir.

Buradaki k değeri akademik çalışmalarda genel olarak 2, 5 veya 10 olarak seçilmektedir. K değeri 10 olarak seçilirse veri kümesi hemen hemen 10 eşit parçaya bölünür. Bu parçaların 9'u eğitim kümesi için kullanılırken geri kalanı test kümesi için kullanılır. Sonuç olarak bu süreç her bir parça test verisi olacak şekilde 10 kez tekrarlanır. Modelin performansı, her test kümesi için elde edilen doğruluk oranlarının ortalaması alınarak ölçülür.

4.5 Çalışmada Kullanılan Yazılım Araçları ve Tasarım Ortamı

Çalışmanın uygulama aşamasında Tablo 4.2'de bilgileri verilen özellik seçim yöntemleri kullanılarak Tablo 4.1'de verilen veri setleri, özelliklerin önem sırasına göre sıralanmıştır. Önem sırasına göre sıralanmış özelliklerden en önemli olan ilk d tanesi 5 sınıflandırıcıya gönderilmiştir ve her sınıflandırıcının doğruluk oranı hesaplanmıştır.

Deneyler, Windows 10 Pro (64-bit) işletim sistemi yüklü, 16 GB RAM ve 2,60 GHz hızında i7-6700HQ işlemciye sahip bir bilgisayarda MATLAB R2021a programı kullanılarak

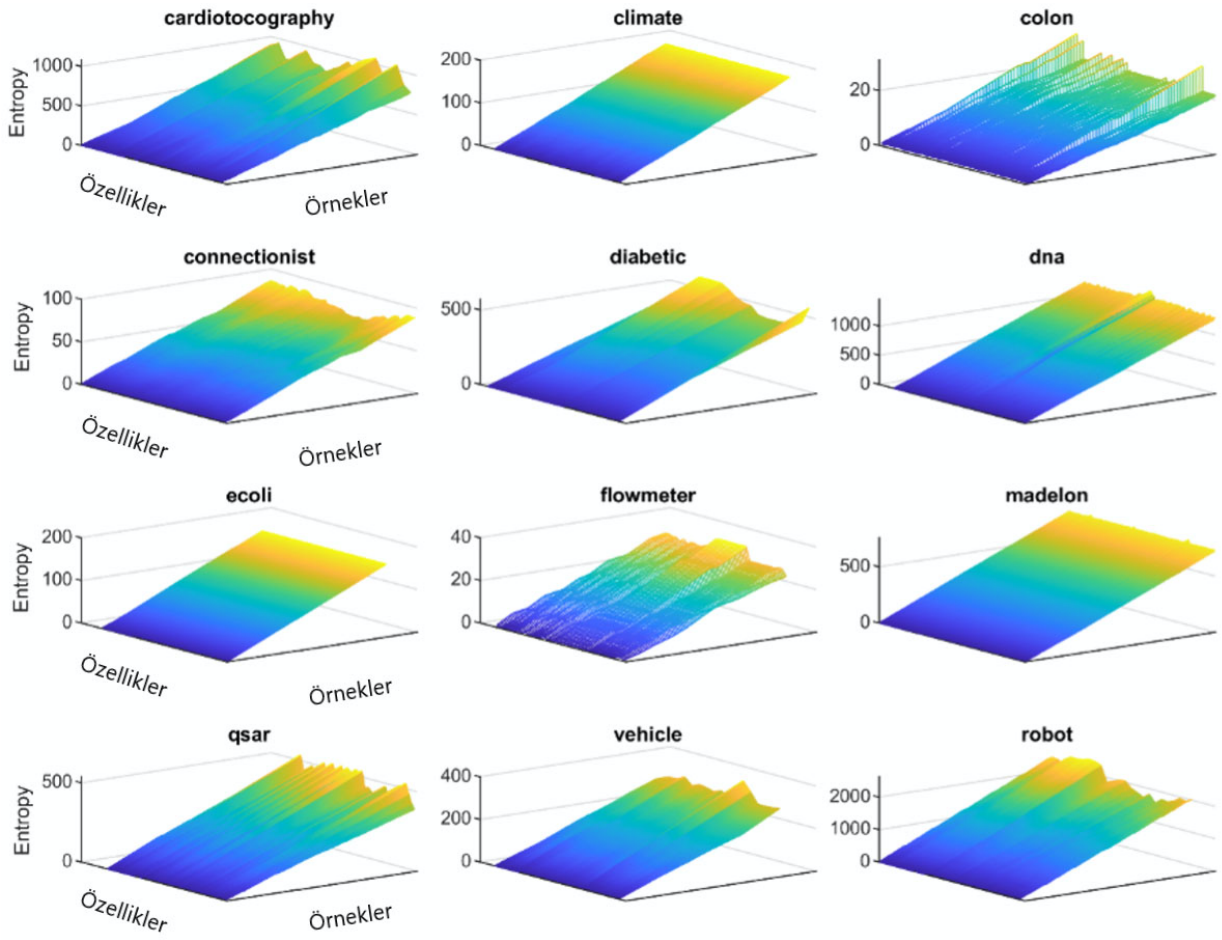
gerçekleştirilmiştir. Değerlendirme için birçok veri kümesinde az sayıda örnek bulunduğundan deneylerin güvenirliğini sağlamak için tüm deneylerde 10 kat çapraz doğrulama uygulanmıştır. Her deney farklı kat kombinasyonları ile farklı eğitim setleri oluşturmak için on kez yapılmıştır.

Bölüm 5'te Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), k-Nearest Neighbors (KNN) ve Naive Bayes (NB) makine öğrenimi algoritmaları kullanılarak Tablo 4.1'de özellikleri verilen veri setleri ile sınıflandırma açısından sonuçlar gösterilmiştir. Ardından, sonuçlar çalışma zamanı açısından karşılaştırılmış.

5. BULGULAR

Bu bölümde önerilen algoritmanın performansı, sınıflandırma deneyleri yoluyla değerlendirilmiştir. Öncelikle veri setlerinin kümülatif ve shannon entropi değerleri açısından değişimleri gösterilmiştir. Daha sonrasında tüm veri setleri üzerinde denetimsiz özellik seçim algoritmalarının seçtiği özelliklere göre 5 sınıflandırıcının doğruluk oranları karşılaştırılmıştır.

Şekil 5.1'de örneklerin sayısına bağlı olarak özelliklerin kümülatif entropisindeki değişim gösterilmektedir.

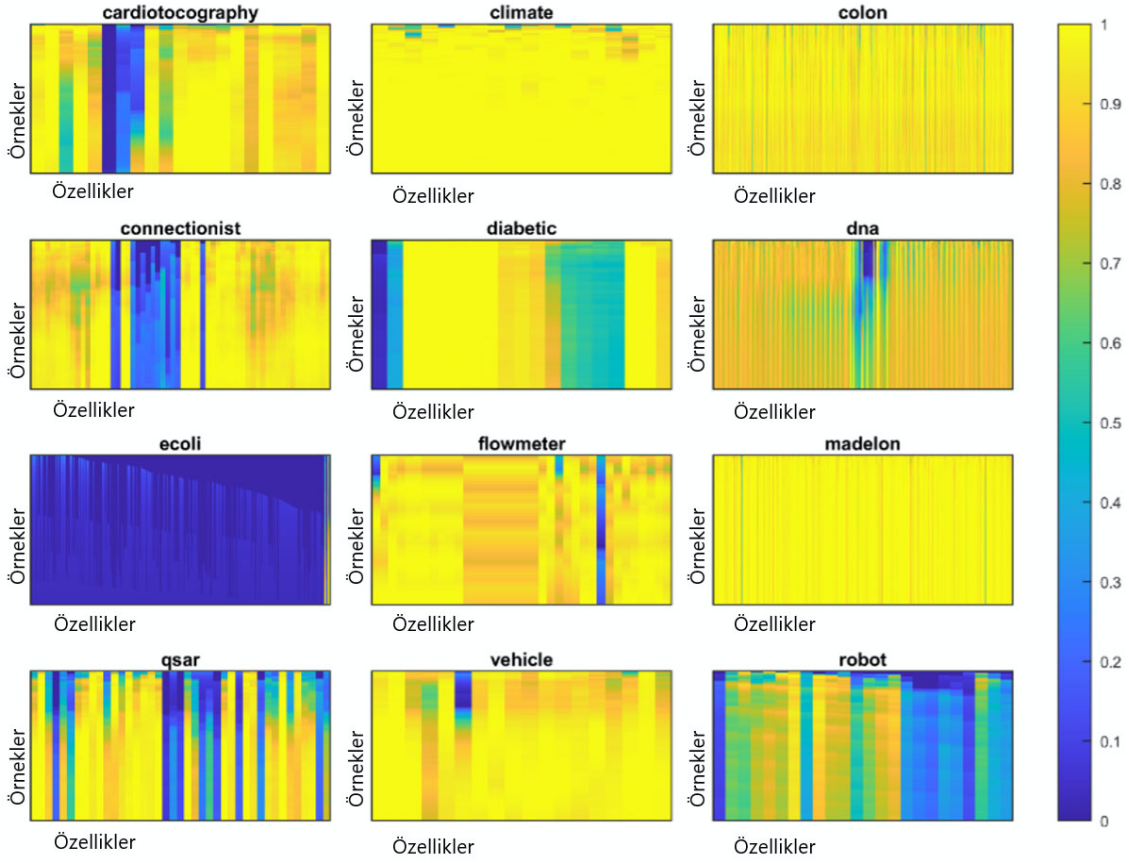


Şekil 5.1: Örneklerin sayısı açısından özelliklerin kümülatif entropilerinin değişimi.

Kümülatif dağılım fonksiyonunun entropisi, olasılık dağılımından çekilen bir rastgele değişkene kadar değişkenleri temsil etmek için gereken bit sayısını belirtir. Başka bir açıdan

bakıldığında, X'in sık rastlanan değerleri en az bit ile temsil edilirken seyrek olanlar daha fazla bit ile ifade edilir. Bu nedenle önerilen algoritmanın ilk aşaması ihtiyaç duyulan en az bit sayısına sahip bir özelliğin en büyük öneme sahip olduğu varsayımına dayanır.

