T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOMEDİKAL VERİLERİN AKILLI SİSTEMLER İLE SINIFLANDIRMA BAŞARIMLARININ ANALİZİ

DOKTORA TEZİ

Akın ÖZÇİFT
(07113204)

Anabilim Dalı: Elektrik-Elektronik Mühendisliği
Programı : Devreler-Sistemler

Tez Danışmanı: Yrd. Doç. Dr. Arif GÜLTEN

Tezin Enstitüye Verildiği Tarih: 18.01.2011

Şubat, 2011

ÖNSÖZ

Bilgisayar destekli hastalık teşhis sistemleri, akıllı sistemler adı verilen yapay zeka algoritmaları ile hastalıklarla ilgili uzmanların teşhis tecrübelerini birleştirerek elde edilen yardımcı yazılımlardır.

Literatürde, 1970'li yıllardan başlayarak bilgisayar destekli hastalık teşhis sistemlerine rastlanmaktadır. Günümüz itibariyle bu alanda binlerce ticari yazılımın kullanılmasının yanında, sadece bu konuya ayrılmış edilmiş çok sayıda dergi ve bu alanda yapılan yüzbinlerce akademik çalışma bulunmaktadır.

Bilgisayar destekli hastalık teşhis sistemlerinin hassasiyeti, bu yazılımların tasarımında kullanılan akıllı sistem algoritmalarının başarımı ile doğru orantılıdır. Daha açık bir ifadeyle, bir teşhis yazılımının başarımı, yazılımın tasarımında kullanılan analiz algoritmasının başarımına bağlıdır.

Bu çalışma temel olarak, farklı biyomedikal verilerin analizinde kullanılan, çok sayıda akıllı sistem algoritmasının başarımına etki eden faktörleri incelemektedir. Bu çalışma ile elde edilen sonuçların, bilgisayar destekli hastalık teşhisi alanında çalışan araştırmacılara yardımcı olacağı düşünülmektedir.

Öncelikle bana kendisiyle çalışma fırsatını veren, bu tez çalışması boyunca ilgisini ve samimi desteğini esirgemeyen danışman hocam, Sayın Yrd.Doç. Arif GÜLTEN'e teşekkürlerimi sunarım.

Ayrıca akıllı sistemler konusuna ilgi duymamı sağlayan ve bu konuda ufuk açıcı desteklerini her zaman hissettiğim, Sayın Doç. Dr. Mehmet KAYA'ya, manevi desteği ile her zaman yanımda olan Sayın Mehmet Emin VURAL'a, anneme, babama ve Fırat Üniversitesi Elektrik-Elektronik Mühendisliği Bölümündeki tüm değerli hocalarıma teşekkürü bir borç bilirim.

Akın ÖZÇİFT
ELAZIĞ-2011

İÇİNDEKİLER

Sayfa No

ÖZET

Medikal enformatik çalışma alanları içinde en popüler alanlardan birisi, biyomedikal verilerin bilgisayar destekli analizidir. Bilgisayar destekli hastalık teşhis sistemi adı verilen uzman sistemler, hastalık teşhisi karar aşamalarında uzman tıp personeline destek olmaktadır. Tıbbi teşhis sürecinin hassasiyetin korunarak hızlandırılması ancak uzman tıp personelinden elde edilen bilgi-beceriyle donanmış uzman yazılım sistemleri ile mümkündür. Literatürde bu amaca uygun olarak tıbbi verilerin akıllı sistem destekli yazılımlar ile analizine ait çok sayıda çalışma yapılmış ve yapılmaktadır.

Akıllı sistemler, uzman tıbbi teşhis sistemlerinin en önemli parçasını oluştururlar. Bir uzman tıbbi teşhis yazılımının hassasiyeti, yazılımın çekirdeği olan akıllı sistemin performansı ile birebir doğru orantılıdır. Bu nedenle, akıllı sistemlerin biyomedikal verilerin analizindeki başarımını etkileyen faktörlerin belirlenmesi, başarımı yüksek tıbbi teşhis yazılımlarının geliştirilmesinde önemli bir aşamadır.

Bu tez çalışması, rahim kanseri, prostat kanseri, Parkinson, dermatoloji ve diyabet gibi medikal verilerin otuz beş akıllı sistem algoritmasıyla analizi ve bu algoritmaların sınıflandırma başarım faktörlerinin belirlenmesi hedefiyle gerçekleştirilmiştir. Akıllı sistemlerin medikal veri analizindeki başarımını etkileyebilecek çok sayıda faktörden öznitelik seçimi, verilerin ön-işlemesi, algoritma parametrelerinin değişiminin performansa etkisi ve algoritmaların topluluk öğrenmesi modellerinin geliştirilmesi incelediğimiz faktörlerdir. Yapılan çok sayıda deneysel çalışma bahsi geçen faktörler ile akıllı sistem algoritmalarının sınıflandırma başarımı arasında korelasyon olduğunu göstermiştir. Bu şekilde tıbbi teşhis hassasiyetinde kullanılmak amacıyla üretilecek uzman sistem yazılımlarına ait performansın, öznitelik seçimi, verinin ön-işlemesi ve topluluk öğrenmesi teknikleriyle arttırılabileceği hesapsal olarak gösterilmiştir.

Bu çalışmada deneysel olarak test edilen akıllı sistem algoritmaları ve öznitelik seçim yöntemleri Matlab, Weka ve Microsoft Visual Studio yazılım geliştirme ortamlarında gerçekleştirilmiştir.

Anahtar Kelimeler: Öznitelik Seçimi, Topluluk Öğrenmesi, Ön-işleme, Kütle Spektrometrisi, Biyomedikal Veri İşleme.

SUMMARY

Analysis of Performances of Intelligent Systems for Biomedical Data

One of the most popular areas of medical informatics is computer assisted analysis of biomedical data. Expert systems that are so called computer based disease diagnosis systems support medicians in disase diagnosis decision making. Fastening the medical decison phase with preserving accuracy is possible with the expert systems that are trained with medicians knowledge and experience. In the literature, there is an aboundant of related work that are suitable for the task of supportive expert systems.

Intelligent systems constitute an important part of expert medical decision systems. An expert medical decision system's accuracy depends on the performance of the intelligent system that is the kernel of the software. Therefore, the accuracy of the expert system is one to one correspondent with the performance of the intelligent system. Hence, it is important to determine the factors that affect the performance of the intelligent systems while analyzing medical data, in order to develop high accurate medical decision systems.

This thesis is fulfilled with the aim of determining performance factors of about thirty intteligent system algorithms while analyzing ovarian cancer, prostate cancer, Parkinson disease, dermatology and diabet datasets. We selected feature selection, data pre-processing, algorithm parameter changes and ensemble learning out of so many performance effecting factors for medical data analysis. As a result of many experiments carried out, a correlation is found with the mentioned factors and the performances of the intelligent systems. In this way, it is proved computationaly that the performance of the experts systems to be developed as medical decision systems migth be improved with feature selection, data pre-processing, algorithm parameter change and ensemble learning.

In this work, the experimented intelligent system algoritms and feature selection strategies are realized using Matlab, Weka and Microsoft Visual Studio software development environments.

Keywords: Feature Selection, Ensemble Laerning, Data Pre-processing, Mass Spectrometry, Biomedical Data Processing.

ŞEKİLLER LİSTESİ

Sayfa No

TABLOLAR LİSTESİ

Sayfa No

SEMBOLLER LİSTESİ

ACC	**:**	Accuracy
ACO	**:**	Ant Colony Optimization
ANN	**:**	Artificial Neural Networks
BMU	**:**	Best Matching Unit
BN	**:**	Bayes Network
BFS	**:**	Best First Search
ELA	**:**	Ensemble Learning Algorithms
Fm	**:**	F-measure
KE	**:**	Kappa's Error
KNN	**:**	K-Nearest Neighboor
CBFS	**:**	Correlation Based Feature Selection
LDA	**:**	Linear Discriminant Analysis
LR	**:**	Logistics Regression
MCC	**:**	Matthew's Correlation Coefficient
MSE	**:**	Mean Squared Error
PCA	**:**	Principal Component Analysis
PLS	**:**	Piecewise Least Squares
PSO	**:**	Particle Swarm OptİmizatIon
PYT	**:**	Pozitron Yayılım Tomografileri
ROC	**:**	Receiver Operating Characteristics
SOM	**:**	Self Organizing Maps
SVM	**:**	Support Vector Machines
TNP	**:**	Tek Nükleotid Polimorfizmi
UTÖS	**:**	Uygunluk Tabanlı Öznitelik Seçimi
WT	**:**	Wavelet Transform
$f(t)$	**:**	Kütle Spektrometri Sinyali
$B(t)$	**:**	Baz Çizgisi
$S(t)$	**:**	Orijinal Protein Sinyali
$\varepsilon(t)$	**:**	Gürültü
(k/y)	**:**	Kütle/Yük
D	**:**	Öznitelik Başlangıç Seti
X_{best}	**:**	En İyi Alt Öznitelik Seti
$J(X_k)$	**:**	J Algoritmasının Öznitelikleri Sınıflandırması
δ	**:**	Eşik Değeri
$d(X,Y)$	**:**	X ve Y Komşuluğu
$\phi(y_i)$	**:**	Sigmoid Fonksiyonu
$e_j(n)$	**:**	J Düğümüne Ait Hata
X_i (t)	**:**	Parçacığın Pozisyonu
V_i (t)	**:**	Parçacığın Hızı
P_i (t)	**:**	Parçacığa Ait Pozisyon Bilgisi

1. GİRİŞ

1.1. Genel Bilgi

Biyomedikal hastalıkların teşhisinde bilgisayar destekli uzman sistemlerin kullanılma eğilimi literatürde bu alanda oldukça fazla sayıda çalışma yapılması sonucunu vermiştir. Bir biyomedikal teşhis sisteminde en önemli parametreler sistemin kullanım kolaylığı ve algoritmanın hastalığı teşhis hassasiyetidir. Bilgisayar tabanlı uzman sistemler hastalık teşhisini akıllı sistem algoritmalarını temel alarak gerçekleştirirler. Bu tez çalışması, bir hastalığın doğru teşhis hassasiyetinin önemi çerçevesinde, farklı türde bir grup medikal veriyi esas alarak literatürde sıklıkla kullanılan çok sayıda akıllı sistem algoritmasının sınıflandırma başarımına etki eden faktörleri belirlemeyi hedeflemektedir.

1.2. Tezin Amacı

Kanser gibi hastalıkların erken teşhis edilmesi, bu tür ölümcül hastalıkların zamanında tedavisi açısından büyük önem taşımaktadır. Son yıllarda kanser türlerinin yaygınlaşması ve kanserden ölümlerin artması, bu tür hastalıkların erken teşhisini daha önemli hale getirmiştir. İnsan genomunun tamamının dizilenmiş olması kanser ve benzeri hastalıkların genetik nedenlerinin bulunmasına önemli katkı sağlamıştır [1]. Genel olarak kanser benzeri hastalıkların teşhisinde iki yöntem kullanılmaktadır. Geleneksel yöntemler biyopsi denilen ve şüpheli dokudan parça alma şeklindeki cerrahi teknikleri kullanarak hastalık teşhisini öngörürken, daha yeni teknikler ise kan serumu gibi biyolojik sıvılardaki hastalığa ait biyolojik işaretçileri kullanarak tanı koymaya çalışırlar [2].

Kanser teşhisi için kullanılan biyolojik serum teknikleri genel olarak hastalardan alınan sıvılarda protein örüntüleri ararlar [3]. Proteomik, protein moleküllerinin yapısı ve fonksiyonu ile ilgili bir bilim dalı olarak, kanser gibi hastalıkların genetik nedenlerini de belirlemeye çalışır. Canlıların vücutlarındaki her protein bir genin ifadesidir yani bir gene karşılık gelir. Bu ilişki, protein örüntü tanıma teknikleri ile hastalıklı ve sağlıklı kişilere ait serum örneklerindeki çok sayıda gen dizisinin bir tek deneyle aynı anda ifade edilmesini sağlar. Bu yaklaşımla, dizilere karşılık gelen protein moleküllerinin analiz edilerek karşılaştırılmasını esas alan teknikler geliştirilmiştir [4]. Kütle spektrometrisi olarak bilinen

bu teknikler hastalıklı ve sağlıklı dokulardaki protein moleküllerini kütlelerine göre sınıflandırarak eldeki veriyi analiz esasına göre çalışırlar [5].

Kütle spektrometrisinin, prostat, göğüs ve rahim kanserinin teşhisinde kullanılanılabilmesi için [6], genel olarak birbiriyle bağlantılı şu aşamalar takip edilir:

i)Deney sonucunda ortaya çıkan verinin ön-işlemesinin yapılması; Bu aşamada ham veri baz çizgisi doğrulaması, normalizasyon ve tepe değerinin muhafaza edilerek gürültünün giderilmesi şeklindeki alt adımları kapsar [7].

ii)Yüksek boyutlu veriden sınıflandırıcılar için uygun özelliklerin seçilmesi [8] ikinci adımı oluşturur.

iii)Uygun sınıflandırıcılar yardımıyla verilerin sınıflandırılması ve sınıflandırma başarımının ölçülmesi [9] hastalık teşhisindeki son adımdır.

Kütle spektrometri verisi, gürültüye ve cihaz hassasiyetine bağlı deneysel hatalara açık bir veri türüdür. Bu nedenle, ilk olarak verinin gürültüden temizlenmesi ve normalizasyonla hatalara karşı hassasiyetinin azaltılması gerekir [10].

Spektrometri verisi, yüksek boyutlu bir veridir [11]. Bu nedenle, çalışmada kullandığımız rahim ve göğüs kanseri verilerinin [12], dördüncü bölümde incelenen boyut indirgeme yöntemleriyle boyutu azaltılmalı ve akıllı sistem algoritmalarının hassasiyetlerini kaybetmeden işleyebileceği şekle getirilmelidir.

Çalışmamızda ön-işleme ve boyut indirgeme aşamalarından geçirilen kütle spektrometri verisinin sağlıklı-hasta şeklinde sınıflandırılması Yapay Sinir Ağları (Artificial Neural Networks, ANN), Genetik Algoritma (Genetic Algorithm, GA), Destek Vektör Makineleri (Support Vector Machine, SVM), Doğrusal Ayrıklık Analizi (Linear Discriminant Analysis, LDA) ve K-En Yakın Komşu (K-nearest Neighbour, KNN) ve Temel Bileşen Analizi (Principal Component Analysis, PCA) gibi öğreticili sınıflandırıcılar kullanılarak gerçekleştirilmiştir [13]. Sınıflandırıcıların başarım performansları veya istatistiksel hassasiyetleri, sensitivite ve spesifite gibi istatistiksel metrikler kullanılarak ölçülmüştür [14]. Kütle spektrometri verisinin sınıflandırmasına ait sonuçlar Bölüm 7'de verilecektir.

Bu çalışmada sınıflandırılma başarımı incelenecek bir diğer grup biyomedikal veri, hem rahim hem de prostat kanserine ait kütle spektrometri verilerinin bir araya getirildiği yüksek boyutlu deneysel bir veri seti olan Arcene, altı farklı deri hastalığının teşhisinde kullanılan dermatoloji verisi, Parkinson hastalarının konuşmalarından yola çıkarak hastalık teşhisinin yapıldığı biyomedikal veri setleridir. Parkinson ve dermatoloji veri setleri kütle

spektrometrisi gibi ön-işleme adımlarına ihtiyaç duymamakla birlikte, bu veri setlerinin yüksek boyutlu yapısından dolayı sınıflandırma algoritmalarınca test edilmeden önce boyutları düşürülmüştür. Bu verilerin analizinde, kütle spektrometrisinde bahsi geçen akıllı sistem algoritmalarının yanında Bayes Ağları (Bayes Networks, BN), karar ağaçları, Lojistik sınıflandırıcılar ve değişik topluluk öğrenme algoritmaları da kullanılmıştır.

Kütle spektrometrisi, Parkinson ve dermatoloji gibi yüksek boyutlu verilerden başka veri çeşitliliğini arttırmak için düşük boyutlu medikal verilerden göğüs kanseri, mamografi, diyabet, protein lokalizasyon problemine ait veri setleri ve genomik örüntü tanıma problemine ait bir grup veri test edilecek diğer veriler olarak seçilmiştir [15]. Arcene ile birlikte bu veri setleri bir öğreticisiz ANN türü olan Kendine Organize Haritalar (Self Organizing Map, SOM) algoritması [16] yardımıyla sınıflandırılarak bu algoritmanın sınıflandırma performansı, hem kümeleme kalitesi hem de sınıflandırma başarımı yönünden incelenmiştir.

Bu bölümde bahsi geçen tüm veri setlerinin matematiksel yapısı, verilerin alındığı kaynaklarla ilgili detaylı bilgi, Bölüm 2'de yer almaktadır.

Bu çalışmanın ilk hedefi, kütle spektrometrisi verilerinin ön-işleme adımlarının, en sık kullanılan akıllı sistem algoritmalarının sınıflandırma başarımlarına nasıl etki ettiğini hesapsal şekilde göstermektir.

Çalışmamızın ikinci hedefi, öğreticisiz bir sistem olan ve eğitim süresi, verinin boyutuna ve kayıt sayısına bağlı olarak ciddi şekilde artan SOM algoritmasına ait eğitim aşamasının, kümeleme kalitesini değiştirmeden bir sürü zekası algoritması türü olan PSO algoritmasıyla [17] optimizasyonudur.

Çalışmamızın üçüncü hedefi ise dermatoloji ve Parkinson hastalıklarının hassas şekilde sınıflandırılmasını sağlayacak öznitelik seçim yöntemlerinin geliştirilmesi ve bu öznitelikler yardımıyla literatürdeki en yüksek sınıflandırma başarımının elde edilmesidir.

1.3. Tezde Geliştirilenler

Bu tez çalışmasının biyomedikal veri işleme ve akıllı sistemler literatürüne katkıları aşağıdaki gibi özetlenebilir:

Kütle spektrometrisi verisinin ön-işlemesinde kullanılan literatürdeki en yaygın üç adım olan, baz çizgisi doğrulaması, normalizasyon ve gürültü giderme adımları, rahim ve prostat kanseri verilerine, tek tek, ikişerli gruplar halinde ve her üçü bir arada olmak üzere

yedi farklı şekilde uygulanmıştır. Elde edilen yedi farklı veri sık kullanılan dört sınıflandırma algoritmasına verilerek, her adımın sınıflandırıcı performansına etkisi somut şekilde gösterilmiştir.

Dermatoloji veri setine ait öznitelikler topluluk öğrenme yöntemiyle (ensemble learning) seçilerek elde edilen düşük boyutlu veri BN, SVM, ANN, Basit Lojistik (Simple Logistic, SL) ve Fonksiyonel Ağaç (Functional Tree, FT) algoritmalarıyla sınıflandırılmıştır. Öznitelik seçiminde literatürde ilk kez rotasyonlu karar ağacı grup sınıflandırıcısı kullanılmış ve sınıflandırmada % 99'luk bir başarım elde edilmiştir. Aynı veri setine ait bir diğer deneysel çalışmada BN ile seçilen özniteliklerle yapılan sınıflandırmada % 99,2'lik bir başarım elde edilmiştir. Bu sonuç literatürde dermatoloji veri seti için elde edilen en yüksek sınıflandırma başarımı olarak belirlendi.

Parkinson veri setine ait öznitelikler, dermatoloji veri setine ait özniteliklerin seçiminde kullanılan rotasyonlu karar ağacı grup sınıflandırıcısı ile seçildi. Bu özniteliklerin sınıflandırılmasında KNN algoritması kullanılmış ve % 98'lik bir sınıflandırma başarımı elde edilmiştir.

Parkinson hastalığına ait öznitelikler zarflayıcı SVM tabanlı algoritma ile seçilerek elde edilen özniteliklerin RFEL algoritması tabanlı KNN uyarlaması ile sınıflandırılmasında % 97'lik bir sınıflandırma başarımı elde edilmiştir.

Parkinson hastalığına ait önemli öznitelikler korelasyon tabanlı öznitelik seçme yöntemi ile ayrılmıştır. Daha sonra bu öznitelikler 30 farklı akıllı sistem algoritması için oluşturulan, rotasyoncu topluluk öğrenme algoritması (Rotation Forest Ensemble Learning, RFEL) modellerinin sınıflandırma başarımınının etkisisini hesaplamak için kullanılmışlardır. Akıllı sistem algoritmalarının 26 tanesinde RFEL modellerinin sınıflandırma başarımını belirgin şekilde arttırdığı gösterilmiştir.

Arcen, göğüs kanseri, diyabet, mamografi, protein lokalizasyon site verileri ve genomik verilerden oluşan bir grup veri SOM algoritması ile kümelenmiştir. Verilerin algoritma tarafından kümelenmesi sırasında geçen zaman kayıt edilmiştir. SOM algoritması, literatürde ilk kez Parçacık Sürü Optimizasyon (Particle Swarm Optimization, PSO) algoritması ile optimize edildikten sonra aynı veriler yeni algoritma ile kümelenmiş ve bu işlem sırasında geçen zaman kayıt edilmiştir. Optimize edilmiş SOM algoritmasının klasik algoritmaya göre kümeleme ve sınıflandırma performansı karşılaştırılmıştır. SOM algoritmasının eğitim aşaması kısaltılırken kümeleme ve sınıflandırma performansında

ciddi bir değişiklik gözlenmediği farklı metrikler yardımıyla hesapsal şekilde gösterilmiştir.

1.4. Tezin İçeriği

Tezin bundan sonraki bölümleri aşağıdaki gibi düzenlenmiştir:

Bölüm 2'de, bu tez çalışmasında ele alınan akıllı sistem algoritmalarının sınıflandırma başarımını test etmekte kullanılan biyomedikal verilerin yapısı açıklanmıştır.

Bölüm 3'de, rahim ve prostat kanserine ait kütle spektrometri verilerinin sınıflandırılması için gerekli ilk aşama olan ön-işleme adımları izah edilmiştir.

Bölüm 4'de, biyomedikal verilerde öznitelik seçme problemi incelenerek, bu çalışmada kullanılan veri setleri için tercih edilen öznitelik seçme yöntemleri ile ilgili bilgi verilmiştir.

Bölüm 5'de, performansı (sınıflandırma başarımı) incelenen akıllı sistem algoritmalarına ait açıklamalara yer verilmiştir.

Bölüm 6'da, akıllı sistem algoritmalarının sınıflandırma başarımını hesapsal şekilde ölçmekte kullanılan metrikleri izah edilmiştir.

Bölüm 7'de, tez çalışmasına konu olan akıllı sistem algoritmalarının başarımlarını gösteren deneysel sonuçlara yer verilmiştir.

Bölüm 8'de bu tezde elde edilen sonuçlar tartışılmış ve benzer çalışmalar yapacak araştırmacılara bazı öneriler sunulmuştur.

2. BU TEZ ÇALIŞMASINDA KULLANILAN BİYOMEDİKAL VERİLER

Bu çalışmada genel olarak veriye ait öznitelik sayısı dikkate alındığında iki tür veri kullanılmıştır. Bunlar yüksek boyutlu olan prostat, rahim kanseri, Arcene kütle spektrometrisi verileri ile Parkinson ve dermatoloji veri setleridirler. İkinci grup veri setleri ise göreceli olarak daha düşük boyutlu olan mamografi, göğüs kanseri ve diyabet gibi hastalık teşhis verilerinin yanında protein lokalizasyon tespit problemine ait veri seti ile bir grup genomik örüntü tanıma problemine ait DNA dizileridir. Literatürde öznitelik seçme probleminin uygulandığı alanlarda yüksek boyutlu veri göreceli bir kavram olarak kullanıldığı için çalışmamızda öznitelik sayısı 5-10 seviyesinde olan veriler düşük boyutlu, 20-30 ve daha fazla özniteliğe sahip veriler ise yüksek boyutlu veriler şeklinde sınıflandırılmıştır. İzleyen bölümde öncelikle yüksek boyutlu biyomedikal verilerin kaynaklarına ait bilgi verilecek ve daha sonra çalışmada kullanılan bahsi geçen verilerin genel yapısı izah edilecektir.

2.1. Yüksek Boyutlu Biyomedikal Verilerin Kaynağı ve Yapısı

Fizik bilimleri, doğa bilimleri, biyomedikal teknolojiler ve diğer bilimsel çalışmalar işlenmesi gereken çok miktarda veri üretirler [18]. İnsan sağlığını yakından ilgilendiren biyomedikal alanında çok sayıda hastalığın teşhisi için farklı teşhis teknikleri geliştirilmiştir. Bu teknikler hastalardan alınan kan, doku, serum gibi örnekler veya kullanılan genetik materyale bağlı olarak kısmi farklılıklar gösterseler de, özellikle kanser teşhislerinde kullanılan kütle spektrometrisi [19] gibi teknikler artan cihaz çözünürlüğü ile beraber yüksek boyutlu veri üretirler. Bu çalışmada ele alınan ve kanser teşhisi için kullanılan kütle spektrometrisi tekniği [20] dışında yine hastalık teşhisinde kullanılan ve yüksek boyutlu veri üreten belli başlı biyomedikal teknikler şunlardır:

Tek Nükleotid Polimorfizmi (TNP): TNP'ler insanlar arasındaki genetik farklılıkların kaynaklarıdırlar. Her bir TNP bir DNA bloğundaki farklı bir tek nükleotid (Örneğin tüm dizide Sitozin yerine Timin gelmesi) olarak tanımlanırlar [21]. Sağlıklı ve hastalıklı bireylerde, ilgili DNA dizisindeki farklı TNP'lerin bulunması sonucunda nükleotid değişiminin hastalığa sebep olup olmadığı araştırılır [22].

Gen İfadeleri: Bir gende kodlanmış bilginin protein gibi bir gen ürününü sentezlemek için kullanılması işlemi o genin ifade edilmesi olarak tanımlanır. Gen ifadelerindeki genetik farklılıkların hastalıkların kaynağı olup olmayacağını belirlemek için yine hastalıklı bireyler ve sağlıklı bireylerin genlerinin ifadeleri ve hastalıklarla ilgili medikal bilgi birleştirilerek arada korelasyon olup olmadığı incelenmektedir [23].

Mikrodiziler: İnsan genomundaki dizilimin anlamlı bir bilgiye dönüştürülmesi için kullanılan en önemli tekniklerden birisi mikrodizilerdir. Bu teknoloji, bir DNA dizisindeki genin fonksiyonunun ya da işlevinin bulunmasını mümkün kılar [24]. Mikrodiziler moleküler tıp biliminin hastalıklara ait gen profillerinin elde edilmesi ve böylece hastalıklara ait genetik arka planın öğrenilmesi amacıyla kullanılırlar [25].

Pozitron Yayılım Tomografileri (PYT): Bir tür nükleer tıp görüntüleme metodu olan bu yöntemle canlı vücudundaki biyolojik aktivitenin üç boyutlu resmi elde edilir. Nükleer olarak takip edilebilen bir tür radyoaktif izotopun ilgili dokuda yoğunlaşmasından sonra, görüntüleme cihazı ile konarak PYT tarayıcısından kişiye ait veri toplanır ve bu veriler hastalık teşhisinde kullanılırlar [26].

İzleyen bölümde kütle spektrometrisi başta olmak üzere yüksek boyutlu verilerin yapısına ait genel bilgilere yer verilecektir.

2.2. Yüksek Boyutlu Verilerin Genel Yapısı

Geleneksel sınıflandırma problemleri, bir gözleme (örneğin başarılı, başarısız) veya bir medikal örneğe (örneğin hasta, sağlam) karşılık gelen değişkenlerle uğraşırlar. Sınıflandırma problemi, sınıflandırma etiketine en çok katkıda bulunan değişkenlerden bir veya bir kaçını belirlemeyi hedefler [27]. Geleneksel sınıflandırma problemlerinin matematiksel yapısı öznitelik sayısı olarak bilinen ve p ile sembolize edilen sütun sayılarının göreceli olarak sınırsız olduğu ve N ile temsil edilen satır sayısının yüzler mertebesinde olduğu bir matris şeklindedir [28]. Örneğin göğüs kanseri teşhisinde kullanılan ve düşük boyutlu bir veri olan mamografi tekniğine ait verinin bir bölümü Tablo 2.1'de gösterilmektedir.

Düşük boyutlu bir veri olan mamografi verisinin [29] sınıflandırılması, bir algoritma yardımıyla sınıf etiketi olarak bilinen durumun (bireyin kanserli veya sağlam olması) beş öznitelik cinsinden en çok hangisi veya hangileri ile ilgili olduğunun bulunması şeklinde

ifade edilebilir. Bu şekilde bir şüpheli bireyin beş adet ölçüme bağlı olarak hastalıklı veya sağlıklı olma tahminini gerçekleştirecek bir akıllı sistem tasarlanabilir.

Yüksek boyutlu veri setlerinde N değeri birkaç yüz satır mertebesindeyken, p onlarca hatta onbinlerce sütun değerine sahiptirler [30]. Geleneksel düşük boyutlu sınıflandırma problemlerinde sınıf etiketi mamografi örneğinde olduğu gibi az sayıdaki birkaç değişkene bağlı olarak belirlenir. Sınıflandırma algoritmaları düşük boyutlu verilerin sınıflandırılması esasına göre çalıştıkları için, kütle spektrometrisi örneğinde olduğu gibi yüksek boyutlu bir verinin analizi, p sayısının on binler mertebesinden birkaç yüz mertebesine indirgenmesini gerekli kılar [31]. Rahim kanseri kütle spektrometrisine ait verinin çok küçük bir bölümü Tablo 2.2'de gösterilmiştir.

Tablo 2.1. Göğüs kanseri mamografi deney sonuçları

Radyasyon Şiddeti	Yaş	Şekil	Ağırlık	Doku Yoğunluğu	Durum
5	67	3	5	3	Kanser
5	58	4	5	3	Kanser
4	28	1	1	3	Sağlam
4	36	3	1	2	Sağlam
4	60	2	1	2	Sağlam
4	54	1	1	3	Sağlam
3	52	3	4	3	Sağlam
5	57	1	5	3	Kanser
5	76	1	4	3	Kanser
3	42	2	1	3	Kanser

Tablo 2.2. Rahim kanseri kütle spektrometri verisi

	(k/y) 1	(k/y) 2	. . . . .	(k/y) 15000	Sınıf
Y1	0,0014	0,0031	. . . . .	0,0013	Kanser
Y2	0,0022	0,0019	. . . . .	0,0034	Sağlam
Y3	0,0078	0,0091	. . . . .	0,0079	Kanser
Y4	0,0087	0,0123	. . . . .	0,0088	Kanser
.	.	.	. . . . .	.	Sağlam
.	.	.	. . . . .	.	Kanser
Y 214	19845	19931	. . . . .	19833	Sağlam
Y 215	19856	19943	. . . . .	19872	Sağlam
Y 216	19877	19957	. . . . .	19895	Kanser

Tablo 2.2' de görülen rahim kanseri verilerinin [32] sınıflandırma problemi olarak ele alınabilmesi için, öznitelik seçme teknikleriyle verinin niteliğini kaybetmeden boyutunun düşürülmesi gerekir. Öte yandan boyutu onlar mertebesinde olan veri setleri

için de sınıflandırma performansının arttırılması için zorunlu olmasa dahi Kütle Spektrometrisi verilerinde olduğu şekilde boyut düşürme teknikleri kullanılabilmektedir. Özel olarak yüksek boyutlu verilerde kullanılsa da, tüm medikal verilerin bir şekilde öznitelik seçme veya öznitelik dönüşümü teknikleri ile boyutunun düşürülmesi gerekebilir. Bu çalışmada gerek yüksek boyutlu verilerin boyutunun düşürülmesinde ve gerekse sınıflandırma başarım performansının arttırılmasında kullanılan boyut düşürme teknikleri Bölüm 4'te detaylı şekilde ele alınacaktır.

2.3. Boyutları İndirgenerek Kullanılan Biyomedikal Verilerin Genel Yapısı

Bu çalışmamızda algoritmalarımızı test amacıyla kullanılan farklı kanser türlerine ait yüksek boyutlu kütle spektrometrisi verilerinin [12] yapısı aşağıdaki tablolarda gösterilmiştir.

Tablo 2.3. Rahim, prostat kanserlerine ve Arcen'e ait verilerin yapısı

Veri Seti	Çözünürlük	Örnek Sayısı		Veri Boyutu	
		Kanser	Kontrol	Öznitelik Sayısı	Örnek Sayısı
Rahim	Yüksek	121	95	15000	216
Prostat	Düşük	253	69	15154	322
Rahim	Yüksek	162	91	15154	253
Arcene	Düşük	390	310	10000	700

Tablo 2.3'de gösterilen Arcene [15], yine kütle spektrometrisi kaynaklı bir test verisidir. Arcene, SOM öğreticisiz akıllı sisteminin PSO tekniği ile eniyileştirme başarımını ölçmekte kullanılmıştır.

Tablo 2.4. Dermatoloji ve Parkinson hastalıklarının genel yapısı

Veri Seti	Örnek Sayısı			Öznitelik Sayısı
	Hastalık	Kontrol	Sınıfsız	
Dermatoloji	366	-	-	33
Parkinson	147	48	-	26

Bu çalışmada kanser dışında ayrıca dermatoloji ve Parkinson olmak üzere yüksek boyutlu iki farklı medikal verinin sınıflandırma başarımları incelenmiştir. Bu verilerin genel yapısı ise Tablo 2.4'de gösterilmiştir.

Dermatoloji ve Parkinson verilerinin algoritmalar ile analizinden önce boyutları özellik seçme yöntemleri ile düşürülmüş ve sınıflandırma başarımını en çok arttıran öznitelikler elde edilmiştir.

Tablo 2.4'de yer alan, Dermatoloji hastalığına ait 366 adet verinin tamamı altı farklı tipte deri hastalığına aittir ve bu veri seti belirtiler (öznitelikler) yardımıyla bu hastalıkların hassas şekilde birbirinden ayrılabilmesi için kullanılmıştır.

Parkinson ve dermatoloji hastalıklarına ait veriler UCI makine öğrenmesi veritabanından [15] alınan test amaçlı medikal verilerdir.

2.4. Bu Çalışmada Kullanılan Diğer Veri Setleri

Bu çalışmada öğreticili sistemlerin sınıflandırma başarımının ölçülmesinde kütle spektrometrisi deneylerine ait prostat, rahim kanseri verileri ile dermatoloji ve Parkinson hastalıklarına ait veriler kullanılmıştır. Öğreticisiz bir sistem olan SOM'ların kümeleme performansının ölçülmesinde Tablo 2.3'de verilen Arcene dışında, mamografi, diyabet, göğüs kanseri, protein lokalizasyon tespit verileri ile genomik veri setleri kullanılmıştır. İzleyen tablolarda bu veri setlerine ait örnek sayısı ve sahip oldukları öznitelik sayısı gösterilmiştir.

Tablo 2.5. Diyabet, göğüs kanseri ve mamografi iki sınıflı veri setleri

Veri Seti	Örnek Sayısı		Öznitelik Sayısı
	Hastalık	Kontrol	
Diyabet	500	268	8
Göğüs Kanseri	241	453	10
Mamografi	516	445	6

Tablo 2.5'de yer alan iki sınıflı veri setleri yine UCI makine öğrenmesi veritabanından [15] alınmıştır. SOM algoritmasının kümeleme başarımının ölçülmesinde veri çeşitliliğini arttırmak için sınıf sayısı ikiden fazla olan protein lokalizasyon sitelerini bulmakta kullanılan *maya* ve *koli basili* veri setlerinin yapısı Tablo 2.6'da verilmiştir.

Tablo 2.6. Maya ve Koli Basili protein lokalizasyon site verilerinin yapısı

Veri Seti	Sınıf Sayısı	Öznitelik Sayısı	Veri Sayısı
Maya	10	8	1484
Koli Basili	8	8	336

Karmaşıklığı azaltmak için, Tablo 2.6'da her sınıfa ait örnek sayısına yer verilmemiş bunun yerine toplam örnek sayısı tabloya dahil edilmiştir.

Tablo 2.6'daki veriler UCI veritabanından alınmış olup [15], bu veri setleri protein lokalizasyon sitesi adı verilen yapıların belirlenmesi amacıyla kullanılmaktadır. Protein lokalizasyon siteleri, proteinlerin hücre içindeki pozisyonlarını ifade eden bölgelerdir. Bu siteler proteinin hücre içindeki fonksiyonlarını bilmekte kullanılmaktadırlar [33].

Bir tür genomik örüntü elde etme problemi olan, motiflerin bulunması için kullanılan ve dört farklı canlıya ait olan veri setleri ise Tompa ve diğ.[34] çalışmasından alınmıştır. Motifler, DNA dizilerinde tekrar eden ve önemli biyolojik aktivitelerde rol oynadığı düşünülen kısa nükleotid dizileridirler [35]. Bu çalışmada SOM kümeleme başarımını test amacıyla kullandığımız genomik verilerin yapısı Tablo 2.7'de gösterilmiştir.

Bu bölümde içerdikleri öznitelik sayısını referans alarak gruplanan *yüksek* ve *düşük* boyutlu iki grup veri farklı medikal alanlardan seçilmiş ve bu şekilde test edilen akıllı sistemlerin sınıflandırma başarımlarının ölçümünde güvenilirlik sağlanmaya çalışılmıştır.

Tablo 2.7. Genomik örüntü bulmakta kullanılan test amaçlı veriler

Veri Seti	Canlı Türü	Veri Uzunluğu	Motif Uzunluğu	Örüntü Sayısı
CBF1	Maya	12159	7	65
LEXA	Koli Basili	4715	20	8
DM05	Meyve Sineği	7466	12	14
HM17	İnsan	5328	16	10

Bir sonraki bölümde, kütle spektrometri verisine ait ön-işleme adımlarına yer verilecektir.

3. KÜTLE SPEKTROMETRİSİ VERİLERİNİN ÖN-İŞLENMESİ

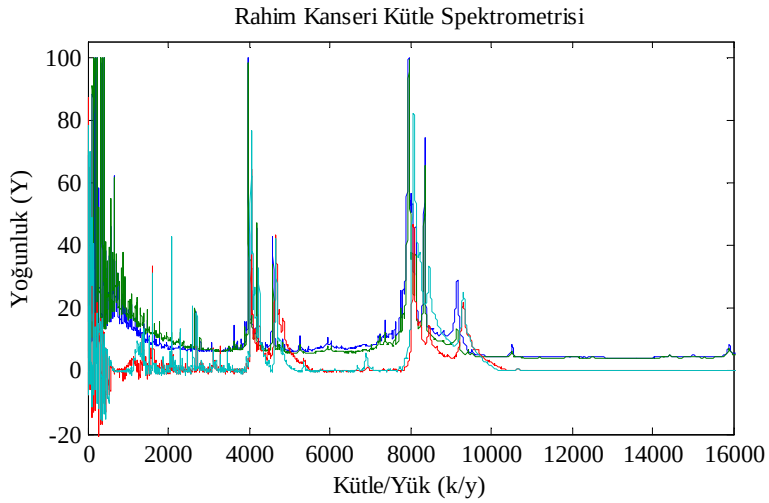
Kütle spektrometrisi, serum ve üre gibi biyolojik sıvılardaki protein yapıları arasında belirgin protein örüntüleri aramakta yaygın şekilde kullanılan yeni bir biyomedikal tekniktir. Protein örüntüleri genel olarak hastalıklı-sağlıklı gibi iki ayrı sınıfa ait biyolojik örneğin kıyaslanması ile elde edilmektedir. Bulunan protein örüntüleri, medikal teşhis sistemlerinin geliştirilmesi ve hastalıkların gelişim seyrinin tespiti gibi alanlarda kullanılma potansiyeli taşımaktadırlar [36,37].

Kütle spektrometrisi, bilinmeyen bileşenlerin içeriğinin belirlenmesi, bileşiklerin içeriğinin nicel seviyelerinin ölçülmesi ve moleküllerin kimyasal yapılarının açığa kavuşturulmasında güçlü bir tekniktir. Kütle spektrometri deneyinin gerçekleştirilmesine ait kısa bilgi ve ortaya çıkan ham verinin ön-işleme adımları izleyen alt bölümlerde izah edilecektir.

3.1. Kütle Spektrometri Deneyi

Bir kütle spektrometrisi deneyi için hastalardan alınan örnekler, enerji emiş gücü yüksek matris adı verilen *sinamik* asit türevleri ile karıştırılırlar [38]. Bu tür asitler, lazer taramasında ortaya çıkan yüksek enerjiyi emerek örneklerdeki proteinlerin bozulmasını önlemek için kullanılırlar. Matrisle karıştırılan örnekler bir metal levha üzerine konur ve karışımın çözücüsü vakum ortamında birbirinden ayrılarak kristalize matris/protein karışımı elde edilir. Bu karışım nitrojen lazeri ile tarandığında, matris aldığı enerji ile gaz fazına geçerek iyonize olurken, karışımda yer alan protein molekülleri de matrisle birlikte buharlaşır ve yüklenmiş olurlar [39]. İyonize olmuş protein molekülleri, kendilerine elektriksel bir alanın uygulandığı, doğrusal uçuş cihazına yönlendirilirler. Uygulanan elektriksel alan nedeniyle iyonize olmuş protein molekülleri vakum ortamında kendilerini tespit eden bir detektöre çarpıncaya dek uçar ve bu uçuş süresi kayıt edilir. Uygulanan elektrik alanın şiddeti ve uçuş tüpünün uzunluğu bilindiği için, uçan protein iyonlarının detektör tarafından tespit edilmesine kadar geçen süre protein moleküllerinin ağırlığına bağlı olacaktır [40]. Farklı protein molekülleri farklı kütlelere sahip olduğu için, elde edilen veriler tüm örneklerin kütle/yük (k/y) dağılımına karşılık gelecektir. Bu spektrum, detektör tarafından yakalanan iyon sayısı (yoğunluk-Y) ve buna karşılık gelen k/y değerleri

ile meydana gelir [41]. Düşük çözünürlüklü rahim kanserine ait örnek bir yoğunluk-kütle/yük dağılım grafiği Şekil 3.1'de gösterilmektedir. Orijinal rahim kanseri dağılımı dört farklı oturumda gerçekleştirilen deneyin sonuçlarını birleştirdiği için birbirine benzeyen dağılımlardan oluşmaktadır. Ön-işleme adımlarının etkisinin görsel olarak daha net gözlenebilmesi için baz çizgisinin doğrultulması ve gürültünün giderilmesi adımları sadece tek sinyal için gerçekleştirilmiştir. Bölüm 3.4'de ele alınan normalizasyon adımı dört spektrumun birbiri arasındaki göreceli sinyal genliği farkını gidermek için yapıldığından, şekillerde tüm sinyaller gösterilecek ancak takibi kolaylaştırmak için tüm dağılımın sadece bir bölümüne yer verilecektir.



Şekil 3.1. Düşük Çözünürlüklü Rahim Kanseri Kütle Spektrometrisi

Kütle spektrometri deneyi kısa ve belirli bir zaman aralığında örneklerin lazerle tarandığı ardışık lazer ateşlemeleri ile gerçekleştirilir. Her lazer ateşlenmesi sırasında ortaya çıkan iyon sayısı binlerce vektörden oluşan bir veri üretir. Tipik bir deney birkaç milisaniye içinde tamamlanırken her bir lazer ateşlemesi birkaç nano-saniye sürer. Bu durumda karakteristik bir kütle spektrometri deneyi ortalama olarak 10 000 ila 100 000 arasında ham özniteliğe sahip olur [42]. Matematiksel bir ifadeyle özgün bir deney sonucunda, 200-300 civarında biyolojik örnek satırına karşılık 10000-100000 arasında öznitelikten oluşan bir matris elde edilir.

Ham kütle spektrometri verisinin biyomedikal anlamda yorumlanabilmesi, bir grup ön-işleme adımın gerçekleştirilmesine bağlıdır. Genel olarak literatürde sıklıkla kullanılan

ön-işleme adımları; elektronik cihazlardan veya kimyasal reaksiyonlardan kaynaklanan gürültünün filtrelenmesi, deneyin ilk aşamalarında iyonize olmuş matris moleküllerinin detektörü yüklemesiyle ortaya çıkan baz çizgisinin giderilmesi ve örneklerin anlamlı şekilde kıyaslanabilmesi için verinin normalizasyonu şeklindedir [43].

Ön-işlemeden geçirilmemiş ham bir kütle spektrometri verisi matematiksel olarak Denklem 3.1 ile ifade edilebilir [44].

$$f(t) = B(t) + N \cdot S(t) + \varepsilon(t) \tag{3.1}$$

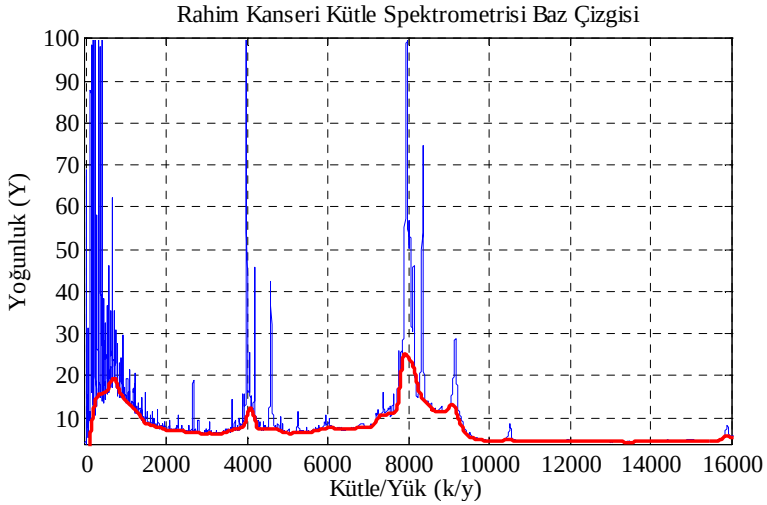
Bu denklemde, $f(t)$ kütle spektrometri sinyalini, $B(t)$ baz çizgisini, $S(t)$ orijinal protein sinyalini, N normalizasyon faktörünü ve $\varepsilon(t)$ gürültüyü temsil eder. Denklem 3.1 dikkate alındığında ön-işleme sürecinde $S(t)$ dağılımını elde etmek için baz çizgisi $B(t)$ ve gürültü ifadesi $\varepsilon(t)$'nin dağılımdan ayrılması ve daha sonra dağılımın normalize edilmesi gerekir. İzleyen bölümde, iki farklı çözünürlüklü rahim kanseri ve prostat kanserine ait kütle spektrometri verilerine uygulanan üç ön-işleme adımı sırasıyla incelenecektir. Ön-işleme adımları her üç kanser türüne aynı şekilde uygulanmış olmakla beraber, detaylı açıklamalar düşük çözünürlüklü rahim kanserine ait dağılım üzerinde gösterilecektir. Yüksek çözünürlüklü rahim ve prostat kanserlerine ait uygulamalara tekrarı önlemek için yer verilmeyecektir.

3.2. Baz Çizgisinin Giderilmesi

Kütle spektrometrisi deneyinde detektör kendisine çarpan iyonize moleküllerden dolayı aşırı yüklenir ve bu yüklenme tüm dağılımın sıfır ekseninden yukarı doğru kaymasına neden olur. Asıl sinyalin üzerine bindiği bu üstel çizgiye kütle spektrometri tekniğinde baz çizgisi adı verilir. Denklem 3.1'de $B(t)$ ile gösterilen bu kaymanın orijinal protein dağılımını bozmadan spektrometriden ayrılması önemli bir ön-işleme adımıdır.

Spektrometride baz çizgisinin giderilmesi için regresyon tabanlı matematiksel yaklaşımlar kullanılmaktadır [45]. Bu matematiksel yöntemler öncelikle tüm dağılımı kütle/yük (k/y) boyunca pencerelere bölerek o pencere içinde ortalama bir baz çizgisi noktası tahmin ederler. Sonraki adımda interpolasyon ile her pencere için tahmin edilen

baz çizgisi noktalarından geçen bir eğri elde edilir [46]. Bu şekilde rahim kanseri dağılımı için oluşturulan örnek baz çizgisi Şekil 3.2'de kırmızı çizgi ile gösterilmiştir.



Şekil 3.2. Rahim kanseri spektrometrisi için tahmini baz çizgisi

Baz çizgisinin giderilmesi için iki aşamalı bir algoritma kullanılır:

i) Şekil 3.2'de görülen ve kırmızı ile çizilen üstel dağılımın mavi ile çizilen ana spektrumdan doğru şekilde ayırt edilmesi için matematiksel eğri uydurma teknikleri kullanılır ve böylece baz çizgisi dağılımı elde edilir.

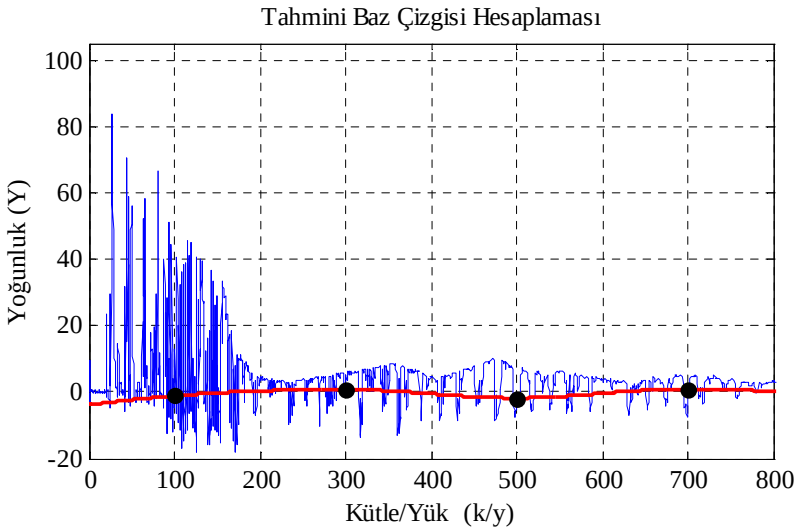
ii) Tespit edilen baz çizgisi kütle spektrometrisinden çıkarılarak kütle spektrumunun orijinal haline (sıfır noktasına) yaklaşması sağlanır.

Bu çalışmada rahim kanseri veri seti için baz çizgisi doğrultması şu şekilde gerçekleştirilmiştir:

i) Rahim kanseri verisine ait baz çizgisi eğrisinin geçeceği ortalama noktaların belirlenmesi için tüm dağılım kütle/yoğunluk ekseni boyunca 200 k/y genişliğinde pencerelere bölünür.

ii) Her 200 k/y genişliğindeki pencerede yer alan asıl sinyale ait noktaların ortalaması alınarak o pencerede baz çizgisi için bir referans nokta elde edilir. Daha sonra, o noktalardan geçen bir eğri uydurulur. Bu çalışmada eğri uydurma işlemi için kübik spline yöntemi kullanılmıştır. Şekil 3.3'te her pencere için hesaplanan ortalama noktalar siyah ile gösterilmiş ve bu noktalardan geçen uydurulmuş kübik spline eğrisi kırmızı ile çizilmiştir.

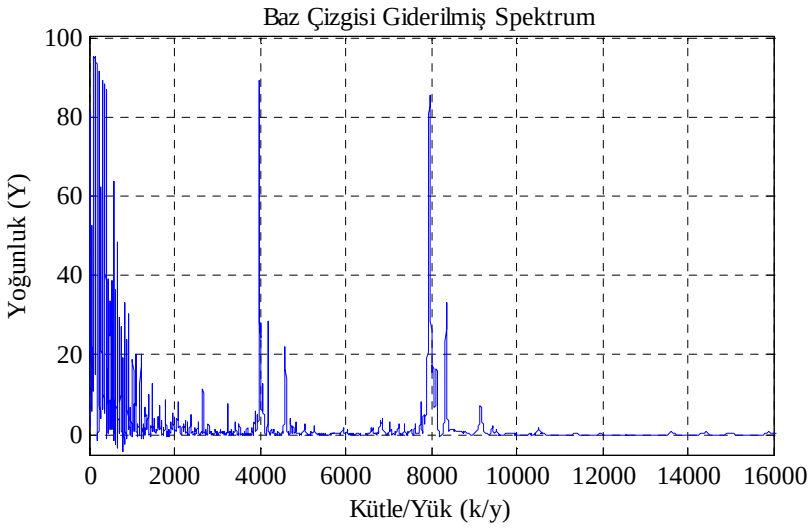
iii) Elde edilen baz çizgisi, asıl dağılımdan çıkarılarak sinyal sıfır eksenine yaklaştırılır. Rahim kanseri dağılımına ait baz çizgisi düzeltilmiş yeni kütle spektrometrisi Şekil 3.4'te gösterilmiştir.



Şekil 3.3. Rahim kanseri için hesaplanan baz çizgisi

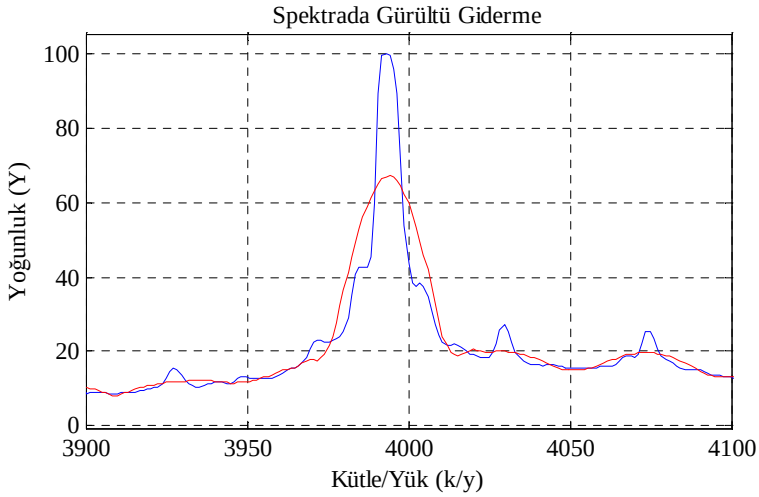
3.3. Gürültünün Giderilmesi

Standart bir kütle spektrometrisi sinyali genel olarak elektronik cihazlardan veya örneklerdeki kimyasal reaksiyonlardan kaynaklanan bir gürültü içerir. Denklem3.1'de ε(t) ile gösterilen bu bileşenin protein sinyalinden ayrılması sırasında, sınıflandırma için gerekli olan protein sinyalindeki tepe değerlerinin hassaslığının korunması önem taşır. Gürültü bileşenini gidermek için en yaygın kullanılan yöntemler istatistiksel bir yöntem olan yerel ağırlıklı çizim yumuşatma, dalgacık yardımıyla filtreleme ve Savitzky-Golay filtrelemesidir [47]. Bu çalışmada gürültünün giderilmesi için, bir *en küçük kareler polinomu* yöntemi olan Savitzky-Golay filtreleme algoritması kullanılmıştır. Bu yöntem, k/y değerlerini referans alan bir kayan pencere yöntemi olup, her penceredeki verinin ortalama değerini hesaplar ve hesaplanan noktalara göre ikinci dereceden bir polinom uydurur:



Şekil 3.4. Baz çizgisi giderilmiş rahim kanseri kütle spektrometrisi

Çalışmamızda, Rahim kanseri dağılımı 35 k/y uzunluğunda pencerelere bölünmüş ve bu pencerelerdeki kütle spektrometri verilerinin ortalama değeri hesaplanarak eğri uydurmak için kullanılacak tahmini noktalar elde edilmiştir. Daha sonra tahmini noktalardan ikinci dereceden bir polinom uydurulmuş ve dağılım daha düzgün bir formda yeniden elde edilmiştir.



Şekil 3.5. Rahim kanseri spektrumundan gürültünün giderilmesi

Şekil 3.5'de gürültüden arındırılarak yumuşatılan eğri kırmızı ile ve gürültü içeren orijinal rahim kanseri dağılımı mavi ile çizdirilmiştir.

3.4. Spektrumun Normalizasyonu

Kütle spektrometrisine ait verilerinin sınıflandırma analizinin en doğru biçimde gerçekleştirilebilmesi ön-işleme adımlarının sağlıklı şekilde yapılması ile yakından ilgilidir. Bir kütle spektrumunda, protein moleküllerinin yıkıma uğrayarak zamana bağlı değişiminden veya detektör hassaslığının değişiminden kaynaklanan ve ölçülen iyon yoğunluklarını etkileyen sistematik etkiler gözlenir. Kütle spektrometrisi deneyinde kullanılan ve dağılımı elde edilen örnekler arasında sağlıklı bir yoğunluk karşılaştırması yapabilmek için örneklerdeki sistematik hataların bulunması ve daha sonra da bu hataların normalize edilmesi gerekir [48]. Normalizasyon bu şekliyle tüm dağılımdaki tepe değerlerinin kullanıcı tarafından belirlenen bir maksimum değere göre yeniden ölçeklenmesi olarak ifade edilir [49]. Örnekler arasındaki sistematik hatalar, doğru bir kıyaslama yapılmasına engel olur. Spektranın normalizasyonu, gerçek protein yoğunluklarına karşılık gelen tepe değerlerin bir tür gürültü kabul edilebilecek sistematik hatalardan ayrılması olarak tanımlanır [50]. Sınıflandırma problemi olarak ele alındığında, kütle spektrometrisinde amaç hastalıklı ve sağlıklı örneklerdeki protein yoğunluklarının kıyaslanması şeklinde ifade edilir. Bu durumda, sinyaldeki protein yoğunluklarını temsil eden tepe değerlerinin sınıflandırma çalışmasını etkilemeyecek şekilde tüm dağılımın göz önüne alınarak normalizasyonuna ihtiyaç duyulur. Bu açıdan bakıldığında normalizasyon kütle spektrometri sinyalinin sınıflandırılmasını kolaylaştıracak tarzda sınıfsal şekilde güçlendirilmesini sağlar [51]. Tipik bir spektrometri sinyali şu şekilde normalize edilir:

i)Öncelikle tüm spektrometri sinyalinin altındaki alan eğri altında kalan alan (EAA) yaklaşımı ile tüm dağılımın alanı bulunur [52].

ii)Spektrometrinin ya da sinyal eğrisinin altında kalan alan sinyalin ortalama değerine bölünür [53].

Baz çizgisinin düzeltilmesi ve gürültünün giderilmesinden sonra gerçek protein sinyali $S(t)$, $f(t)$ ham sinyal ve N normalizasyon faktörü olmak üzere Denklem 3.2'deki gibi elde edilir.

$$S(t) = f(t)/N \tag{3.2}$$

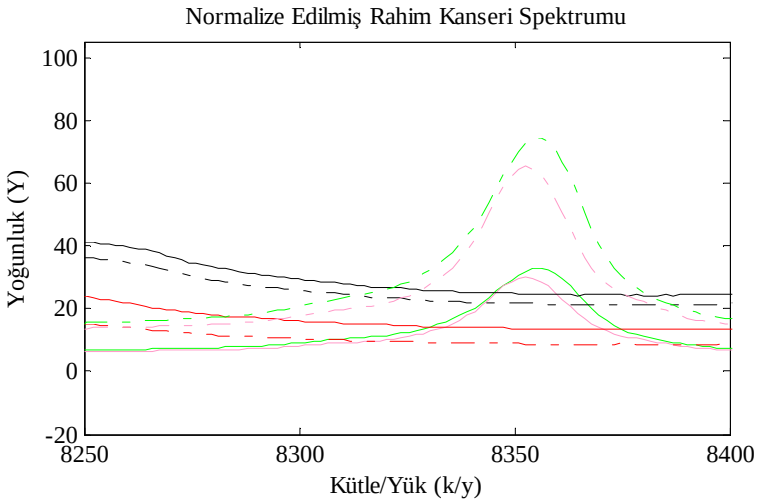
Baz çizgisi ve gürültü giderildikten sonra kalan normalize edilmemiş sinyal ($T(t)$)kendisini oluşturan yoğunluklar türünden Denklem 3.3' deki gibi gösterilebilir.

$$T(t) = [x_1, x_2, ..., x_n] \tag{3.3}$$

Denklem 3.3'te, vektörler k/y eksenini temsil etmektedir. $T(t)$ sinyalinin çizdiği eğrinin altında kalan alan yaklaşık olarak $EAA(\sum T(t))$ ve sinyali oluşturan toplam protein iyonu yoğunluklarına ait ortalama (tüm yoğunluk dağılımını ortadan ikiye bölen değer) değer $\mu_{1/2}(T(t))$ olmak üzere normalizasyon faktörü (N), Denklem 3.4 ile verilmiştir.

$$N = \frac{EAA(\sum_{i=0}^{\infty} T(t))}{\mu_{1/2}(T(t))} = \frac{\int_0^{\infty} I(y) \cdot d(k/y)}{\mu_{1/2}(T(t))} \tag{3.4}$$

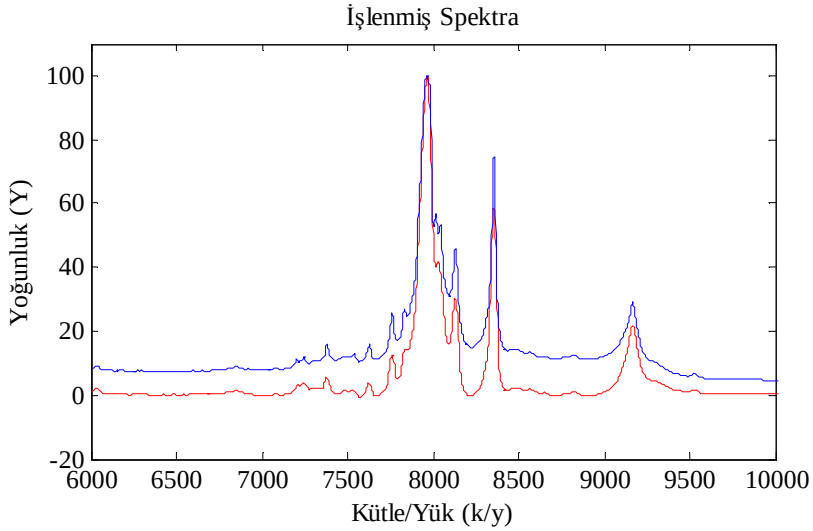
Rahim kanseri spektrumunun normalizasyon öncesinde ve sonrasındaki görünümleri Şekil 3.6'daki gibidir.



Şekil 3.6. Normalize edilmiş rahim kanseri spektrumu

Şekil 3.6'da dört sinyal siyah, kırmızı, yeşil ve pembe renklerle çizilmiştir. Burada, kesikli çizgiler orijinal spektrumu, sürekli çizgiler sinyallerin normalize edilmiş halini göstermektedir. Şekil 3.6'daki sinyallerin yoğunluk eksenindeki 8350 k/y civarına dikkat edildiğinde, yüksek yoğunluklu pembe, yeşil çizgilerin yoğunluğunun azalmasına karşın, düşük yoğunluklu siyah ve kırmızı çizgilerin yoğunluğunun arttığı görülmektedir. Normalizasyonun amacı sinyallerin birbirine kıyasla yoğunluklarının dengelenmesidir. Bu perspektiften sinyallerin arzu edilen şekilde yeniden ölçeklendirildiği açıkça fark edilmektedir.

Düşük çözünürlüklü rahim kanserine ait ön-işleme adımlarının ardışık olarak dağılıma uygulanmasıyla elde edilen sonuç Şekil 3.7'de gösterilmiştir. Şekil 3.7'de mavi çizim orijinal spektrayı, kırmızı çizim ise gürültüden arındırılmış, baz çizgisi giderilmiş ve normalize edilmiş $S(t)$ sinyalini göstermektedir.



Şekil 3.7. Ön-işleme adımları gerçekleştirilmiş rahim kanseri spektrumu

Bu bölümde izah edilen ön-işleme adımları iki rahim kanseri türüne ve prostat kanserine izleyen adımlardaki gibi uygulanmıştır:

i) Her üç veri seti baz çizgisi doğrulamasına tabi tutulmuştur.

ii)Daha sonra gürültü işleme adımı uygulanmış ve elde edilen sonuçlar bu şekliyle saklanmıştır.

iii)Son aşamada ise üç tür veri normalize edilmiş ve bu veriler saklanmıştır.

İlk aşamada, baz çizgisi düzeltme, gürültünün giderilmesi ve normalizasyon adımları kütle spektrometrisi verilerine birbirinden bağımsız şekilde ayrı ayrı uygulanmıştır. Daha sonraki adımda, ön-işleme adımları verilere ikişer ikişer uygulanarak elde edilen sonuçlar bu şekilde saklanmıştır. Son adımda ise, ön-işleme adımlarının üçü her veriye arka arkaya uygulanmış ve bağımlı şekilde ele alınarak elde edilen sonuçlar kayıt edilmiştir. Ön-işleme adımlarının uygulanmasından elde edilen bu sonuçlar, farklı sınıflandırıcılarla analiz edilerek, her ön-işleme adımının tek tek veya birlikte sınıflandırma başarımına nasıl etki ettiği incelenmiştir. Yapılan deneylerle ilgili sonuçlar ve ön-işleme adımlarının sınıflandırma başarımına etkileri Bölüm 7'de detaylı olarak verilecektir. İzleyen bölümde yüksek boyutlu verilerde ve yüksek boyutlu verilerde boyut indirgeme yöntemleri ele alınacaktır.

4. YÜKSEK BOYUTLU VERİLERDE BOYUT İNDİRGEME PROBLEMİ

Yeni nesil biyomedikal teknolojiler bir hastadan alınan örnekte veya hastada aynı anda yüzlerce hatta binlerce özelliği ölçebilecek şekilde imkanlar sunmaya başlamışlardır. Kütle spektrometrisi, gen teknolojileri, kardiyoloji teknikleri ve tomografi teknolojileri yüksek boyutlu veri üreten biyomedikal teşhis sistemlerine örnek olarak verilebilirler. Sınıflandırma problemi ekseninde ele alındığında bu teknolojiler bir hastaya veya bir hastadan alınan örneğe karşın onlar mertebesinden yüz binler mertebesine dek öznitelik oluşturabilirler. Matematiksel bir ifadeyle aralarında $p >> N$ bağıntısı olan N örnek sayısına karşılık çok sayıda p öznitelik ya da yüksek boyutlu veri üretirler. Bir yüksek boyutlu veri p değerinin N örnek sayısına göre büyük olmasıyla tanımlanmasına karşın bu iki parametrenin aralarında somut bir bağıntı tanımlamak mümkün değildir. Sınıflandırma problemlerinde temel kriter sınıflandırma başarımını düşürmeyen en küçük öznitelik sayısını elde etmektir. Bu eksende düşünüldüğünde birkaç özniteliğe sahip bir veri setinin daha az öznitelikle sınıflandırma başarımı arttırılabiliyorsa bu durumda öznitelik indirgeme işlemi gerçekleştirilmelidir.

Akıllı sistemler veya istatistiksel tekniklerin çoğunda başarımını arttırmak, yüksek sayıdaki öznitelik yerine daha az sayıda öznitelik kullanmakla mümkündür. Bu nedenle çalışmamızda kullandığımız kanser kütle spektrometrisi, Parkinson ve dermatoloji gibi yüksek boyutlu verilerin geleneksel algoritmalar ile analizi sırasında veri öncelikle olabildiğince düşük boyuta indirgenmiştir [54]. Çoğu sınıflandırma probleminde uygun özniteliklerin seçilmesi, sınıflandırmada tüm özniteliklerin kullanılmasına göre daha yüksek sınıflandırma başarımının elde edilmesini sağlamaktadır. Bu bağlamda boyut indirgeme yüksek boyutlu verilerde hastalık teşhisinde geleneksel sınıflandırıcı algoritmaların kullanılabilmesini mümkün kılarken, sınıflandırma başarımının arttırılması için de bir ön şarttır [55].

4.1 Kütle Spektrometrisi Veri Analizinde Boyut İndirgeme

Bu çalışmada kullanılan Parkinson ve dermatoloji veri setlerinden farklı olarak kütle spektrometrisinin analizinin gerçekleştirilebilmesi için ilk aşama olarak verinin ön-işleme

adımlarına tabi tutulması gerekir. İkinci adımda ise gerek kütle spektrometrisi verileri gerekse diğer veri setlerinin veri boyutunun azaltılması ve daha sonra akıllı sistem algoritmalarıyla sınıflandırılması gerekmektedir. Yüksek boyutlu verilerin boyutunun azaltılması farklı teknikler yardımıyla gerçekleştirilmektedir. İzleyen bölümde önce boyut indirgeme problemi tanımlanacak ve daha sonra boyutun azaltılması için kullanılan tekniklerden bahsedilecektir.

4.2. Boyut İndirgeme Probleminin Tanımlanması

Boyut indirgeme problemi matematiksel olarak şu şekilde ifade edilebilir: Bir grup n vektörü $\{x_1, x_2, ..., x_n\}\ \varepsilon\ R^p$ şeklinde verilsin. Bu vektördeki bilgiyi maksimum şekilde saklayabilecek bir daha düşük boyutlu bir vektör kümesini bulmak, boyut indirgeme şeklinde tanımlanır. Matematiksel ifadeyle boyut indirgeme $p > p'$ şartını sağlayan ve sınıflandırma başarımının artması gibi bir kritere bağlı olarak $\{z_1, z_2, ..., z_n\}\ \varepsilon\ R^{p'}$ ile ifade edilen daha az sayıda özniteliğe sahip yeni bir vektör dizisi elde etmek şeklinde tanımlanır [56].