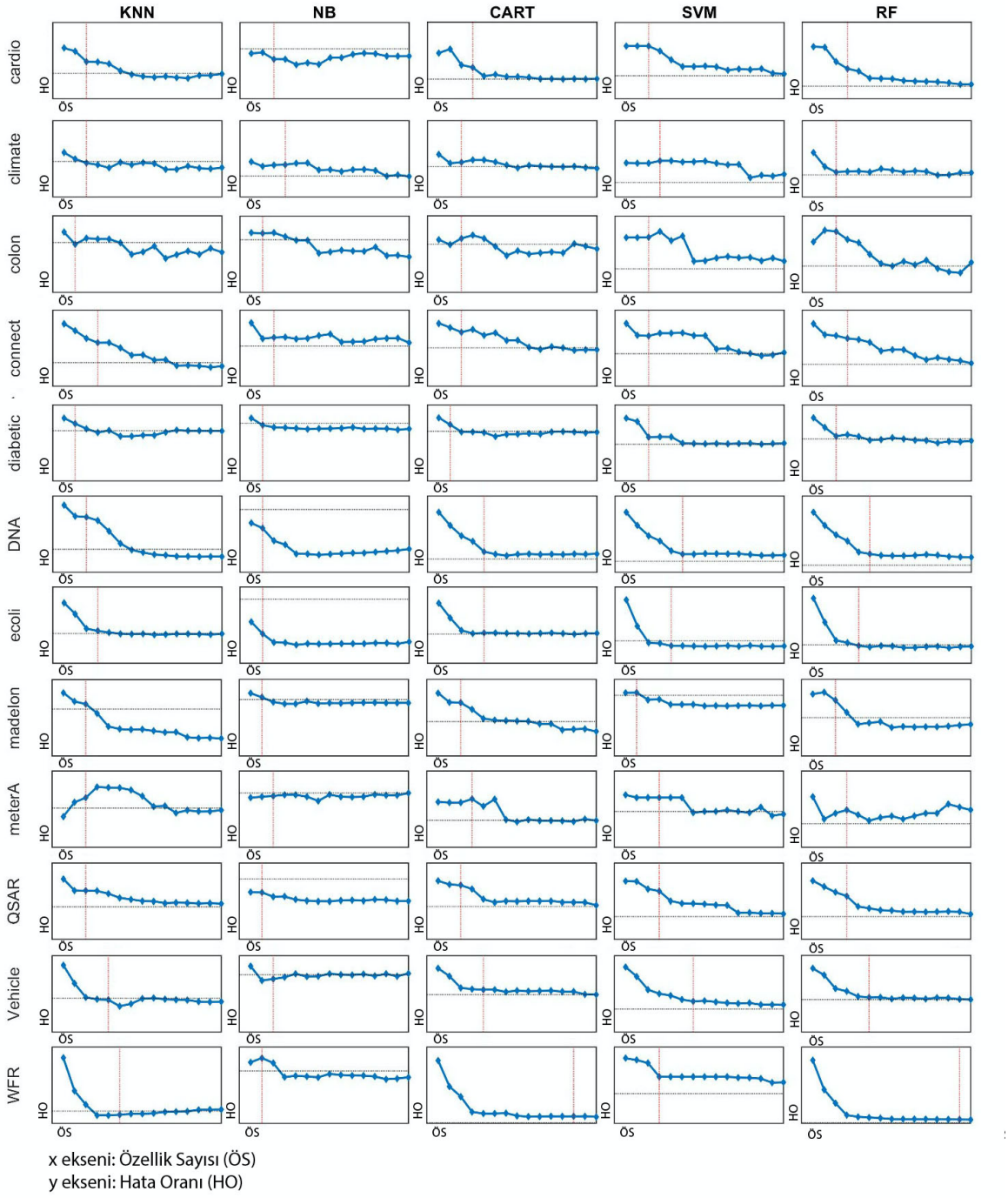
Şekil 5.2'de örnek sayısına bağlı olarak her boyutta dağılımın simetrisinin Shannon entropisindeki değişim gösterilmektedir.



Şekil 5.2: Örnek sayısı açısından her boyuttaki dağılımın simetrisinin Shannon entropisindeki değişim.

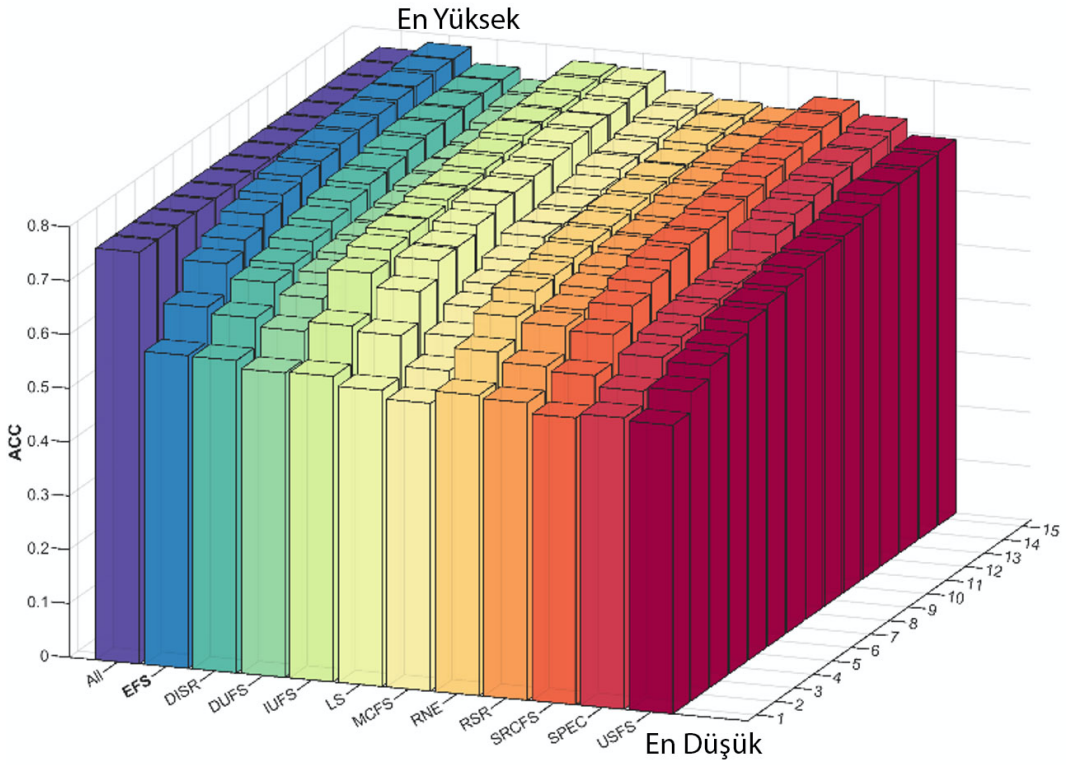
Şekil 5.3'te veri kümelerindeki özellik sayısı açısından minimum hata oranındaki değişim gösterilmektedir. Sonuçlardan, seçilen özellik sayısına bağlı olarak, hata oranı tüm giriş verilerinin hata oranına yakın ve global minimum hata oranından büyük olan d' özellik sayısını gözlemleyebiliriz. Bu bir alt sınırdır. Ayrıca, özelliklerin birleşiminden oluşan öngörülemeyen ilişkiler nedeniyle, optimum özellik sayısı için bir üst sınıra karar vermek zordur. Ancak alt sınırdan başlayarak belirli bir adımda (yani optimize edilmiş bir yinelemeli ileri arama) ilerleme yoluyla global bir hata oranı aranabilir. Bu nedenle, olası tüm alt kümeleri kontrol etmeye gerek yoktur. Sonuçlara göre bir alt sınırdan başlayarak birkaç adımda minimum hata oranlarına ulaşabiliriz. Başka bir deyişle, global minimuma ulaşmak

için çok sayıda özellik aramak çoğu zaman gerekli değildir. Üç yüksek boyutlu veri seti üzerindeki sonuçlar da bu durumu doğrulamaktadır. Ancak bu durumun tüm veri setlerine de genelleyemeyeceğinin altını çizmek gerekir. Bu nedenle birçok özelliğin (örn., $d-1$) analizinin yapılabileceğini belirtmek gerekir.



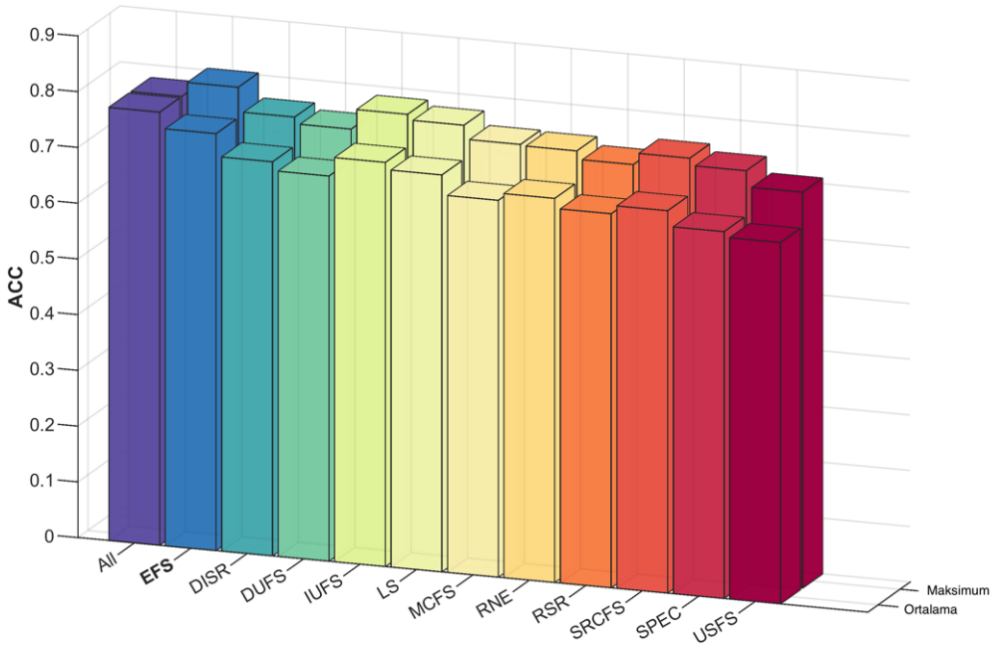
Şekil 5.3: Minimum hata oranının veri setlerindeki özellik sayısına göre değişimi (Yatay siyah kesikli çizgiler tüm girdi verisinin hata oranını göstermektedir. Dikey kırmızı kesikli çizgiler Denklem (13) ile elde edilen özelliklerin sayısını göstermektedir. Yatay eksen (x) seçilen özellik sayısını, dikey eksen (y) ise hata oranını ifade eder).

Şekil 5.4'te tüm veri setlerinde beş sınıflandırıcının ortalamasına göre gözetimsiz özellik seçim yöntemlerinin karşılaştırmalı sonuçları gösterilmektedir. Sonuçlara göre Önerilen Yöntem, IUFS ve LS, sırasıyla 0,783, 0,774 ve 0,771 ile istatistiksel olarak anlamlı en yüksek doğruluk oranına sahiptir. USFS, 0,695 ile en düşük doğruluk oranına sahiptir. Ek olarak EFS, IUFS ve LS, 0,765 doğruluk oranına sahip taban çizgisini (baseline) aşmaktadır. Son olarak, EFS, IUFS ve LS istatistiksel anlamlılık ile ortalama en yüksek doğruluk oranını sunar.