Boyut indirgeme probleminde iki temel yaklaşım vardır:

i)İlk yaklaşım, öznitelikler (p boyut vektörlerinin) birleştirilerek veya dönüştürülerek daha az sayıda vektör elde edilmesini sağlayan dönüşüm tekniklerinden oluşur [57]. Bu teknikler özniteliklerin tamamını kullanarak boyut düşürme esası üzerine çalışırlar.

ii)Bir diğer boyut indirgeme yaklaşımı ise tüm öznitelikler arasından bir kriter yardımıyla özniteliklerin bir alt grubunu seçerek çalışır. Bir diğer ifadeyle öznitelik seçme, istatistiksel seçme yöntemleri kullanılarak *p* vektörlerinden bir bölümünün daha belirleyici veya daha ayırıcı bilgi taşıyan vektörler şeklinde seçilmesidir [58].

4.2.1. Öznitelik Dönüştürme

Öznitelik dönüştürme teknikleri R^p vektör uzayından $R^{p'}$ uzayına vektörler arasındaki ilişkileri yeniden tanımlayarak $p > p'$ olacak şekilde daha az sayıda öznitelik elde etmeyi sağlayan yöntemleri kapsar. Öznitelik dönüştürme teknikleri öğreticili ve öğreticisiz olmak üzere iki grupta ele alınır.

i)Öğreticisiz Dönüştürme: Yüksek boyutlu verilerin düşük boyuta indirgenmesinde çok değişik yöntemler olmakla birlikte literatürde bu alanda en sık kullanılan iki yöntem *PCA* ve *Dalgacık Dönüşümü* (Wavelet Transform, WT) teknikleridir [59-60]. Bu dönüşümlerin ortak özelliği dönüşüm sırasında sınıf etiketleri (örneğin hastalıklı-sağlıklı) ayırımını dikkate almamalarıdır.

ii)Öğreticili Dönüştürme: Boyut dönüştürmesi yaparken verilerdeki sınıf yapısını dikkate alan bu yöntemlerden en çok bilineni LDA metodudur [59]. Yine bu grupta bir diğer öğreticili dönüştürme tekniği olan *Parçalı En-küçük Kareler* (Piecewise Least Squares, PLS) yöntemi de bir tür regresyondur. Bu teknik dönüştürmeyi yaparken giriş uzayındaki vektörlerden varyansı en çok küçülten bir alt-kümeyi seçer [61].

Bu çalışmada öznitelik dönüştürme yöntemi olarak öğreticisiz dönüştürme teknikleri arasından literatürde en çok kullanılan PCA seçilmiştir. PCA öznitelik dönüştürme işlemi şu şekilde çalışmaktadır:

PCA bir yüksek boyutlu verinin boyutunu orijinal veride öznitelikler arasındaki varyasyonu olabildiğince muhafaza ederek düşürür. Bu dönüşüm $X = [x_1, x_2, ..., x_p]$ orijinal veriye ait öznitelikler arasında lineer kombinasyonlar oluşturarak daha düşük boyutlu bir $T = [t_1, t_2, ..., t_k]$ öznitelik dizisi elde etmeyi hedefler. Burada giriş vektörü X boyutu ile yeni öznitelik dizisi T vektörünün boyutu arasında $p > k$ ilişkisi bulunur. PCA, Denklem 4.1'de tanımlandığı gibi giriş özeliklerinin arasındaki varyansı maksimize edecek şekilde, giriş özniteliklerinin doğrusal kombinasyonlarını alır.

$$u_k = \arg\max_{u^T u=1}(Var(Xu)) \tag{4.1}$$

Bu ardışık işlem doğrusal kombinasyonların kovaryansının sıfır olması şartına bağlıdır. Daha açık bir ifadeyle, PCA'nın oluşturduğu doğrusal kombinasyonlar $u_i^T S_X u_j = 0, \forall\ 1 \le i < j$ ortogonallık şartı sağlanacak şekilde oluşturulur. Bu şekilde $Cov(Xu_i, Xu_j) = 0,\ i \ne j$ sağlanmış olur. Bu şart altında elde edilen u_i katsayıları kullanılarak X giriş vektörlerinden $t_i = Xu_i$ eşitliği uyarınca 'k' adet daha az sayıda yeni öznitelik elde edilmiş olur. Elde edilen bu yeni öznitelikler *Temel Bileşenler* olarak adlandırılırlar [62].

Kütle spektrometrisi kanser verilerinin boyutları düşürülürken öğreticisiz bir yöntem olan PCA tekniğinin yanı sıra aynı verilere sınıf etiketlerini dikkate alan ve öğreticili bir dönüştürme yöntemi olan LDA tekniği de uygulanmıştır. LDA hem bir boyut dönüştürme hem de bir sınıflandırma yöntemi olarak kullanılabilmektedir [63]. Literatürde en çok kullanılan öğreticili dönüştürme yöntemi olan LDA'nın yapısı şu şekildedir:

LDA, X giriş vektörlerini, PCA yöntemine benzer şekilde U gibi bir dönüşüm yardımıyla boyutu daha küçük olan T vektörüne indirger. Matematiksel bir ifadeyle, $LDA(T) = XU$ eşitliğinde T vektörünü elde için edebilmek için sınıflar arasındaki ayrımı artıran ve aynı sınıf içindeki varyansı azaltan U doğrusal dönüşümü gerçekleştirilir [63]. Ancak bu dönüşümün uygulanabilmesi için kütle spektrometrisi örneğinde olduğu gibi verinin öncelikle PCA gibi bir dönüşüm veya bir öznitelik seçme yöntemiyle boyutunun U dönüşümüne izin verecek şekilde düzenlenmesi gerekir [64]. Bu çalışmada üç kütle spektrometri verisinin önce PCA dönüşümü uygulanarak boyutları düşürülmüş ve daha sonra bu verilere LDA dönüşümü uygulanmıştır. Bu deneylere ait sınıflandırma sonuçları Bölüm 7'de detaylı olarak verilmiştir.

4.2.2. Öznitelik Seçme

Yüksek boyutlu verilerde öznitelik sayısının azaltılmasındaki diğer yöntemler öznitelik seçme olarak bilinen teknikleri kapsar. Bu gruptaki teknikler, öznitelik dönüşümünden farklı olarak sisteme girilen öznitelikleri dönüştürmeden belli bir seçme kriterine bağlı olarak elemek şeklinde uygulanırlar [65]. Öznitelik seçme yöntemleri tek değişkenli yöntemler, çok değişkenli yöntemler ve zarflayıcılar olmak üzere üç grupta incelenirler. Tek değişkenli teknikler, giriş uzayındaki vektörlerin sınıflandırma başarımına etkilerini hesaplarken, etkin öznitelikleri birbirinden bağımsız gibi kabul eder ve istatistiksel bir kritere göre tüm öznitelikleri sıralarlar. Bu sıralama sonucunda ortaya çıkan öznitelik listesinden, uygulayıcı tarafından belirlenen sayıda öznitelik seçilerek bir test algoritmasında bu seçim sınanır. Çok değişkenli tekniklerde giriş uzayındaki vektörlerden sınıflandırma amacıyla seçileceklerin birbirine bağlı olabileceği varsayımıyla öznitelikleri seçilirler. Zarflayıcılar ise, GA gibi bir arama algoritması ile bir sınıflandırıcının birleşmesinden oluşan tekniklerdir. Öznitelik sayısının birkaç yüz civarında olması durumunda GA kullanılabilirken, kütle spektrometrisi örneğinde olduğu gibi on binlerce özniteliği olan bir veri seti için tek değişkenli seçme yöntemiyle özniteliklerin sayısı 200-

300 civarına indirgenir. Daha sonra GA ile bu öznitelikler arasından seçim yapılır. Tüm özniteliklerin GA tarafından seçilmesi çok ciddi bir hesaplama yüküne karşılık geleceği için kütle spektrometrisi gibi verilerde GA algoritmasından önce tek değişkenli öznitelik seçme yöntemlerinin uygulanması bir zorunluluktur. Bu şekilde GA algoritmasının sınıflandırma performansı en yüksek olan birkaç özniteliği seçmesi sağlanır [66].

i)*Tek değişkenli öznitelik seçimi*: Bu yöntemde, bir veri setindeki iki ayrı sınıftaki özniteliklerin bir istatistiksel yönteme bağlı olarak farklarının ölçülmesi ve bu farkı en çok gösteren özniteliklerin sınıflandırmada kullanmak için ayrılması işlemi gerçekleştirilir. Bu sıralama işlemi için t-testi, Kolmogorov-Smirnov testi, Mann-Whitney testi ve P-testi gibi istatistiksel kriterler kullanılır [67]. Testlere göre sıralanan özniteliklerden, kullanıcının belirlediği bir eşik değerinin üzerinde kalanlar seçildikten sonra sınıflandırma işleminde kullanılırlar. Bu çalışmada kütle spektrometrisine ait özniteliklerden en belirgin olan 150 özniteliğin seçimi için t-testi kullanılmıştır. Sıralama işleminde kullanılan t-testine ait matematiksel ifade Denklem 4.2 ile verilmiştir.

$$t = \left(\frac{2Fn_1n_2}{n_1 + n_2}\right)^{1/2} \tag{4.2}$$

Denklem 4.2'deki F, Fisher oranı olarak bilinir ve iki sınıflı bir veri seti için Denklem 4.3'deki ifadeyle hesaplanır.

$$F_j = \frac{(1m_j - 2m_j)^2}{(1v_j - 2v_j)} \tag{4.3}$$

Denklem 4.3'de m, iki sınıflı veri için sınıf ortalamalarını, v ise sınıflara ait varyansları temsil eder [67]. Özniteliklerin sınıf ayırabilme yeterlilikleri t-testi'nden aldıkları değerle ölçülür ve böylece en belirgin öznitelik seçilmiş olur. Çalışmamızda, kütle spektrometri verisindeki on binlerce öznitelik t-testi yardımıyla filtrelenmiş ve sınıflandırma çalışmalarında kullanılmışlardır. Bu özniteliklerin seçilmesi sırasında önce tüm özniteliklere t-testi uygulanmış ve daha sonra bu özniteliklerin ait oldukları grubun ortalamasından ne kadar farklı olduğu bulunmuştur. Böylece en belirgin 150 öznitelik seçilmiştir [68].

ii)Çok değişkenli öznitelik seçimi:

Tek değişkenli öznitelik seçimi öznitelikleri birbirinden bağımsız olarak kabul eder ve seçme işlemini ona göre yaparken, çok değişkenli teknikler özniteliklerin birbirleriyle bağlantılı olabileceğini de dikkate alarak seçim işlemini gerçekleştirirler [69]. Burada temel sorun daha önce değinildiği gibi yüksek boyutlu bir veri setinin sınıflandırma işleminde hangi özniteliklerinin seçileceğinin belirlenmesindeki hesapsal yükün fazla olmasıdır. Öznitelik sayısı arttıkça, bu öznitelikler içinden alt-grup olarak hangilerinin birlikte seçileceği ciddi bir hesaplama yükü problemidir. Örneğin *p* özelliği olan bir yüksek boyutlu veri kümesinde kullanılabilecek $(2^p - 1)$ adet alt-grup öznitelik vardır [56]. Bunun için özniteliklerin seçiminde sezgisel araştırma yöntemleri kullanılır. Literatürde sezgisel araştırma yöntemi olarak en yaygın kullanılan araştırma stratejileri En İyi İlk Arama (Best First Search, BFS), Tepe Tırmanma Araması (Hill Climbing Search, HCS), Benzetimli Tavlama (Simulated Annealing, SA) ile Genetik Arama (Genetic Search, GS) gibi yöntemlerdir [70]. Bu algoritmalar öznitelik uzayında seçme yaparken genellikle *ileri seçme* ve *geri eleme* yöntemleri gibi iki seçme yöntemini veya bu iki yöntemin sentezinden oluşan karma teknikleri kullanırlar. İleri seçim yönteminde algoritma boş bir öznitelik kümesi ile başlar ve bu kümeye giriş verilerinden rastgele öznitelikleri seçerek devam eder. Seçilen öznitelikler ise bir performans endeksine bağlı olarak değerlendirilir ve yöntemin başarımı kontrol edilir. Geri eleme ise tüm giriş özniteliklerini seçim kümesine dahil ederek öznitelikleri çıkartır ve buna karşın bir performans kriterinin değişimini gözler. Böylece en az belirleyici öznitelikler giriş uzayından elenerek, en belirleyici öznitelikler elde edilmiş olur. Her iki yöntem de hızlı ve aşırı uygunluk problemine karşı duyarsızdır [70]. İleri seçme ve geri eleme yöntemleri ile birden fazla öznitelikli alt gruplar seçilirken her iki sınıf arasındaki ayrıklığı artıracak uygun özniteliklerin seçilmesi gerekir.

Öznitelikler arasındaki bağıntıyı dikkate alarak öznitelik seçimini gerçekleştiren yöntemler *korelasyon* ve *ortak bilgi* metriklerini kullanırlar. Bu metrikleri kullanarak öznitelik seçimi yapan örnek iki yöntem *Uygunluk Tabanlı Öznitelik Seçimi* (Mutual Information Based Feature Selection, MIFS) ve *Korelasyon-Tabanlı Öznitelik Seçimi* (Correlation Based Feature Selection, CFS) yöntemleridir. Bu çalışmada kullandığımız öznitelik seçim yöntemlerinden biri olan CFS yöntemin yapısı şu şekildedir:

Öznitelik seçiminde özniteliklerin önemi aralarındaki korelasyona bağlı olarak belirlenebilir. CFS sezgisel bir yöntem olarak aynı sınıf etiketine işaret eden özniteliklerin birbiriyle korelasyonunun yüksek olduğunu varsayan bir algoritmadır [71]. Bu hipotez

özniteliklerin bir bölümünü elimine ederken bir grup özniteliği ise aralarındaki korelasyona bağlı olarak korur. Öznitelikler ve sınıf etiketi arasındaki ilişki şartlı entropi ile ölçülür. Matematiksel olarak eğer X ve Y, R_x ve R_y aralığında rastgele iki değişken ise Y'nin X'e bağlı entropisini hesaplamak için Denklem 4.4 ve 4.5 kullanılır[71].

$$H(Y) = -\sum_{y \in R_y} p(y) \log\ p(y) \tag{4.4}$$

$$H(Y \mid X) = -\sum_{x \in R_x} p(x) \sum_{y \in R_y} p(y \mid x) \log\ p(y \mid x) \tag{4.5}$$

Denklem 4.4 ve 4.5 kullanılarak, Y ile X arasındaki korelasyon Denklem 4.6 ile hesaplanır.

$$C(Y \mid X) = \frac{H(Y) - H(Y \mid X)}{H(Y)} \tag{4.6}$$

Bu korelasyon ölçüm yöntemiyle öznitelikler arasındaki ilgi derecesi ölçülerek en ideal öznitelik alt grubu seçilmeye çalışılır.

iii)*Zarflayıcılarla öznitelik seçimi*: Öznitelik seçimi için zarflayıcı yaklaşımı çekirdek bir sınıflandırıcı ve bu sınıflandırıcıyı çevreleyen bir arama algoritmasından meydana gelir. Arama algoritmasının sezgisel olarak ileri seçme veya geri eleme yöntemleri ile sınıflandırıcıya girdiği özniteliklerin sınıflandırıcı performansına etkisi göz önünde bulundurularak en yüksek başarımı sağlayan öznitelikler ayrılmış olur.

Bir zarflayıcı öznitelik seçme algoritmasının genel yapısı Şekil 4.1'de gösterilmektedir. Burada D ilk öznitelik değerleri ile birlikte eğitim veri setini, X_{best} seçilecek olan en iyi öznitelik alt grubunu ve $J(X_k)$ çalışma zamanındaki X_k öznitelik alt grubunun bir akıllı sistem algoritması (A) yardımıyla başarımını ölçen değerlendirme fonksiyonunu temsil eder.

Şekil 4.1'de görüldüğü gibi, her bir öznitelik arama yinelemesinde X^* alt grup özniteliğinin performansı çekirdek sınıflandırıcının başarımıyla ölçülür. Sınıflandırıcının X^* alt grup özniteliğine karşılık ürettiği başarım ile önceki adımlarda elde edilen en iyi öznitelik gruplarıyla elde edilen başarım kıyaslanarak, daha yüksek başarım sağlayan

öznitelikler gruba dahil edilirken diğerleri gruptan çıkarılır. Bir diğer ifadeyle, eğer elde edilen sınıflandırma başarımı öncekinden daha iyi olursa bu durumda $X_k = X^*$ kabul edilerek öznitelik grubu güncellenmiş olur. Bu işlem daha önceden tanımlanmış δ gibi bir kritere ulaşılıncaya kadar devam eder [72].

```
1: input D, X0, δ
2: Xk ← rand(X0)
3: K ← |Xk|
4: λ ← J(A(Xk,D))
5:   while δ < λ OR t < maxiterations do
6:     K ← K + 1; X* # değerini ara
7:     Xk ← X*
8:     λ ← J(A(Xk,D))
9:     # eğitim zamanını arttır
10:    t ← t + 1
11:  end while
12: output Xbest = Xk
```

Şekil 4.1. Genel zarflayıcı öznitelik seçme algoritması

Zarflayıcılar basit filtreleme yaklaşımlarının aksine öznitelikler arasındaki muhtemel bağımlılıkları dikkate alırlar. Zarflayıcı öznitelik algoritmaları hesapsal olarak daha maliyetli olmalarına rağmen sınıflandırılan verinin bilgi içeriğini korumada filtreleme yöntemlerine göre daha başarılıdırlar.

4.2.3. Özellik Seçiminde Zarflayıcı Teknikleri

Öznitelik seçme problemi temel olarak bir optimizasyon problemidir. Bu nedenle, bir zarflayıcı öznitelik seçim tekniği bir tür optimizasyon algoritması şeklinde düşünülebilir. Bir zarflayıcı temel olarak bir arama algoritması ve aday öznitelikler arasından en uygun olanlarını seçmekte kullanılacak çekirdek sınıflandırıcıdan meydana gelir. Burada optimize edilecek fonksiyon sınıflandırıcı algoritmanın performansı veya bu performansa bağlı bir başka parametredir. Zarflayıcı algoritmaların araştırma algoritmaları, çoğunlukla sezgisel bir yapıya sahiptir. Arama algoritmalarının sezgiselliği öznitelik indirgeme problemleri için bir zorunluluktur. Tüm özniteliklerin performansının sırasıyla alt gruplar halinde denenmesi öznitelik arttıkça çok ciddi bir hesapsal yüke karşılık gelir. Örneğin N tane

özniteliğe sahip bir arama uzayında sıralı arama için 2^N adet muhtemel öznitelik alt grubu bulunur. Bu da bir zarflayıcı algoritma için çekirdek sınıflandırıcının 2^N defa eğitilmesi demektir ve bu yaklaşım ciddi eğitim zamanlarına ve yüksek miktarda hesapsal yüküne karşılık gelir [73]. Bu nedenle her zarflayıcı öznitelik seçme algoritması GS ve BFS gibi hesaplama yükünü azaltan bir sezgisel arama yöntemiyle birlikte tasarlanır. Araştırma algoritmaları genellikle ileri seçme veya geri eleme gibi bir seçme stratejisi ile birlikte kullanılırlar [73]. İzleyen bölümde iki tür zarflayıcı algoritma hakkında bilgi verilecektir.

4.2.3.1. Genetik Zarflayıcı Algoritma İle Öznitelik Seçimi

GA'nın temel yapısı Bölüm 5'te detaylı olarak incelenecektir. Bu alt bölümde GS yardımıyla öznitelik seçme işleminin nasıl gerçekleştirildiğine değinilecektir. GA sistemleri optimizasyon problemlerinde yoğun olarak kullanılırlar. Bu çalışmada GA dermatoloji veri setindeki 35 öznitelikten en etkin 17 tanesinin seçiminde kullanılmıştır.

GS tabanlı zarflayıcı algoritmamızın çekirdeği bir BN sınıflandırıcısı olup, özniteliklerin seçimi bu sınıflandırıcının performansını maksimize eden öznitelikler olarak seçilmişlerdir. Öznitelik seçme algoritmamız genel olarak rastgele seçilmiş bir grup öznitelikle çalıştırılır. Bu öznitelikler *çaprazlama*, *mutasyon* ve *seçme* operatörleri ile geliştirilerek optimum öznitelik alt grubu elde edilmeye çalışılır [74]. GA yardımıyla arama yapabilmek için öznitelikler, uzunluğu 35 bit olan ikilik tabanda vektörler olarak kodlanırlar. Vektör üzerindeki her bir 1 veya 0 değeri ilgili özniteliğin seçilip seçilmediğini temsil eder. Algoritma, rastgele seçilmiş özniteliklerden türetilmiş 20 kromozomdan oluşan öznitelik havuzu ile başlatılır. Bireyler, bir noktadan ikilik çaprazlama ve ikilik mutasyona tabi tutulurlar. Her bir turda özniteliklerin seçimi için rulet döngüsü kullanılarak yeni nesiller oluşturulur. Bir kromozomum seçilme yeterliliği o kromozomda bulunan özniteliklerin BN sınıflandırıcısının başarımı ile belirlenir. Bölüm 5'te ele alınacak olan BN sınıflandırıcısı, GA algoritmasının uygunluk fonksiyonudur.

4.2.3.2. Topluluk Sınıflandırıcı Algoritması ile Zarflayıcı Öznitelik Seçimi

Topluluk öğrenme teknikleri bir sınıflandırıcının performansını artırmak amacıyla bir sınıflandırma probleminde tek bir güçlü sınıflandırıcı yerine çok sayıda zayıf

sınıflandırıcıyı kullanarak daha yüksek bir sınıflandırma başarımına ulaşmaya çalışırlar. Zarflayıcı öznitelik seçme teknikleri temelde çekirdek algoritmanın sınıflandırma başarımını arttıran öznitelikleri bulmaya çalışırlar. Bu durumda zarflayıcıda bir tek çekirdek sınıflandırıcı yerine birçok sınıflandırıcıdan oluşan bir topluluk sınıflandırıcı algoritmasının kullanılması, daha kaliteli öznitelik seçimlerinin yapılmasını sağlayacaktır. Çalışma prensibi Bölüm 5'te anlatılacak olan RFEL [75], Parkinson veri setine ait 22 öznitelikten 7 tanesini seçmek için zarflayıcının çekirdek algoritması olarak kullanılmıştır. Bu yaklaşımda arama algoritması olarak BFS algoritması ileri seçme tekniği ile birlikte kullanılmıştır. Parkinson veri setine ait özniteliklerin seçiminden sonra elde edilen sınıflandırma başarımları Bölüm 7'de detaylı şekilde verilecektir.

Parkinson veri setine ait özniteliklerin seçiminde zarflayıcı algoritmanın çekirdek sınıflandırıcısı olarak RFEL algoritmasının yanında karşılaştırma yapabilmek için, Yapay Verinin Yeniden Etiketlenmesi Tabanlı Ayrık Topluluğu (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Data, Decorate), Adaptif Teşvik Topluluğu (Adaptive Boosting, Adaboost), Öz Yükleme Topluluğu (Bootstrap Aggregating, Bagging) [134] gibi topluluk öğrenme algoritmaları kullanılmıştır. Bu algoritmalarla üretilen zarflayıcıların öznitelik seçme performansı ve buna ait sonuçlar Bölüm 7'de verilecektir. Bahsi geçen topluluk öğrenme algoritmalarına ait açıklamalara ise Bölüm 5'te yer verilecektir.

Bir sonraki bölümde bu teze kaynaklık etmiş olan ve başarımları incelenen akıllı sistem algoritmalarına ait özet bilginin yanı sıra SOM algoritmasının eğitim süresinin optimizasyonunda kullanılmış olan PSO ve topluluk öğrenme algoritmalarına yer verilecektir.

5. AKILLI SİSTEM HESAPLAMA TEKNİKLERİ

Bu bölümde akıllı sistem algoritmaları ve hesaplama tekniklerine genel bir giriş yapılacaktır. Akıllı sistem algoritmaları incelenirken ağırlıklı olarak bu çalışmada kullanılan yöntemler ele alınacaktır. Çalışmamız biyomedikal verilerin sınıflandırma başarımını incelemek esasına göre şekillendirildiği için öncelikle sınıflandırma problemine giriş yapılacak ve daha sonra akıllı sistem algoritmaları öğreticili ve öğreticisiz sistemler olarak iki ana başlık altında incelenecektir. Diğer taraftan bir optimizasyon yöntemi olarak PSO yöntemi, evrimsel bir algoritma olan GA ve topluluk öğrenme algoritmaları konu bütünlüğünü sağlamak adına üçüncü bir grup olarak ele alınacaktır.

5.1. Veri Sınıflandırma Problemi

Kütle spektrometrisi, genetik, tıbbi görüntüleme, mikrodiziler ve benzeri biyomedikal teknolojilerin kullanımı arttıkça üretilen bilginin yoğunluğu çok ciddi boyutlara ulaşmıştır. Bu verilerin insanlar tarafından işlenmesini kolaylaştırmak veya uzmanlara yardımcı olmak için bilişsel hesap yöntemlerini içeren akıllı sistem algoritmalarına ihtiyaç duyulmuştur [76]. Biyomedikal veriler akıllı sistem algoritmaları yardımıyla analiz edilirken genellikle eldeki verilerin hastalıklı-sağlıklı şeklinde sınıflandırılması ve verinin anlam kazanması amaçlanmıştır. Bu bağlamda, bir sınıflandırma problemi çözülürken öğreticili ve öğreticisiz algoritmalar kullanılır. Genel bir ifadeyle tanımlamak gerekirse; bir grup verinin içindeki matematiksel ilişkiler yardımıyla alt gruplar veya kümeler bulma işlemi öğreticisiz öğrenme ile sınıflandırma olarak adlandırılırken, eldeki sınıfsal olarak etiketlenmiş verileri bir algoritmanın eğitiminde kullanarak yeni bir durumu sınıflandırmak için kullanma işlemine öğreticili sınıflandırma adı verilir [77]. Bu ayrıma karşın sınıflandırma terimi daha çok öğreticili sınıflandırma tanımlamasına karşılık gelirken, öğreticisiz sınıflandırma ise kümeleme kavramıyla tanımlanır. İzleyen alt bölümlerde öğreticili sınıflandırma problemi ve ilgili algoritmalarla öğreticisiz sınıflandırma problemi ve bu problemin çözümü için kullanılan algoritmalar ele alınacaktır. Öncelikle öğreticili sınıflandırma algoritmalarının yapısı ve modeli ile ilgili bilgiler verilecek ve daha sonra öğreticisiz öğrenme yöntemleri tartışılacaktır.

5.2. Öğreticili Sınıflandırma

Sınıflandırma problemlerinde kullanılan yaklaşımların temelinde kullanılan kavramları şu şekilde incelemek mümkündür; sınıflandırılmak istenen her veri kümesi istatistikte durum olarak bilinen verilerden oluşur. Her veri nesnesi ona ait öznitelik adı verilen gözlemlere bağlı olarak ifade edilir. Örneğin dermatoloji hastalığına ait veride her bir birey için hastalığı tanımlayan 35 civarında öznitelik bulunur. Her bir şüpheliye ait veri kümesine ise hastalığın türü veya şüphelinin sağlıklı olduğuna dair bir sınıf etiketi karşılık gelir. İstatistiksel yöntemler, bu veri kümesinde her bir duruma karşılık gelen sınıf etiketiyle, o durum arasındaki ilişkileri matematiksel bağıntılarla ortaya koymaya çalışırlar [78].

Öğreticili öğrenme sınıf etiketi bilinen verilerden faydalanarak bu veriler ile verilere karşılık gelen sınıflar arasında sınıflandırma kuralları geliştirmek şeklinde tanımlanır. Öğreticisiz öğrenmeyi öğreticili yöntemlerden ayıran en belirgin özellik öğreticisiz sınıflandırma yöntemlerinde verilerin birbirinden sınıf etiketleri yardımıyla ayrılmaması ve bu sınıf etiketlerinin veri kümesinden öğrenilmeye çalışılmasıdır.

Öğreticili öğrenme yöntemleri sınıflandırma kurallarını türetirken, eldeki veriyi eğitim ve test amaçlı olarak bölerek kullanırlar. Eğitim amacıyla ayrılan veriler yardımıyla eğitilen bir akıllı sistem algoritmasının başarımı, test amacıyla ayrılmış ve sınıfları bilinen verilerin hangi doğrulukla sınıflandırıldığına bağlı olarak ölçülür [78]. Eğitilen sınıflandırıcının başarımının sağlaması için değişik teknikler kullanılmakta olup, bu çalışmada ele alınan sınıflandırma başarımlarının ölçümünde kullanılan metrikler Bölüm 6 da detaylı şekilde incelenecektir.

Matematiksel bir ifadeyle, iki gruplu bir öğreticili öğrenme probleminde 5.1 eşitliği ile ifade edilen X özniteliğin şekillendirdiği N adet örnek veya durum bulunur.

$$X = \{x_1, x_2, ..., x_n\} \tag{5.1}$$

Burada amaç N adet sınıflandırılacak verinin bir bölümünün test amacıyla ayrıldıktan sonra algoritmanın $C \in \{c_1, c_2\}$ şeklindeki bir sınıf paralelinde eğitilmesi ve daha sonra test verilerinin Tablo 5.1'de gösterildiği gibi atanmasıdır. Bu atama işlemi sınıflandırma olarak bilinir.

Tablo 5.1. Sınıflandırma Problemi

	Veriler	**X**	**…**	**X**	**C**
Eğitim	D_1	x_1	…	x_n	c_1
	D_2	x_1	…	x_n	c_2
	D_3	x_1	…	x_n	c_1
	D_{N-m}	x_1	…	x_n	c_2
Test	D_{N-m-1}	x_1	…	x_n	?
	D_{N-m-2}	x_1	…	x_n	?
	D_{N-m-3}	x_1	…	x_n	?
	D_N	x_1	…	x_n	?

Sınıflandırma problemleri ikiden fazla sınıf içerebilirler. Ancak sınıflandırma prensibi iki sınıflı problem ile aynıdır. Verilerin eğitim ve test şeklinde bölünmesi için kullanılan değişik yöntemler bulunmaktadır. Bu yöntemlerle ilgili detaylı bilgi de yine bölüm 6'da verilecektir.

Literatürde, en yaygın olarak kullanılan öğreticili algoritmalar BN, Lojistik Regresyon (Logistics Regression, LR), LDA, Karar Ağaçları (Decision Trees, DT), KNN, ANN (veya kısaca NN), SVM olarak sayılabilir [79]. Ancak bu çalışmada bunlara ek olarak çok sayıda öğreticili akıllı sistem algoritması kullanılmıştır. Kullanılan tüm algoritmalarla ilgili detaylı bilgi bu çalışmanın maksadına uygun olmadığı için benzer algoritmalar bir grup altında toplanacak ve her algoritma bu başlık etrafında öz bir bilgi ile incelenerek okuyucu detaylı bilgiler için referanslara yönlendirilecektir. Bu çalışmada geliştirilen algoritmaların tamamına yakını bir açık kaynak kodlu makine öğrenmesi ve veri madenciliği yazılımı olan Bilgi Analizi Waikato Ortamı (Waikato Environment for Knowledge Analysis, WEKA) ortamında geliştirilmiştir [80]. Algoritmaların gruplanmasında Java tabanlı bir akıllı sistem geliştirme ortamı olan WEKA yazılımının sınıfsal nesne modeli referans olarak alınacaktır.

5.2.1 Öğreticili Sınıflandırma Algoritmaları

Bu çalışmada kullanılan öğreticili sınıflandırma algoritmaları altı grupta incelenmiştir. Öğreticili sınıflandırma algoritmaları olarak kullanılan 30 algoritmanın tez çalışması boyunca kullanılan orijinal adlarının kısaltmaları ve bu algoritmalara ait özet bilgi aşağıda yer almaktadır.

BN temel olarak ilgili değişkenler arasındaki olasılıksal ilişkilerin kodlandığı bir grafik modeldir. Bu yapıda değişkenler düğüm olarak kodlanırken olasılıksal ilişkiler düğümler arasında çizgiler olarak gösterilirler. Daha açık bir ifadeyle düğümler rastgele

değişkenleri, çizgiler ise değişkenler arasındaki şartlı bağlılıkları temsil ederler. Herhangi bir şekilde bağlı olmayan bir düğümün ise bu gösterimde diğer değişkenlerden şartlı olarak bağımsız olduğu anlamına gelir [81].

α ve β gibi iki değişken için, Bayes teoremi şartlı olasılığı Denklem 5.2 ile tanımlanmaktadır.

$$P(\alpha / \beta) = \frac{P(\beta / \alpha)P(\alpha)}{P(\beta)} \tag{5.2}$$

Ancak BN'de $X = \{X_1, ..., X_n\}$ gibi çok sayıda değişken için zincir kuralı olarak bilinen kural yardımıyla bir düğüme bağlı değişkenlerin ortak olasılığı Denklem 5.3'teki gibi hesaplanır.

$$P(X) = \prod_{i=1}^{n} P(X_i \mid Parents(X_i)) \tag{5.3}$$

Denklemde *parents* fonksiyonu belirgin bir düğümün komşuluğunu, *n* ise BN yapısındaki düğüm sayısını ifade etmektedir [82].

BN sınıflandırıcılarının eğitimi için kullanılacak öğrenme algoritması iki bileşenden oluşur. Bu bileşenler veriye bağlı olarak ağı hesaplayacak bir fonksiyon ile ağ uzayını araştıracak bir arama metodu şeklindedir [83].

BN'de öncelikle her bir öznitelik için sınıf etiketini de içeren bir tarzda ağ düğümleri belirlenir. Ağın yapısının öğrenilmesi, mümkün olan tüm bağlantıların gezilerek buna bağlı şekilde şartlı ihtimal tablolarının elde edilmesine bağlıdır. Basit düzeyde bir BN yapısının eğitilmesi bir uzmanın yardımıyla mümkün olsa da çoğu durumda ağlar oldukça karmaşık bir yapıya sahip olup ağın tanımlanabilmesi oldukça güçtür. Bu tür durumlar için ağın yapısına ait parametrelerin makine öğrenmesi yardımıyla analiz edilen verinin kendisinden elde edilmesi gerekir. Bu amaçla kullanılan BN öğrenme algoritmaları K2, tabu arama, SA, GS ve karar ağacı öğrenmesidir [83]. BN yapısı olarak bilinen düğümler arası bağlantıların öğrenilmesinden sonra ihtimal tablolarındaki değerlerin tahmin edilmesi gerekmektedir. Bunun için eğitim verisindeki öznitelik değerlerinin göreceli kombinasyonları referans alınır. BN öğrenilirken her mümkün olan her yapı için tabloların güncellenmesi gerekmektedir. Bunun gerçekleştirilmesi ise verinin her seferinde yeni yapıya uygun

taranması ile mümkündür. Bu taramanın ardışık şekilde kolayca gerçekleştirilmesi için çok boyutlu bir ağaç yapısı kullanılır [84].

Bu çalışmada ele alınan veri setlerinden biri olan Parkinson veri setinin analizi için çok sayıda Bayes tabanlı akıllı sistem kullanılmış ve bu algoritmaların performansları incelenmiştir. Bu amaçla kullanılan diğer iki BN, naif Bayes (Naive Bayes, NB) ve basit naif Bayes (Simple Naive Bayes, SNB) şeklindedir. BN algoritmasının türevlerinden olan NB ve SNB ele aldıkları tüm öznitelikleri birbirinden bağımsız kabul ederler [85]. Algoritmalar hakkında daha detaylı detaylı bilgiye [86]'dan ulaşılabilir.

Bir diğer öğreticili akıllı sistem yaklaşımı *Tembel Öğreniciler* veya *Durum-Tabanlı Öğreniciler* olarak bilinen sınıflandırma algoritmalarını içerir [87]. Durum-tabanlı sınıflandırma algoritmaları aceleci algoritmalar olarak bilinen karar ağaçları, BN ve ANN ağlarının tersine eğitim safhasında daha az hesapsal yüke gereksinim duyarlarken test safhasında daha çok zaman gerektirirler. En temel haliyle durum-tabanlı algoritma, literatürde çok kullanılan akıllı sistemlerden biri olan KNN algoritmasından türetilmiştir. KNN, aynı sınıf etiketi taşıyan örneklerin birbirine diğer örneklere göre daha yakın olduğunu ya da bu örneklerin birbirinin komşuluğunda bulunduğunu varsayar. Bu şekliyle sınıf etiketine sahip örneklerin KNN komşuluk prensibiyle gruplanmasından sonra sınıfı bilinmeyen test örneğinin ait olduğu sınıf eldeki örneklerle komşuluk karşılaştırmasına tabi tutulmasıyla bulunur [88].

KNN prensibinde genel olarak her bir örnek *n* boyutlu uzayda birer nokta gibi düşünülürken, her bir boyut ise örnekleri niteleyen bir özniteliğe karşılık gelir. Bu şekilde eğitim örnekleri *n* boyutlu örnek uzayında saklanırlar. Sınıfı bilinmeyen bir örnek için KNN örnek uzayını araştırarak en yakın örneği bulur. KNN algoritmasında yakınlık ya da komşuluk Öklid uzaklığı ile hesaplanır. $X = (x_1, x_2, ..., x_n)$ ve $Y = (y_1, y_2, ...y_n)$ gibi iki nokta için Öklid uzaklığı Denklem 5.4'teki gibi hesaplanır.

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5.4}$$

Sınıflandırma için kullanılan komşuluk metriği aynı sınıfta olan örnekler arasındaki mesafeyi minimize ederken farklı sınıftaki örneklerin arasındaki mesafeyi de maksimize eder. Bu amaçla en sık kullanılan metrikler Minkowsky, Manhattan ve Öklid komşuluk ölçümleridir [89].

Durum-tabanlı algoritmalar, her yeni örneği eğitim safhasında elde ettikleri ve hafızada tuttukları eski örneklerle kıyaslayarak çalışma zamanında sınıflandırma gerçekleştirirler. Bu yaklaşım sınıflandırmakta kullanacağı modeli eğitimde kullandığı örneklerin kendisinden elde eder. Diğer algoritmaların tersine eğitimdeki örneklerin bir kısmı eğitimden sonra değiştirilse de, durum-tabanlı algoritmalar tekrar eğitime tabi tutulmadan ürettikleri eski modellerini yeni durumlarda da kullanabilirler [90].

Durum-tabanlı algoritmalar örnek sayısı arttıkça kullandıkları hafızanın artması ve algoritmanın ürettiği sınıflandırma modelinin karmaşıklaşmasından dolayı eğitimde kullanılan hafızada saklanmış örneklerin bir bölümünün sınıflandırma hassaslığını bozmadan filtrelenmesi gerekir. Bunun amaçla, hafızada tutulan örnek sayısını % 80 oranında azaltarak durum-tabanlı sınıflandırma modelinin karmaşıklığını ve hesapsal yükünü azaltan yardımcı algoritmalar kullanılırlar [91]. Durum-tabanlı algoritmaların karmaşıklığını azaltmak için kullanılan diğer yöntemler ise öznitelik seçim teknikleridir.

Durum-tabanlı algoritmaların çok sayıda türevleri olup bu çalışmada kullanılan tembel öğrenici algoritmalar KNN, K-Yıldız (KSTAR), Durum-Tabanlı k (Instance-Based k, IBk) ve Durum-Tabanlı Bir (Instance-Based 1, IB1) şeklindedir. Bu algoritmalar hakkında detaylı bilgiye [92]'de yer verilmiştir.

Kural tabanlı sınıflandırıcılar eğitim verisinden bu amaç için üretilmiş algoritmalar yardımıyla elde ettikleri kuralları kullanarak sınıflandırma yaparlar. Sınıflandırma kuralları her bir sınıfı Ayrıştırıcı Normal Form (Disjunctive Normal Form, DNF) olarak bilinen bir şekle dönüştürerek elde edilirler [93]. Burada amaç eğitim verisine uygun en az sayıda kuralı elde etmektir. Daha açık bir ifadeyle amaç eğitim verisindeki sınıflandırmayı en küçük kural takımıyla sembolize edebilmektir. Eğitim sırasında elde edilen kuralların çoğu öğrenilmiş örneklerin *hatırlanması* şeklindedir. Bir yeni örneğin sınıflandırılması veya yeni kurallar türetilmesi için kural-tabanlı algoritmalara Ayır-Fethet (Seperate-Conquer, SC) stratejileri eklenir [94]. Bu strateji eğitimde kullanılan örneklerin bir bölümünü açıklayan bir kuralı araştırır ve daha sonra ayrılan örnekleri daha fazla kural öğrenerek ardışık şekilde birleştirir. Bu nedenle karar kurallarının sağlıklı sonuçlar üretebilmesi önemlidir. Kuralların kalitesi hem üretilme hem de sınıflandırma kalitesinin ölçülmesiyle elde edilir. Bu amaçla kullanılan kurallardan birisi literatürde J-ölçümü adıyla bilinir [95]. Kural tabanlı sistemler iki durumlu problemlerin çözümünde karar ağaçlarına göre daha etkindirler. Kural tabanlı algoritmaların sınıflandırma performansı özniteliklerin birleşimleriyle geliştirilebilmektedir [96].

Çalışmamızda kullanılan kural-tabanlı sistemler, Birleştirici Kural (Conjunctive Rule, CR), Karar Tablosu (Decision Table, DS), Java Tekrarlayan Artımlı Budama (Java Repeated Incremental Pruning, JRIP), İç içe olmayan Genelleştirilmiş Örnekler (Non-Nested Generalized Exemplars, NNGE), Bir Kural (One Rule, OneR), Dalgalı Düşen Kural Öğrenici (Ripple Down Rule Learner, RIDOR) ve Sıfır Kural (Zero Rule, ZeroR) algoritmalarıdır. Bu algoritmaların temel yapıları kural-tabanlı algoritma mantığına göre şekillendirilmiş olup detaylı bilgiye WEKA yazılımına ait dokümanlardan erişilebilmektedir [97].

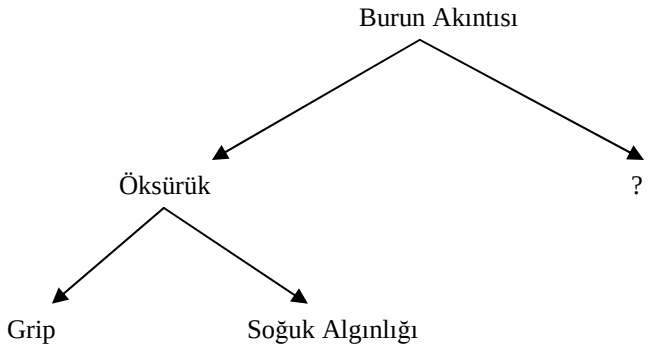
Literatürde sınıflandırma amacıyla kullanılan bir diğer algoritma grubu *karar ağacı* adı verilen algoritmalardan oluşmaktadır. Geniş verilerin sınıflandırılmasında başarıyla kullanılan karar ağaçları özniteliklere bağlı olarak verileri sınıflandırma başarımını yükseltecek şekilde bir kök düğümden ardışık şekilde dallara ayırarak gruplandırır. Dallanmanın yapılacağı özniteliğin seçimi her özniteliğin hedef sınıf etiketlerinin sınıflandırma başarımına ait bilgi içeriğiyle ölçülür. Entropi ile ölçülen bilgi içeriği entropi küçüldükçe artacağından tüm özniteliklerden entropisi en küçük olan öznitelik, kök düğüm olarak seçilir. Her adımda entropi hesaplama ve dallanma işlemi ardışık olarak yapılır ve ağaç tamamlanmış olur. Entropi hesabı için literatürde Denklem 5.5 kullanılmaktadır.

$$Entropi(A) = \sum_{i=1}^{2} \frac{\sum_{j=1}^{c} f_{ij}}{R} \sum_{j=1}^{c} P(c_{ij}) Ln(P(c_{ij})) \tag{5.5}$$

Denklem 5.5'te, c sınıf etiket sayısı, f_{ij} ifadesi i dalında j sınıf etiketinin sıklığı, R her iki daldaki toplam örnek sayısıdır. $P(c_{ij})$ ise Denklem 5.6 ile gösterir [98].

$$P(c_{ij}) = \frac{f_{ij}}{\sum_{k=1}^{c} f_{kj}} \tag{5.6}$$

Grip ile soğuk algınlığı hastalıklarını birbirinden ayıran örnek bir karar ağacı Şekil 5.1'de gösterilmektedir [99].



Şekil 5.1. Basit bir hastalık teşhis karar ağacı

Karar ağaçları mantığına dayanan ve literatürde yaygın olarak kullanılan algoritmalar C4.5 ve ID3 yapılarıdır. Bu çalışmada kullanılan karar ağacı algoritmaları ticari C4.5 algoritmasının Java versiyonu olan J48, ID3, ADT, En İyi İlk Karar Ağacı (Best First Decision Tree, BFT), FT, Lojistik Model Ağacı (Logistic Model Tree, LMT), Lojistik Değişken Karar Ağacı (Logistic Alternating Decision Tree, LADT), ve Rastgele Ağaç (Random Tree, RT) algoritmalarıdır [100,101].

SVM algoritması, diğer algoritmalarda olduğu gibi eğitildikten sonra özniteliklere bağlı olarak test verisini sınıflandıracak bir model üretir. SVM deneysel sınıflandırma hata oranını minimize ederken geometrik uzaklığı da eşzamanlı olarak maksimize eder ve bu nedenle maksimum uzaklık (maximum margin) sınıflandırıcıları olarak bilinirler. Eğitim verisi olarak $(x_i, y_i), i = 1,....,l$ örnek çiftleri verilsin. Burada $x_i \in R^n$ ve $y \in \{1,-1\}^l$ şartını sağlarken, bir optimizasyon algoritması olan SVM Denklem 5.7'nin çözümüne ihtiyaç duyar.

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{5.7}$$

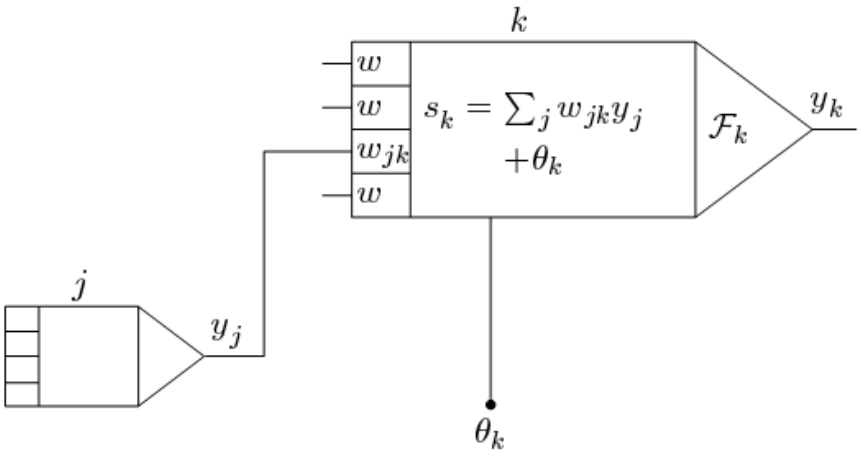
Denklem 5.7, $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$ şartına bağlı olarak çözülmelidir. Bu problemde, x_i eğitim vektörleri φ fonksiyonu yardımıyla daha yüksek bir boyuta transfer edilirler. SVM algoritması daha yüksek boyutta sınıfları birbirinden ayırabilecek maksimum marjine sahip bir doğrusal hiper-düzlem bulmaya çalışır. Denklem 5.7'de $C > 0$ olup hata terimine ait karar verme katsayısıdır [102]. Denklem 5.7'nin bu şekildeki çözümü sadece doğrusal SVM sınıflandırıcılarının oluşumuna izin verir. Ancak çekirdek

kolaylığı (*kernel trick*) yardımıyla doğrusal olmayan sınıflandırıcıların oluşturulması mümkündür [103]. Çekirdek ile $\varphi(x_i)$ dönüşümü arasındaki bağıntı Denklem 5.8 ile verilir.

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \tag{5.8}$$

Denklem 5.8 çekirdek fonksiyonu olarak bilinir. Bu amaçla literatürde yaygın şekilde kullanılan fonksiyonlar; polinom tabanlı fonksiyonlar, ve sigmoid çekirdek fonksiyonlarıdır [104].

Nöron adı verilen sinir hücrelerinin çalışma prensibini modelleyerek elde edilen akıllı sistemler *ANN* algoritmalarının temelini oluştururlar [105]. Bir nöronun matematiksel modeli Şekil 7.2'de gösterilmiştir [106].



Şekil 5.2. Bir Matematiksel Nöron Modeli

Bu modelde *j* gibi bir nöronun çıkışından gelen ve *k* nöronunun girişi olan y_j sinyali w_{jk} ağırlığına bağlı olarak θ_k düzeltme katsayısı (bias) değeriyle toplandıktan sonra, F_k transfer fonksiyonuna bağlı olarak sonraki nörona bir çıkış sinyali üretilir [106]. Bir diğer ifadeyle giriş değerlerinin her biri bir bağlantı ağırlığıyla çarpılır ve doğrusal olarak toplandıktan sonra, doğrusal veya doğrusal olmayan bir transfer fonksiyonunda işlenerek bir çıktı üretilir. Bu çıktı çok katmanlı bir ağ yapısında kendisinden sonra gelen diğer nörona giriş bilgisi olarak verilir [107]. Geleneksel bir nöron ağında giriş katmanı, saklı

katman ve çıkış katmanı olmak üzere üç katman bulunur. Bu yapıda amaç bir öğrenme algoritması veya kuralı yardımıyla örneğin bir sınıflandırma problemi için çözüm üretmektir. Bu sinir ağlarının eğitiminde temel olarak öğreticili, öğreticisiz ve takviyeli öğrenme yaklaşımları kullanılmaktadır [108]. Öğreticili öğrenme temel olarak hedef değere ulaşmak için ağdaki çıkış değeri ile hedef değerin arasındaki farkı, yani hatayı azaltmaya çalışan bir ortalama küçük kareler hata toplamını minimize eder. Farkın küçültülmesi için kullanılan yaklaşım meyilli düşme (Gradient Descent,GD) yaklaşımlarını içeren bir grup yaygın ağ türü çok katmanlı perceptron (Multi Layer Perceptron, MLP) adını alır [109].

MLP algoritmasını diğer ANN algoritmalarından ayıran en önemli özellik kullanılan transfer fonksiyonunun doğrusal olmayan bir fonksiyon olmasıdır. Bu amaçla en sık kullanılan fonksiyon Denklem 5.9'da gösterilen sigmoid fonksiyonudur.

$$\phi(y_i) = \frac{1}{(1+e^{-v_i})} \tag{5.9}$$

Denklem 5.9'da, y, i düğümünün çıktısını ve v_i ise bu düğümün girişlerinin ağırlıklı ortalamasını gösterir. Bir doğrusal perseptron sadece birbirinden doğrusal olarak ayrılabilen verileri sınıflandırabilirken MLP doğrusal olarak birbirinden olmayan verileri de birbirinden ayırabilmektedir [110]. Daha açık bir ifadeyle MLP gizli katmanında doğrusal olmayan aktivasyon fonksiyonlarını içerdiği için doğrusal olmayan gruplamaları yapabilmektedir. MLP'de öğrenme bir giriş örneği ile çıkış sinyali arasındaki farka bağlı olarak düğümler arasındaki ağırlıkların güncellenmesi ile gerçekleştirilir.

Daha açık bir ifadeyle j çıkış düğümündeki n verisine ait hata Denklem 5.10 ile verilir.

$$e_j(n) = d_j(n) - y_j(n) \tag{5.10}$$

Denklem 5.10'da, d hedef değeri ve y perceptron tarafından üretilen değeri temsil eder. Denklem 5.10'daki hata terimini düğümler arasındaki ağırlıkları güncelleyerek minimize etmek için Denklem 5.11 kullanılır.

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{5.11}$$

Her bir ağırlığın değişimi Denklem 5.12 ile verilen meyilli düşme ya da GD ile hesaplanır.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_i(n) \tag{5.12}$$

Denklem 5.12'de y_i bir önceki adımdaki düğümün çıktısını η ise öğrenme oranını temsil eder [111].

ANN algoritmaları kendilerine literatürde yaygın şekilde kullanım alanı bulmuşlardır. Bu çalışmada MLP dışında kullanılan bir diğer ağ, Radyal Temel Fonksiyonlu Ağ (Radial Basis Function Network, RBF) algoritmasıdır. Bu algoritma ve genel olarak ANN hakkında detaylı bilgi Bishop [108] tarafından yapılan çalışmada bulunabilir.

Lojistik tabanlı algoritmalar sınıflandırma problemlerinde kullanılan diğer akıllı sistemler arasında yer alırlar. Bir lojistik sınıflandırıcı temel olarak karar ağacı algoritmasının LR fonksiyonlarıyla bütünleştirilmiş şeklidir. Bir diğer ifadeyle, bir lojistik sınıflandırıcı, yapraklarında LR fonksiyonunun kullanıldığı bir karar ağacıdır. Doğrusal lojistik regresyon, j sınıf etiketine karşılık gelecek şekilde doğrusal x fonksiyonlarıyla sınıf olasılıklarını hesaplar. Bunun için $\Pr(G = j \mid X = x)$ ifadesi kullanılır. LR modeli Denklem 5.13 ile tanımlanır.

$$\Pr(G = j \mid X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}, \sum_{k=1}^{J} F_k(x) = 0 \tag{5.13}$$

Denklem 5.13'te, $F_j(x) = (\beta_j)^T.x$ şeklinde doğrusal regresyon fonksiyonlarıdır. Bu Bu fonksiyonlar, β_j parametreleri için bulunan maksimum benzerlik parametre tahminleri yardımıyla elde edilirler. Bu tahminleri bulmak için *LogitBoost* algoritması kullanılır [112]. LogitBoost algoritması kullanıldığında Denklem 5.13 ifadesi yerine

$F_j(x) = \sum_m f_{mj}(x)$ ifadesi elde edilir. Burada f_{mj} girdi değişkenleri için en küçük kareler yöntemi ile uydurulan fonksiyonları göstermektedir [113].

Bu çalışmada logistik regresyon (Logistik Regression, LR) ve bu algoritmanın basit doğrusal regresyonlar şeklinde uyarlandığı basit regresyon (Simple Regression, SR) algoritmaları sınıflandırma problemlerinde kullanılmıştır [114].

Bu bölümde tez çalışması içinde kullanılan öğreticili akıllı sistem algoritmalarına ait özet bilgi verilmiş olup izleyen bölümde yine bu çalışmada kullanılan öğreticisiz öğrenme algoritmalarına kısaca değinilecektir.

5.3. Öğreticisiz Sınıflandırma

Sınıflandırma algoritmaları en genel haliyle öğreticili ve öğreticisiz olmak üzere iki yaklaşım ile sınıflandırma işlemini gerçekleştirirler. Bir algoritma yeni örnekleri önceden tanımlanmış belirgin bir sınıf etiketine uygun olarak sınıflandırıyorsa bu algoritma öğreticili bir sınıflandırma gerçekleştirmektedir. Öğreticisiz bir algoritma ise her yeni örneği eldeki verilerin kendi içindeki ilişkilerine uygun olarak elde ettiği önceden tanımlanmamış bir sınıfa veya kümeye atayarak çalışır. Öğreticili sınıflandırma yöntemlerinde algoritmanın eğitimi önceden tanımlanmış sınıflar gerektirirken, öğreticisiz sınıflandırmada gruplar benzer örneklerin bir araya toplandığı kümeler şeklinde eğitim sırasında elde edilirler [115]. Daha açık bir ifadeyle kümeleme veriyi birbirine benzer nesnelerin bir grupta bulunduğu sınıflara bölmek şeklinde tanımlanır. Verinin kümeler halinde gösterilmesi, sınıflandırılan veri içindeki gizli ilişkilerin ortaya çıkarılarak daha basit şekilde ifade edilmesi sonucunu verir. Bu bakışla kümeleme incelenen veri içindeki gizli ilişkileri görsel şekilde sunmak olarak tanımlanabilir [116].

Matematiksel olarak ifade etmek gerekirse, $x_i = (x_{i1}, x_{i2}, ..., x_{id}) \in A$ ifadesi d elemanlı ve A öznitelik uzayında tanımlanan bir X veri setini temsil etsin. Burada A uzayı $i = 1:N$ aralığında özniteliğe karşılık gelir ve her bir x_{il} örneği A_l özniteliği ile tanımlanır. Veri örneklerinin öznitelikle tanımlanması kavramsal olarak $d \times N$ şeklinde bir matrise karşılık gelir. Öznitelik uzayına ait alt küme $C = \prod C_l \subset A, C_l \subseteq A_l, l = 1:d$ kartezyen çarpımıyla ifade edilir. Buradan veri setindeki alt kümeler $X = C_1, ..., C_k \wedge (C_{j1} \cap C_{j2})$ şeklinde tanımlanır. Teorik olarak kümeler arası kesişme boş küme olmakla beraber pratikte kullanılan algoritmaların yapısına bağlı olarak kümeler kısmen iç içe geçebilir

veya aynı örnek birkaç kümede bulunabilir [117]. İzleyen alt bölümde öğreticisiz sınıflandırma algoritmaları ve özellikle SOM algoritması ile ilgili özet bilgiye yer verilecektir.

5.3.1. Öğreticisiz Sınıflandırma Algoritmaları

Öğreticisiz sınıflandırma algoritmaları önceki bölümde bahsedildiği gibi veriler içinde birbirine benzeyen veya benzemeyen grupları veya kümeleri bulmaya çalışırlar. Bu anlamda kümeleme, verileri birbirlerine olan benzerliklerine göre gruplamak olarak tanımlanır [118]. Literatürde bir öğreticisiz sınıflandırma yaklaşımı olan kümeleme moleküler biyolojiden (mikrodiziler, gen ifade analizi vb.) arama motorlarında bilgiyi sınıflandırmaya ve ticari uygulamalara dek çok farklı alanlarda kullanılırlar. Bu amaç için kullanılan yaygın kümeleme algoritmalarından bazıları *K-means*, *Fuzzy C-means* ve *SOM*'dur. Bu algoritmaları birbirine tercih ederken her bir algoritmanın belli bir veriyi kümeleme kalitesi belirleyicidir. Bunun için farklı metrikler geliştirilmiştir. Genel olarak bir algoritmanın kümeleme kalitesi aynı kümeye gruplanan nesnelerin birbirine benzerliği ile doğru orantılıdır. Başka bir ifadeyle kümeleme kalitesi kümeler arasındaki benzemezlik oranının artmasıyla yine aynı şekilde artış gösterecektir [118].

Kümeleme algoritmaları tüm avantajlarına karşın geniş veri setlerinde ciddi hesapsal güce ihtiyaç duyarlar. Bu algoritmalardan yaygın şekilde kullanılan SOM algoritmasında veri boyutu büyüdükçe eğitilmesi gereken SOM boyutu büyür ve dolayısıyla eğitim zamanı uzar. Eğitim zamanının kısaltılması SOM algoritmasının kullanımını kolaylaştıracağından çalışmadaki hedeflerimizden birisi SOM sınıflandırma performansını değiştirmeden eğitim zamanını kısaltmaktır. SOM algoritmasının çalışma mantığı verildikten sonra bu optimizasyon yaklaşımından bahsedilecektir [119].

SOM algoritması sinir ağı temeline dayanan kümeleme ve veri içindeki örüntüleri görselleştirme amacıyla yaygın şekilde kullanılan bir öğreticisiz sınıflandırma sistemidir. SOM algoritması bir tür veri sıkıştırma tekniği olarak yüksek boyutlu veriyi alarak çıkışında bu verinin düşük boyutlu bir haritasını oluşturur [120]. Bu dönüşüm yüksek boyutlu verilerin görselleştirilmesi ve yorumlanmasında faydalıdır. SOM yapısal olarak birbirine karşılıklı bağlı düğüm adı verilen nöronlardan oluşur. Çok farklı SOM yapıları olsa da en yaygın topoloji matris benzeri iki boyutlu yapıdır [121].

SOM eğitilirken her bir düğüm vektörü kümelenecek giriş vektörünün boyutuna uygun şekilde rastgele değerlerle başlatılır. Burada amaç ardışık şekilde sisteme girilen vektörlerin her bir düğümde kendisine en yakın vektörü güncelleyerek o vektörün ağırlıklarını kendisine benzetmesidir. Burada en önemli ara adım giriş vektörüne uygun düğümün nasıl bulunacağıdır. Yarışmacı öğrenme teorisinde kazanan düğüm olarak da bilinen bu en iyi uyan birim (Best Matching Unit, BMU) her bir giriş vektörü için ayrıca bulunmalıdır [122]. Bu manada BMU, giriş vektörüne en yakın düğüm vektörü olacaktır. BMU'nun bulunması her bir giriş vektörünün, düğüm vektörleriyle Minkowski metriği, Mahalanobis uzaklığı veya Öklid mesafesi gibi bir uzaklık ölçümüyle kıyaslanarak en yakın olan düğümün BMU olarak atanmasıyla yapılır [119]. Tüm eğitim vektörlerinin sisteme bu şekilde girilmesi ile bir çevrimlik eğitim tamamlanır. Bu eğitim çevrimi kullanıcı tarafından uygun görünen bir sayı kadar tekrar edilir. SOM boyutunun kaliteli kümeleme için giriş verisine uygun şekilde ayarlanması, giriş verisi büyüdükçe boyutun da artırılması gerekir. Bu şekilde SOM algoritmasının eğitiminin veri boyutu ve dolayısıyla seçilen SOM boyutuyla yakından ilgili olduğu görülecektir. BMU seçiminin boyut arttıkça büyüyeceği ve eğitim için harcanan zamanın uzayacağı açıktır. Kohonen, BMU seçimi için tüm düğümlerin gezilmesi prosedürünü değiştirecek bir yöntemin SOM eğitim zamanını kısaltacağını ifade etmiştir [119]. Bu çalışmada ele alınan PSO ile eğitim süresini kısaltan yaklaşım ve bu yaklaşıma ait sonuçlar Bölüm 7'de detaylı şekilde verilecektir. Ancak konuyu açıklığa kavuşturmak adına kullanılan yöntem basitçe şu şekildedir:

BMU düğümünü en kısa şekilde tüm SOM düğümlerini gezmeden bulmak eğitim süresinin kısaltılması için ön şarttır. Veriye göre değişmekle beraber kaliteli bir SOM eğitimi ortalama 100 çevrimlik bir eğitim döngüsüne ihtiyaç duyar. Her bir SOM düğümüne eşlik etmesi için tasarlanan PSO vektörleri ilk birkaç çevrim boyunca normal şekilde eğitilen SOM ağında hangi vektörün hangi düğüme gittiğini hafızasına alır. Böylece her bir giriş vektörü için tüm SOM düğümlerinin gezilmesi yerine, giriş vektörlerinin daha önce girdiği düğümlerin çevresinde BMU araştırması yapar. Bu şekilde devam eden eğitim normal şekilde BMU araştırması yapan eğitime göre çok daha kısa sürede bitirilmiş olur.

Literatürde yaygın şekilde kullanılan SOM algoritmasının büyük boyutlu verilerin kümelenmesi sırasında eğitim için uzun harcayacağı zaman açıktır. Bu çalışmada elde edilen optimizasyon yaklaşımı orijinal SOM algoritmasının yerine geçmese de Bölüm 7'de

görüleceği gibi oldukça iyi sonuçlar üretmiş olup kestirme sonuç üretme ihtiyacı duyulduğunda işlevsel olacaktır.

5.4. PSO Algoritması

Sürü zekası algoritmaları arı, karınca ve termit gibi canlıların kolektif davranışlarını modelleyen sistemlerdir [123]. Sürü zekasını modelleyen en popüler iki algoritma arı koloni optimizasyon (Ant Colony Optimization, ACO) ve PSO algoritmalarıdır. Verimli bir optimizasyon algoritması olan ve bu çalışmada SOM eğitim zamanını optimize etmekte kullanılan PSO, Kennedy tarafından geliştirilmiştir [124]. PSO birbiriyle etkileşen elemanların bir arama uzayında bir problemi optimize etmesi şeklinde tanımlanabilir [125]. PSO algoritmasında bir arama uzayında uçarken belli bir zamanda her bir parçanın bir pozisyonu ve göreceli bir hızı bulunmaktadır. Parçacığın orijine göreceli pozisyonu, çözümü aranan problemin muhtemel çözümlerinden birisi olarak düşünülür. Parçacıkların arama uzayında yeni bir pozisyona hareketi yeni çözümlerin üretilmesi anlamına gelir. Elde edilen çözümlerin uygunluğu bir uygunluk fonksiyonu ile ölçülür [126].

PSO algoritması başladığında, parçacıkların pozisyonu ve göreceli hızları rastgele değerlerle başlatılır. Parçacıkların pozisyonu ve hızları sırasıyla $X_i\ (t)$ ve $V_i\ (t)$ ile ifade edilir [127]. Bu iki durum dışında parçacıklar gezintileri sırasında elde ettikleri en iyi pozisyonu ve komşu parçacıkların en iyi pozisyonlarını tutan bir hafızaya sahiptirler. Algoritma geliştikçe parçacıkların pozisyonları, hafızada tutulan $P_i\ (t)$ pozisyon bilgisi ile etkileşirler. Bu bilgi parçacıkların tamamının pozisyon bilgisi kullanılarak hesaplanırsa o zaman global en iyi (Global Best, Gbest) P_g, eğer parçacığın daha küçük komşuluğunda olan bir kısım parçacıkların pozisyonuna bakılarak hesaplanırsa da yerel en iyi (Local Best, Lbest) P_l adıyla bilinir [128].

D boyutlu problem uzayında PSO algoritmasının her bir döngüsünde parçacıklar, P_l ve P_g pozisyon bilgisini kullanarak hızlarını ve ardışık pozisyonlarını Denklem 5.14 ve 5.15 ile hesaplarlar [129].

$$V_{id}(t+1) = V_{id}(t) + c1(P_{ld} - X_{id}(t)) + c2(P_{gd} - X_{id}(t)) \tag{5.14}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{5.15}$$

Parçacıkların güncelleme işlemi maksimum değerleri 2 olan *c1* ve *c2* katsayılarıyla ayarlanmaktadır [130].

5.5. Topluluk Öğrenmesi Algoritmaları

Akıllı sistemler literatüründe daha hassas sınıflandırma başarımları için tek bir sınıflandırıcı yerine çok sınıflandırıcının bir arada kullanıldığı modeller geliştirilmiştir. Bu çalışmaları içine alan algoritmalar topluluk öğrenme algoritmaları (Ensemble Learning Algorithms, ELA) olarak isimlendirilirler. Topluluk öğrenme algoritmaları sınıflandırma başarımları göreceli olarak düşük olan zayıf algoritmalardan topluluklar oluşturarak daha yüksek başarımlı bir algoritma elde etmeye çalışırlar. Öğreticili öğrenme algoritmaları bir problemi çözerken çözüm uzayını uygun bir hipotez bulmak için araştırırlar. Topluluk öğrenme algoritmaları ise aynı problem için birden fazla hipotez üreterek ortalama bir sonuç elde etmeye çalışırlar [131].

Topluluk öğrenmesi algoritmaları aynı anda birçok sınıflandırıcıyı eğiterek çalıştıkları için bu algoritmalar ciddi hesapsal yük getirirler. Bu yükü azaltmak adına topluluk öğrenmesi algoritması uygulanacak baz sınıflandırıcılar çoğunlukla karar ağacı gibi hızlı sınıflandırıcılardan seçilirler [132,133].

Topluluk öğrenme veya çoklu sınıflandırıcı yaklaşımları geniş ölçekli verilerin analizinde özellikle tercih edilirler. Bu tür bir verinin sınıflandırılmasında, veri öncelikle alt kümelere bölünür ve her bir alt küme için ayrı bir sınıflandırıcı eğitilerek tüm sonuç bu algoritmaların ayrı ayrı bulduğu sonuçların bir tür ortalaması olarak elde edilir. Çoklu sınıflandırıcıların ürettiği bu sonuçlar değişik birleştirme yaklaşımlarıyla bir araya getirilip tek sonuca ulaşılır [134].

Literatürde yaygın şekilde kullanılan topluluk öğrenme algoritmaları Bagging, Boosting, Adaboost, Decorate ve RFEL yöntemleridirler.