Şekil 5.4: Tüm veri setlerinde beş sınıflandırıcının ortalamasına göre Gözetimsiz Özellik Seçim yöntemlerinin karşılaştırmalı sonuçları.

Şekil 5.5'te ortalama doğruluk ve maksimum doğruluk oranları açısından tüm sınıflandırma deneylerinde gözetimsiz özellik seçim algoritmalarının ortalama sonuçları gösterilmektedir. Sonuçlardan Önerilen Yöntem, ortalama açısından 0,748 ile en yüksek doğruluk oranına ve maksimum açısından 0,803 ile en yüksek doğruluk oranına sahiptir. Tüm özellikler göz önüne alındığında ortalama doğruluk oranı 0,777'dir. İkinci en iyi sonuçlar, ortalama ve maksimum açısından 0,725 ve 0,783 ile IUFS'ye aittir. Son olarak, üçüncü en iyi sonuçlar ortalama ve maksimum açısından 0,712 ve 0,773 ile LS'ye aittir. EFS, IUFS ve LS'ye ait sonuçlar istatistiksel olarak diğerlerinden daha anlamlıdır.



Şekil 5.5: On iki veri seti üzerinde beş sınıflandırıcıya ait sonuçlar dikkate alınarak Gözetimsiz Özellik Seçim algoritmalarının Maksimum ve Ortalama doğruluk oranı açısından performansı.

Sonuçlar daha ayrıntılı olarak ele alındığında deneysel çalışmada kullanılan diğer gözetimsiz özellik seçim algoritmalarına karşı önerilen algoritmanın sayısal sonuçları Tablo 5.1, Tablo 5.2, Tablo 5.3, Tablo 5.4 ve Tablo 5.5'te gösterilmektedir.

Tablo 5.1'de KNN sınıflandırıcı açısından ortalama sınıflandırma doğrulukları gösterilmektedir. Önerilen algoritmanın 12 veri setinden 5'inde diğerlerine göre daha yüksek doğruluk oranına sahip olduğu görülmektedir.

Tablo 5.1: Önerilen yöntem ve diğer özellik seçim algoritmalarının ortalama sınıflandırma doğrulukları (KNN sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	**0,869**	0,853	0,855	0,792	0,867	0,831	0,858	0,856	0,854	0,800	0,867
2	0,878	0,861	0,852	**0,882**	0,869	0,865	0,866	0,853	0,878	0,865	0,860
3	**0,657**	0,561	0,599	0,594	0,494	0,650	0,641	0,491	0,545	0,568	0,631
4	**0,748**	0,695	0,736	0,737	0,727	0,555	0,697	0,702	0,680	0,571	0,727
5	**0,618**	0,594	0,597	0,600	0,614	0,573	0,608	0,605	0,606	0,573	0,591
6	**0,654**	0,626	0,322	0,591	0,636	0,321	0,265	0,468	0,264	0,534	0,291
7	0,766	**0,776**	0,755	0,765	0,758	0,635	0,775	0,426	0,768	0,426	0,426
8	0,632	0,526	0,499	0,580	0,722	0,517	0,499	0,504	0,759	**0,761**	0,498
9	0,720	0,702	0,689	0,714	0,734	0,712	0,702	0,688	0,691	**0,772**	0,705
10	0,737	**0,768**	0,745	0,713	0,704	0,724	0,765	0,766	0,530	0,485	0,724
11	0,631	**0,635**	0,615	0,554	0,572	0,569	0,608	0,625	0,628	0,520	0,598
12	0,845	0,837	0,847	0,880	**0,887**	0,858	0,872	0,847	0,859	0,870	0,856

Tablo 5.2'de NB sınıflandırıcısı açısından algoritmaların ortalama sınıflandırma doğrulukları gösterilmektedir. Bu tabloda da aynı şekilde önerilen yöntem ile 12 veri setinden 5'inde diğerlerine göre daha yüksek doğruluk oranına sahip sonuçlar elde edilmiştir.

Tablo 5.2: Önerilen yöntem ve diğer özellik seçim algoritmalarının ortalama sınıflandırma doğrulukları (NB sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	**0,823**	0,811	0,808	0,652	0,807	0,681	0,633	0,793	0,802	0,529	0,567
2	0,922	0,917	0,918	**0,929**	0,914	0,918	0,919	0,913	0,925	0,918	0,920
3	**0,694**	0,606	0,664	0,622	0,521	0,653	0,681	0,540	0,595	0,648	0,688
4	0,616	0,621	0,653	0,640	0,629	0,577	0,580	0,633	0,619	0,586	**0,667**
5	0,593	0,511	0,569	0,556	**0,602**	0,526	0,558	0,541	0,560	0,513	0,495
6	**0,810**	0,804	0,519	0,744	0,807	0,553	0,519	0,752	0,519	0,519	0,519
7	0,794	**0,805**	0,782	0,803	0,745	0,590	**0,805**	0,426	0,769	0,426	0,308
8	0,583	0,512	0,505	0,560	0,592	0,510	0,503	0,515	0,590	**0,594**	0,495
9	**0,769**	0,666	0,669	0,761	0,472	0,711	0,667	0,709	0,542	0,617	0,519
10	**0,748**	0,617	0,649	0,680	0,603	0,737	0,566	0,675	0,541	0,615	0,733
11	0,461	0,467	0,450	0,451	0,404	0,420	0,448	0,423	**0,468**	0,427	0,438
12	0,547	0,490	0,507	0,555	0,503	0,557	0,492	0,482	**0,559**	0,436	0,456

Tablo 5.3'te CART sınıflandırıcısı kullanılarak elde edilen ortalama sınıflandırma doğrulukları gösterilmektedir. Bu tabloya göre önerilen algoritmanın 12 veri setinden 3'ünde diğerlerine göre daha yüksek doğruluk oranına sahip olduğu görülmektedir.

Tablo 5.3: Önerilen yöntem ve diğer özellik seçim algoritmalarının ortalama sınıflandırma doğrulukları (CART sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,867	0,871	0,860	0,846	**0,877**	0,847	0,871	0,870	0,868	0,838	0,867
2	**0,894**	0,867	0,860	0,888	0,869	0,872	0,872	0,865	0,884	0,876	0,872
3	0,708	0,616	**0,720**	0,669	0,567	0,628	0,696	0,582	0,568	0,644	0,665
4	0,648	0,639	0,668	0,671	0,650	0,559	0,620	0,653	0,626	0,571	**0,682**
5	0,613	0,592	0,597	0,601	0,619	0,573	**0,631**	0,593	0,620	0,581	0,582
6	**0,842**	0,837	0,539	0,796	0,834	0,601	0,516	0,782	0,516	0,692	0,515
7	0,776	**0,784**	0,756	0,768	0,767	0,651	0,770	0,426	0,775	0,426	0,426
8	0,655	0,507	0,498	0,583	0,676	0,508	0,500	0,495	0,715	**0,717**	0,499
9	**0,878**	0,706	0,702	0,854	0,636	0,840	0,730	0,733	0,679	0,762	0,649
10	0,766	0,770	0,760	0,770	0,777	0,776	0,772	**0,788**	0,708	0,715	0,775
11	0,657	**0,663**	0,662	0,654	0,608	0,642	0,653	0,650	0,651	0,596	0,660
12	0,890	0,844	0,871	**0,929**	0,926	0,903	0,903	0,863	0,907	0,895	0,883

Tablo 5.4'te SVM sınıflandırıcı kullanılarak algoritmalar tarafından elde edilen ortalama sınıflandırma doğrulukları gösterilmektedir. Bu tabloda önerilen algoritmanın 12 veri setinden 3'ünde diğerlerine göre daha yüksek doğruluk oranına sahip olduğu görülmektedir.

Tablo 5.4: Önerilen yöntem ve diğer özellik seçim algoritmalarının ortalama sınıflandırma doğrulukları (SVM sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,848	0,839	0,828	0,835	0,847	**0,852**	0,851	0,833	0,831	0,841	0,833
2	0,923	0,921	0,921	**0,934**	0,915	0,918	0,920	0,917	0,926	0,917	0,925
3	0,727	0,624	**0,737**	0,602	0,623	0,648	0,732	0,606	0,628	0,666	0,697
4	0,666	0,616	**0,684**	0,657	0,664	0,617	0,580	0,650	0,632	0,646	0,677
5	0,686	0,642	0,669	0,642	**0,687**	0,608	0,683	0,658	0,669	0,612	0,631
6	**0,842**	0,835	0,541	0,802	0,836	0,598	0,519	0,776	0,519	0,692	0,519
7	0,829	**0,834**	0,815	0,823	0,812	0,699	0,831	0,426	0,821	0,426	0,426
8	0,595	0,518	0,505	0,569	0,598	0,515	0,489	0,513	**0,602**	0,601	0,496
9	**0,828**	0,600	0,598	0,794	0,603	0,700	0,697	0,714	0,652	0,689	0,646
10	0,793	0,782	0,783	0,782	0,767	0,801	**0,806**	0,797	0,712	0,698	0,794
11	**0,689**	0,667	0,674	0,674	0,587	0,617	0,670	0,650	0,638	0,607	0,651
12	0,568	0,525	0,552	**0,622**	0,612	0,571	0,576	0,558	0,584	0,587	0,589

Son olarak, Tablo 5.5'te algoritmaların RF sınıflandırıcı açısından ortalama sınıflandırma doğrulukları gösterilmektedir. Bu tabloda önerilen algoritmanın 12 veri setinden 4'ünde diğerlerine göre daha yüksek doğruluk oranına sahip olduğu görülmektedir.