Bagging, temel olarak bir öz yükleme topluluk algoritmasıdır. Öz yükleme bir veri setinin kopyalarının kullanılarak eğitim sırasında ayrıklığın sağlanmasına karşılık gelir. Diğer bir ifadeyle veri setine ait alt veri kümelerinin *yerinin değiştirilmesi* ile eğitim için çekilip kullanılmasıdır. Her alt veri kümesi, topluluğu oluşturan sınıflandırıcılardan birinin

eğitiminde kullanılır ve gerçek sonuç, topluluğu oluşturan algoritmaların sonuçlarının salt çoğunluğuna bağlı olarak elde edilir. Daha açık bir ifadeyle belli bir örnek veriye ait sınıf etiketinin elde edilmesi topluluk içinde yer alan sınıflandırıcıların çoğunluğunun o belirli veri için elde ettiği sınıfların *basit çoğunluk kuralı* (simple majority rule) yardımıyla değerlendirilmesiyle gerçekleştirilir [135].

Boosting, algoritması topluluk öğrenmesi aşamasında veriyi yeniden örnekleme yaklaşımı ve daha sonuçların basit çoğunluk kuralı yardımıyla değerlendirilmesi anlamında *Bagging* algoritmasına benzeyen bir algoritmadır. Boosting temel olarak Bagging algoritmasına göre yeniden örnekleme yaklaşımı özelliği ile farklılık gösterir. Boosting algoritmasında her bir yinelemede örneğin C1, C2 ve C3 gibi üç zayıf sınıflandırıcı oluşturulur. C1 eğitim verisinin rastgele bir alt kümesi ile eğitilirken, C2 sınıflandırıcısı C1 tarafından doğru sınıflandırılan verinin yarısı ve yanlış sınıflandırılan verinin yarısı ile eğitilir. C3 sınıflandırıcısı ise C1 ve C2 algoritmalarının sınıfı üzerinde anlaşamadıkları örnekler ile eğitilir. Bu algoritmaların sonuçları, aynı şekilde basit çoğunluk kuralı yardımıyla birleştirilerek eğitimde kullanılan örnekler için net sınıflandırma sonuçlarına ulaşılır.

Adaboost, bir boosting algoritma türevi ile aynı yaklaşıma sahiptir. Adaboost algoritmasının normal boosting algoritmasından farkı, ikili sınıflandırma problemlerinin yanında çok sınıflı problemleri ve regresyon tabanlı problemleri de ela almasıdır [135].

Bir diğer topluluk öğrenme algoritması olan *Decorate,* topluluğu oluşturan sınıflandırıcıların ayrıklığını sağlamak için var olan eğitim örneklerine ek olarak rastgele üretilen yapay örnekler kullanır. Bu şekilde topluluğu oluşturan sınıflandırıcılar eğitilerek bir topluluk öğrenme stratejisi geliştirir [136].

RFEL sınıflandırma algoritması göreceli olarak yeni bir topluluk öğrenme algoritması olup PCA tekniğine dayanan bir yapıdadır. RFEL algoritması sınıflandırıcı ayrıklığını desteklerken, sınıflandırma hassasiyetini topluluk öğrenme tekniği ile artırmağa çalışır. PCA, topluluğu oluşturan her bir sınıflandırıcıya bir rotasyoncu öznitelik ekseniyle ayrıklığı temin eder. Bunun yanında sınıflandırma hassaslığını korumak için algoritma tüm temel bileşenleri muhafaza ederken topluluktaki her bir sınıflandırıcıyı tüm veri seti ile eğitir [137]. Geleneksel RFEL algoritmasında rotasyon karar ağaçları yardımıyla temin edilir. Karar ağaçları öznitelik rotasyonuna karşı hassas oldukları için RFEL algoritması için uygundurlar [138].

Bu çalışmada *Adaboost*, *Decorate*, *Bagging* ve *RFEL* algoritmalarından J48 karar ağacı sınıflandırıcısı ile üretilen topluluk öğrenmesi algoritmaları dermatoloji hastalığına ait özniteliklerin seçiminde kullanılmışlardır.

5.6. Genetik Algoritma

GA sistemleri, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve optimizasyon yöntemi olarak bilinirler. Çok boyutlu uzayda, en iyinin hayatta kalması ilkesine göre çözüm uzayını arayan bu algoritmanın temel ilkeleri, ilk kez Michigan Üniversitesi'nden John Holland tarafından ortaya atılmıştır [139]. Holland'ın 1975 yılındaki bu çalışmaları 'Adaptation in Natural and Artificial Systems' çalışmasında yer almıştır [140]. Evrimsel süreçler de yine ilk defa adı geçen araştırmacı tarafından GA sistemlerinde optimizasyon problemleri için kullanmıştır.

Bir GA üzerinde çalıştığı probleme bir tek çözüm yerine çoklu çözümleri içeren bir çözüm kümesi üretir. Bu yaklaşım arama uzayında birçok noktanın değerlendirilmesi ve daha sağlıklı bir sonucun elde edilmesine yardımcı olur. GA ile elde edilen çoklu çözümler, ayrık çözüm vektörleri şeklinde elde edilirler [141].

Bir GA yapısal olarak evrimsel süreçlerin bilgisayar ortamında gerçeklenmesi olarak düşünülebilir. Bu süreçler yardımıyla tek çözüm ya da tek en iyi yerine birbirinden farklı çok sayıda iyi veya çözüm elde edilir. Çözüm kümesi de diyebileceğimiz bu çözümler topluluğu GA terminolojisinde popülasyon olarak bilinir. Bir popülasyon çoğunlukla vektörel şekilde ifade edilen kromozom veya birey adı verilen yapılardan oluşur. Bu mantıkla her bireyin içindeki elemana gen adı verilir. Nüfus bireyleri evrimsel süreçler tarafından değiştirilir, yenilenirler. Problemin çözümünü gösteren bireylerin, problem için bir çözüm olup olamayacaklarını bir uygunluk fonksiyonu belirler. Uygunluk fonksiyonuna uygulanan bireyler skorlarına göre sıralanarak bunlar arasındaki yüksek değerli bireyler kendi aralarında yeni bir nesil üretmeleri için çaprazlanırlar. Bu işlemler sırasında skoru ya da uygunluk değeri düşük olan bireyler popülasyondan ayrılırlar. Bu süreç, uygunluk değeri yüksek bireylerin sayısını artırarak arama uzayının zenginleşmesini ve çözüm kümesinin niteliğinin artmasını sağlar. Bir probleme ait en iyi çözümün bulunması için, bireyler doğru şekilde gösterilmeli, etkin bir uygunluk fonksiyonu belirlenmeli, genetik operatörlerin seçimi probleme uygun olmalıdır [142].

Bu şartların sağlanması durumda çözüm kümesi problem için bir noktada birleşerek belli bir optimizasyon aralığında, uygun bir çözüme ulaşacaktır. GA sistemleri, Kütle Spektrometri analizi gibi büyük arama uzayına sahip problemlerin çözümünde işlevseldirler. Problem çözümünde en iyi sonucu garantilemese de, optimal sonucu elde etmekte GA sistemleri işlevseldirler. Genel olarak optimizasyon tabanlı problemlerin çözümünde oldukça başarılı olan GA sistemleri, çözüm yöntemi bilinen ve doğru çözümü kesin ve hassas şekilde elde edilebilen problemlerde kullanılmazlar. GA sistemleri, arama uzayının büyük olduğu, eldeki veriyle çözüme ulaşılamadığı, çözümün zor olduğu ve problemin matematiksel olarak ifadesinin yapılamadığı durumlarda kullanılırlar [143].

Kütle Spektrometrisi deneylerinin öznitelik uzayının ya da öznitelik arama uzayının büyük olduğu düşünüldüğünde, GA sistemlerinin bu geniş uzayda uygun alt küme seçiminde işlevsel olduğu ve bu seçimi optimal şekilde gerçekleştirebileceği görülür [144].

Çalışmamızda, GA bir KNN sınıflandırıcısı ile birlikte zarflayıcı tabanlı özellik seçimi mantığıyla kullanılmıştır. Burada KNN sınıflandırıcısı, GA tarafından seçilen (k/y) altkümelerinin ne oranda işlevsel olduğunu belirleyen uygunluk fonksiyonu olarak kullanılmaktadır. KNN sınıflandırıcısı, arama uzayı olarak seçilen (k/y) değişken değerlerinden GA sisteminin seçerek kendisine verdiği alt kümelerin, kanserli-kansersiz örnek ayrımını ne derece doğru yaptığını değerlendirir. Hata miktarını minimize eden altküme optimum çözüm kümesi olarak seçilir.

Algoritmada 100 olarak belirlenen popülasyon sayısı ise, rastgele seçilen (k/y) değerlerine karşılık gelen Y değerleridir. Algoritma 50 kuşak oluşturacak şekilde çalıştırılarak bu çalışma sonunda ortaya çıkan (k/y) altkümesi seçilen özellikler olarak elde edilmiş olur.

Uygunluk fonksiyonu olarak kullandığımız KNN sınıflandırıcısı ise şu mantıkla sınıflandırma yapar; KNN öncelikle bir veri seti ile eğitilir. Daha sonra yeni noktanın önceden belirlenen “k” en yakın komşusuna bakarak noktanın dahil olacağı sınıf belirlenir. Bir başka ifade ile KNN daha önce eğitildiği ve öğrendiği özniteliklere bakarak, her yeni eklenen özniteliğin öğrendiği gruplardaki özniteliklere yakınlığına bakar. Öklit uzaklık ölçümü veya benzeri bir ölçümle bu uzaklığı belirler ve sınıflandırılması istenen özelliği buna göre bir gruba atar.

İzleyen bölüm olan, Bölüm 6’da bahsi geçen öğreticili ve öğreticisiz algoritmaların sınıflandırma başarımlarının hesapsal şekilde incelenmesi için bu çalışmada kullanılan metriklere yer verilecektir.

6. SINIFLANDIRMA BAŞARIMI ÖLÇÜM METRİKLERİ

Genel olarak sınıflandırma problemi farklı akıllı sistem teknikleriyle ele alınır. Her sınıflandırma probleminin çözümü daha önce o problemi ele alan çözüm algoritmalarıyla veya aynı çalışma içindeki diğer algoritmaların başarımlarıyla belli bir standarda uygun olarak kıyaslanmalıdır. Her algoritma belli parametrelere bağlı olarak belirli sonuçlar ürettiği için tüm sonuçları değerlendirmek için güvenilir karşılaştırma veya başarım ölçüm metriklerine ihtiyaç duyulmaktadır. Deneysel çalışma sonuçlarının istatistiksel sağlaması sonuçlarla ilgili çıkarımların doğruluğu için önem taşır [145].

Sınıflandırma başarımları öğreticili ve öğreticisiz sistemler için genel olarak farklı istatistiksel yaklaşımlarla ölçülür. Bu manada başarım metrikleri öncelikle öğreticili sistemler için açıklanırken daha sonra kümeleme tabanlı kalite metrikleri ele alınacaktır. Öğreticili algoritmanın kullanıldığı çalışmalarda aynı problemin çözümünde kullanılan algoritmalar arasındaki sınıflandırma başarımlarının kıyaslanması için literatürde değişik yöntemler önerilse de en yaygın olan sınıflandırma başarım ölçüm metrikleri; genel doğruluk (Accuracy, ACC), f-ölçümü (F-measure, Fm), Matthew'ün Korelasyon Katsayısı (Matthew's Correlation Coefficient, MCC), ROC eğrisi (Receiver Operating Characteristics Curve, ROC Curve), ROC eğrisi altında kalan alan (Area Under the ROC Curve, AUC), ortalama hata kareleri (Mean Squared Error, MSE), ortalama hata karelerinin kökü (Root Mean Squared Error, RMSE), Kappa Hatası (Kappa's Error, KE) ve k-katlı çapraz sağlama (K-fold Cross Validation) şeklindedir [146].

İzleyen bölümde öncelikle öğreticili sistemlerin sınıflandırma başarımı metrikleri ve Karışıklık Matrisi (Confusion Matrix, CM) ve daha sonra da öğreticisiz sistemlere ait başarım metrikleri SOM özelinde açıklanacaktır.

6.1. Karışıklık Matrisi

CM bir akıllı sistem algoritmasına ait sınıflandırma başarımı hakkında bütüncül bir yaklaşımla bilgi veren bir matris modelidir. CM yapısı sınıflandırıcının başarımı ve yapılan test ile yakından ilgilidir. Bu şekilde bir sınıflandırma başarımının CM olarak türetilmesi, deneysel çalışma ile ilgili tüm başarım metriklerinin kolayca hesaplanmasını mümkün kılar

[147]. Bir CM yapısal olarak Eşitlik 6.1'de olduğu gibi ifade edilir. CM'de her elemanın sınıflandırma başarımı ile ilgili özel bir anlamı vardır. Bu elemanlar diğer başarım metriklerinin tanımlanmasında referans olarak kullanılırlar.

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{6.1}$$

Eşitlik 6.1'de gösterilen ve dört elemana sahip olan CM'de her eleman, gerçek pozitif (True Positive, TP), yalancı pozitif (False Positive, FN), yalancı negatif (False Negative, FN) ve gerçek negatif (True Negative, TN) şeklinde tanımlanır. Daha açık bir ifadeyle matrisin her bir elemanı bir deneysel gözlemin sonucuna göre gerçek değer ile tahmin edilen değer arasındaki ilişkiyi gösterir [148]. İkilik sınıflandırma mantığı çerçevesinde sınıflandırmanın pozitif (Positive) ve negatif (Negative) olarak yapıldığı varsayılarak CM elemanlarının anlamı şu şekildedir:

i)TP: Gözlenen örneğin gerçek değerinin pozitif olması ve algoritma tarafından pozitif olarak doğru tahmin edilmesi durumudur.

ii)TN: Gözlenen örneğin gerçek değerinin negatif olması ve algoritma tarafından negatif olarak doğru tahmin edilmesi durumudur.

iii)FP: Gözlenen örneğin gerçek değerinin negatif olması ve algoritma tarafından pozitif olarak yanlış tahmin edilmesi durumudur.

iv)FN: Gözlenen örneğin gerçek değerinin pozitif olması ve algoritma tarafından negatif olarak yanlış tahmin edilmesi durumudur.

6.2. Öğreticili Sınıflandırma Başarım Metrikleri

Öğreticili sınıflandırma algoritmalarının başarımlarını ölçmek veya algoritmaların başarımlarını matematiksel olarak karşılaştırmak için bu çalışmada kullanılan metrikler, *ACC, Fm, AUC, RMSE, KE* ve CV sırasıyla ele alınacaktır. Bu metriklerin bir bölümü *TN*, *TP*, *FN* ve *FP* ifadelerinin yanı sıra metriklerin türetiminde kullanılan duyarlılık (Sensitivity, Sn), özgüllük (Specificity, Sp), artı yorum gücü (Positive Predictive Value, PPV) ve eksi yorum gücü (Negative Predictive Value, NPV) ifadeleri şu şekilde tanımlanır [149]:

i) Sn: Duyarlılık olarak bilinen bu metrik gerçek pozitiflerin, tüm pozitif tahminlerine oranıdır ve Denklem 6.2 ile tanımlanır.

$$Sn = TP/(TP + FN) \tag{6.2}$$

ii) Sp: Özgüllük olarak bilinen bu metrik gerçek negatiflerin, tüm negatif tahminlerine oranıdır ve Denklem 6.3 ile tanımlanır.

$$Sp = TN/(TN + FP) \tag{6.3}$$

iii) PPV: Artı yorum gücü olarak bilinen bu metrik pozitif olarak yapılan tahminlerin ne kadarının gerçek pozitif olduğunu bulan orandır ve Denklem 6.4 ile tanımlanır.

$$PPV = TP/(TP + FP) \tag{6.4}$$

iv) NPV: Eksi yorum gücü olarak bilinen bu metrik negatif olarak yapılan tahminlerin ne kadarının gerçek negatif olduğunu bulan orandır ve Denklem 6.5 ile elde edilir.

$$NPV = TN/(TN + FN) \tag{6.5}$$

Genel doğruluk, literatürde sınıflandırma performanslarını karşılaştırmak ve bir algoritmanın başarımını ölçmek için en yaygın kullanılan metrik olup, bu metrik bir sınıflandırıcının başarımına ait toplam verimliliği veya etkinliği olarak görülür. Genel doğruluk bir sınıflandırma modelinin doğru tahmin yapabilme gücünü veya ne kadar yanlış tahmin yaptığını ölçer [150]. ACC metriği matematiksel olarak Denklem 6.6 ile tanımlanır ve bu çalışmada algoritmalarımızın hastalık teşhis başarımlarını ölçmekte en sık kullandığımız hesaplama yöntemidir [151].

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \tag{6.6}$$

Genel doğruluk oranı yüzde olarak 100 değerine yaklaştıkça algoritmanın başarımının arttığı kabul edilir. Deneysel çalışmalarda algoritmanın başarımını kolayca yorumlanabilir bir şekilde hesapladığı için özellikle hastalık teşhis uzman sistemlerinin tasarımında yoğun olarak kullanılan bir metriktir [152].

F-ölçümü olarak bilinen metrik artı yorum gücü (PPV) ve duyarlılık (Sn) metriklerinin birleşimidir [153]. Duyarlılık bir algoritmanın pozitif sınıf etiketlerini belirleyebilme etkinliğini ölçerken, artı yorum gücü ile algoritmanın pozitif tahminleri ile pozitif sınıf etiketleri arasındaki uyum belirlenir. Bu şekilde F-ölçümü sınıflandırılacak verinin pozitif sınıfı ile algoritmanın pozitif tahminleri arasındaki bağıntıyı veya algoritmanın bir tür doğru tahmin yapabilme gücünü ölçer [154]. Bu şekilde *ACC* metriğine benzeyen *Fm* matematiksel olarak 6.7 ifadesiyle gösterilir [155].

$$Fm = \frac{(1+\beta^2)\times Sn\times PPV}{(\beta^2\times PPV)+Sn} \tag{6.7}$$

Denklem 6.7'de β değeri sıfır ile bir arasında değişmekte olup bu şekilde *Sn* veya *PPV* metriklerinin sonuca etkinliğinin ayarlanması mümkündür. Çoğunlukla β değeri 1 olarak seçildiğinden 6.7 ifadesi, Denklem 6.8'e dönüşür.

$$Fm = \frac{2\times Sn\times PPV}{PPV+Sn} \tag{6.8}$$

Genel doğruluk metriği, ACC'nin kullanıldığı çalışmalarda bu metriğin ürettiği sonuçların güvenirliğini desteklemek için ek bir metrik kullanılır. Literatürde bu amaçla Fm gibi bir ek metrik destekleyici olarak ACC ile birlikte kullanılır.

Ortalama hataların kareleri metriği literatürde sınıflandırma başarımlarının ölçümünde kullanılan bir başka metriktir. Bu metrik tahmin ile gerçek değer arasındaki farkın ölçülmesi işlevini yerine getirir. MSE bir tür risk fonksiyonu olup sınıflandırıcının tahmini ile gerçek değer arasındaki farkların ya da hataların karelerinin ortalamasıdır. MSE sınıflandırıcının varyansı ile algoritmanın sapmasını birleştiren bir metriktir. Bu tanıma bağlı olarak RMSE, MSE ifadesinin kökü olarak tanımlanır. Buna göre RMSE, bir algoritma tarafından tahmin edilen değerler ile gerçek değerler arasındaki farkın ölçümünü hesaplar. RMSE artı tahmin değeri ölçümü için iyi bir metrik olup bir sınıflandırıcının

doğru tahmin gücünü tek başına ölçebilir. RMSE sıfır ile bir arasında değerler alırken sıfıra yaklaşan değerler başarımı ölçülen algoritmanın daha verimli olduğunu gösterir [156]. İki gruplu bir sınıflandırma problemi için RMSE, Denklem 6.9 ile hesaplanır.

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + ... + (p_n - a_n)^2}{n}} \tag{6.9}$$

Denklem 6.9'da, *p* tahmin edilen değeri, *a* ise gerçek değeri temsil etmektedir. Bu çalışmada sınıflandırıcıların hastalık teşhisindeki başarımlarını ölçerken algoritmaların performansı aynı zamanda RMSE metriği kullanılarak gözlenmiştir.

KE bir veri setinde tahmin edilen sınıflandırma ile gözlem arasındaki uygunluğu ölçen bir metriktir. Bir başka ifadeyle, tahmin edilen sınıf etiketinin şans eseri olma ihtimali ile gerçek tahmin olup olmadığı bu metrik tarafından ölçülür. Bu şekilde bir sınıflandırıcının başarımının şans eseri olup olmadığı ve dolayısıyla ölçüm sonucunun güvenirliliği KE ile ölçülür [157]. Bir algoritmanın sadece tahminlerinin doğruluğunu ve yanlışlığını dikkate alan ACC değeri dikkate alınırsa sınıflandırıcının rastgele performansı yanıltıcı olabilir. Elde edilen sınıflandırma başarımının şans eseri olmadığının bir tür kanıtı olan KE değeri bire yaklaştıkça başarımın güvenirliği artarken, sıfıra yaklaştıkça başarımın güvenirliği azalır [158]. KE metriğini hesaplamak için Denklem 6.10 kullanılmaktadır.

$$KE = \frac{p_0 - p_c}{1 - p_c} \tag{6.10}$$

Denklem 6.10'da p_0 toplam uyuşma olasılığını ve p_c ise şans olasılığını ifade etmekte ve KE bu iki değere bağlı hesaplanmaktadır.

Sınıflandırma başarımında yaygın kullanılan bir diğer önemli metrik ROC eğrisi altında kalan veya literatürdeki bilinen adıyla *AUC* hesabıdır. ROC karakteristik eğrileri önce radyo sinyallerinin çalışılmasında kullanılmış ve sonrasında medikal teşhis sistemleri gibi çok değişik sınıflandırma problemleriyle birlikte uygulanmaya başlanmıştır. ROC eğrileri yakın zamanda akıllı sistem algoritmaların performansının incelenmesi konusunda vazgeçilmez metriklerden birisi haline gelmiştir [159]. Bir ROC eğrisi ikilik bir sınıflandırma problemi için the TPR'yi, FPR'nin bir fonksiyonu olarak çizer. Bu şekilde

bir sınıflandırıcının geniş ölçekli performansı elde edilmiş olur. Ortaya çıkan bu eğrinin altında kalan alan AUC olarak bilinir. AUC tek bir değer ürettiği için ROC eğrisine göre yorumu daha kolay bir metriktir [160]. AUC temel olarak bir sınıflandırıcının performansını yanlış sınıflandırmadan kaçma yeteneği olarak ölçer. Bu çalışmada AUC kullanılan sınıflandırma algoritmalarının başarımının ölçülmesinde sıklıkla kullanılan bir metriktir ve Denklem 6.11 ile hesaplanır.

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{6.11}$$

Sınıflandırma başarımının ölçümü sırasında algoritmanın verinin bir bölümüyle eğitimi ve daha sonra kalanıyla test edilmesi esastır. Çünkü aynı verinin hem eğitim hem de test için kullanımı aşırı uyumluluk (overfitting) sorununun ortaya çıkmasına neden olur. Bir diğer ifadeyle algoritma gerçek başarımdan ziyade mükemmel ancak aldatıcı sonuçlar üretir. Bu nedenle verinin belli bir standart ile eğitim ve test için bölünmesi gerekir. Literatürde bu anlamda en etkili veri bölümleme tekniği K-Katlı Çapraz Sağlama (K-Fold Cross Validation) veya Çapraz Sağlama (Cross Validation, CV) olarak bilinir [161].

K-fold CV tekniğinde *D* veri seti birbirinden farklı k bölüme ayrılır. Sınıflandırıcı bu bölümlerden ilk bölümü test için ayırarak kalan bölümleri eğitimde kullanır ve buna uygun performans hesaplanır. İkinci turda ikinci dilim test için ayrılırken kalan tüm bölümler eğitim için kullanılır. Bu işlem k bölümünün ardışık olarak izah edildiği gibi kullanılana dek devam eder. Bir sınıflandırıcının başarımı her turda elde edilen başarımların ortalaması şeklinde hesaplanır [162]. Literatürde en yaygın kullanılan CV hesabı verinin 10 parçaya ayrıldığı 10-katlı çapraz sağlama (10-fold Cross Validation) ile gerçekleştirilen hesaplamadır. Bu çalışmada tüm öğreticili algoritmaların başarımları hesaplanırken 10-katlı çapraz sağlama tekniğinden faydalanılmıştır. İzleyen bölümde öğreticisiz sınıflandırma veya kümeleme algoritmalarının kalitesi ve birbiriyle kıyası için gereken metrikler hakkında bilgi verilecektir.

6.3. Öğreticisiz Sınıflandırma Kalite ve Başarım Metrikleri

Öğreticisiz bir algoritma olarak sınıflandırma veya kümeleme başarımı incelenen SOM algoritmasının kalitesini ölçmek amacıyla literatürde yaygın şekilde kullanılan

metriklere ek olarak ele alınan genomik örüntü verileri için kullanılan kalite metrikleri de bu bölümde incelenecektir. SOM algoritmasına ait metrikler incelenirken, iki gruplu bir inceleme yapılacaktır. Buna göre ilk grupta SOM algoritmasına ait kümeleme tabanlı başarım metriklerine, ikinci grupta ise sınıflandırma-örüntü tanıma başarımına ait metriklere yer verilecektir.

6.3.1. SOM Algoritmasının Kümeleme Kalite Metrikleri

Bu bölümde SOM algoritması ve SOM algoritmasının optimize edilmiş halinin kümeleme başarımlarının karşılaştırılması için kullanılan kalite metrikleri incelenecektir. SOM algoritmasının kümeleme ya da haritalama başarımının ölçümünde öğreticili ve öğreticisiz metrikler kullanılmaktadır. Öğreticisiz ölçümler veya iç metrikler veri etiketlerine ihtiyaç duymadan bir kümeye atanmış veri vektörlerinin birbirine ne kadar benzediğini ve kümelerin birbirine ayrıklık miktarını ölçerler. Öğreticili ölçümler ise sınıflandırılan veya kümelenen verilerin etiketlerini kullanarak performans tabanlı ölçümler gerçekleştirirler. [163]. Literatürde sıklıkla karşılaşılan SOM kalite metrikleri Topografi Hatası (Topographic Error, TE), Topografik Çarpım (Topographic Product, TP), İç-Küme Hatası (Inter-cluster Error, IE), Kuantizasyon Hatası (Quantization Error, QE) ve Küme Kohezyonu (Clustering Cohesion, CC) şeklindedir.

Kuantizasyon Hatası (QE): Bir geniş ölçekli veri setinin daha az sayıda vektör ile temsil edilme kalitesini ölçer [164]. SOM algoritması için QE her bir örnek vektörün ait olduğu kümenin merkezine ortalama uzaklığı olarak ölçülür ve bu metrik Denklem 6.12 ile hesaplanır.

$$e_q(m) = \sum_{x \varepsilon C_m} \| x - m \| \tag{6.12}$$

Denklem 6.12'de C küme merkezini, m ilgili kümenin ağırlık vektörünü ve x o kümedeki örnek vektörleri temsil eder. Buna göre QE değerinin küçülmesi ilgili kümenin ağırlık vektörü ile aralarındaki farkın küçülmesi veya vektörlerin küme içindeki benzerliğinin artarak kümeleme kalitesinin artması anlamına gelir.

Topografik Hata (TE): Bir SOM haritasının kalitesi topografik hata fonksiyonu ile ölçülür. Topografik hata giriş vektörlerinin çıkışa projeksiyonu sırasında giriş verilerine ait

topolojinin ne kadar korunduğunun ölçüsüdür. Bir diğer ifade ile bu ölçüm giriş ile çıkış arasındaki topolojilerin farkını ölçer [165]. TE, Denklem 6.13 ile hesaplanır.

$$E_t = \frac{\sum_{k=1}^{N} u(x_k)}{N} \tag{6.13}$$

Denklem 6.13'te $u(x)$ SOM haritasında bir giriş vektörünün en yakın küme merkezine olan uzaklığını temsil etmektedir.

Küme Kohezyonu (CC): Kümeleme algoritmalarının kalite ölçüm yöntemlerinden birisi bir küme içindeki vektörlerin birbirine ne kadar benzer olduğunun ölçülmesidir. Küme kohezyonu olarak bilinen bu metrik bir küme içindeki vektörlerin benzerliği arttıkça kümeleme kalitesinin arttığını gösterir. Kohezyon, bir küme içindeki kareler toplamı ile ölçülür ve Denklem 6.14 ile hesaplanır.

$$C_{cohesion} = \sum_{i} \sum_{x \varepsilon C_i} (x - m_i)^2 \tag{6.14}$$

Denklem 6.14'te C_i i kümesinin numarasını, x örnek vektörü ve m_i ise ilgili kümenin ağırlık vektörünü temsil eder [118].

İç-küme Uzaklık Hatası: Bir kümeleme algoritmasının kalitesini ölçmekte kullanılan bir diğer metrik iç-küme uzaklığıdır. İç-küme uzaklığı minimize edildikçe kümeleme kalitesi artar. İç-küme uzaklığı, bir küme içindeki tüm vektör çiftleri arasındaki uzaklık olarak tanımlanır [163]. Ortalama iç-küme uzaklığı, { S_1, S_2, S_3…, S_i,…, Su} noktalarına sahip C_1 küme merkezi için Denklem 6.15 ile hesaplanır.

$$I_{av} = \left(\frac{\sum_{i=1}^{u} \sum_{j=1}^{u} D(S_i, S_j)}{u^2}\right) \tag{6.15}$$

Denklem 6.15'te $D\ (S_i, Sj)$ düğümdeki veri vektörleri arasındaki uzaklığı, u ise düğüm sayısını temsil eder.

6.3.2. SOM Algoritmasının Başarım Metrikleri

SOM algoritmasının çözümünde kullanıldığı problemlerin kümeleme ya da haritalama başarımının kalitesi, kullanılan verinin yapısına bağlı olarak bahsi geçen metriklerle ölçülür. Bu amaçla SOM algoritması ile analiz edilen ve Bölüm 2'de izah edilen çeşitli medikal verilerin ne kadar başarıyla kümelendiği veya sınıflandırıldığını nicelik olarak ölçebilmek için ekstra metrikler kullanılmıştır. SOM başarım metrikleri olarak bölüm 6.1'de bahsedilen *ACC*, Sn, Sp, PPV ve eksi yorum gücü NPV metriklerine ek olarak Performans Katsayısı (Performance Coefficient, PC), MCC ve olasılık aşımı (Probability Excess, PE) indeksleridir. Bu metriklerden SOM performans katsayısı Denklem 6.16 ile hesaplanır [166].

$$PC = \frac{TP}{(TP + FN + FP)} \tag{6.16}$$

Literatürde çok sıklıkla kullanılan bir diğer başarım metriği *MCC* olarak bilinir. Bu metrik Denklem 6.17 ile hesaplanır [167].

$$MCC = \frac{(TP \cdot TN - FN \cdot FP)}{\sqrt{[(TP + FN)(TN + FP)(TP + FP)(TN + FN)]}} \tag{6.17}$$

Diğer bir SOM sınıflandırma başarım metriği ise olasılık aşımı olarak bilinir ve Denklem 6.18 ile hesaplanır [168].

$$PE = Sn + Sp - 1 \tag{6.18}$$

Bu bölümde öğreticili ve öğreticisiz akıllı sistemlerin değişik koşullarda performansını analiz ederken kullanılan sınıflandırma başarım metrikleri incelenmiştir. Bir

sonraki bölümde ise bu tezde gerçekleştirilen deneysel çalışmalar ve incelenen algoritmaların başarımlarına ait sonuçlar verilecektir.

7. MEDİKAL VERİLERİN SINIFLANDIRMA BAŞARIMININ ANALİZİ

Bu bölümde, daha önce bölüm 5'te izah edilen akıllı sistem algoritmalarıyla analizi gerçekleştirilen Parkinson, dermatoloji, diyabet, mamografi, genomik örüntüler ve kütle spektrometrisi tabanlı kanser hastalıkları verilerinin sınıflandırma başarımına ait sonuçlara verilecektir. Sonuçların bir bütünlük oluşturması açısından tüm deneysel veriler dört ayrı grupta toplanmıştır. Bunlar RFEL uyarlamalarının sınıflandırma başarımına etkisi, öznitelik seçiminin performansa etkisi, kütle spektrometri verilerinin ön-işleme adımlarının algoritma sonuçları üzerindeki belirleyici etkisi ve SOM algoritmasının PSO ile optimizasyonuna ait etkilerin incelenmesi şeklinde sıralanabilir.

7.1. RFEL Algoritmasıyla Parkinson Hastalığının Teşhisi

Akıllı sistem algoritmalarının sınıflandırma performansının arttırılması uzman hastalık teşhis sistemlerinin tasarımı için önem taşımaktadır. Parkinson hastalığının bir örnek veri seti olarak seçildiği ve bu hastalığın (dolayısıyla başka hastalıkların) teşhis başarımının bir topluluk öğrenme yöntemi olan, RFEL algoritması uyarlaması ile arttırıldığı hesapsal şekilde gösterilmiştir. Bu amaçla WEKA yazılımından seçilen 30 adet sınıflandırıcının her birinin hem RFEL uyarlamasında hem de yalnız başlarına Parkinson hastalığının teşhisinde gösterdikleri performans, hassaslık ACC, KE ve AUC metrikleri ile birlikte değerlendirilmiştir. Bu deneysel çalışmaya ait bilgiler izleyen alt bölümde açıklanmıştır.

7.1.1. Sınıflandırma Sonuçları

Parkinson hastalığının teşhis başarımını test etmek için olabildiğince farklı türdeki 30 akıllı sistem algoritması seçilmiştir. Bu algoritmalara ait orijinal isimler Tablo 7.1'de verilmiştir. Tablo 7.1'deki algoritmalara ait RFEL topluluk öğrenmesi uyarlamalarının başarımına ait KE, ACC ve AUC başarım metriklerinin değerleri ise Tablo 7.2' de verilmiştir. Algoritma isimlerinin kullanımını kolaylaştırmak için her algoritma Tablo 7.1'de sıralandığı numara ile temsil edilmiş ve böylece başarım sonuçlarının grafiksel gösterimi kolaylaştırılmıştır.

Tablo 7.1. WEKA yazılımından seçilen akıllı sistem algoritmaları

No	Algoritma	No	Algoritma	No	Algoritma
1	BLR	11	FLR	21	ADT
2	BayesNet	12	HiperPipes	22	BFDT
3	Naive Bayes	13	VFI	23	Decision Stump
4	Logistic	14	Conjunctive Rule	24	FL
5	MLP	15	RIPPER	25	J48 (C4.5)
6	RBF Network	16	Nnge	26	Decision Tree
7	Simple Logistic	17	OneR	27	LMT
8	SVM	18	PART	28	Random Tree
9	KSTAR (K*)	19	RIPPLE	29	FDTL
10	LWL	20	ZeroR	30	Chart

Tablo 7.2. Akıllı sistem algoritmalarının RFEL uyarlanmasına ait başarımları

Alg.	ACC(%)	*e*ACC(%)	Fark(%)	Kappa	*e*Kappa	Fark	AUC	*e*AUC	Fark
1	75.4	86.2	10.8	0.41	0.56	0.15	0.648	0.845	0.197
2	82.6	85.1	2.5	0.53	0.58	0.05	0.827	0.847	0.02
3	77.4	75.9	**-1.5**	0.49	0.41	-0.08	0.788	0.768	-0.02
4	84.1	83.1	**-1**	0.56	0.53	-0.03	0.837	0.828	-0.009
5	89.7	90.3	0.6	0.73	0.74	0.01	0.898	0.903	0.005
6	87.7	87.7	**0**	0.64	0.64	0	0.873	0.872	-0.001
7	85.1	86.2	1.1	0.56	0.59	0.03	0.843	0.854	0.011
8	87.2	87.7	0.5	0.58	0.61	0.03	0.856	0.863	0.007
9	91.8	**93.8**	2	0.78	0.85	0.07	0.917	0.94	0.023
10	84.1	88.2	4.1	0.55	0.63	0.08	0.838	0.871	0.033
11	82.6	85.6	3	0.53	0.61	0.08	0.827	0.855	0.028
12	84.1	86.6	2.5	0.45	0.52	0.07	0.824	0.857	0.033
13	73.3	78.5	5.2	0.42	0.51	0.09	0.751	0.797	0.046
14	80.5	81	0.5	0.36	0.32	-0.04	0.777	0.766	-0.011
15	88.2	89.7	1.5	0.65	0.71	0.06	0.877	0.894	0.017
16	88.2	88.8	0.6	0.65	0.66	0.01	0.877	0.879	0.002
17	86.2	86.7	0.5	0.58	0.57	-0.01	0.852	0.849	-0.003
18	81.1	92.3	11.2	0.46	0.79	0.33	0.805	0.922	0.117
19	87.2	89.7	2.5	0.65	0.66	0.01	0.872	0.881	0.009
20	75.4	75.4	**0**	0.37	0.37	0	0.648	0.648	0
21	88.7	92.3	3.6	0.7	0.78	0.08	0.888	0.922	0.034
22	89.2	90.8	1.6	0.69	0.73	0.04	0.888	0.904	0.016
23	83.1	85.6	2.5	0.52	0.55	0.03	0.826	0.844	0.018
24	83.6	87.7	4.1	0.52	0.66	0.14	0.829	0.876	0.047
25	85.6	89.7	4.1	0.62	0.72	0.1	0.857	0.896	0.039
26	88.7	92.3	3.6	0.71	0.79	0.08	0.888	0.922	0.034
27	86.2	88.2	2	0.61	0.68	0.07	0.856	0.881	0.025
28	84.6	91.3	6.7	0.57	0.71	0.14	0.844	0.893	0.049
29	82.6	85.7	3.1	0.54	0.61	0.07	0.827	0.855	0.028
30	88.7	91.8	3.1	0.68	0.78	0.1	0.856	0.892	0.036
ORT			2.7	0.57	0.63		0.833	0.861	

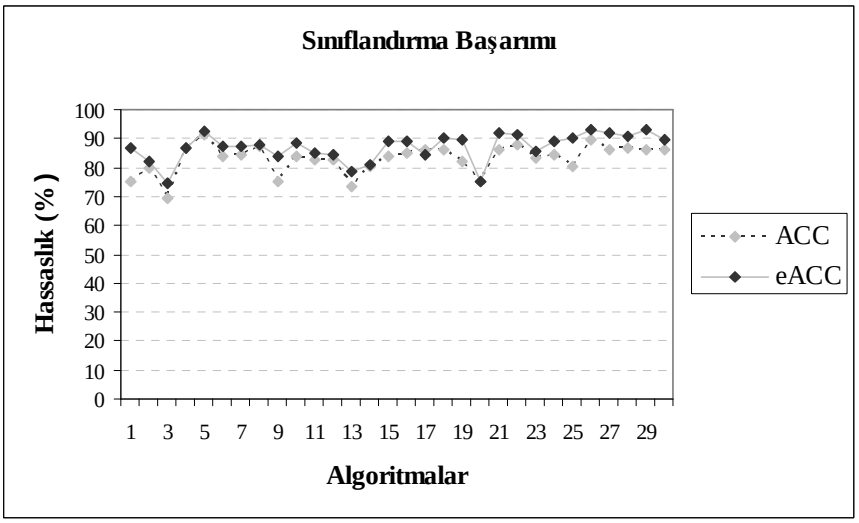
Tablo 7.2'de, '*e*' öneki akıllı sistem algoritmasının RFEL uyarlamasına ait ölçümünü sembolize etmektedir. Yine Tablo 7.2'de 'ORT' kısaltması 'ortalama' anlamında kullanılmıştır.

Tablo 7.2, KE, ACC ve AUC metrikleri paralelinde incelediğinde 30 temel sınıflandırıcının 26 adedinde RFEL topluluk öğrenme uyarlamasının performansı arttırdığı görülmektedir. Diğer 4 algoritmanın ikisinde sınıflandırma performansında ACC değeri üzerinden değişme görülmemekte diğer ikisinde ise azalma görülmektedir. Daha açık bir ifadeyle 6 ve 20 numaralı algoritmalarda (RBF ve ZeroR) bir performans değişimi olmazken, 3 ve 4 numaralı algoritmalarda (Naive Bayes ve Logistic) sınıflandırma başarımının düştüğü görülmektedir. Bu iki algoritma için ACC değerleri % 77,4'ten % 75,9'a ve % 84,1'den %83,1'e düşmüştür. Algoritmaların performans kaybının doğruluğu KE ve AUC değerlerindeki azalma ile desteklenmektedir. Bölüm 6'da izah edildiği gibi her iki metrik 0 ile 1 arasında değerler almaktadır ve metriklerin 1'e yaklaşması ölçümün hassaslığının ya da sınıflandırma başarımının arttığını göstermektedir.

Akıllı sistem algoritmalarının RFEL uyarlamalarının başarım hassaslığı dikkate alındığında tüm algoritmaların başarımında ortalama % 2.7'lik bir artış görünmektedir. Bu artış akıllı sistem algoritmalarının RFEL algoritması ile birlikte oluşturulan *topluluk öğrenmesi* türevlerinin başarısı anlamına gelmektedir. Öte yandan Tablo 1'de yer alan {1, 10, 13, 18, 24, 25, 28} numaralı ve yüksek başarımlı algoritmalar bu ortalamadan çıkarıldığında bile dahi algoritmaların başarımında ortalama % 1.6'lık artış görülmektedir. Bu başarım artışı, RFEL algoritmasının akıllı sistem algoritmalarına ait uyarlamalardaki başarımını göstermektedir.

Tablo 7.1'de elde edilen sınıflandırma başarımlarının güvenirliliği KE değerleriyle örtüşmektedir. Bir sınıflandırma probleminde başarımın güvenirliliği KE değerinin 1 değerine yaklaşmasıyla artmaktadır. Başarım sonuçları KE değerine göre incelendiğinde, sınıflandırma performansı yükselen topluluk öğrenmesi algoritmalarına ait KE değerlerinin topluluk öğrenmesi uyarlanmamış algoritmalarla kıyaslandığında yükseldiği gözlenmektedir. Böylece RFEL algoritmasıyla elde edilen topluluk tabanlı algoritmaların başarımlarının şans eseri olmadığı ve yöntemin güvenilir olduğu sonucuna varılabilir. RFEL modeli uygulanmış ve uygulanmamış olan algoritmaların Parkinson hastalığına ait sınıflandırma başarımlarının karşılaştırmalı grafiği Şekil 7.1'de gösterilmiştir.

Hastalık teşhis sistemlerinin başarımının arttırılması için kullanılan yöntemlerin literatürde önem kazandığı düşünüldüğünde RFEL tabanlı akıllı sistem algoritmalarının başarımının yüksekliği benzer problemler için bu yöntemin sağlıklı olarak işleyeceğini hesapsal olarak göstermektedir.



Şekil 7.1. Akıllı sistem algoritmalarının topluluk öğrenme modeliyle başarımları

7.2. Öznitelik Seçiminin Sınıflandırma Başarımına Etkisi

Bu bölümde RFEL, BN ve SVM tekniklerinin öznitelik seçimine uygulanmasının, sınıflandırma performansına olan etkisi incelenecektir. Tüm deneysel çalışmalarda öznitelik seçimi olmadan yapılan sınıflandırmaların başarımı ve bahsi geçen tekniklerle öznitelik seçimi yapılarak elde edilen sınıflandırmaların başarımı kıyaslanmıştır. Böylece öznitelik seçiminin sınıflandırma başarımına etkisi gözlemlenmiş olup aynı zamanda literatürde ilk kez öznitelik seçme probleminde kullanılan rotasyoncu topluluk algoritmasının öznitelik seçimindeki başarımı incelenmiştir.

7.2.1. SVM Öznitelik Seçiminin Parkinson Hastalığına Uygulanması

Bu deneysel çalışmada 31 Parkinson hastasına ait 195 ses kaydının girildiği 23 öznitelikli veri seti kullanılmıştır. Çalışmada bu 23 özniteliğin tamamının kullanılmasıyla sınıflandırma yapmanın sınıflandırma performansını olumsuz etkileyebileceği düşünülerek SVM algoritması yardımıyla öznitelik seçimi gerçekleştirilmiştir. Bu şekilde 23 öznitelikten en değerli 10 öznitelik seçilerek bu özniteliklerin değeri akıllı sistemler aracılığıyla sınıflandırılarak ölçülmüştür. Seçim işlemi sırasında, öznitelik havuzundan rastgele sayıda öznitelik alınarak, SVM algoritmasına verilmiş ve öznitelikler SVM algoritmasında ürettikleri başarıma göre büyükten küçüğe doğru sıralanmışlardır. Parkinson hastalığına ait özniteliklerden seçilen en etkin 10 öznitelik Tablo 7.3'te

verilmiştir. Tablo 7.3'teki özniteliklerin numarası sınıflandırma performansında en etkin özniteliğin en üstte yer alması anlamına gelmektedir.

Tablo 7.3. SVM ile seçilen en etkin öznitelikler

No	Öznitelik Adı	No	Öznitelik Adı
1	Spread1	6	MDVP_APQ
2	MDVP_Fo_Hz	7	DFA
3	D2	8	HNR
4	Spread2	9	PPE
5	MDVP_Fhi_Hz	10	RPDE

SVM algoritmasının eğitimi sırasında 10'lu CV tekniği kullanılarak kalitesi düşük öznitelliklerin seçilmesi engellenmiştir.

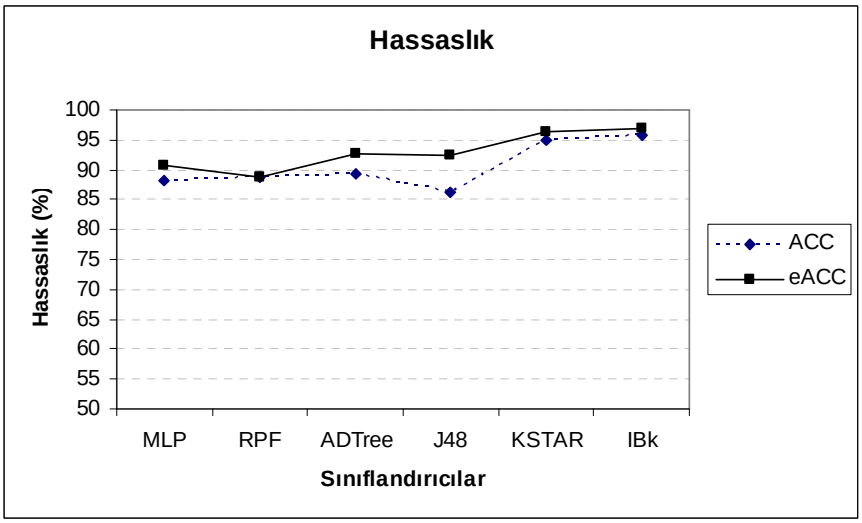
Seçilen öznitelikler MLP, RBF, ADT, J48 (C4.5 Java sürümü), KSTAR ve IBk şeklinde farklı yapıda 6 algoritma seçilerek sınıflandırılmıştır. Her sınıflandırıcının RFEL uyarlaması ACC değerini ve dolayısıyla başarımı arttırmıştır. Gerek sınıflandırıcıların normal başarımı gerekse topluluk öğrenme başarımları Tablo 7.4'te verilmiştir. Hesapsal başarımların doğruluğu, Tablo 7.4'te verilen KE ve AUC değerleri ilede desteklenmiştir.

Tablo 7.4'teki değerlerde 'e' öneki RFEL algoritma uyarlamalarına karşılık olarak kullanılmıştır. Tablo 7.4'teki 'Fark' sütunlarında gözlenen KE ve AUC değerlerindeki pozitif artış elde edilen başarımların güvenirliliğini desteklemektedir.

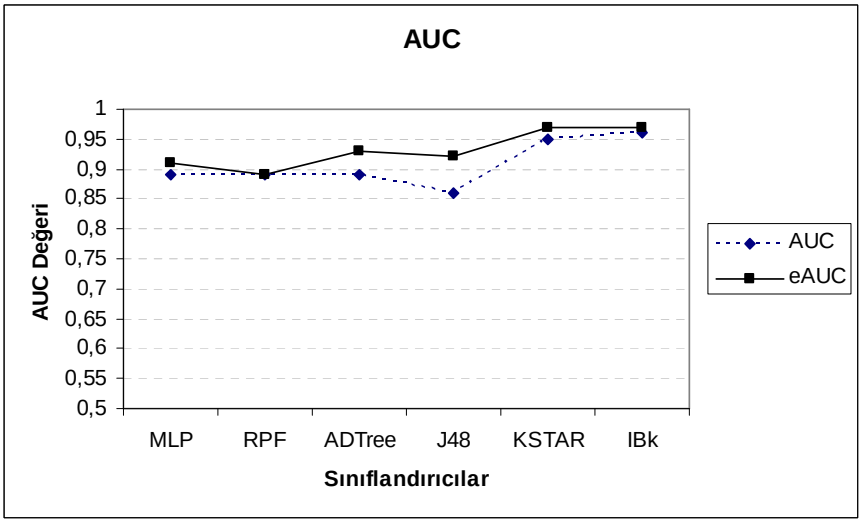
Tablo 7.4. Algoritmaların Parkinson hastalığını teşhis performansı

Algoritma	ACC(%)	*e*ACC(%)	Fark (%)	Kappa	*e*Kappa	Fark	AUC	*e*AUC	Fark
MLP	88.21	90.8	2.59	0.69	0.75	0.06	0.89	0.91	0.02
RPF	88.71	88.71	0	0.68	0.68	0	0.89	0.89	0
ADTree	89.23	92.82	3.59	0.71	0.81	0.1	0.89	0.93	0.04
J48	86.15	92.3	6.15	0.63	0.78	0.15	0.86	0.92	0.06
KSTAR	**94.91**	**96.41**	1.51	0.86	0.91	0.05	0.95	0.97	0.02
IBk	**95.89**	**96.93**	1.04	0.89	0.92	0.03	0.96	0.97	0.01

Tablo 7.4'teki değerlerin toplu şekilde yorumlanmasını kolaylaştırmak için algoritmaların ve topluluk öğrenmesi karşılıklarının hassaslıkları ile AUC değerleri sırasıyla Şekil 7.2 ve Şekil 7.3'te grafiksel olarak verilmiştir. Gerek Tablo 7.4 değerleri, gerekse Şekil 7.2 ve Şekil 7.3 incelendiğinde RBF dışında kalan algoritmaların Parkinson hastalığını teşhis hassasiyetlerinin RFEL uyarlamasıyla arttığı görülmektedir.



Şekil 7.2. Algoritmaların Parkinson hastalığını teşhis hassasiyeti



Şekil 7.3. Algoritmaların Parkinson hastalığını teşhis hassasiyetinin AUC değerleri

Literatürde Parkinson hastalığının teşhisini inceleyen iki çalışmanın [168,169] elde ettikleri hassasiyet sırasıyla %91,4 ve %92,9 şeklindedir. Tablo 7.4 incelendiğinde KSTAR ve IBk algoritmalarının % 94,91 ve % 95,89 hassaslık değerleriyle daha iyi sonuç verdiği görülmektedir. Bu sonuçlar SVM algoritmasına dayanan öznitelik seçiminin başarımını hesapsal şekilde göstermektedir. Öte yandan Tablo 7.4'teki algoritmalara RFEL uyarlamasıyla elde edilen başarım sonuçlara yine aynı tabloda yer verilmiştir. RFEL uyarlamasıyla, KSTAR ve IBk algoritmalarında % 96,41 ve %

96,93 şeklinde daha yüksek bir başarıma ulaşıldığı Tablo 7.4'te açıkça görülmektedir.

7.2.2. RFEL Öznitelik Seçiminin Parkinson Hastalığının Teşhisine Uygulanması

Bölüm 7.1'de, RFEL algoritması öznitelik seçimiyle sınıflandırma başarımının arttırılmasında kullanılmış ve algoritmanın dermatoloji veri setinde başarımı gösterilmişti. Bu bağlamda çalışmamızda ele alınan bir diğer veri seti olan Parkinson hastalığına ait 23 öznitelikten en belirgin özniteliklerin seçilmesinde SVM yerine RFEL tekniği kullanılmıştır. Tekniğin başarımını değerlendirmek için RFEL algoritmasına ek olarak Adaboost, Decorate, Bagging ve Dagging topluluk öğrenme algoritmaları yardımıyla Parkinson hastalığına ait öznitelik seçimi yapılmıştır. Böylece literatürde sıklıkla kullanılan topluluk öğrenme algoritmalarının öznitelik seçme performansı karşılaştırmalı olarak incelenmiştir.

Bu öznitelik seçimi yöntemimizde, değişik topluluk öğrenme algoritmaları zarflayıcı öznitelik seçiminin çekirdek algoritması olarak kullanılmıştır. Öznitelik seçiminde hesapsal yükü azaltmak için zarflayıcı algoritmalarımız, BFS arama tekniğini ve ileri seçme stratejisini kullanarak tüm öznitelik gruplarını aramak yerine optimum sayıya daha kısa bir zamanda ulaşmıştır. Algoritmalar çalıştırıldıklarında 10 çaprazlı-sağlama yöntemiyle kontrol edilerek aşırı uyum problemine karşı korunma sağlanmıştır. Algoritmaların Parkinson hastalığına ait verilerden seçtikleri öznitelikler Tablo 7.5'te gösterilmiştir.

Öznitelik seçme algoritmalarının çalıştırılmasından sonra elde edilen boyutu azaltılmış veri setinin bulduğu özniteliklerin gücü veya değerini test etmek için veriler sırasıyla MLP, RBF, KSTAR, IBk, NNGE, RIDOR, ADT ve CART (Classification and Regression Trees) algoritmalarına verilmişlerdir. Bu sekiz algoritmanın başarımlarının yanında sınıflandırma performansını destekleyen RMSE, KE ve AUC metrikleri hesaplanmıştır. MLP, RBF, KSTAR, IBk, NNGE, RIDOR, ADT ve CART algoritmaları, öznitelik seçimiyle elde edilen verileri kullanarak çalıştırılmış ve elde edilen KE, RMSE, ACC, AUC değerleri sırasıyla Tablo 7.6, Tablo 7.7 ve Tablo 7.8 ve Tablo 7.9'da verilmiştir.

Tablo 7.5. Topluluk öğrenmesi algoritmaları tarafından seçilen öznitelikler

Öznitelikler	Rotation Forest	Decorate	Bagging	Dagging	Adaboost
MDVP:Fo(Hz)	✓	✓	✓		
MDVP:Fhi(Hz)				✓	✓
MDVP:Flo(Hz)					
MDVP:Jitter(%)	✓				
MDVP:Jitter(Abs)		✓		✓	✓
MDVP:RAP					
MDVP:PPQ					
Jitter:DDP			✓		
MDVP:Shimmer			✓		
MDVP:Shimmer(dB)					
Shimmer:APQ3	✓	✓			
Shimmer:APQ5			✓		
MDVP:APQ					
Shimmer:DDA	✓		✓		
NHR					
HNR			✓		
RPDE	✓				
D2				✓	✓
DFA	✓				
spread1				✓	✓
spread2					
PPE	✓	✓	✓	✓	✓

Tablo 7.6. Topluluk algoritması Parkinson özniteliklerinin sınıflandırılmasına ait Kappa değerleri

Topluluk Algoritması	Kappa Değeri							
	MLP	**RBF**	**IBK**	**KSTAR**	**NNGE**	**RIDOR**	**ADT**	**CHART**
Rotation Forest	0.719	**0.651**	**0.944**	**0.824**	**0.827**	**0.716**	**0.763**	**0.791**
Decorate	0.735	0.643	0.675	0.712	0.769	0.643	0.739	**0.791**
Bagging	**0.779**	0.468	0.723	0.799	0.757	0.468	0.719	0.704
Dagging	0.735	0.643	0.675	0.712	0.769	0.643	0.739	**0.791**
Adaboost	0.696	0.554	0.668	0.753	0.682	0.678	0.699	0.657

Tablo 7.6'daki KE değerleri ekseninde incelendiğinde, RFEL algoritmasının diğer topluluk öğrenme algoritmalarına kıyasen sekiz algoritmanın yedisinde ya en yüksek ya da aynı değere ulaşarak, KE değerleri bakımından performansı arttırdığı görülmektedir.

Tablo 7.7. Topluluk algoritması Parkinson özniteliklerinin sınıflandırılmasına ait RMSE Değerleri

Topluluk Algoritması	RMSE Değeri							
	MLP	**RBF**	**IBK**	**KSTAR**	**NNGE**	**RIDOR**	**ADT**	**CHART**
Rotation Forest	0.293	**0.312**	**0.143**	**0.222**	**0.248**	0.321	0.287	**0.277**
Decorate	0.255	**0.312**	0.358	0.279	0.286	**0.312**	**0.283**	**0.277**
Bagging	**0.249**	0.361	0.328	0.249	0.295	0.361	0.296	0.327
Dagging	0.307	0.343	0.351	0.289	0.336	0.343	0.294	0.318
Adaboost	0.306	0.343	0.351	0.288	0.336	0.336	0.284	0.319

Bir algoritmanın sınıflandırma başarımının RMSE değerlerinin küçülmesiyle arttığı düşünülürse, Tablo 7.7'de RFEL algoritmasının sekiz algoritmanın beşinde daha başarılı olduğu görülmektedir. Bu sonuca en yakın algoritma dört değerle Decorate algoritmasıdır. Ancak bu algoritma da, Rotasyoncu topluluk algoritmasının beş değerine karşılık dört iyi RMSE değerine sahiptir.

Literatürde sınıflandırma başarımlarını ölçmek için yaygın olarak kullanılan diğer bir metrik de, Tablo 7.8'de ACC ile temsil edilen hassaslıktır. Elde edilen değerler incelendiğinde, RFEL yönteminin sekiz algoritmanın yedisinde diğer algoritmalarla ya aynı ya da daha yüksek değere ulaştığı görülmektedir. Bu metrikte Rotasyoncu öğrenme algoritması hassaslık anlamında diğer tüm topluluk öğrenme algoritmalarına göre ciddi bir başarım göstermiştir.

Tablo 7.8. Topluluk algoritması Parkinson özniteliklerinin sınıflandırılmasına ait ACC Değerleri

Topluluk Algoritması	**ACC Değerleri**							
	MLP	**RBF**	**IBK**	**KSTAR**	**NNGE**	**RIDOR**	**ADT**	**CHART**
RFEL	89.74	**88.21**	**97.95**	**93.33**	**93.85**	**89.75**	**91.28**	**92.31**
Decorate	89.74	87.69	87.18	89.23	91.79	87.69	90.26	**92.31**
Bagging	**91.79**	83.58	89.23	92.30	91.28	83.59	89.74	88.72
Dagging	88.72	84.62	87.69	91.28	88.72	84.62	89.23	88.72
Adaboost	88.72	84.62	87.69	91.28	88.72	88.72	89.23	88.72

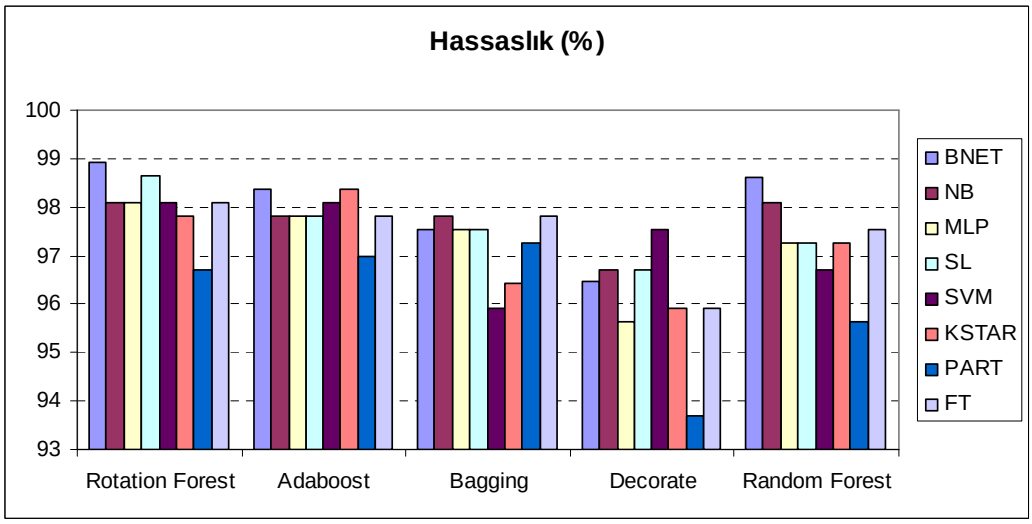
Topluluk öğrenme algoritmalarının Parkinson hastalığının en etkin özniteliklerinin seçimindeki etkisinin ölçümünde kullanılan son metrik AUC değeridir. AUC değeri herhangi bir sınıflandırıcı algoritmanın başarımının ölçülmesinde hassasiyet kadar etkin bir değerdir ve AUC değeri bire yaklaştıkça başarım artışını ifade etmektedir. Tablo 7.9 bu kriterle incelendiğinde Rotasyoncu Topluluk öğrenme stratejisinin sekiz algoritmanın yedisinde diğer topluluk öğrenme algoritmalarından daha iyi bir AUC değerine sahip olduğu görülmektedir.

Tablo 7.9. Topluluk algoritması Parkinson özniteliklerinin sınıflandırılmasına ait AUC değerleri

Topluluk Algoritması	**AUC Değeri**							
	MLP	**RF**	**IBK**	**KSTAR**	**NNGE**	**RIDOR**	**ADT**	**CHART**
RFEL	0.897	**0.875**	**0.979**	**0.934**	**0.937**	**0.896**	**0.913**	**0.923**
Decorate	0.899	0.872	0.875	0.893	0.916	0.872	0.903	**0.923**
Bagging	**0.918**	0.815	0.895	0.924	0.911	0.815	0.897	0.889
Dagging	0.882	0.840	0.877	0.910	0.885	0.840	0.888	0.879
Adaboost	0.887	0.840	0.877	0.910	0.885	0.884	0.890	0.879

Sınıflandırma performansını en somut şekilde gösteren ACC metriğine ait değerler, Şekil 7.4'teki grafikten kolayca kıyaslanabilir.

Sınıflandırıcı algoritmaların başarımını arttırmakla ilgili çalışmalar arasında en çok değinilen alanlardan birisi öznitelik seçim problemidir. Bu amaçla çok değişik yöntemler kullanılmakta ve geliştirilmekte olup topluluk öğrenme algoritmalarının bu alanda kullanımı güncel çalışmalar arasındadır. RFEL algoritmasının literatürde en sık kullanılan topluluk öğrenme algoritmalarıyla öznitelik seçme performansı anlamında kıyaslanması, bu algoritmanın dört metrikte belirgin olarak diğer öğrenme algoritmalarına göre daha başarılı olduğunu göstermiştir. Bu yöntemin benzer sınıflandırma problemlerinde kullanılmasının faydalı olacağı düşünülmektedir. Algoritmanın Parkinson hastalığının öznitelik seçiminde etkin olması nedeniyle aynı algoritma dermatoloji veri setine uygulanmış ve deneysel sonuçlar bir sonraki alt bölümde verilmiştir.



Şekil 7.4. Topluluk öğrenme algoritmalarının öznitelik seçme performansı

7.2.3. RFEL Öznitelik Seçiminin Dermatoloji Hastalığına Uygulanması

Rotasyoncu topluluk öğrenme algoritması akıllı sistem algoritmalarında sınıflandırma başarımını artırmak üzere kullanılmış bir algoritmadır. Bu algoritmanın öznitelik seçme problemi için başarıyla kullanılabileceği Parkinson hastalığı çalışmamızda gösterilmiştir. Bir diğer medikal veri analiz çalışmamızda Rotasyoncu topluluk algoritmasının dermatoloji hastalığına ait öznitelikleri seçme ve dolayısıyla elde edilen

özniteliklerin sınıflandırma başarımına etkisi incelenmiştir. Yine literatürde sıklıkla kullanılan topluluk öğrenme algoritmaları Rotasyoncu topluluk algoritması ile karşılaştırılarak algoritmanın performansı hesapsal şekilde ortaya konmuştur. Dermatoloji hastalığının 33 özniteliği arasından 12 öznitelik seçilerek verinin boyutu indirgenmiş ve daha sonra elde edilen veriler BN, NB, MLP, SL, SVM, KSTAR, Karar Ağaçlı Kural Öğrenmesi (Decision Tree Based Rule Laerner, PART) ve FT algoritmalarına verilerek sınıflandırma başarımı ölçülmüştür. Bu sınıflandırmaların sonuçları ACC metriğinin yanında RMSE, ve KE metrikleriyle de hesaplanarak elde edilen sonuçların doğruluğu desteklenmiştir. Bahsi geçen sınıflandırma algoritmalarının performanslarının ölçümünü veya başarımlarının doğruluğunu destekleyici olarak kullanılan RMSE, KE, AUC ve ACC metriklerine ait sonuçlar Tablo 7.10, Tablo 7.11 ve Tablo 7.12'de verilmiştir.

Tablo 7.10. Topluluk algoritması dermatoloji özniteliklerinin sınıflandırılmasına ait RMSE değerleri

Topluluk Algoritması	**RMSE Değeri**							
	BNET	**NB**	**MLP**	**SL**	**SVM**	**KSTAR**	**PART**	**FT**
RFEL	**0.0651**	0.0797	0.0836	**0.0822**	**0.3107**	0.0816	**0.0991**	**0.0760**
Adaboost	0.0745	**0.0733**	**0.0777**	0.0968	0.3108	**0.0811**	0.0999	0.0851
Bagging	0.0864	0.0813	0.0880	0.0906	0.3117	0.1067	0.0953	0.0861
Decorate	0.0882	0.0915	0.1080	0.0991	0.3109	0.0951	0.1356	0.1026
Random Forest	0.0714	0.0817	0.0860	0.0957	0.3117	0.0893	0.1096	0.0851
Öznitelik Seçimi Yok	0.0658	0.0832	0.0822	0.0915	0.3111	0.1169	0.1145	0.0845

Tablo 7.10'daki sekiz sınıflandırıcının performansı RMSE değerleri üzerinden incelendiğinde en küçük beş değerin Rotasyoncu topluluk algoritması, kalan üç değerin Adaboost algoritması tarafından elde edildiği görülmektedir. Tablo 7.10, bu metrik için Rotasyoncu topluluk algoritmasının diğer topluluk öğrenmesi algoritmalarına göre daha başarılı olduğunu göstermektedir.

Tablo 7.11. Topluluk algoritması dermatoloji özniteliklerinin sınıflandırılmasına ait Kappa değerleri

Topluluk Algoritması	**Kappa Değeri (KE)**							
	BNET	**NB**	**MLP**	**SL**	**SVM**	**KSTAR**	**PART**	**FT**
RFEL	**0.9863**	**0.9761**	**0.9760**	**0.9829**	**0.9760**	0.9726	0.9589	**0.9761**
Adaboost	0.9795	0.9726	0.9726	0.9726	**0.9760**	**0.9795**	0.9624	0.9727
Bagging	0.9692	0.9726	0.9692	0.9692	0.9486	0.9555	**0.9658**	0.9726
Decorate	0.9555	0.9590	0.9453	0.9590	0.9692	0.9486	0.9212	0.9487
Random Forest	0.9829	**0.9761**	0.9658	0.9658	0.9589	0.9658	0.9453	0.9692
Öznitelik Seçimi Yok	0.9761	0.9693	0.9692	0.9658	0.9624	0.9317	0.9488	0.9692

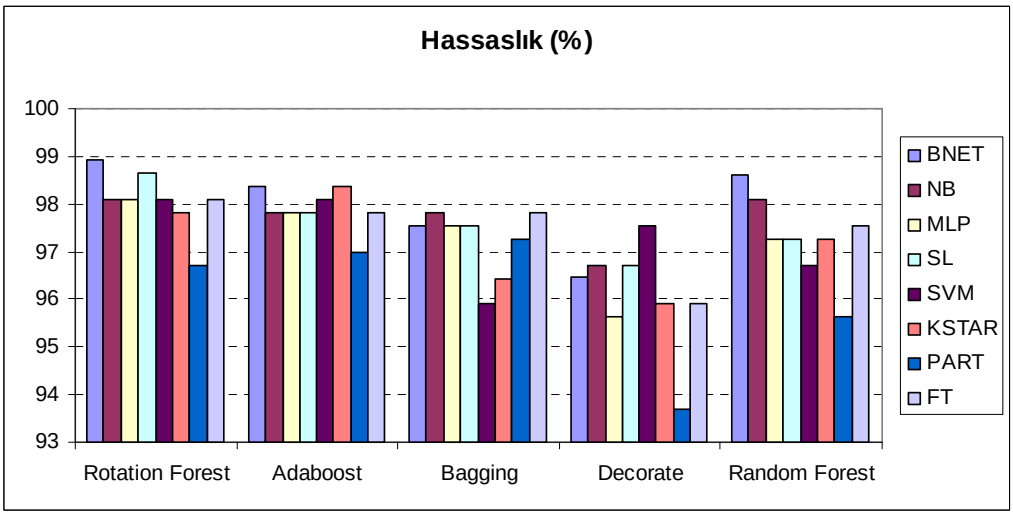
Tablo 7.10'da görülen bir diğer belirgin sonuç ise, öznitelik seçiminin özniteliklerin tamamının kullanıldığı sınıflandırmaya göre RMSE değeri üzerinden belirgin bir iyileşme gösterdiğidir.