Tablo 5.5: Önerilen yöntem ve diğer özellik seçim algoritmalarının ortalama sınıflandırma doğrulukları (RF sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,897	0,890	0,883	0,879	**0,899**	0,874	0,896	0,897	0,886	0,861	0,889
2	**0,917**	0,909	0,908	0,915	0,903	0,910	0,909	0,905	0,914	0,906	0,910
3	**0,741**	0,629	0,726	0,665	0,575	0,688	0,716	0,576	0,578	0,665	0,673
4	0,717	0,698	0,731	**0,740**	0,723	0,613	0,684	0,694	0,679	0,618	0,728
5	0,663	0,619	0,643	0,648	**0,671**	0,600	0,667	0,634	0,657	0,609	0,622
6	**0,847**	0,841	0,554	0,802	0,837	0,606	0,524	0,786	0,524	0,690	0,532
7	0,808	**0,826**	0,802	0,815	0,806	0,696	0,814	0,426	0,810	0,426	0,426
8	0,706	0,515	0,498	0,618	0,738	0,510	0,503	0,497	0,774	**0,780**	0,496
9	0,866	0,720	0,705	**0,877**	0,648	0,846	0,780	0,779	0,719	0,770	0,654
10	0,807	0,812	0,799	0,812	0,813	0,814	0,818	**0,824**	0,714	0,727	0,814
11	**0,699**	**0,699**	0,696	0,686	0,647	0,674	0,691	0,686	0,689	0,627	0,691
12	0,912	0,887	0,901	**0,937**	0,935	0,918	0,924	0,898	0,922	0,916	0,910

Beş tablodaki tüm sonuçlar göz önüne alındığında önerilen yöntem 60 deneyden 20'sinde en yüksek ortalama doğruluğu sağlarken, DISR 60 deneyden 9'unda en yüksek ortalama doğruluğa sahiptir. Özetlemek gerekirse tüm sınıflandırıcılar üzerinde en çok ortalama doğruluklar önerilen yöntem ile elde edilmiştir.

Sonuçlar maksimum sınıflandırma doğruluğu açısından daha ayrıntılı olarak ele alındığında, önerilen algoritmanın deneysel çalışmada kullanılan diğer gözetimsiz özellik seçim algoritmalarına karşı sonuçları Tablo 5.6, Tablo 5.7, Tablo 5.8, Tablo 5.9 ve Tablo 5.10'da gösterilmektedir.

Tablo 5.6, KNN sınıflandırıcı açısından maksimum sınıflandırma doğruluklarını içerir. Bu tabloda önerilen algoritmanın ve SPEC'in 12 veri setinden 3'ünde en yüksek maksimum sınıflandırma doğruluğuna ulaştığı görülmektedir.

Tablo 5.6: Önerilen algoritma ve diğer özellik seçim algoritmalarının maksimum sınıflandırma doğrulukları (KNN sınıflandırıcı).

Veri seti	**Algoritma**										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,902	0,883	**0,919**	0,858	0,902	0,875	0,894	0,886	0,903	0,872	0,916
2	0,909	0,889	0,910	**0,911**	0,891	0,898	0,892	0,882	0,905	0,885	0,892
3	0,765	0,623	0,706	0,674	0,534	**0,789**	0,729	0,595	0,650	0,677	0,752
4	**0,857**	0,806	0,792	0,826	0,782	0,629	0,830	0,769	0,774	0,600	0,782
5	0,660	**0,675**	0,632	0,645	0,653	0,665	0,652	0,644	0,668	0,636	0,670
6	**0,821**	0,777	0,530	0,743	0,815	0,452	0,369	0,731	0,368	0,745	0,415
7	0,809	0,817	0,815	0,814	0,815	0,749	**0,820**	0,426	0,818	0,426	0,426
8	0,700	0,572	0,510	0,590	0,823	0,551	0,521	0,533	0,865	**0,867**	0,532
9	0,809	0,737	0,766	0,791	0,798	0,766	0,764	0,741	0,787	**0,838**	0,779
10	0,776	**0,809**	0,798	0,758	0,783	0,788	0,806	0,806	0,746	0,754	0,807
11	**0,727**	0,704	0,678	0,639	0,650	0,668	0,673	0,694	0,709	0,640	0,686
12	0,924	0,888	0,896	0,935	0,929	0,928	0,923	0,891	0,934	**0,937**	0,924

Tablo 5.7, NB sınıflandırıcı açısından maksimum sınıflandırma doğruluklarını göstermektedir. Bu tabloda önerilen algoritmanın ve DISR'nin 12 veri setinden 2'sinde en yüksek maksimum sınıflandırma doğruluğuna ulaştığı görülmektedir. Ayrıca sonuçlara göre IUFS 12 veri setinden 5'inde, USFS ise 3'ünde en yüksek maksimum sınıflandırma doğruluğuna ulaşmıştır.

Tablo 5.7: Önerilen algoritma ve diğer özellik seçim algoritmalarının maksimum sınıflandırma doğrulukları (NB sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,845	**0,850**	0,829	0,796	0,838	0,808	0,794	0,829	0,821	0,756	0,758
2	0,939	0,935	0,939	0,949	0,923	0,949	0,947	0,934	0,946	0,943	**0,950**
3	0,766	0,635	0,726	0,645	0,645	0,697	0,708	0,645	0,645	0,706	**0,789**
4	0,655	0,663	0,679	**0,736**	0,681	0,620	0,620	0,690	0,636	0,632	0,694
5	0,610	0,543	0,607	**0,644**	0,626	0,553	0,604	0,573	0,597	0,568	0,559
6	**0,865**	**0,865**	0,520	0,826	0,861	0,606	0,519	0,800	0,519	0,519	0,519
7	**0,848**	0,840	0,847	**0,848**	0,841	0,769	0,842	0,426	0,843	0,426	0,426
8	0,615	0,548	0,515	0,571	**0,619**	0,543	0,511	0,546	0,609	0,616	0,536
9	0,791	0,684	0,686	**0,820**	0,597	0,808	0,707	0,799	0,602	0,752	0,613
10	0,763	0,734	0,720	0,737	0,696	0,767	0,728	0,745	0,667	0,692	**0,784**
11	0,512	0,512	0,511	0,505	0,428	0,468	0,482	**0,526**	0,515	0,462	0,468
12	0,599	0,536	0,591	**0,622**	0,573	0,619	0,574	0,533	0,598	0,502	0,522

Tablo 5.8'de CART sınıflandırıcısına göre özellik seçme algoritmalarının maksimum sınıflandırma doğrulukları gösterilmektedir. Bu tabloda önerilen algoritmanın ve LS'nin 12 veri setinden 2'sinde en yüksek maksimum sınıflandırma doğruluğunu elde ettiği görülmektedir. Ek olarak bu tabloda RSR ve SRCFS 12 veri setinden 3'ünde en yüksek maksimum sınıflandırma doğruluğuna sahiptir.

Tablo 5.8: Önerilen algoritma ve diğer özellik seçim algoritmalarının maksimum sınıflandırma doğrulukları (CART sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,914	0,895	0,920	0,898	**0,925**	0,891	0,917	0,917	**0,925**	0,913	0,904
2	0,910	0,906	0,908	0,913	0,890	0,914	0,912	0,906	0,911	0,906	**0,915**
3	**0,808**	0,653	0,800	0,752	0,645	0,739	0,776	0,656	0,645	0,744	0,774
4	0,725	0,703	0,688	0,747	0,681	0,606	0,681	**0,749**	0,666	0,608	0,731
5	0,652	0,642	0,630	0,622	0,654	0,628	0,693	0,623	**0,694**	0,617	0,633
6	**0,894**	0,888	0,626	0,839	0,892	0,637	0,523	0,845	0,521	0,835	0,521
7	0,817	0,818	0,812	0,810	0,818	0,776	**0,820**	0,426	0,818	0,426	0,426
8	0,713	0,542	0,514	0,596	0,758	0,541	0,511	0,519	**0,814**	0,810	0,534
9	0,909	0,738	0,756	0,908	0,684	**0,932**	0,838	0,808	0,807	0,862	0,754
10	0,810	0,806	0,811	0,802	0,820	0,810	0,815	**0,828**	0,805	0,794	0,805
11	0,704	0,710	0,707	0,707	0,706	0,702	0,699	**0,720**	0,717	0,685	0,704
12	0,951	0,896	0,959	0,991	**0,995**	0,972	0,994	0,915	0,966	0,994	0,963

Tablo 5.9, SVM sınıflandırıcı kullanılarak algoritmalar tarafından elde edilen maksimum sınıflandırma doğruluklarını içermektedir. Bu tabloda önerilen algoritmanın 12 veri setinden 4'ünde diğerlerine göre daha yüksek maksimum doğruluk elde ettiği görülmektedir.

Tablo 5.9: Önerilen algoritma ve diğer özellik seçim algoritmalarının maksimum sınıflandırma doğrulukları (SVM sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	**0,895**	0,877	0,887	0,890	0,888	0,885	0,890	0,879	0,883	0,884	0,867
2	0,952	0,950	0,949	**0,961**	0,922	0,959	0,944	0,941	0,951	0,950	0,958
3	0,798	0,663	**0,824**	0,694	0,705	0,713	0,811	0,756	0,660	0,805	0,763
4	**0,769**	0,671	0,707	0,762	0,697	0,704	0,683	0,724	0,671	0,704	0,702
5	0,724	**0,729**	0,726	0,706	0,723	0,727	0,723	0,714	0,728	0,724	0,727
6	**0,893**	0,882	0,633	0,836	0,887	0,629	0,519	0,842	0,519	0,830	0,519
7	0,869	**0,874**	0,870	0,869	0,871	0,818	0,873	0,426	0,871	0,426	0,426
8	0,618	0,554	0,515	0,574	0,619	0,547	0,498	0,537	0,619	**0,621**	0,545
9	0,868	0,626	0,598	0,856	0,672	0,845	0,805	**0,885**	0,782	0,793	0,810
10	**0,858**	0,849	0,834	0,822	0,845	0,855	0,857	0,845	0,817	0,813	0,844
11	0,761	**0,794**	0,777	**0,794**	0,774	0,790	0,793	0,765	0,774	0,741	0,793
12	0,640	0,602	0,647	0,728	0,709	0,669	0,673	0,635	0,669	**0,733**	0,715

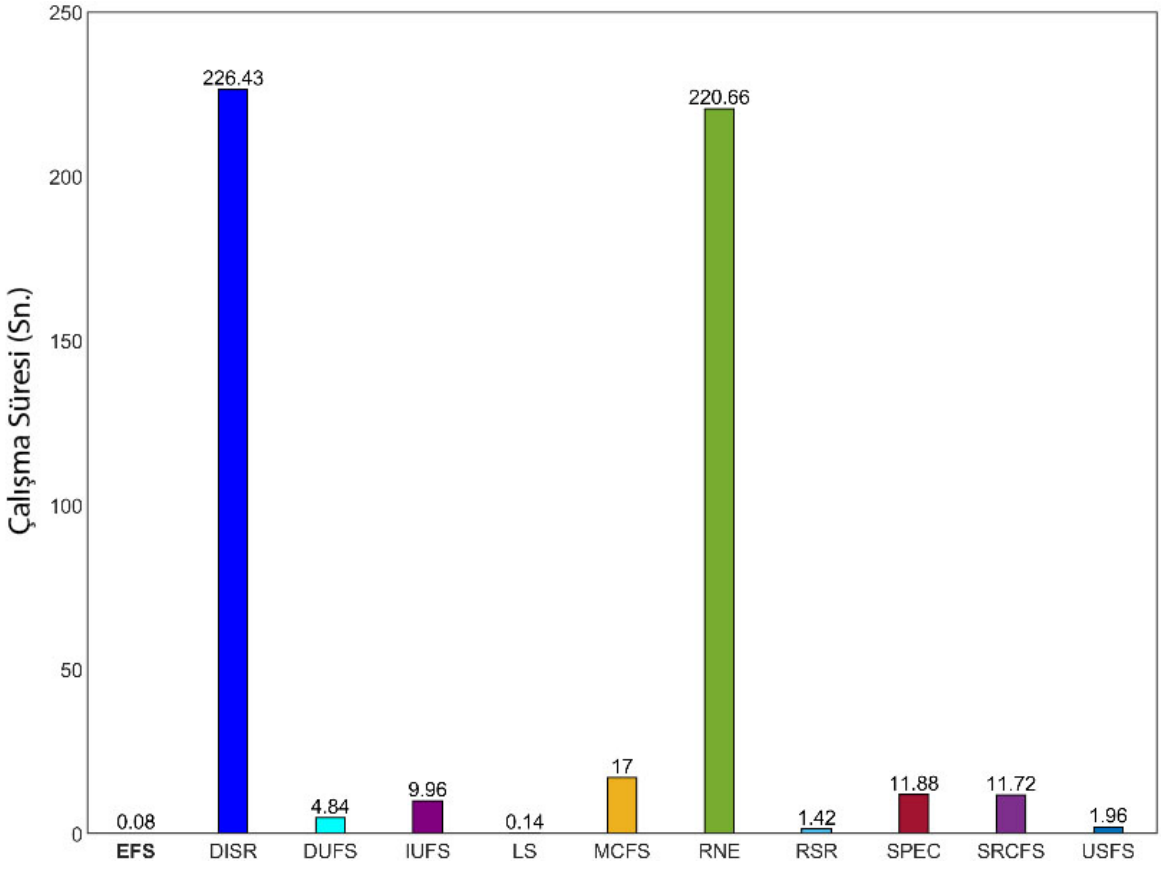
Son olarak, Tablo 5.10 algoritmaların RF sınıflandırıcı açısından maksimum sınıflandırma doğruluklarını göstermektedir. İlgili tabloya göre Önerilen Algoritma, LS, RNE ve SRCFS'nin 12 veri setinden 2'sinde diğer algoritmalardan daha yüksek maksimum sınıflandırma doğruluğuna sahip olduğu görülmektedir. Ek olarak bu tabloya göre IUFS 12 veri setinden 3'ünde en yüksek maksimum sınıflandırma doğruluğuna sahiptir.

Tablo 5.10: Önerilen algoritma ve diğer özellik seçim algoritmalarının maksimum sınıflandırma doğrulukları (RF sınıflandırıcı).

Veri seti	Algoritma										
	EFS	DISR	DUFS	IUFS	LS	MCFS	RNE	RSR	SRCFS	SPEC	USFS
1	0,940	0,922	0,942	0,929	0,945	0,923	**0,947**	0,942	0,941	0,941	0,935
2	0,932	0,925	0,927	**0,936**	0,918	0,928	0,930	0,927	0,931	0,931	0,931
3	**0,863**	0,742	0,798	0,742	0,645	0,798	0,815	0,734	0,645	0,855	0,806
4	0,829	0,791	0,764	**0,837**	0,767	0,709	0,788	0,788	0,767	0,702	0,786
5	0,708	0,682	0,695	0,679	**0,710**	0,683	0,704	0,680	0,703	0,684	0,700
6	**0,904**	0,891	0,663	0,845	0,902	0,659	0,553	0,862	0,548	0,844	0,576
7	0,865	**0,882**	0,875	0,878	0,881	0,820	0,872	0,426	0,875	0,426	0,426
8	0,776	0,567	0,525	0,639	0,844	0,556	0,516	0,533	**0,889**	0,887	0,544
9	0,897	0,770	0,816	**0,943**	0,707	**0,943**	0,874	0,902	0,874	0,879	0,764
10	0,858	0,862	0,864	0,845	0,858	0,855	**0,869**	**0,869**	0,833	0,805	0,855
11	0,746	0,754	0,749	0,752	0,754	0,744	0,750	0,745	**0,762**	0,737	0,754
12	0,971	0,939	0,975	0,995	**0,997**	0,982	0,996	0,951	0,980	0,996	0,983

Beş tablodaki tüm sonuçlar göz önüne alındığında önerilen yöntem 60 deneyden 13'ünde, IUFS ise 11'inde en yüksek maksimum doğruluğa sahiptir. Sonuç olarak önerilen yöntem ile tüm sınıflandırıcılar üzerinde en çok maksimum sınıflandırma doğruluğuna ulaşılmıştır.

Şekil 5.6'da çalışma süresi açısından Gözetimsiz Özellik Seçim Yöntemlerinin karşılaştırmalı sonuçları gösterilmektedir. Sonuçlara göre ortalama çalışma süreleri açısından Önerilen Yöntem ve LS ortalama çalışma süreleri 1 saniyenin altında olan yöntemlerdir. Diğer bir deyişle deneysel çalışmada kullanılan diğer yöntemlere göre en hızlı gözetimsiz özellik seçim yöntemleridir. DISR ve RNE'nin tüm yüksek boyutlu veri kümelerinde yavaş çalıştığı, SRCFS ve SPEC'in büyük miktarda veriye sahip veri kümelerinde performansının düşük olduğu gözlemlenmiş. Öte yandan MCFS, RSR, IUFS, DUFS ve USFS çalışma süreleri açısından iyi performans sergilemişlerdir.



Şekil 5.6: Ortalama çalışma süresi açısından Gözetimsiz Özellik Seçim Yöntemlerinin karşılaştırmalı sonuçları.

Sonuç olarak, önerilen yöntem deneysel çalışmaya göre çalışma süreleri açısından en hızlı gözetimsiz özellik seçim yöntemi olmuştur. LS ise ikinci sırada yer almaktadır. Önerilen Yöntem (EFS), hem doğruluk oranına bakıldığında hem de çalışma süresi açısından, diğer yöntemlere göre daha yüksek başarı sağlamıştır.

6. TARTIŞMA ve ÖNERİLER

Bu tez çalışmasında özellik seçimi için tek değişkenli ve filtre yaklaşımına dayanan EFS adlı yeni bir gözetimsiz özellik seçim yöntemi önerilmektedir. Önerilen yöntem ile dağılımın kümülatif entropisi hesaplanarak özellikler sıralanmış daha sonra dağılımın simetrisi tarafından hesaplanan Shannon entropisi bulunarak özellikler sıralanmıştır. Son olarak bu iki sıralamanın ortalaması alınarak özelliklerin önem sırası bulunmuştur. Algoritmanın performansını ölçmek için farklı teknikleri kullanan tek değişkenli 10 gözetimsiz özellik seçim yöntemi ile veri setleri üzerinde özellik seçim işlemleri gerçekleştirilmiştir. Seçilen özellikler iyi bilinen 5 farklı sınıflandırıcıya gönderilmiş ve doğruluk oranları ölçülmüştür. Elde edilen doğruluk oranlarından maksimum ve ortalama doğruluk değerleri karşılaştırılmış. Yapılan deneysel çalışmada veri setlerinden önerilen yöntem ile belirlenen özelikler ile yapılan sınıflandırma sonuçlarında 60 deneyden 20'sinde en yüksek ortalama doğruluk oranları alınmıştır. Onu 9 en yüksek ortalama doğruluk oranı ile DISR takip etmektedir.

Random Forest (RF), Classification and Regression Trees (CART), Support Vector Machine (SVM), k-Nearest Neighbors (KNN) ve Naive Bayes (NB) makine öğrenimi algoritmaları kullanılarak gerçekleştirilen tüm deneylerde, önerilen EFS algoritmasının yüksek performanslar sergilediğini görülmektedir.

Yapılan deneysel çalışmada önerilen yöntemin zaman karmaşıklığının oldukça düşük olduğu bu nedenle yüksek boyutlu veri kümelerinde özellik seçimini hızla gerçekleştirebildiği gözlemlenmektedir.

Önerilen yöntemin ortalama çalışma süresinin 0,08 saniye ile deneylerde kullanılan diğer gözetimsiz özellik seçim yöntemlerine göre daha hızlı olduğu gözlemlenmiştir. Onu 0,14 saniye ortalama çalışma süresi ile LS algoritması takip etmektedir. Bu düşük çalışma süreleri yöntemin yüksek boyutlu veri setlerinde önemli derece hızlı çalıştığını göstermektedir.

EFS yönteminin çalışması için herhangi bir parametreye ihtiyaç duymaması uygulanmasını kolaylaştırmaktadır.

Önerilen yöntemin deneylerde kullanılan veri setleri üzerinde öne çıkan katkılar ise şu şekildedir:

- Önerilen yöntemin deneysel çalışmada kullanılan diğer özellik seçim yöntemlerine göre daha hızlı çalıştığı gözlemlenmiştir.
- Önerilen yöntem, sınıflandırma algoritmaları ve veri seti etki alanlarından bağımsız olarak, seçilen özellik sayısı artarken çoğunlukla ortalama olarak en yüksek doğruluk oranını sağlamayı sürdürmektedir.
- Önerilen yöntemin çalışması için herhangi bir parametreye ihtiyacı yoktur.
- Uygulanması kolay bir yöntemdir.

Yapay veri setlerinde ve farklı alanlardan on bir gerçek dünya veri setinde yapılan testlerde, algoritmanın yüksek doğruluk oranları sağladığı görülmüştür. Özellikle, EFS'nin öne çıkan özelliği, özellik sayısı arttığında bile ortalama ve maksimum doğruluk oranlarında yüksek başarı sağlamaya devam etmesidir. Bu durum, algoritmanın geniş bir uygulama yelpazesine sahip olduğunu ve çeşitli veri setlerinde iyi sonuçlar elde etme yeteneğini vurgulamaktadır.

Bu tez çalışmasında EFS algoritmasının etkinliğini göstererek önemli bir başlangıç sağlanmaktadır. Gelecekteki yapılacak çalışmalar ile algoritmanın daha geniş bir veri yelpazesinde ve farklı uygulama alanlarında nasıl performans gösterdiği daha ayrıntılı bir şekilde incelenebilir. Ayrıca, EFS'nin diğer özellik seçim yöntemleriyle karşılaştırmalı analizlerinin yapılarak sonuçların incelenmesi, algoritmanın rekabet avantajlarını daha iyi anlaşılmasına yardımcı olabilir. Bu bağlamda, EFS'nin kullandığı veri türleri, boyutları ve özellikleri üzerine yapılacak derinlemesine analizler, algoritmanın kullanım sınırlamalarını çok daha iyi anlamak ve geliştirmek için değerli bilgiler sağlayabilir.