Tablo 7.11 Kappa değeri açısından incelendiğinde, sekiz sınıflandırıcının altı tanesinde, Rotasyoncu Topluluk algoritmasının daha iyi sonuç verdiği, diğer topluluk öğrenme algoritmalarının bir veya iki değerde Rotasyoncu Topluluk algoritmasından daha iyi sonuç verdiği görülmektedir. KE değerleri, Rotasyoncu Topluluk algoritmasının seçtiği özniteliklerin diğer öğrenme algoritmalarına göre daha yüksek nitelikli olduğunu göstermektedir.

Tablo 7.12. Topluluk algoritması dermatoloji özniteliklerinin sınıflandırılmasına ait ACC değerleri

Topluluk Algoritması	**Hassaslık (%)**							
	BNET	**NB**	**MLP**	**SL**	**SVM**	**KSTAR**	**PART**	**FT**
RFEL	**98.91**	**98.08**	**98.08**	**98.64**	**98.08**	97.81	96.72	**98.08**
Adaboost	98.36	97.81	97.81	97.81	**98.08**	**98.36**	96.99	97.81
Bagging	97.54	97.81	97.54	97.54	95.90	96.44	**97.26**	97.81
Decorate	96.45	96.72	95.62	96.72	97.54	95.90	93.71	95.90
Random Forest	98.63	**98.08**	97.26	97.26	96.72	97.26	95.62	97.54
Öznitelik Seçimi Yok	98.08	97.54	97.54	97.26	96.99	94.53	95.90	97.54

Tablo 7.12'daki, ACC değerlerine göre, RFEL algoritmasının seçtiği özniteliklerin performansı, sekiz algoritmanın altısında diğer topluluk öğrenme algoritmalarına göre ya aynı veya daha yüksektir. Sonuçlar Şekil 7.5'te karşılaştırmalı olarak verilmiştir.



Şekil 7.5. Topluluk algoritmalarının seçtikleri özniteliklerin hassaslıklarının karşılaştırılması

Tablo 7.10, Tablo 7.11 ve Tablo 7.12'de elde edilen sonuçlar, Şekil 7.5 ile birlikte değerlendirildiğinde RFEL algoritmasının dermatoloji hastalığı için seçtiği özniteliklerin başarımının yüsek olduğunu göstermektedir. Parkinson hastalığında benzer bir sonucu veren algoritmanın bu gibi sınıflandırma problemlerinde başarıyla kullanılabileceğini göstermektedir.

Bir sonraki alt bölümde öznitelik seçiminde dermatoloji veri setinin zarflanmış BN yardımıyla analizi ile ilgili deneysel sonuçlara yer verilecektir.

7.2.4. BN Öznitelik Seçiminin Dermatoloji Hastalığının Teşhisine Uygulanması

GA ile zarflanmış BN sınıflandırıcısı yardımıyla dermatoloji hastalığına ait 33 öznitelikten en etkin özniteliklerin seçilmesi bu alandaki bir diğer çalışmamızdır. Dermatoloji hastalığının öznitelik sayısının düşürülmesiyle elde edilen yeni veri setinin sınıflandırma performansı bu yöntemin başarımını gösterir. BN sınıflandırıcısının GA ile zarflanmasının yanında yöntemin etkinliğinin analizi için BN sırasıyla BFS ve Ardışık Kayan Arama (Sequential Floating Search, SFS) algoritmaları yardımıyla zarflanmış ve bu algoritmalar dermatoloji hastalığına ait özniteliklerin seçilmesinde kullanılmıştır.

GW, BFW ve SFW algoritmalarının seçtiği özniteliklerle beş ayrı sınıflandırıcı algoritma (SVM, MLP, SL, FT, BN) kullanılarak yapılan sınıflandırmanın başarım sonuçları, yöntemlerin öznitelik seçim performansını hesapsal şekilde göstermiştir. Zarflayıcı algoritmalar ve sınıflandırma algoritmaları Tablo 7.13'te verilmiştir.

Tablo 7.13. Zarflayıcı algoritmalar ve sınıflandırma algoritmaları

No	Algoritmalar	Kısaltmaları
1	Genetik Zarflayıcı (Genetic Wrapper)	GW
2	En İyi Birinci Zarflayıcı (Best First Wrapper)	BFW
3	Ardışık Kayan Zarflayıcı (Sequential Floating Wrapper)	SFW
4	Destek Vektör Makinesi (Support Vector Machine)	SVM
5	Çok Katmanlı Perseptron (Multi Layer Perceptron)	MLP
6	Basit Lojistik (Simple Logistics)	SL
7	Fonksiyonel Karar Ağacı (Functional Decision Tree)	FT
8	Bayes Ağı (Bayes Network)	BN

Zarflayıcı algoritmaların her üçünün (GW, BFW ve SFW) çalıştırılmasıyla üretilen öznitelikler beş ayrı sınıflandırıcı algoritma ile sınıflandırılmış ve yöntemlerin öznitelik

seçme başarımı, beş algoritmanın ürettiği ACC ve KE değerleri hesaplanarak izlenmiştir. Zarflayıcı algoritmaların her iki metrikteki başarımına ait sonuçlar Tablo 7.14'te verilmiştir.

Tablo 7.14. Dermatoloji hastalığı özniteliklerinin zarflayıcılarla seçim başarımı

Zarflayıcı Algoritmalar	**Algoritmalar**									
	SVM		**MLP**		**SL**		**FT**		**BN**	
	ACC	**KE**	**ACC**	**KE**	**ACC**	**KE**	**ACC**	**KE**	**ACC**	**KE**
Genetic	97.3	0.96	**97.8**	**0.97**	**97.8**	**0.97**	**98.1**	**0.98**	**99.2**	**0.99**
Best First	**97.5**	**0.97**	95.6	0.94	97.0	0.96	96.0	0.96	98.9	0.98
Sequential Float	95.3	0.94	97.5	0.96	**97.8**	**0.97**	97.8	0.97	98.9	0.98
Öznitelik Seçimsiz	97.0	0.96	97.5	0.96	97.2	0.96	97.5	0.96	98.1	0.97

Tablo 7.14, hem ACC hem de KE metrikleri ile birlikte incelendiğinde GW yönteminin, beş sınıflandırıcının dördünde diğer zarflayıcılara göre sınıflandırma başarımında daha iyi veya aynı sonucu ürettiğini göstermektedir. Bu sonuçlar, BN tabanlı GW öznitelik seçiminin başarımını ortaya koymaktadır. Tablo 7.14'teki sonuçlardan en yükseği olan % 99,18 değeri BN algoritması tarafından üretilmiştir. Bu değer dermatoloji hastalığını baz alan çalışmalar arasında literatürdeki en yüksek değerdir [172].

Bu çalışmanın gösterdiği bir başka değerli sonuç GA türü zarflayıcı algoritmaların öznitelik seçme performansının, kullanılan mutasyon ve çaprazlama parametrelerinin değerlerine bağlı olduğunu göstermesidir. Çalışmamızda, mutasyon oranı sabit tutulup çaprazlama oranını 0,6 değerinden 0,9 değerine artırdığımızda seçilen öznitelikler değişmiş ve BN sınıflandırıcısının performansı düşmüş ve algoritma % 98,90'lik ACC değerini üretmiştir.

Bir sonraki bölümde farklı kanserlere ait kütle spektrometrisi verilerinin ön-işleme adımlarının sınıflandırma başarımına etkisi incelenecek ve çalışmamıza ait deneysel sonuçlar verilecektir.

7.3. Önişleme Adımlarının Kütle Spektrometrisi Verisinin Analizine Etkisi

Çeşitli kanser hastalıklarına ait kütle spektrometrisi verilerinin sınıflandırılması için verilerin öncelikle baz çizgisi doğrulanması, normalizasyon ve gürültüden arındırılması gibi ön işleme adımlarından geçirilmesi gerekmektedir. Sonraki aşamada ise verilerin öznitelik seçimi ile boyutunun düşürülmesi ve sonrasında akıllı sistem algoritmaları

yardımıyla elde edilen verilerin sınıflandırılması gerekmektedir. Bu bölümde, kütle spektrometrisi ön-işleme adımlarının sınıflandırma başarımına etkisi hesapsal şekilde gösterilmiştir.

Üç ön-işleme adımından geçirilen veriler GA ile birlikte kullanılan KNN (GA-KNN), PCA-LDA, SVM ve ANN algoritmaları yardımıyla sınıflandırılmışlardır. Ön-işleme adımlarının etkisinin hesapsal şekilde ortaya konabilmesi için sınıflandırmadan önce her ön işleme verilere önce tek tek, sonra ikişerli ve sonra üçü birden uygulanarak sınıflandırma algoritmalarına verilmişlerdir. Ön-işleme adımlarının sonuçlarının daha görülebilmesi için veriler ön-işleme adımı uygulanmadan da sınıflandırılmıştır. Yüksek çözünürlüklü rahim kanseri, prostat kanseri ve düşük çözünürlüklü rahim kanseri verilerinin bu şekilde analizi ile ilgili sonuçlar sırasıyla Tablo 7.15, Tablo 7.16 ve Tablo 7.17'de verilmiştir.

Tablo 7.15. Kütle spektrometrisi verilerinde ön-işleme adımlarının sınıflandırma başarımına etkisi

Veri Seti	**Sınıflandırıcı**	**Hassaslık (%)**			
		Ham Veri	**Baz-Çizgisi**	**Normalizasyon**	**Gürültü Giderme**
Yüksek Çözünürlüklü Rahim Kanseri	GA-KNN	84.63	85.74	86.43	90.23
	PCA-LDA	95.69	98.04	98.95	97.84
	SVM	92.46	92.67	93.95	94.21
	NN	84.64	86.51	89.76	92.09
Prostat Kanseri	GA-KNN	77.51	80.59	84.72	85.87
	PCA-LDA	93.22	94.03	95.56	94.84
	SVM	87.99	88.98	89.07	92.17
	NN	86.24	89.06	89.99	91.87
Düşük Çözünürlüklü Rahim Kanseri	GA-KNN	91.66	95.85	92.69	96.25
	PCA-LDA	94.82	99.80	99.90	99.44
	SVM	91.32	99.47	99.42	98.89
	NN	97.20	98.40	100.00	98.80

Tablo 7.15 incelendiğinde ham yani ön-işleme uygulanmayan veriler ile tek tek ön-işleme adımı uygulanan verilerin sınıflandırma performansları kıyaslandığında her işleme adımının sınıflandırma performanslarında ham veriye oranla net bir artış meydana getirdiği görülmektedir.

Her bir ön-işleme adımının sınıflandırma performansına etkisini gösteren sonuçlar ise Tablo 7.16'da verilmiştir.

Tablo 7.16. Ön-işleme Adımlarının Sınıflandırma Başarımına Ortalama Katkısı

Veri Seti	Sınıflandırma	Baz-Çizgisi	Normalizasyon	Gürültü Giderme
Yüksek Çözünürlüklü Rahim Kanseri	GA-KNN	1.11	1.8	5.6
	PCA-LDA	2.35	3.26	2.15
	SVM	0.21	1.49	1.75
	NN	1.87	5.12	7.45
Prostat Kanseri	GA-KNN	3.08	7.21	8.36
	PCA-LDA	0.81	2.34	1.62
	SVM	0.99	1.08	4.18
	NN	2.82	3.75	5.63
Düşük Çözünürlüklü Rahim Kanseri	GA-KNN	4.19	1.03	5.59
	PCA-LDA	4.18	5.08	4.62
	SVM	8.15	8.1	7.57
	NN	1.2	2.8	1.6
Ortalama Değişim		2.58	3.59	4.68

Tablo 7.16'nın gösterdiği gibi ön-işleme adımlarından gürültü giderme, normalizasyon ve baz çizgisi doğrulama, tüm sınıflandırma başarımlarında % 2,58 ile % 4,68 arasında bir artışa sağlamışlardır. Gürültü giderme adımı, her üç veri setine uygulanan dört algoritmanın sınıflandırma performansında ortalama % 4,68'lik bir artış sonucunu vermiştir. Sonraki aşamada, yöntemlerin sınıflandırma başarımına etkisini belirlemek için her ön-işleme adımı ikişerli ve üçlü bir grupla birbirlerine bağlı şekilde veri setlerine uygulanmış ve bu ölçümlere ait sonuçlar Tablo 7.17'de verilmiştir.

Tablo 7.17. Ön-işleme Adımlarının Birlikte Gerçekleştirilmesinin Başarıma Etkisi

Veri Seti	Sınıflandırma	Ön-işleme Adımları Birbirine Bağlıyken			
		BÇ & NRM	BÇ & GG	NRM & GG	BÇ & GG & NRM
Yüksek Çözünürlüklü Rahim Kanseri	GA-KNN	86,53	89,68	88,47	89,40
	PCA-LDA	98,12	95,12	94,75	94,91
	SVM	93,05	93,58	92,49	92,01
	NN	85,11	89,76	93,48	86,51
Prostat Kanseri	GA-KNN	84,32	83,69	82,64	84,35
	PCA-LDA	93,72	92,23	94,45	93,78
	SVM	88,53	89,83	88,26	89,76
	NN	89,06	86,25	87,50	89,68
Düşük Çözünürlüklü Rahim Kanseri	GA-KNN	89,96	96,40	85,81	94,11
	PCA-LDA	99,68	99,98	99,58	99,84
	SVM	98,46	99,37	95,19	99,34
	NN	99,60	99,60	98,80	100,00
	Ortalama	92,18	92,96	91,78	92,81

Tablo 7.17, kütle spetkrometrisi verilerinin ham (ön-işleme uygulanmamış) ve işlenmiş (ön-işleme uygulanmış) durumlarına ait sınıflandırılma başarımlarının ortalama değerleri gösterilmiştir. Bu tabloda, ön-işleme adımı uygulanmayan ham verinin sınıflandırma başarım ortalaması % 89.78, baz çizgisi ve gürültü giderme adımları uygulan veri için % 92.96, her üç ön-işleme adımının birlikte uygulandığı veri için % 92.81'lik bir başarım elde edilmiştir.

Bu çalışma kütle spektrometrisi verisinin ön-işleme adımları ile sınıflandırma başarımı arasında yakın bir ilişki olduğunu hesapsal şekilde göstermiştir. Bu şekilde kütle spektrometrisi verilerinin sınıflandırılma performansının yükseltilmesi için ön-işleme adımlarının dikkate alınması gerektiği niceliksel olarak ortaya konmuştur.

7.4. SOM Eğitim Süresinin Optimizasyonu ve Sınıflandırma Başarımının İlişkisi

Öğreticisiz bir akıllı kümeleme-sınıflandırma sistemi olan SOM algoritması veri setleri büyüdükçe eğitim süresi artan bir yapıya sahiptir. SOM algoritmasını oluşturan tüm düğümlerin giriş vektörlerine bağlı olarak eğitilmesi bir tam eğitim süresi olarak bilinir. Uygulamadan uygulamaya değişmekle birlikte bir SOM eğitimi ortalama olarak literatürdeki yaklaşımlara göre yüz civarında tam eğitim süresi ile tamamlanır. SOM algoritmasının bu görece uzun eğitim süresinin kümeleme başarımını düşürmeden optimizasyonu ve bu optimizasyonla sınıflandırma başarımının, kümeleme kalitesinin değişimi literatürde az incelenen konular arasındadır. Bu tez çalışmasında öğreticili akıllı sistemlerin sınıflandırma performansı ile ilgili çok sayıda deney gerçekleştirilmiştir. Bu nedenle öğreticisiz bir sistem olan ve literatürde çok yoğun şekilde kullanılan SOM algoritmasının sınıflandırma başarımının incelenmesi ve uzun süren eğitim süresinin optimizasyonu bir diğer çalışma olarak seçilmiştir.

SOM algoritmasının sınıflandırma başarımını olabildiğince koruyarak PSO algoritmasıyla eğitim süresinin optimizasyonuna ait deneysel sonuçlara bu bölümde yer verilecektir. PSO ile optimize edilmiş algoritmamız sürü optimizeli organize haritalar yaklaşımından (Sworm Optimized Organizing Map, SWOM) konu içerisinde SWOM olarak kısaltılacaktır. Kohonen tarafından önerilen klasik optimizasyon yöntemi de SWOM algoritmasının performansını karşılaştırmak adına çalışmada dinamik kendine organize haritalar (Dynamic Self Organizing Map, DSOM) algoritmasından DSOM olarak bahsedilecektir.

SOM, SWOM ve DSOM algoritmalarının başarımları algoritmanın kümeleme kalitesi ve sınıflandırma başarımı anlamında iki grup metrikle ölçülecektir. Kümeleme kalitesi metrikleri QE, TE, prototip tabanlı kohezyon (Prototype Based Cohesion, PBC) ve grafik tabanlı kohezyon (Graph Based Cohesion, GBC) şeklindedir. Sınıflandırma performansları için kullanılacak metrikler ise, Sn, Sp ve MCC şeklindedir.

Çalışmalarda kullanılan veri setlerinin adları ve tablolarda yer alan kısaltmaları şu şekildedir: Genomik örüntü veri setleri LEXA, HM17, DM05 ve CBF1 olarak isimlendirilmiştir. Bu örüntüler sırasıyla koli basili, insan, meyve sineği ve maya canlılarına aittir. Hastalık veri setleri ise diyabet, mamografi ve Arcene veri setlerine aittir. Bunlardan Arcene kütle spektrometrisi verilerinden hazırlanmış ve performans amaçlı testler için kullanılan bir kanser grubu veri setidir. Bir diğer grup veri seti ise koli basili ve maya canlılarına ait içerisinde sırasıyla on ve sekiz ayrı sınıf barındıran, çalışmada koli basili ve maya olarak kısaltılan protein lokalizasyon site verileridir.

Her üç algoritma bu üç sınıf veri seti için eğitilmiş ve algoritmaların eğitim zamanları, kümeleme performans metrikleri ve sınıflandırma başarım indeksleri ölçülmüşlerdir.

SOM, SWOM ve DSOM algoritmalarının genomik örüntüler için eğitilmesi sırasında yapılan ölçümler ve sonuçları sırasıyla Tablo 7.18, Tablo 7.19 ve Tablo 7.20'de yer almıştır. Tablolarda algoritma isimleri gerektiğinde SOM için S, SWOM için SW ve DSOM için DS olarak kısaltılmışlardır.

Tablo 7.18. SOM, SWOM ve DSOM algoritmalarının genomik örüntü kümeleme performansları

Metrik	LEXA			HM17			DM05			CBF1		
	S	SW	DS	S	SW	DS	S	SW	DS	S	SW	DS
QE	**0.490**	0.501	**0.496**	**0.511**	**0.522**	0.526	**0.618**	**0.622**	0.625	**0.566**	0.571	0.576
TE	**0.921**	**0.922**	0.927	**0.877**	**0.883**	0.889	**0.870**	**0.871**	0.875	**0.864**	0.870	**0.866**
PBC	0.223	**0.222**	0.233	**0.301**	**0.301**	0.304	**0.414**	0.421	**0.420**	**0.341**	**0.359**	0.362
GBC	**0.591**	0.621	**0.616**	**0.605**	0.609	**0.607**	**0.641**	**0.651**	0.653	**0.567**	**0.601**	0.603

Tablo 7.18 metrikleri sıfıra yaklaştıkça algoritmaların kümeleme performansı daha iyi kabul edilmektedir. Buna göre klasik SOM algoritması kümeleme metriklerinde SWOM ve DSOM algoritmalarından daha iyi olmakla en iyi birinci ve en iyi ikinci değerler ele toplam 16 ölçümden 14'ünde SOM, 11'inde SWOM ve 5 tanesinde DSOM başarılı sonuçlar üretmiştir. SOM algoritması beklendiği gibi en iyi sonuçların çoğunu

üretirken, SWOM algoritması SOM algoritmasına en yakın sonuçları üreterek bu metrikler için DSOM'a göre başarısını göstermiştir.

Tablo 7.19. SOM, SWOM ve DSOM algoritmalarının örüntü bulma performansları

Örüntü Bulma Başarımı	LEXA			HM17			DM05			CBF1		
	S	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**
Sn	**0.37**	0.27	**0.31**	**0.33**	**0.50**	0.22	**0.11**	**0.12**	0.09	0.58	**0.71**	**0.59**
Sp	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
MCC	**0.45**	0.32	**0.34**	**0.33**	**0.48**	0.19	**0.13**	**0.13**	0.11	0.72	**0.78**	**0.72**

Tablo 7.19 birbirine çok yakın Sp değerleri cinsinden üç algoritma arasında belirgin bir fark göstermese de, bir değerine yakınlık anlamında daha iyi olma kriteriyle bakıldığında Sn ve MCC metrikleri kolay yorumlanabilir sonuçlar üretmişlerdir. Buna göre Tablo 7.19'da en iyi birinci ve ikinci değerler olarak performans ele alındığında SOM 12 değerden 6'sını, SWOM bu değerlerden yine altısını, DSOM ise dört tanesini almıştır. Tüm algoritmalar içinde SWOM bu altı değerin altısında da en iyi performansa ulaşmış ve sınıflandırma performansı anlamında SOM algoritmasından daha iyi sonuçlar üretmiştir. Elde edilen sonuçlara bakıldığında SOM ve SWOM algoritma sonuçlarının DSOM algoritmasına göre birbirine daha yakın olduğu görülmektedir.

Her üç algoritmanın genomik örüntü veri setleri için eğitilmesi sırasında elde edilen eğitim zamanları Tablo 7.20'de verilmiştir.

Tablo 7.20. SOM, SWOM ve DSOM algoritmalarının Genomik Örüntü Tanıma Eğitim zamanları

Epoç	LEXA			HM17			DM05			CBF1		
	S	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**
10	51	37	30	107	93	81	126	119	108	197	150	132
50	259	52	48	539	274	257	644	329	327	969	354	352
100	513	70	62	1102	478	469	1242	573	562	1993	625	618

Tablo 7.20'deki sonuçlara bakıldığında gerek SWOM ve gerekse DSOM algoritmalarının örüntü tanıma probleminde SOM algoritmasına göre çok ciddi zaman kazancı sağladıkları ve kabaca birbirinin aynı değerler ürettikleri gözlenmiştir.

Medikal veri seti grubundaki veriler ile yine üç algoritma eğitilmiş ve bu deneylere ait sonuçlar Tablo 7.21, Tablo 7.22 ve Tablo 7.23'te verilmiştir.

Tablo 7.21. Algoritmaların biyomedikal verilerle eğitimine ait kümeleme metrikleri

Veriseti	**Algoritmalar**	**Kümeleme Kalite Metrikleri**			
		QE	**TE**	**GBC**	**PBC**
Diyabet	SOM	**0.059**	**0.154**	**0.054**	**0.054**
	SWOM	**0.060**	**0.154**	**0.055**	**0.051**
	DSOM	0.070	0.201	0.074	0.068
Mamografi	SOM	**0.058**	**0.128**	**0.053**	**0.054**
	SWOM	0.060	0.138	0.049	0.052
	DSOM	**0.054**	**0.137**	**0.072**	**0.051**
Arcene	SOM	**0.599**	**0.050**	**0.414**	**0.500**
	SWOM	**0.611**	**0.061**	**0.473**	**0.529**
	DSOM	0.680	0.066	0.511	0.558

Kümeleme metrikleri anlamında algoritmaların performansının yorumlanması için Tablo 7.21 incelendiğinde, 12 değerin hepsinde SOM en iyi birinci değeri alırken SWOM bu 12 değerin 8 tanesinde en iyi ikinci değeri almış buna karşın DSOM 12 değerin 4 tanesinde en iyi ikinci değer elde etmiştir. Bu sonuçlar SWOM'un kümeleme metriklerinin çoğunda SOM'a yakın değerler ürettiği ve bu anlamda DSOM'a göre daha başarılı olduğunu göstermektedir.

Tablo 7.22. SOM, SWOM ve DSOM algoritmalarının sınıflandırma başarımı

Verisetleri	Algoritmalar	Sınıflandırma Başarımı		
		Sn	Sp	MCC
Diyabet	SOM	0.717	0.761	0.479
	SWOM	**0.727**	**0.772**	**0.500**
	DSOM	**0.731**	**0.775**	**0.507**
Mamografi	SOM	0.813	0.811	0.625
	SWOM	**0.819**	**0.816**	**0.635**
	DSOM	**0.814**	**0.811**	**0.625**
Arcene	SOM	**0.901**	**0.902**	**0.801**
	SWOM	**0.909**	**0.911**	**0.821**
	DSOM	0.897	0.900	0.797

Tablo 7.22 medikal setlerin performansını Sn, Sp ve MCC değerleri ile ölçmektedir. Tablo 22 incelendiğinde 12 değerden en iyi 6 birinci değeri SWOM algoritması alırken, DSOM sadece 3 değerde birinci olabilmiştir. SOM ise bu değerlerin hiç birinde en iyi olamamıştır. Deney sonuçları incelendiğinde SWOM'un hem SOM hem de DSOM'a göre

sınıflandırma performansı anlamında daha başarılı olduğu ancak değerler arasında çok farklılık olmadığı görülmektedir. Her üç algoritmanın eğitimi için harcanan zamana ait değerler ise Tablo 7.23'te verilmiştir.

Tablo 7.23. SOM, SWOM ve DSOM algoritmalarının eğitim zamanı

Epoç	Diyabet			Mamografi			Arcene		
	S	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**
10	0.39	0.28	0.23	0.27	0.25	0.23	155	123	114
50	1.45	1.02	0.89	1.23	0.94	0.86	987	418	391
100	2.88	1.83	1.63	2.53	1.67	1.56	1554	599	546

Tablo 7.23 her üç algoritma için harcanan zamanı referans kabul ederek incelendiğinde SWOM ve DSOM algoritmalarının eğitim sürelerini SOM'a göre çok ciddi şekilde kısalttıkları görülmektedir.

Algoritmaların test edilmesi için kullanılan koli basili ve maya canlılarına ait protein lokalizasyon kümelerinin belirlenmesi ile ilgili yapılan deneyler ve sonuçları Tablo 7.24, Tablo 7.25, Tablo 7.26'da ve Tablo 7.27'de verilmiştir.

Tablo 7.24'teki sonuçların ortalamalarına 10 sınıflı maya verisi üzerinden bakıldığında SWOM algoritmasının 3 sınıflandırma performans metriğinin ikisinde, SOM algoritmasının bir tanesinde başarılı olduğu, DSOM'un ise bu metriklerde sıralamaya giremediği görülür. Benzer şekilde, Tablo 25'te SWOM yine üç ortalama değerin ikisinde, SOM algoritması bir tanesinde başarılı olurken, DSOM bu alanda yeterli başarıyı göstermemiştir.

Tablo 7.24. Algoritmaların koli basili protein lokalizasyon site sınıflandırma performansı

Sınıf	Sn			Sp			MCC		
	S	**SW**	**DS**	**S**	**SW**	**DS**	**S**	**SW**	**DS**
Mit	0.590	0.635	0.586	0.919	0.928	0.918	0.509	0.563	0.504
Nuc	0.578	0.517	0.538	0.828	0.803	0.812	0.406	0.321	0.350
Cyt	0.632	0.656	0.699	0.833	0.844	0.863	0.466	0.500	0.563
me1	0.659	0.795	0.704	0.989	0.993	0.990	0.648	0.789	0.695
Exc	0.371	0.542	0.542	0.984	0.988	0.988	0.356	0.531	0.531
me2	0.549	0.352	0.313	0.983	0.976	0.975	0.532	0.329	0.289
me3	0.773	0.668	0.601	0.971	0.959	0.950	0.744	0.627	0.552
Vac	0	0	0.033	0.979	0.979	0.980	-0.020	-0.020	0.013
Pox	0.550	0.550	0.550	0.993	0.993	0.993	0.543	0.543	0.543
Erl	0	0.550	0	0.996	0.993	0.996	-0.003	0.543	-0.003
AVG.	0.470	**0.526**	0.457	**0.948**	0.946	0.947	0.418	**0.473**	0.404

Her üç algoritmaya ait kümeleme performansları Tablo 7.26'da verilmiştir.

Tablo 7.25. Algoritmaların maya protein lokalizasyon site sınıflandırma performansı

Sınıf	**Sn**			**Sp**			**MCC**		
	S	SW	DS	S	SW	DS	S	SW	DS
Cp	0.844	0.883	0.831	0.953	0.965	0.949	0.797	0.848	0.780
İm	0.685	0.922	0.909	0.963	0.976	0.972	0.649	0.898	0.882
imS	0.500	0	0	0.997	0.994	0.994	0.497	-0.005	-0.005
imL	0	0.500	0.500	0.994	0.997	0.997	-0.005	0.497	0.497
imU	0.685	0.542	0.714	0.963	0.946	0.966	0.649	0.489	0.681
Om	0.800	0.900	0.628	0.987	0.993	0.956	0.787	0.893	0.585
omL	0	0.600	0	0.994	0.953	0.994	-0.005	0.553	-0.005
Pp	0.788	0.600	0.884	0.961	0.953	0.978	0.749	0.553	0.863
AVG.	0.538	**0.618**	0.558	**0.976**	0.972	0.976	0.514	**0.591**	0.534

Tablo 7.26. Algoritmaların koli basili ve maya kümeleme performansları

Veri Seti	**Algoritmalar**	**Kümeleme Kalite Metrikleri**			
		QE	**TE**	**GBC**	**PBC**
Maya	SOM	**0.00792**	**0.11118**	**0.03173**	**0.00748**
	SWOM	0.00853	0.15161	**0.03520**	**0.00742**
	DSOM	0.00836	0.14690	0.04777	0.00781
Koli Basili	SOM	**0.02426**	**0.13988**	**0.07577**	**0.02362**
	SWOM	**0.02619**	**0.25595**	**0.07593**	0.02571
	DSOM	0.02625	0.27321	0.07907	**0.02524**

Tablo 7.26 en iyi birinci ve ikinci değerler cinsinden incelendiğinde beklendiği gibi klasik SOM kümeleme performanslarında 8 değerden 8'inde ya birinci ya da ikinci değeri elde ederken, SWOM tüm değerlerden 5'inde ya birinci ya da ikinci değeri alarak SOM'un kalitesine yaklaşmıştır. DSOM algoritması ise sadece bir değerde SOM algoritmasına yakınlaşmış ve en iyi ikinci değeri elde etmiştir.

Algoritmaların eğitim süresi ile ilgili karşılaştırmayı yapabilmek için, her üç algoritma eğitilirken geçen zaman Tablo 7.27'de verilmiştir.

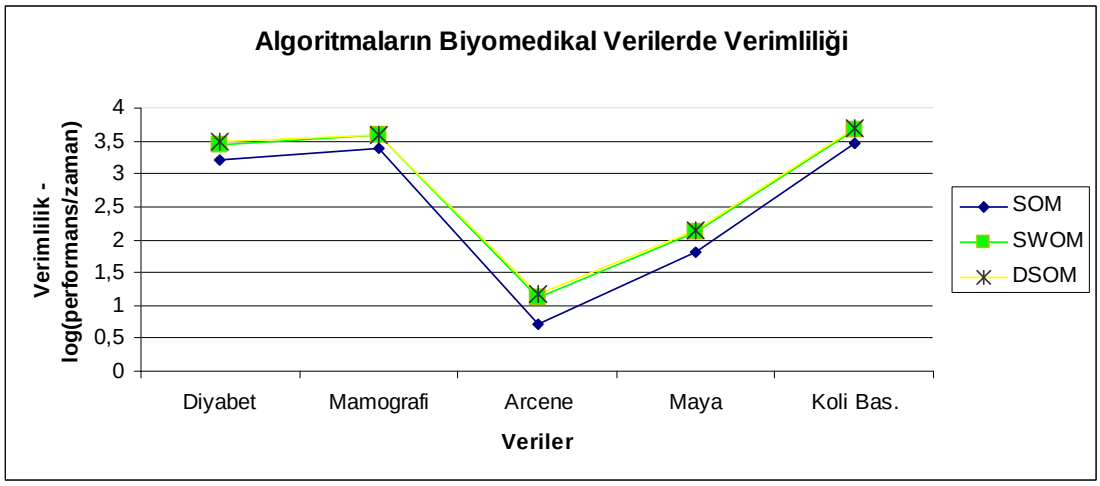
Tablo 7.27. Koli basili ve maya için algoritmaların eğitim süreleri

Epoç	**Koli Basili**			**Maya**		
	S	**SW**	**DS**	**S**	**SW**	**DS**
10	0.36	0.31	0.28	14.31	10.94	9.91
50	1.72	1.13	1.08	71.10	48.94	45.83
100	3.41	2.09	1.95	141.16	71.66	69.73

Tablo 7.27 eğitim süreleri için kıyaslandığında DSOM algoritmasının SWOM algoritmasına göre daha kısa bir eğitim süresince eğitimini tamamladığı, SOM algoritmasının ise beklendiği şekilde her iki algoritmadan çok daha uzun sürede eğitimini tamamladığı görülmektedir.

Algoritmaların eğitim zamanı ve sınıflandırma performanslarının beraber kıyaslanması optimizasyonun için daha sağlıklı bir bakış açısı sunacaktır. Bu nedenle her üç algoritmanın hem eğitim zamanı hem de sınıflandırma başarımlarını dikkate alan birleştirilmiş performanslarına ait iki şekil türetilmiştir. Şekil 7.6 tüm medikal veri setler için karşılaştırmayı temsil ederken, Şekil 7.7 genomik veri setleri için kıyaslamayı göstermektedir.

Biyomedikal verilerde her üç algoritmanın zaman-sınıflandırma başarımı performanslarının ele alındığı durumda SWOM ve DSOM optimizasyonlarının Şekil 7.6'da hemen hemen aynı performansa sahip olduğu gözlenir. Bu anlamda SWOM algoritmasının yeni bir SOM optimizasyon tekniği olarak biyomedikal verilerde daha önce önerilen DSOM optimizasyonu kadar başarılı olduğu gözlenir.