7. KAYNAKLAR

[1] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review. Data classification: Algorithms and applications, Encyclopedia of Machine Learning and Data Mining, 2016, http://dx.doi.org/10.1007/978-1-4899-7502-7_101-1.

[2] K. Arslan, Eğitimde Yapay Zeka ve Uygulamaları. Batı Anadolu Eğitim Bilimleri Dergisi, 2017, 11(1), 71-88.

[3] T. Erbayram, M. Erişoğlu, Yeni bir özellik seçim yöntemi ve özellik seçim yöntemlerinin sınıflama performanslarının karşılaştırılması, Nicel Bilimler Dergisi, 2021, 3(1), 72-90, https://doi.org/10.51541/nicel.909876.

[4] G. Reddy, M. Reddy, K. Lakshmanna, vd., Analysis of dimensionality reduction techniques on big data, IEEE, Volume: 8, 2020, s. 54776 - 54788.

[5] F. Anowar, S. Sadaoui, B. Selim, Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE), Computer Science Review, 2021, 1-10.

[6] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, Data classification: Algorithms and applications, 2014, 37.

[7] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, J. F. Martínez-Trinidad, A systematic evaluation of filter Unsupervised Feature Selection methods, Expert Syst Appl 162:113745, 2020, https://doi.org/10.1016/j.eswa.2020.113745.

[8] Z. A. Zhao, H. Liu, Spectral Feature Selection for Data Mining. Chapman and Hall/CRC, 2011.

[9] S.K. PAL, P. Mitra, Pattern Recognition Algorithms for Data Mining, 1st. Ed. Chapman & Hall/CRC, 2004.

[10] A. Di Crescenzo, M. Longobardi, On cumulative entropies, J Stat Plan Inference 139:4072–4087, 2009, https://doi.org/10.1016/j.jspi.2009.05.038.

[11] C. E. Shannon, A Mathematical Theory of Communication. Bell Syst Tech J 27:379–423, 1948, https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[12] Yapay Zekâ Nedir? https://aix.web.tr/yapay-zeka-nedir/ [Online] Erişim tarihi: 16 Aralık 2023.

[13] G. Scalia, Machine Learning for Scientific Data Analysis, In: Piroddi, L. (eds) Special Topics in Information Technology. SpringerBriefs in Applied Sciences and Technology, Springer, Cham, 2022, https://doi.org/10.1007/978-3-030-85918-3_10.

[14] G. N. College, vd., A review on machine learning, International Journal of Science and Research Archive, 2023, 09(01), 281–285 https://doi.org/10.30574/ijsra.2023.9.1.0410.

[15] H. Liu and H. Motoda, editors. Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer Academic Publishers, 1998, Second Printing, 2001.

[16] M. Alimoussa, A. Porebski, N. Vandenbroucke, R. Thami, S. El Fkihi, Clustering-based sequential feature selection approach for high dimensional data classification, In: Proceedings of the 16th international joint conference on computer vision, imaging and computer graphics theory and applications, Science and Technology Publications, 122–132, 2021, https://doi.org/10.5220/0010259501220132.

[17] L. Ladha, T. Deepa, Feature Selection Methods And Algorithms, International Journal on Computer Science and Engineering, 2011, 3(5), 1787-1797.

[18] S. Wang, J. Tang, H. Liu, Feature Selection. In: Sammut, C., Webb, G. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA, 2016, https://doi.org/10.1007/978-1-4899-7502-7_101-1.

[19] W. Zhang, F. Zhu, Y. Lv, C. Tan, W. Liu, X. Zhang, F. Wang, AdapGL: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks, Transportation Research Part C: Emerging Technologies, Volume 139, 2022, 103659, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2022.103659.

[20] M. G. Parsa, H. Zare, M. Ghatee, Unsupervised feature selection based on adaptive similarity learning and subspace clustering. Eng Appl Artif Intell 95:103855, 2020, https://doi.org/10.1016/j.engappai.2020.103855.

[21] D. Tomar, Y. Prasad, M. K. Thakur and K. K. Biswas, "Feature Selection Using Autoencoders," 2017 International Conference on Machine Learning and Data Science (MLDS), Noida, India, 2017, pp. 56-60, https://doi.org/10.1109/MLDS.2017.20.

[22] N. J. Martarelli, & M. S. Nagano, Unsupervised feature selection based on bio-inspired approaches. Swarm and Evolutionary Computation, 52, 100618, 2020.

[23] D. M. Witten, & R. Tibshirani, A framework for feature selection in clustering. Journal of the American Statistical Association, 105(490), 713-726, 2010.

[24] E. Hancer, B. Xue, and M. Zhang, “Differential evolution for filter feature selection based on information theory and feature ranking,” Knowledge-Based Systems, vol. 140, pp. 103–119, Jan. 2018, doi: https://doi.org/10.1016/j.knosys.2017.10.028.

[25] W. Fan, H. Sallay, Nizar Bouguila, and Sami Bourouis, “A hierarchical Dirichlet process mixture of generalized Dirichlet distributions for feature selection,” Computers & Electrical Engineering, vol. 43, pp. 48–65, Apr. 2015, doi: https://doi.org/10.1016/j.compeleceng.2015.03.018.

[26] J. Tang, X. Hu, H. Gao, and H. Liu, “Discriminant Analysis for Unsupervised Feature Selection,” Apr. 2014, doi: https://doi.org/10.1137/1.9781611973440.107.

[27] M.-Y. Zhai, R.-H. Yu, S.-F. Zhang, and J.-H. Zhai, “Feature selection based on extreme learning machine,” Jul. 2012, doi: https://doi.org/10.1109/icmlc.2012.6358904.

[28] Yassine Akhiat, M. Chahhou, and A. Zinedine, “Feature Selection Based on Graph Representation,” 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Oct. 2018, doi: https://doi.org/10.1109/cist.2018.8596467.

[29] S. Nagpal, S. Arora, S. Dey, and None Shreya, “Feature Selection using Gravitational Search Algorithm for Biomedical Data,” Procedia Computer Science, vol. 115, pp. 258–265, Jan. 2017, doi: https://doi.org/10.1016/j.procs.2017.09.133.

[30] S. Adams, P. A. Beling and R. Cogill, "Feature Selection for Hidden Markov Models and Hidden Semi-Markov Models," in IEEE Access, vol. 4, pp. 1642-1657, 2016, https://doi.org/10.1109/ACCESS.2016.2552478.

[31] T. Wang, Z. Hu, and H. Liu, “A unified view of feature selection based on Hilbert-Schmidt independence criterion,” Chemometrics and Intelligent Laboratory Systems, vol. 236, pp. 104807–104807, May 2023, https://doi.org/10.1016/j.chemolab.2023.104807.

[32] M. Labbé, L. I. Martínez-Merino, and A. M. Rodríguez-Chía, “Mixed integer linear programming for feature selection in support vector machine,” Discrete Applied Mathematics, vol. 261, pp. 276–304, May 2019, https://doi.org/10.1016/j.dam.2018.10.025.

[33] A. Ivanov and G. Riccardi, "Kolmogorov-Smirnov test for feature selection in emotion recognition from speech," 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 5125-5128, https://doi.org/10.1109/ICASSP.2012.6289074.

[34] in KNN, “Features selection in KNN,” Data Science Stack Exchange, Dec. 04, 2018. https://datascience.stackexchange.com/questions/42121/features-selection-in-knn, [Online] Erişim tarihi: 14 Mart 2023.

[35] X. He, D. Cai, and P. Niyogi, “Laplacian Score for Feature Selection,” Advances in Neural Information Processing Systems, vol. 18, 2024.

[36] C. Tang vd., “Unsupervised feature selection via latent representation learning and manifold regularization,” Neural Networks, vol. 117, pp. 163–178, Sep. 2019, https://doi.org/10.1016/j.neunet.2019.04.015.

[37] H. Zhao, L. Du, J. Wei and Y. Fan, "Local Sensitive Dual Concept Factorization for Unsupervised Feature Selection," in IEEE Access, vol. 8, pp. 133128-133143, 2020, https://doi.org/10.1109/ACCESS.2020.3010862.

[38] X. Liu, L. Wang, J. Zhang, J. Yin and H. Liu, "Global and Local Structure Preservation for Feature Selection," in IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 6, pp. 1083-1095, June 2014, https://doi.org/10.1109/TNNLS.2013.2287275.

[39] J. Zhu, J. Chen, B. Xu, H. Yang, and F. Nie, “Fast orthogonal locality-preserving projections for unsupervised feature selection,” Neurocomputing, vol. 531, pp. 100–113, Apr. 2023, https://doi.org/10.1016/j.neucom.2023.02.021.

[40] X. Chen, R. Chen, Q. Wu, F. Nie, M. Yang and R. Mao, "Semisupervised Feature Selection via Structured Manifold Learning," in IEEE Transactions on Cybernetics, vol. 52, no. 7, pp. 5756-5766, July 2022, https://doi.org/10.1109/TCYB.2021.3052847.

[41] M. Qi, T. Wang, F. Liu, B. Zhang, J. Wang, and Y. Yi, “Unsupervised feature selection by regularized matrix factorization,” Neurocomputing, vol. 273, pp. 593–610, Jan. 2018, https://doi.org/10.1016/j.neucom.2017.08.047.

[42] R. Ge vd., “McTwo: a two-step feature selection algorithm based on maximal information coefficient,” BMC Bioinformatics, vol. 17, no. 1, Mar. 2016, https://doi.org/10.1186/s12859-016-0990-0.

[43] P. Agrawal, H. F. Abutarboush, T. Ganesh and A. W. Mohamed, "Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)," in IEEE Access, vol. 9, pp. 26766-26791, 2021, https://doi.org/10.1109/ACCESS.2021.3056407.

[44] T. Zhang et al., "Sleep Staging Using Plausibility Score: A Novel Feature Selection Method Based on Metric Learning," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 2, pp. 577-590, Feb. 2021, https://doi.org/10.1109/JBHI.2020.2993644.

[45] J. R. Vergara and P. A. Estévez, “A review of feature selection methods based on mutual information,” Neural Computing and Applications, vol. 24, no. 1, pp. 175–186, Mar. 2013, https://doi.org/10.1007/s00521-013-1368-0.

[46] W. Fan and Nizar Bouguila, “Spherical data clustering and feature selection through nonparametric Bayesian mixture models with von Mises distributions,” Engineering Applications of Artificial Intelligence, vol. 94, pp. 103781–103781, Sep. 2020, https://doi.org/10.1016/j.engappai.2020.103781.

[47] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, “An improved particle swarm optimization for feature selection,” Journal of Bionic Engineering, vol. 8, no. 2, pp. 191–200, Jun. 2011, https://doi.org/10.1016/s1672-6529(11)60020-6.

[48] F. Song, Z. Guo and D. Mei, "Feature Selection Using Principal Component Analysis," 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, Yichang, China, 2010, pp. 27-30, https://doi.org/10.1109/ICSEM.2010.14.

[49] “Linear regression–based feature selection for microarray data classification,” International Journal of Data Mining and Bioinformatics, 2015, https://www.inderscienceonline.com/doi/abs/10.1504/IJDMB.2015.066776 , [Online] Erişim tarihi: 13 Şubat 2023.

[50] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S. C. K. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit 48:438–446, 2020, https://doi.org/10.1016/j.patcog.2014.08.006.

[51] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining – KDD ’10. ACM Press, New York, New York, USA, p 333, 2010.

[52] T. Chen, Y. Guo, S. Hao, Unsupervised feature selection based on joint spectral learning and general sparse regression, Neural Comput Appl 32:6581–6589, 2020.

[53] G. A. Giraldi, Paulo Sergio Rodrigues, E. C. Kitani, João Ricardo Sato, and C. E. Thomaz, “Statistical learning approaches for discriminant features selection,” Journal of the Brazilian Computer Society, vol. 14, no. 2, pp. 7–22, Jun. 2008, https://doi.org/10.1007/bf03192556.

[54] R. Shang, K. Xu, L. Jiao, Subspace learning for unsupervised feature selection via adaptive structure learning and rank approximation, Neurocomputing 413:72–84, 2020, https://doi.org/10.1016/j.neucom.2020.06.111.

[55] X. Lin, C. Li, W. Ren, X. Luo, and Y. Qi, “A new feature selection method based on symmetrical uncertainty and interaction gain,” Computational Biology and Chemistry, vol. 83, pp. 107149–107149, Dec. 2019, https://doi.org/10.1016/j.compbiolchem.2019.107149.

[56] M. R. Sikonja and I. Kononenko, Theoretical and empirical analysis of Relief and ReliefF. Machine Learning, 53:23, 69, 2003, https://doi.org/10.1023/A:1025667309714.

[57] Z. Xu, I. King, M. R. -T. Lyu and R. Jin, "Discriminative Semi-Supervised Feature Selection Via Manifold Regularization," in IEEE Transactions on Neural Networks, vol. 21, no. 7, pp. 1033-1047, July 2010, https://doi.org/10.1109/TNN.2010.2047114.

[58] Y. Liu, K. Liu, C. Zhang, J. Wang, X. Wang, Unsupervised feature selection via Diversity-induced Self-representation, Neurocomputing 219:350–363, 2017, https://doi.org/10.1016/j.neucom.2016.09.043.

[59] E. Hancer, B. Xue, M. Zhang, A survey on feature selection approaches for clustering, Artif Intell Rev 53:4519–4545, 2020, https://doi.org/10.1007/s10462-019-09800-w.

[60] C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B. Qaqish & M. Yousef, Review of feature selection approaches based on grouping of features, PeerJ, 11, e15666, 2023, https://doi.org/10.7717/PEERJ.15666/FIG-4.

[61] H. Budak, Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2018, 22, 21-23.

[62] R. Kohavi and G.H. John, Wrappers for feature subset selection, Artificial intelligence, 97(1-2):273–324, 1997.

[63] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pp. 491-502, 2005, https://doi.org/10.1109/TKDE.2005.66.

[64] Saúl Solorio-Fernández, J. Ariel Carrasco-Ochoa, and José Fco. Martínez-Trinidad, “A review of unsupervised feature selection methods,” Artificial Intelligence Review, vol. 53, no. 2, pp. 907–948, Jan. 2019, https://doi.org/10.1007/s10462-019-09682-y.

[65] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, In: NIPS’05: Proceedings of the 18th International Conference on Neural Information Processing Systems, pp 507–514, 2005.

[66] Y. Liu, D. Ye, W. Li, H. Wang, Y. Gao, Robust neighborhood embedding for unsupervised feature selection. Knowledge-Based Syst 193:105462, 2020, https://doi.org/10.1016/j.knosys.2019.105462.

[67] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on Machine learning – ICML ’07. ACM Press, New York, New York, USA, pp 1151–1157, 2007.

[68] K. Henni, N. Mezghani, A. Mitiche, Cluster Density Properties Define a Graph for Effective Pattern Feature Selection, IEEE Access 8:62841–62854, 2020, https://doi.org/10.1109/ACCESS.2020.2981265.

[69] J. Miao, Y. Ping, Z. Chen, X. B. Jin, P. Li, L. Niu, Unsupervised feature selection by non-convex regularized self-representation, Expert Syst Appl 173:114643, 2021, https://doi.org/10.1016/j.eswa.2021.114643.

[70] S. L. Huang, L. Zhang, L. Zheng, An information-theoretic approach to unsupervised feature selection for high-dimensional data, In: 2017 IEEE Information Theory Workshop (ITW), IEEE, pp 434–438, 2017.

[71] H. Lim, D. W. Kim, Pairwise dependence-based unsupervised feature selection, Pattern Recognit 111:107663, 2021, https://doi.org/10.1016/j.patcog.2020.107663.

[72] D. Huang, X. Cai, C. D. Wang, Unsupervised feature selection with multi-subspace randomization and collaboration, Knowledge-Based Systems 182:104856, 2019, https://doi.org/10.1016/j.knosys.2019.07.027.

[73] X. Yan, S. Nazmi, B.A. Erol, A. Homaifar, B. Gebru, E.Tunstel, An efficient unsupervised feature selection procedure through feature clustering, Pattern Recognit Lett 131:277–284, 2020, https://doi.org/10.1016/j.patrec.2019.12.022.

[74] D. Ding, F. Xia, X. Yang, C. Tang, Joint dictionary and graph learning for unsupervised feature selection, Appl Intell 50:1379–1397, 2020, https://doi.org/10.1007/s10489-019-01561-x.

[75] Y. Huang, Z. Shen, F. Cai, T. Li, F. Lv, Adaptive graph-based generalized regression model for unsupervised feature selection, Knowledge-Based Syst 227:107156, 2021, https://doi.org/10.1016/j.knosys.2021.107156.

[76] F. Nie, W. Zhu, X. Li, Structured Graph Optimization for Unsupervised Feature Selection, IEEE Trans Knowl Data Eng 33:1210–1222, 2019, https://doi.org/10.1109/TKDE.2019.2937924.

[77] W. Fan, N. Bouguila, Spherical data clustering and feature selection through nonparametric Bayesian mixture models with von Mises distributions, Eng Appl Artif Intell 94:103781, 2020, https://doi.org/10.1016/j.engappai.2020.103781.

[78] W. Fan, R. Wang, N. Bouguila, Simultaneous positive sequential vectors modeling and unsupervised feature selection via continuous hidden Markov models, Pattern Recognit 119:108073, 2021, https://doi.org/10.1016/j.patcog.2021.108073.

[79] Y. Min, M. Ye, L. Tian, Y. Jian, C. Zhu, S. Yang, Unsupervised feature selection via multi-step markov probability relationship, Neurocomputing 453:241–253, 2021, https://doi.org/10.1016/j.neucom.2021.04.073.

[80] J. S. Wu, M. X. Song, W. Min, J. H. Lai, W. S. Zheng, Joint adaptive manifold and embedding learning for unsupervised feature selection, Pattern Recognit 112:107742, 2021, https://doi.org/10.1016/j.patcog.2020.107742.

[81] N. J. Martarelli, M. S. Nagano, Unsupervised feature selection based on bio-inspired approaches, Swarm Evol Comput 52:100618, 2020, https://doi.org/10.1016/j.swevo.2019.100618.

[82] H. Li, Y. Wang, Y. Li, P. Hu, R. Zhao, Joint local structure preservation and redundancy minimization for unsupervised feature selection, Appl Intell 50:4394–4411, 2020, https://doi.org/10.1007/s10489-020-01800-6.

[83] D. Ding, X. Yang, F. Xia, T. Ma, H. Liu, C. Tang, Unsupervised feature selection via adaptive hypergraph regularized latent representation learning, Neurocomputing 378:79–97, 2020, https://doi.org/10.1016/j.neucom.2019.10.018.

[84] R. Shang, L. Wang, F. Shang, L. Jiao, Y. Li, Dual space latent representation learning for unsupervised feature selection, Pattern Recognit 114:107873, 2021, https://doi.org/10.1016/j.patcog.2021.107873.

[85] F. Wang, L. Zhu, J. Li, H. Chen, H. Zhang, Unsupervised soft-label feature selection, Knowledge-Based Syst 219:106847, 2021, https://doi.org/10.1016/j.knosys.2021.106847.

[86] X. Han, P. Liu, L. Wang, D. Li, Unsupervised feature selection via graph matrix learning and the low-dimensional space learning for classification, Eng Appl Artif Intell 87:103283, 2020, https://doi.org/10.1016/j.engappai.2019.103283.

[87] P. Zhou, L. Du, X. Li, Y. D. Shen, Y. Qian, Unsupervised feature selection with adaptive multiple graph learning, Pattern Recognit 105:107375, 2020, https://doi.org/10.1016/j.patcog.2020.107375.

[88] R. Zhang, X. Li, Unsupervised Feature Selection Via Data Reconstruction and Side Information, IEEE Trans Image Process 29:8097–8106, 2020, https://doi.org/10.1109/TIP.2020.3011253.

[89] H. Zhao, L. Du, J. Wei, Y. Fan, Local Sensitive Dual Concept Factorization for Unsupervised Feature Selection, IEEE Access 8:133128–133143, 2020, https://doi.org/10.1109/ACCESS.2020.3010862.

[90] J. Chen, Y. Zeng, Y. Li, G. B. Huang, Unsupervised feature selection based extreme learning machine for clustering, Neurocomputing 386:198–207, 2020, https://doi.org/10.1016/j.neucom.2019.12.065.

[91] R. Jain, W. Xu, RHDSI: A novel dimensionality reduction based algorithm on high dimensional feature selection with interactions, Inf Sci (Ny) 574:590–605, 2021, https://doi.org/10.1016/j.ins.2021.06.096.

[92] X. Zhang, M. Fan, D. Wang, P. Zhou, D. Tao, Top- k Feature Selection Framework Using Robust 0–1 Integer Programming, IEEE Trans Neural Networks Learn Syst 32:3005–3019, 2021, https://doi.org/10.1109/TNNLS.2020.3009209.