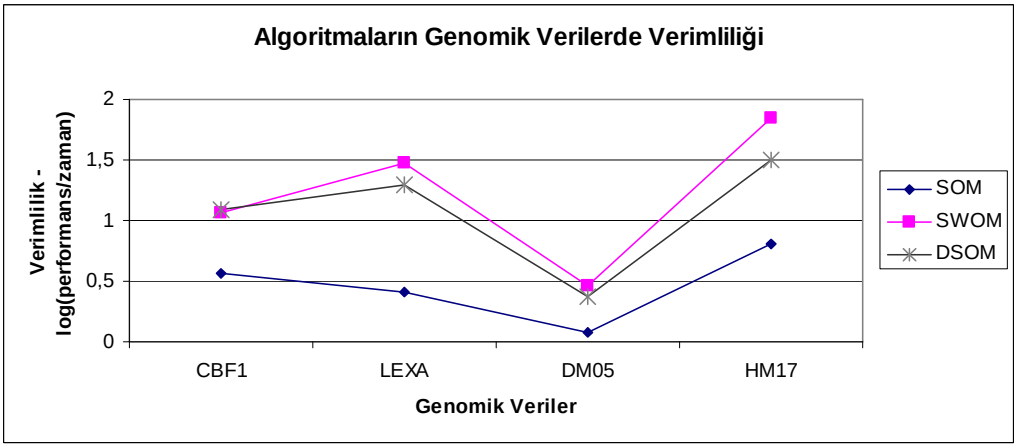


Şekil 7.6. Biyomedikal verilerde SOM, SWOM ve DSOM algoritmalarının toplam verimliliği

Genomik örüntülerde algoritma performanslarının ele alındığı Şekil 7.7'de, SWOM algoritmasının DSOM algoritmasına göre başarımının daha yüksek olduğu gözlenir.

SOM algoritmasının sınıflandırma başarımını azaltmadan görece kümeleme performansındaki beklenen düşüşe rağmen PSO ile optimize edilmesi bu bölümdeki tüm sonuçlar bir bütün olarak ele alındığında başarılı kabul edilmektedir. Başka bir ifade ile

veri setinin boyutu arttıkça SOM algoritmasının ihtiyaç duyacağı yüksek eğitim zamanları yerine daha kısa zamanda SOM'a yakın bir performans gösteren SWOM algoritması ön-izleme anlamında yeterince başarılı sonuçlar üretecektir.



Şekil 7.7. Genomik verilerde SOM, SWOM ve DSOM algoritmalarının toplam verimliliği

Bir sonraki bölümde bu tez çalışmasında çok sayıda akıllı sistem algoritması ve çok çeşitli veri kullanılarak yapılan deneylerde elde edilen sınıflandırma başarımına etki eden faktörler üzerine bir değerlendirme yapılarak elde edilen sonuçlar tartışılacak, analiz edilecektir.

8. SONUÇ VE DEĞERLENDİRME

Bu bölümde tez çalışması boyunca elde edilen sonuçlar ve bu sonuçların değerlendirilmesine yer verilecektir.

8.1. Sonuçların Değerlendirilmesi

Bu çalışmada biyomedikal verilerin uzman sistemler yardımıyla analizinde veya hastalık teşhisinde literatürde sıklıkla kullanılan akıllı sistem algoritmalarının başarımını etkileyen faktörler incelenmiştir. Bölüm 7'de izah edilen deneysel çalışmaların ışığında aşağıdaki somut sonuçlara ulaşılmıştır:

- Kütle spektrometrisi ve benzeri yöntemlerle elde edilen veriler hastalık teşhisinde kullanılmaktadırlar. Bu alanda çalışacak bir uzman sistemin başarımının kütle spektrometrisi verilerinin ön-işleme adımlarına bağlı şekilde değiştiği, artıp-azaldığı hesapsal şekilde gösterilmiş ve elde edilen sonuçlar European Journal of Mass Spectrometry dergisinde yayınlanmıştır [170].
- Biyomedikal verilerde öznitelik seçiminin verileri analiz amacıyla tasarlanacak akıllı sistemlerin performansını değiştirdiği hesapsal şekilde gösterilmiştir. Öznitelik seçimi yapılmadan sınıflandırılan verilere göre değişik öznitelik seçim teknikleri kullanılarak sınıflandırma başarımının arttırılabildiği hesapsal şekilde gösterilmiş olup dermatoloji verileri için yapılan çalışmaya ait sonuçlar Journal of Medical Systems dergisinde yayınlanmıştır [171].
- Bölüm 7'de analiz edilen biyomedikal veriler için tasarlanan akıllı sistem algoritmalarının aynı şartlarda aynı veri seti için farklı başarımlar ürettiği gösterilmiştir. Dermatoloji ve Parkinson hastalıklarının analizinde kullanılan 30 algoritmanın bu veri setlerinin sınıflandırılmasında farklı başarımlar gösterdiği hesapsal olarak ortaya konmuştur.
- Medikal verilerin analizi için kullanılan algoritmaların sınıflandırma başarımlarının her bir algoritmanın parametrelerinin değişiminden etkilendiği

bölüm 7'de açıklanan çalışmalarda hesapsal olarak gösterilmiştir. Örneğin dermatoloji veri setinin analizi için kullanılan GA tabanlı öznitelik seçim yönteminde mutasyon oran ve çaprazlama katsayısı değiştiğinde seçilen öznitelikler ve dolayısıyla algoritma sınıflandırma başarımları değişmiştir. Bu deneyde her iki parametrenin etkisini gözleyebilmek için mutasyon sabit tutulup çaprazlama katsayısı değiştirildiğinde seçilen özniteliklerin değiştiği gözlenmiştir.

- Topluluk öğrenmesi algoritmaları bölüm 7'de izah edildiği gibi genellikle sınıflandırma başarımlarını arttırırlar. RFEL algoritmasının Parkinson hastalığının analizinde kullanılan 30 kadar algoritmanın 26 tanesinin başarımını arttırması bu çıkarımı hesapsal şekilde doğrulamaktadır.
- Öğreticisiz bir öğrenme yöntemi olan ve kümeleme çalışmalarında çok kullanılan SOM algoritmasının veri boyutu büyüdükçe veri analiz süresinin uzaması nedeniyle algoritmanın kalite kaybı olmadan optimize edilmesi ihtiyacını doğurmuştur. Bu şekilde PSO algoritması ile optimize edilen SOM algoritması bölüm 7'de izah edilen çok sayıda medikal veri ile test edilmiş ve eğitim süresinin kısaltılmasına karşın algoritmanın kalitesinin düşmediği hesapsal olarak gösterilmiştir. SOM optimizasyonuna ait deneysel sonuçlar Expert Systems with Applications dergisinde yayınlanmıştır [172].

8.2. Öneriler

Bu çalışmada elde edilen ve bölüm 8.1'de izah edilen sonuçların ışığında bu alanda çalışmayı düşünen araştırmacılara yapılabilecek öneriler ve ileride ele alınabilecek konular aşağıda izah edilmiştir:

- Bir veri setine ait sınıflandırma çalışmaları için uzman sistem tasarımı sırasında veri setine ait şartlar aynı tutularak çok sayıda farklı akıllı sistem algoritması denemeli ve algoritmaların başarımı gözlenmelidir
- Hesapsal veri analizi çalışmalarında sadece sınıflandırma başarımı ACC ölçüm metriği olmamalı ve bu metrik KE, AUC ve RMSE gibi ek metriklerle desteklenmelidir. Bu şekilde elde edilen sınıflandırma başarımının rastgele bir

sonuç olup olmadığı görülecek ve elde edilen sonuçların güvenirliği sağlanmış olacaktır.

- Yüksek başarım elde edilen algoritmalar belirlendikten sonra bu algoritmaların her birinin parametrelerinin değişiminin performansa etkisi ayrıca incelenmeli ve her bir algoritma için optimum parametre değerleri elde edilmelidir.
- Sınıflandırma tabanlı veri analiz problemlerinde öznitelik seçiminin performansa etkisi mutlaka çalışılmalı ve bu şekilde algoritmaların başarımının nasıl değiştiği öznitelik seçimi olmadan elde edilen referans sonuçlarla karşılaştırılmalıdır.

Yüksek başarım elde edilen bir algoritma için topluluk öğrenmesi yöntemleri ile sınıflandırma başarımının artıp artmayacağı ayrıca denenmelidir.

Bu çalışmada elde edilen sonuçlar ileriye dönük olarak iki önemli çalışmanın yapılabilmesini mümkün kılmaktadır:

- Öznitelik seçiminin sınıflandırma başarımına etkisi deneysel olarak gösterilmiştir. Ancak her öznitelik seçim yönteminin başka bir veri seti için aynı şekilde etkin olacağı kesin değildir. Bu nedenle bir veri setine uygulanarak başarısı görülen öznitelik seçim yöntemleri ile veri setinin istatistiksel özellikleri arasında bağıntı incelemeğe değer görünmektedir. Böylece benzer istatistiksel özelliklere sahip tüm veri setleri için daha önce başarımı elde edilmiş öznitelik seçim yönteminin uygulanabileceği ve öznitelik seçim yönteminin seçilmesinin kolaylaşacağı düşünülmektedir. Bu çalışmaları bünyesinde toplayan otomatikleştirilmiş bir yazılımın geliştirilmesi nihai hedef olacaktır.
- Bu çalışmada biyomedikal verilere uygulanan öznitelik seçim yöntemleri ve topluluk öğrenmesi algoritmalarının başka mühendislik alanlarına uygulanarak geliştirilmesi, disiplinler arası çalışma alanlarının araştırılması çalışılabilecek diğer konulardan birisidir.

KAYNAKLAR

[1] **Shau, H., Chandler, G.**, 2003. Proteomic profiling of cancer biomarkers, Briefings in Functional Genomics and Proteomics, **2**, 147-158.

[2] **Boguski, M. S. and McIntosh, M.**, 2003. Biomedical informatics for proteomics, Nature, **422,** 233-237.

[3] **Liebler, D. C.**, 2002. Introduction to proteomics : tools for the new biology. Totowa, NJ, Humana Press.

[4] http://biochem218.stanford.edu/Projects.html/

[5] **Cannataro, M.**, 2008. Computational proteomics: management and analysis of proteomics data, Briefings in Bioinformatics, **9**, 97-101.

[6] **Diamandis, E. P.**, 2003. Proteomic patterns in biological fluids, Clinical Chemistry, **49**, 1272-1275.

[7] **Morris, S.**, 2005. Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum, Bioinformatics, **21**, 1764-1775.

[8] **Liu, Y.**, 2009. Feature extraction and dimensionality reduction for mass spectrometry data, Comput. Biol. Med., **39**, 818-823.

[9] **Barla, A.**, 2008. Machine learning methods for predictive proteomics, Briefings in Bioinformatics, **9**, 119-128.

[10] **Veltri, P.**, 2008, Algorithms and tools for analysis and management of mass spectrometry data, Briefings in Bioinformatics, **9**, 144-155.

[11] **Baek, S.**, 2009. Development of biomarker classifiers from high-dimensional data, Briefings in Bioinformatics, **10**, 537-546.

[12] http://home.ccr.cancer.gov/ncifdaproteomics/default.asp

[13] **Villmann, T.**, 2008. Classification of mass-spectrometric data in clinical proteomics using learning vector quantization methods, Briefings in Bioinformatics, **9**, 129-143.

[14] **Forner, F.**, 2007. Mass spectrometry data analysis in the proteomics era, Current Bioinformatics, **2**, 63-93.

[15] http://archive.ics.uci.edu/ml/

[16] **Hassanien, A. and Mariofanna, G.**, 2008. Computational Intelligence in Solving Bioinformatics Problems: Reviews, Perspectives, and Challenges, Computational Intelligence in Biomedicine and Bioinformatics, 3-47.

[17] **Kennedy, J. and Eberhart, R. C.**, 1995. Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, 1942-1948.

[18] **Cvek, U., Trutschl, M., Cannon, J. C., Scott, R. S., and Rhoads, R. E.**, 2007. 2D and 3D Neural-Network Based Visualization of High-Dimensional Biomedical Data, 11th international Conference information Visualization IEEE Computer Society, 545-550.

[19] **Gullo, F., Ponti, G., Tagarelli, A., Tradigo, G., and Veltri, P.,** 2009. MaSDA: A system for analyzing mass spectrometry data, Comput. Methods Prog. Biomed. **95**, 2.

[20] **He, S., Chen, H., Li, X., and Yao, X.**, 2009. Profiling of Mass Spectrometry Data for Ovarian Cancer Detection Using Negative Correlation Learning, 19th international Conference on Artificial Neural Networks, Lecture Notes In Computer Science, 185-194.

[21] **Salmela, E.**, 2008. Genome-Wide Analysis of Single Nucleotide Polymorphisms Uncovers Population Structure in Northern Europe, PLoS ONE **3**, 3519.
[22] **May, E. E., and Dolan, P.**, 2006. Syndrome-Based Discrimination of Single Nucleotide Polymorphism, Engineering in Medicine and Biology Society, 28th Annual International Conference of the IEEE.
[23] **Dermitzakis, E. T.**, 2008. From gene expression to disease risk, Nat Genet, **40**,492-493.
[24] **Gershon, D.**, 2002. Microarray technology: An array of opportunities, Nature, **416**,885-891.
[25] **Simon, R.**, 2003. Using DNA microarrays for diagnostic and prognostic prediction, Expert Review of Molecular Diagnostics, **3**, 587-595.
[26] **Gould, M. K., and Maclean, C.**, 2001. Accuracy of Positron Emission Tomography for Diagnosis of Pulmonary Nodules and Mass Lesions: A Meta-analysis, JAMA, **285**, 914-924.
[27] **Sajda, P.**, 2006. Machine learning for detection and diagnosis of disease, Annual review of biomedical engineering, **8**, 537-565.
[28] **Tanwani, A. K.**, 2009. Guidelines to Select Machine Learning Scheme for Classification of Biomedical Datasets, Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer-Verlag**,** 128-139.
[29] **M. Elter, R.**, 2007. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process, Medical Physics, **34**, 4164-4172
[30] **Donoho, D. L.**, 2000. High-dimensional data analysis: the curses and blessings of dimensionality, American Mathematical Society Conf. Math Challenges of the 21st Century.
[31] **Loy, C.**, 2006. Dimensionality Reduction of Protein Mass Spectrometry Data Using Random Projection, Neural Information Processing, **4233**, 776-785.
[32] http://home.ccr.cancer.gov/ncifdaproteomics/OvarianCD_PostQAQC
[33] **Scott, M. S.**, 2005. Refining Protein Subcellular Localization, PLoS Comput Biology, **1**, 66.
[34] **Tompa, M.**, 2005. Assessing computational tools for the discovery of transcription factor binding sites, Nature Biotech, **23**, 137-144.
[35] **D'Haeseleer, P.**, 2006. What are DNA sequence motifs, Nature Biotech, **24**, 423-425.
[36] **Zhang, Z.**, 2004. Three biomarkers identified from serum proteomic analysis for the detection of early stage ovarian cancer, Cancer research, **64**, 5882-5890.
[37] **Coombes, K. R.**, 2005. Analysis of Mass Spectrometry Profiles of the Serum Proteome, Clinical Chemistry, **51**, 1-2.
[38] **Liu, Q. and Xiao, Y.**, 2007. MALDI MS: A Practical Guide to Instrumentation, Methods and Applications, ChemBioChem, **8**, 2315-2316.
[39] http://www.psrc.usm.edu/mauritz/maldi.html
[40] http://elchem.kaist.ac.kr/vt/chem-ed/ms/tof.htm
[41] **Guzzi, P. H., and Mazza,T.**, 2005. Preprocessing of Mass Spectrometry Proteomics Data on the Grid, Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, IEEE Computer Society, 549-554.
[42] **Coombes, K., and Baggerly, K.**, 2007. Pre-Processing Mass Spectrometry Data, Fundamentals of Data Mining in Genomics and Proteomics, Kluwer, 79-99.
[43] **Wagner, M., Naik, D. and Pothen, A.**, 2003. Protocols for disease classification from mass spectrometry data, Proteomics, **3**, 1692–1698.

[44] **Armananzas, R., and Garcia, M.**, 2008. Mass spectrometry data analysis: it's all in the preprocessing, Proceedings of the Benelux Bioinformatics Conference, 92.
[45] **Katajamaa, M. and Oresic, M.**, 2007. Data processing for mass spectrometry-based metabolomics, Journal of Chromatography, **1158**, 318-328.
[46] **Williams, B., and Cornett, S.**, 2005. An algorithm for baseline correction of MALDI mass spectra, Proceedings of the 43rd annual Southeast regional conference, ACM, 137-142.
[47] **Bougioukos, P.**, 2007. Prostate Cancer Biomarker Selection through a Novel Combination of Sequential Global Thresholding, 19th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society**,** 85-90.
[48] **Cannataro, M., and Veltri, P.**, 2005. Preprocessing, Management, and Analysis of Mass Spectrometry Proteomics Data, The NETTAB 2005 workshop
[49] **Meuleman, W., and Engwegen, J.**, 2008. Comparison of normalisation methods for surface-enhanced laser desorption and ionisation (SELDI) time-of-flight (TOF) mass spectrometry data, BMC Bioinformatics, **9**(1), 88.
[50] http://www.stat.berkeley.edu/~terry/Group/publications/Final2Gensips2004Sauve
[51] **Norris, J. L., Cornett, D. S.**, 2007. Processing MALDI mass spectra to improve mass spectral direct tissue analysis. International Journal of Mass Spectrometry, **260**, 212-221.
[52] **Ge, G. and Wong, G. W.**, 2008. Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles, BMC Bioinformatics, **9**, 275.
[53] **Marcuson, R., and Burbeck, S.**, 1982. Normalization and reproducibility of mass profiles in the detection of individual differences from urine. Clinical Chemistry, **28**, 1346-1348.
[54] **Simon, R.**, 2003. Supervised analysis when the number of candidate features (p) greatly exceeds the number of cases (n). SIGKDD Explor. Newsl., **5**, 31-36.
[55] **Han, J.**, 2002. How Can Data Mining Help Bio-Data Analysis?, Proceedings of KDD Workshop on Data Mining in Bioinformatics.
[56] **Hilario, M. and Kalousis, A.**, 2008. Approaches to dimensionality reduction in proteomic biomarker studies, Briefings in Bioinformatics, **9**, 102-118.
[57] **Yu, J. S., Ongarello, S.**, 2005. Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data, Bioinformatics **21**, 2200-2209.
[58] **Donoho, D. L.**, 2000. High-dimensional data analysis: the curses and blessings of dimensionality, American Mathematical Society Conf. Math Challenges of the 21st Century.
[59] **Lilien, R., Farid, H.**, 2003. Probabilistic disease classification of expression-dependent proteomic data from mass spectrometry of human serum, JComputBiol, **10**, 925–946.
[60] **Qu, Y., Adam, B.**, 2003. Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensional data, Biometrics, **59**, 143–51.
[61] **Purohit, P., Rocke, D.**, 2003. Discriminant models for high-throughput proteomics mass spectrometer data, Proteomics, **3**, 699–703.
[62] **Zeng, X., Li, Z.**, 2008. Dimension reduction with redundant gene elimination for tumor classification, BMC Bioinformatics, **9**, 1-8.
[63] **Duda, R., and Hart, P.**, 2000. PatternClassification,Wiley, 117-124.

[64] **Cozzolino, D., Smyth, H.**, 2005. Usefulness of chemometrics and mass spectrometry-based electronic nose to classify Australian white wines by their varietal origin, Talanta, **68**, 382-387.
[65] **Verikas, A. and Bacauskiene, M.** ,2002. Feature selection with neural Networks, Pattern Recogn. Lett., **23**, 1323-1335.
[66] **Saeys, Y.**, 2007. A review of feature selection techniques in bioinformatics, Bioinformatics, **23**, 2507-2517.
[67] **Yoshida, H., Leardi, R.** , 2001. Feature selection by genetic algorithms for mass spectral classifiers, Analytica Chimica Acta, **446**, 483-492.
[68] **Tan, N., Fisher, W.**, 2009. Application of multiple statistical tests to enhance mass spectrometry-based biomarker discovery, BMC Bioinformatics, **10**, 144-150.
[69] **Reynès, C., Sabatier, R.**, 2008. A new genetic algorithm in proteomics: Feature selection for SELDI-TOF data, Computational Statistics & Data Analysis **52**, 4380-4394.
[70] **Guyon, I. and Elisseeff, A.** ,2003. An introduction to variable and feature selection, J. Mach. Learn. Res., **3**, 1157-1182.
[71] **Hall, M., and Smith, L.**, 1998. Feature subset selection: A correlation based filter approach. Progress in Connectionist-Based Information Systems, 855-858.
[72] **Peng, Y.**, 2010. A novel feature selection approach for biomedical data classification, Journal of Biomedical Informatics, 15-23.
[73] **Guyon, I.**, 2006. Feature Extraction Foundations and Applications, Studies in Fuzziness and Soft Computing, 119-135.
[74] **Jeffries, N.**, 2004. Performance of a genetic algorithm for mass spectrometry proteomics, BMC Bioinformatics, **5**, 180-183.
[75] **Zhang, C. and Zhang, J.**, 2010. A variant of Rotation Forest for constructing ensemble classifiers, Pattern Analysis Applications, 59–77.
[76] **Baldi, P. and Brunak, S.**, 2001. Bioinformatics: The Machine Learning Approach, second ed., MIT Press, Cambridge, 11-52.
[77] **Harish, B., David, C.**, 2006. Machine learning in bioinformatics: A brief survey and recommendations for practitioners. Computers in biology and medicine, **36**, 1104-1125.
[78] **Larrañaga, P., Calvo, B.**, 2006. Machine learning in bioinformatics." Briefings in Bioinformatics, **7**, 86-112.
[79] **Babita, P., Mishra, R.**, 2009. Knowledge and intelligent computing system in medicine. Comput. Biol. Med. **39**, 215-230.
[80] http://www.cs.waikato.ac.nz/ml/weka/
[81] **Castanon, L.**, 2008. Fault Diagnosis of Industrial Systems with Bayesian Networks and Neural Networks, Advances in Artificial Intelligence, Springer-Verlag, 998-1008.
[82] **Curiac, D., Vasile, G.**, 2009. Bayesian network model for diagnosis of psychiatric diseases, Information Technology Interfaces, Proceedings of 2009 International Conference on.
[83] **Rebane, G. and Pearl, J.**, 1987. The Recovery of Causal Poly-trees from Statistical Data, Proceedings 3rd Workshop on Uncertainty in AI, 222-228.
[84] **Liu, Y., Kamaya, A.**, 2008. A Bayesian classifier for differentiating benign versus malignant thyroid nodules using sonographic features, AMIA Annu Symp Proc., **6,** 419-423.
[85] **Larsen, K.**, 2005. Generalized Naive Bayes Classifiers. SIGKDD Explor. Newsl., **7**, 76-81.

[86] **Friedman, N.**, 1997. Bayesian Network Classifiers learning with probabilistic representations, Machine Learning, 131-163.
[87] **Mitchell, T.**, 1999. Machine learning and data mining. Communications ACM, **42**, 30-36.
[88] **Cover, T., Hart, P.**, 1967. Nearest neighbor pattern classification, IEEE Transactions on Information Theory, **13**, 21–27.
[89] **Aha, D., Kibler, D.**, 1991. Instance-Based Learning Algorithms. Mach. Learn., **6**, 37-66.
[90] **Li, J., Dong, G.**, 2000. Instance-Based Classification by Emerging Patterns, Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Springer-Verlag**,** 191-200.
[91] **Thair, N.**, 2009. Survey of Classification Techniques, Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS 2009.
[92] **Marc, A., David, A.**, 1991. Analysis of instance-based learning algorithms, AAAI-91 Proceedings.
[93] **Beffara, E. and Danos, V.**, 2003. Disjunctive normal forms and local exceptions, SIGPLAN Notes, **38**, 203-211.
[94] **Fürnkranz, J.**, 1999. Separate-and-Conquer Rule Learning, Artif. Intell. Rev. **13**, 3-54.
[95] **Smyth, P. and Goodman, R.**, 1990. Rule induction using information theory, Knowledge Discovery in Databases, MIT Pres, 215-238.
[96] **Kotsiantis, S.** , 2007. Supervised Machine Learning: A Review of Classification Techniques, Informatica, **3**, 249-268.
[97] http://weka.sourceforge.net/doc/
[98] **Ting, S., and Wang, W.**, 2010. RACER: Rule-Associated CasE-based Reasoning for supporting General Practitioners in prescription making, Expert Syst. Appl., **37**, 8079-8089.
[99] **Myatt, G.**, 2006. Making Sense of Data, John Wiley & Sons, 150-172.

[100] **Safavian, S. and Landgrebe, D.**, 1991. A survey of decision tree classifier methodology, Systems, Man and Cybernetics, IEEE Transactions **21**, 660-674.
[101] **Ravi, K. and Ming, D.**, 2007. Decision trees for Classification, World Scientific, 1-15.
[102] **Hsu, C., Chang, C., Lin, C.**, 2003. A practical guide to support vector classification, Tech. Rep., Taipei.
[103] **Boser, B., Guyon, I., and Vapnik, V.**, 1992. A training algorithm for optimal margin classifiers, 5th Annual ACM Workshop, 144-152.
[104] **Scholkopf, B.**, 2003. An introduction to support vector machines, Recent Advances and Trends in Nonparametric Statistics, 3-17.
[105] **Raudys, S.**, 1996. Linear classifiers in perceptron design, Pattern Recognition, 1996., Proceedings of the 13th International Conference.
[106] **Dougherty, M.**, 1995. A review of neural networks applied to transport, Transportation Research, Emerging Technologies, **3**, 247-260.
[107] **Yazıcı, A.**, 2007. Yapay Sinir Ağlarına Genel Bakış, Türkiye Klinikleri Dergisi, 66-73.
[108] **Bishop, C.** , 2009. Neural Networks and Their Applications, Review of Scientific Instruments, 4772 – 4774.

[109] **Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.**, 2007. A Review of Classification Algorithms for EEG-Based Brain-Computer İnterfaces, Journal of Neural Engineering.

[110] **Cybenko, G.**, 1989. Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems, MCSS, **2**, 303–314.

[111] **Haykin, S.**, 1998. Neural Networks: A Comprehensive Foundation, Prentice Hall, 178-179.

[112] **Zhou, X., K., Liu, Y.**, 2004. Cancer classification and prediction using logistic regression with Bayesian gene selection, Journal of Biomedical Informatics, **37**, 249-259.

[113] **Landwehr, N.**, 2009. Logistic Model Trees, University of Waikato web.

[114] **Roos, T., Grünwald, P., Myllymäki, P.**, 2005. On discriminative bayesian network classifiers and logistic regression, Machine Learning, **59**, 267-296.

[115] **Rui, X. and Wunsch, D.**, 2005. Survey of clustering algorithms. Neural Networks, IEEE Transactions, **16**, 645-678.

[116] **Fernandez, E. and Balzarini, M.**, 2007. Improving cluster visualization in self-organizing maps: Application in gene expression data analysis, Comput. Biol. Med. **37**, 1677-1689.

[117] **Berkhin, P.**, 2002. Survey Of Clustering Data Mining Techniques. Tech. rep., Accrue Software, San Jose, CA.

[118] **Kumar, V., and Tan, P.**, 2006. Introduction to Data Mining, Addison Wesley, Pearson International Edition, 487-556.

[119] **Grabmeier, J., Rudolph, A.**, 2002. Techniques of Cluster Algorithms in Data Mining, Data Mining and Knowledge Discovery, **6**, 303-360.

[120] **Kohonen, T.**, 1993. Things You Haven't Heard about the Self-organizing Map, Institute of Electrical and Electronics Engineers, IEEE, 1147-1156.

[121] **Kangas, A. and Kohonen, T.**, 1990. Variants of Self-organizing Maps, Transactions on Neural Networks, **1**, 93-99.

[122] **Hai, K. L., and Duc, M.**, 2004. Some remarks on the Self-organizing Feature Maps. Proceedings of the 2nd International Seminar on the environmental science and technology issues related to the urban and coastal zones development. Ha Long, Vietnam.

[123] **Beni, G., and Wang, U.**, 1989. Swarm intelligence in cellular robotic systems, NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy.

[124] **Kennedy, J. and Eberhart, R.**, 1995. Particle Swarm Optimization, Proceedings of the IEEE International Conference on Neural Networks, 1942-1948.

[125] **Shi, Y. and Eberhart, R.**, 1998. A modified particle swarm optimizer, IEEE International Conference on Evolutionary Computation, 69- 73.

[126] **Xiaohui, C.**, 2005. Document clustering using Particle Swarm Optimization, IEEE, 1-13.

[127] **Das, S.**, 2008. Swarm Intelligence Algorithms in Bioinformatics, Studies in Computational Intelligence, 113–147.

[128] **Omran, M.**, 2005.A Color Image Quantization Algorithm Based on Particle Swarm Optimization, Informatica, **29**, 261–269.

[129] **Sharma, A.**, 2006.Determining Cluster Boundaries using Particle Swarm Optimization, Proceedings of World Academy of Science Engineering and Technology, 250-254.

[130] **Kennedy, J.**, 1997. The particle swarm: social adaptation in information processing systems, IEEE, 303-308.

[131] **Opitz, D. and Maclin, R.**, 1999. Popular ensemble methods: An empirical study, Journal of Artificial Intelligence Research, 169–198.
[132] **Kuncheva L. and Whitaker, C.**, 2003. Measures of diversity in classifier ensembles, Machine Learning, 181-207.
[133] **Ho, T.**, 1995. Random Decision Forests, Proceedings of the Third International Conference on Document Analysis and Recognition, 278-282.
[134] **Polikar, R.**, 2006. Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, 21-45.
[135] **Polikar, R.**, 2009. Ensemble Learning, Scholarpedia, **4**, 2776.
[136] **Melville, P. and Mooney, R.**, 2003. Constructing Diverse Classifier Ensembles using Artificial Training Examples, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 505-510.
[137] **Zhang, C. and Zhang, J.**, 2010. A variant of Rotation Forest for constructing ensemble classifiers, Pattern Analysis Applications, 59–77.
[138] **Rodriguez, J. and Kuncheva, L.**, 2006. Rotation Forest: A New Classifier Ensemble Method, IEEE Transactions On Pattern Analysis And Machine Intelligence, 1619-1630.
[139] **Beasley, D., Bull, D.**, 1993. An Overview of Genetic Algorithms: Part 1, Fundamentals, University Computing, **15**, 58-69.
[140] **Holland, J.**, 1975, Adaption in Natural and Artificial Systems, University of Michigan Pres, 123-135.
[141] **Goldberg, D.E.**, 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, 224-273.
[142] **Man, K. and Tang, K.**, 1996. Genetic Algorithms: Concepts and Applications, IEEE Transactions on Industrial Electronics, **43**, 519-533.
[143] **Poli, R., and Langdon, W.**, 2008. A Field Guide to Genetic Programming.
[144] **Yifeng, L. and Yihui, L.**, 2008. Genetic algorithm based feature selection for mass spectrometry data, BioInformatics and BioEngineering, 118-132.
[145] **Garcia, S.**, 2009. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing, **13**, 959-977.
[146] **Sokolova, M. and Lapalme, G.**, 2009. A systematic analysis of performance measures for classification tasks, Information Processing and Management, **45**, 427-437.
[147] **Ferri, C., and Hern, J.**, 2009. An experimental comparison of performance measures for classification, Pattern Recogn. Lett., **30**, 27-38.
[148] **Dem, J.**, 2006. Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Mach. Learn. Res., **7**, 1-30.
[149] **Akobeng, A.**, 2007. Understanding diagnostic tests : sensitivity, specificity and predictive values, Acta Pediatrica, **96**, 338-341.
[150] **Beygelzimer, A. and Langford, J.**, 2008. Machine Learning Techniques, Reductions Between Prediction Quality Metrics, 3-28.
[151] **Forbes, A.**, 1995. Classification-algorithm evaluation: five performance measures based on confusion matrices, Journal of Clinical Monitoring, **11**, 189-206.
.[152] **Çamlıca, H.**, 2008. Tanı Testlerinde Sınır Değerlerinin Belirlenmesi, Türk Onkoloji Dergisi, 26-33
[153] **Tsong, Y.**, 2004. On the statistical properties of the F-measure, Fourth International Conference on Quality Software, 113-125.

[154] **Hripcsak, G.**, 2005. Agreement the F-measure and reliability in information retrieval, Journal of the American Medical Informatics Association, **12**, 296-298.
[155] **Baeza, R.**, 1999. Modern Information Retrieval, Addison Wesley, 342-347.
[156] **Efron, B.**, 1979. Bootstrap methods: another look at the jackknife, The Annals of Statistics, 1-26.
[157] **Carletta, J.**, 1996. Assessing agreement on classification tasks: the kappa statistic, Comput. Linguist., **22**, 249-254.
[158] **Ben-David, A.**, 2008. Comparison of classification accuracy using Cohen's Weighted Kappa, Expert Systems with Applications, **34**, 825-832.
[159] **Cortes, C. and Mohri, M.**, 2005. Confidence Intervals for the Area under the ROC Curve, Advances in Neural Information Processing Systems, NIPS 2004, **17**.
[160] **Calders, T. and Jaroszewicz, S.**, 2007. Efficient AUC Optimization for Classification, Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases, Springer-Verlag**,** 42-53.
[161] **Ma, S. and Huang, J.**, 2005. Regularized ROC method for disease classification and biomarker selection with microarray data, Bioinformatics, **21**, 4356-4362.
[162] **Delen, D., Walker, G.**, 2005. Predicting breast cancer survivability: a comparison of three data mining methods, Artificial Intelligence in Medicine, **34**, 113-127.
[163] **Amir, A.**, 2007. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set, Pattern Recognition Letters, 110-118.
[164] **Gray, R.**, 1984.Vector Quantization, IEEE ASSP Magazine, 4-29.
[165] **Pölzlbauer, G.**, 2004. Survey and Comparison of Quality Measures for Self-Organizing Maps, Proceedings of the Fifth Workshop on Data Analysis, Elfa Academic Press, 67-82.
[166] **Burset, M., Guigó, R.**, 1996. Evaluation of gene structure prediction programs, Genomics, **34**, 353-67.
[167] **Yang, Z. and Thomson, R.**, 2005. RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins, Bioinformatics Advance Access, 45-56.
[168] **Das, R.**, 2010. A comparison of multiple classification methods for diagnosis of Parkinson disease, Expert Systems with Applications, 37, 1568-1572.
[169] **Little, A., McSharry, E.,** 2009, Suitability of Dysphonia Measurements for Telemonitoring of Parkinson Disease, Biomedical Engineering, IEEE Transactions on, 1015-1022.
[170] **Özçift, A., Gülten, A.,** 2008. Assessing effects of pre-processing mass spectrometry data on classification performance, European Journal of Mass Spectrometry, 267-73.
[171] **Özçift, A., Gülten, A.,** 2010. A Robust Multi-Class Feature Selection Strategy Based on Rotation Forest Ensemble Algorithm for Diagnosis of Erythemato-Squamous Diseases, Journal of Medical Systems.
[172] **Özçift, A., Kaya, M., Gülten, A.,** 2009. Swarm optimized organizing map (SWOM): A swarm intelligence based optimization of self-organizing map, Expert Systems with Applications, 10640-10648.