[93] A. Chaudhuri, D. Samanta, M. Sarma, Two-stage approach to feature set optimization for unsupervised dataset with heterogeneous attributes, Expert Syst Appl 172:114563, 2021, https://doi.org/10.1016/j.eswa.2021.114563.

[94] Y. Zhang, Z. Lu, S. Wang, Unsupervised feature selection via transformed auto-encoder, Knowledge-Based Syst 215:106748, 2021, https://doi.org/10.1016/j.knosys.2021.106748.

[95] O. Bahadır, H. Türkmençalıkoğlu, Bilgi Kuramında Shannon Entropisi ve Uygulamaları, Avrupa Bilim ve Teknoloji Dergisi (32), 491-497, 2021, https://doi.org/10.31590/ejosat.1039771

[96] A. Karci, F. Bilgiç, Karcı ve Shannon Entropilerin Karşılaştırılması, Computer Science, 4(2), 68-73, 2019.

[97] M. Soleimani-damaneh, M. Zarepisheh, Shannon's entropy for combining the efficiency results of different DEA models: Method and application, Expert Systems with Applications, Volume 36, Issue 3, Part 1, 2009, https://doi.org/10.1016/j.eswa.2008.06.031.

[98] Y. Karaca, M. Moonis, Shannon entropy-based complexity quantification of nonlinear stochastic process: diagnostic and predictive spatiotemporal uncertainty of multiple sclerosis subgroups, 2022, https://doi.org/10.1016/B978-0-323-90032-4.00018-3.

[99] H.T. Nguyen, G.S. Rogers, The Cumulative Distribution Function, In: Fundamentals of Mathematical Statistics, Springer Texts in Statistics. Springer, New York, NY,1989, https://doi.org/10.1007/978-1-4612-1013-9_24.

[100] S. G. Tanyer, "The Cumulative Distribution Function for a finite data set," 2012 20th Signal Processing and Communications Applications Conference (SIU), Mugla, Turkey, 2012, pp. 1-3, https://doi.org/10.1109/SIU.2012.6204462.

[101] D. Campos and J. Bernardes, Cardiotocography, UCI Machine Learning Repository, 2010, https://doi.org/10.24432/C51S4N.

[102] D. Lucas, R. Klein, J.Tannahill, D. Ivanova, S. Brandon, D. Domyancic, Y. Zhang, Climate Model Simulation Crashes. UCI Machine Learning Repository. 2013, https://doi.org/10.24432/C5HG71.

[103] [Online] Erişim adresi: https://jundongl.github.io/scikit-feature/datasets.html.

[104] Sejnowski,Terry and R. Gorman, Connectionist Bench (Sonar, Mines vs. Rocks), UCI Machine Learning Repository, https://doi.org/10.24432/C5T01Q.

[105] B. Antal and A. Hajdu, Diabetic Retinopathy Debrecen, UCI Machine Learning Repository. 2014, https://doi.org/10.24432/C5XP4P.

[106] P. Gijsbers, OpenML A worldwide machine learning lab, [Online] Erişim adresi: https://www.openml.org/d/40670

[107] [Online] Erişim adresi: https://github.com/wang-feifei/USFS-code/tree/master/Datasets.

[108] K. Gyamfi and C. Marshall, Ultrasonic flowmeter diagnostics. UCI Machine Learning Repository, 2018, https://doi.org/10.24432/C5B895.

[109] I. Guyon, Madelon. UCI Machine Learning Repository, 2008, https://doi.org/10.24432/C5602H.

[110] K. Mansouri, T. Ringsted, D. Ballabio, R. Todeschini and V. Consonni, QSAR biodegradation, UCI Machine Learning Repository. 2013, https://doi.org/10.24432/C5H60M.

[111] P. Mowforth and B. Shepherd, Statlog (Vehicle Silhouettes), UCI Machine Learning Repository, https://doi.org/10.24432/C5HG6N.

[112] A. Freire, M. Veloso and G. Barreto, Wall-Following Robot Navigation Data, UCI Machine Learning Repository, 2010, https://doi.org/10.24432/C57C8W.

[113] Ö. Akar, O. Güngör, Classification of multispectral images using Random Forest algorithm, Journal of Geodesy and Geoinformation, 1(2), 105-112, 2012.

[114] D. Segura, E. Khatib, R. Barco, Dynamic Packet Duplication for Industrial URLLC. Sensors, 2022, 22. 587, https://doi.org/10.3390/s22020587.

[115] D. Korkmaz, H. E. Çelik, M. KAPAR, Sınıflandırma ve Regresyon Ağaçları ile Rastgele Orman Algoritması Kullanarak Botnet Tespiti: Van Yüzüncü Yıl Üniversitesi Örneği. Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2018, 23(3), 297-307.

[116] K. Özkan, Modelling ecological data using classification and regression tree technique (CART), Turkish Journal of Forestry, c. 13, sy. 1, ss. 1–4, 2012, https://doi.org/10.18182/tjf.94244.

[117] C. Cortes, V. Vapnik, Support-vector networks, Mach Learn 20, 273–297, 1995, https://doi.org/10.1007/BF00994018.

[118] O. YILDIZ, Derin öğrenme yöntemleriyle dermoskopi görüntülerinden melanom tespiti: Kapsamlı bir çalışma, GUMMFD, c. 34, sy. 4, ss. 2241–2260, 2019, https://doi.org/10.17341/gazimmfd.435217.

[119] E. Pearson, Bayes Theorem, Examined in the Light of Experimental Sampling. Biometrika, 17, 388-442, 1925, https://doi.org/10.1093/biomet/17.3-4.388.

EKLER

EKLER

EK A: Önerilen Yöntem (EFS) Matlab Kodları

```
%% Entropi Tabanlı Özellik Seçimi - Entropy-based Feature Selection (EFS)
function [ IDX, Weights ] = EFS( X )
%  Parametre:
%                   'X' - Giriş Verisi
%
%  Çıktılar:
%
%                   'IDX' - Özellik İndeksleri
%
%                   'Weights' - Özelliklerin sıralarına göre ağırlıkları
%
%
%   Örnek:
%                [IDX, Weights] = EFS(X);
%

   [m,d] = size(X);

%% Dağılımın Kümülatif Entropisini hesapla
   p = zeros(m,d);
   for i=1:d
      p(:,i) = cdf('normal',X(:,i),mean(X(:,i)),std(X(:,i)));
   end
   H1 = -sum(p(:,1:end).*log2(p(:,1:end)));
   [~, idx1] = sort(H1,'ascend');


%% Dağılımın simetrisinin Shannon entropisini hesaplayın
   centroid = max([mean(X,'omitnan'); median(X,'omitnan'); mode(X)],[],1);
   Xt = double(X >= centroid);
    prO = sum(Xt)/m;
    prZ = 1-prO;
    H2 = -((prO).*log2(prO) + (prZ).*log2(prZ));
   [~, idx2] = sort(H2,'descend');

%% Her birinin entropi değerlerine göre özelliklerin sıralarını hesapla
    r = zeros(1,d);
   for i=1:d
      r(i) = geomean([find(idx1==i)  find(idx2==i)]);
   end
   [Weights, IDX] = sort(r,'ascend');
end
```

EK B: Demo Programı Çalıştırma Kodları

```
%% EFS Algoritması için DEMO Program
clc;
clear;

%%Bir Veri Seti Seçin
dataset = load('colon');
fns = fieldnames(dataset);
[ X, Y ] = divideTable( dataset.(fns{1}) );

NumOfFea = 8;
tic;
[ idx, Weights ] = EFS( X );
T=toc;
ACC = demo1(X(:,idx(1:NumOfFea)), Y, 'KNN');

clear dataset;
clear fns;
clear NumOfFea;

%%%%%%%%%
function [ACC] = demo1( X, Y, classifier )

   predictions = repmat(Y, 1, 2);
   indices = crossvalind('Kfold', Y, 10);

   for i = 1:10
      fprintf('%d',i);
      test = (indices == i);
      train = ~test;

      trainY = Y(train,:);
      trainX = X(train,:);
      testX = X(test,:);

      switch classifier
         case 'KNN'
            Mdl = fitcknn(trainX, trainY, 'NumNeighbors', 1);
         case 'CART'
            Mdl = fitctree(trainX, trainY);
         case 'NB'
            % "normal", "mn", "kernel", "mvmn".
            Mdl = fitcnb(trainX, trainY, 'DistributionNames', 'normal');
         case 'SVM'
            % "linear", "gaussian", "rbf", "polynomial"
            t = templateSVM('Standardize', true, 'KernelFunction', 'linear');
            Mdl = fitcecoc(trainX, trainY, 'Learners', t);
         case 'RF'
            Mdl = fitcensemble(trainX,trainY,'Method','Bag');
      end
```

```
        predictions(test, 2) = predict(Mdl, testX);
    end
    ACC = sum(predictions(:,1) == predictions(:,2))*100/length(Y);
end

%% Veri kümesini girdi matrisine ve çıktı vektörüne ayırın
function [ X, Y ] = divideTable( DATASET )

    if istable(DATASET)
        X = table2array(DATASET(:,1:end-1));
        Y = categorical(DATASET.Class);
    else
        error('Parametre bir tablo olmalidir, %s.', class(DATASET));
    end
end
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Samet DEMİREL

Doğum tarihi ve yeri :

e-posta :

Öğrenim Bilgileri

Derece	**Okul/Program**	**Yıl**
Yüksek Lisans	Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği	2024
Lisans	Hoca Ahmet Yesevi Uluslararası Türk-Kazak Üniversitesi/Mühendislik Fakültesi/Bilgisayar Mühendisliği	2017
Lisans	Anadolu Üniversitesi/İşletme Fakültesi/İşletme	2010
Ön Lisans	Balıkesir Üniversitesi/ Balıkesir Meslek Yüksekokulu/Bilgisayar Programcılığı	2013
Lise	Balıkesir Anadolu Ticaret Meslek Lisesi	2003

Yayın Listesi

M. T. Sarıtaş, C. Börekci, & S. Demirel, Quality assurance in distance education through data mining, International Journal of Technology in Education and Science (IJTES), 6(3), 443-457, 2022, https://doi.org/10.46328/ijtes.396

S. Demirel, F. Aydın, A New Method Proposal based on Shannon and Cumulative Entropy for Unsupervised Feature Selection, 2n Global Conference on Engineering Research (GLOBCER'22), 391-391, 2022, Online, (Özet Bildiri/Sözlü Sunum)

S. Demirel, F. Aydın, A New Fast Filter-based Unsupervised Feature Selection Algorithm Using Cumulative and Shannon Entropy, Journal of Soft Computing and Artificial Intelligence, 2024, (Makale Değerlendirme Aşamasında)