T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

AYRIKLAŞTIRMA VE OPTİMİZASYON YAKLAŞIMLARI İLE SINIFLANDIRMA ALGORİTMALARININ PERFORMANSININ İYİLEŞTİRİLMESİ

Mohammed Hussein Ibrahim IBRAHIM

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Şubat-2019
KONYA

TEZ KABUL VE ONAYI

Mohammed Hussein Ibrahim IBRAHIM tarafından hazırlanan "Ayrıklaştırma ve optimizasyon yaklaşımları ile sınıflandırma algoritmalarının performansının iyileştirilmesi" adlı tez çalışması 12/02/2019 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof. Dr. Ahmet ARSLAN	.....
Danışman Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU	.....
Üye Doç. Dr. Halife KODAZ	.....
Üye Doç. Dr. Barış KOÇER	.....
Üye Dr. Öğr. Üyesi Ersin KAYA	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mohammed Hussein Ibrahim IBRAHIM

Tarih: 12.02.2019

ÖZET

DOKTORA TEZİ

AYRIKLAŞTIRMA VE OPTİMİZASYON YAKLAŞIMLARI İLE SINIFLANDIRMA ALGORİTMALARININ PERFORMANSININ İYİLEŞTİRİLMESİ

Mohammed Hussein Ibrahim IBRAHIM

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU

2019, 100 Sayfa

Jüri
Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU
Prof. Dr. Ahmet ARSLAN
Doç. Dr. Halife KODAZ
Doç. Dr. Barış KOÇER
Dr. Öğr. Üyesi Ersin KAYA

Sınıflandırma algoritmaları veri madenciliği ve makine öğrenmesi problemlerinin çözümünde en çok kullanılan algoritmalardır. Sınıflandırma algoritmaları, eğitim veri kümelerini istatistiksel ve matematiksel denklemler ile analiz ederek bir sınıflandırma modeli oluştururlar. Oluşturulan sınıflandırma modelinin başarısı test veri kümesi ile değerlendirilir ve bu model sınıfı belli olmayan örneklerin sınıf etiketlerinin tahmininde kullanılır. Günümüzde, sınıflandırma algoritmaları medikal, finans, sahtekârlık tespiti, hava durumu tahmini, bankacılık ve sosyal ağlar başta olmak üzere birçok alana başarılı şekilde uygulanmaktadır. Sınıflandırma algoritmaları, sınıflandırma modellerini oluşturma yöntemlerine göre kural tabanlı, olasılık tabanlı ve ağırlık tabanlı olmak üzere üç kategoriye ayrılabilir. Kural tabanlı ve olasılık tabanlı sınıflandırma algoritmaları genellikle kategorik ve ayrık veri kümeleri üzerinde daha başarılı performans sergilerken, ağırlık tabanlı sınıflandırma algoritmaları genellikle sürekli veri kümeleri üzerinde daha başarılı olurlar.

Bu tez çalışmasında genel olarak sınıflandırma algoritmalarının performansının iyileştirilmesi üzerinde durulmuştur. İlk olarak, kural ve olasılık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek üzere veri madenciliği önişleme tekniği olan ayrıklaştırma işlemi için yeni bir yöntem önerilmiştir. EF-Unique olarak adlandırılan önerilen yeni ayrıklaştırma yöntemi eşit aralıklı, eşit frekanslı ve entropi tabanlı ID3 ayrıklaştırma yöntemleri ile karşılaştırılmıştır. Önerilen yöntemin birçok deneyde diğer yöntemlerden daha başarılı sonuçlar elde ettiği görülmüştür. Ayrıca, EF-Unique yönteminin literatürde sıklıkla kullanılan naive bayes, karar ağaçları, destek vektör makinesi ve k en yakın komşu makine öğrenmesi sınıflandırma algoritmalarının performansını artırdığı gözlemlenmiştir. İkinci olarak, ağırlık tabanlı sınıflandırma algoritması olan yapay sinir ağlarının eğitim işlemi parçacık sürü optimizasyon algoritmasının geliştirilmiş bir versiyonu ile gerçekleştirilmiştir. Önerilen çoklu ortalama (multi mean) parçacık sürü optimizasyon (MM-PSO) algoritması yapay sinir ağının sınıflandırma başarısını artırmıştır. Deneysel çalışmalarda, önerilen MM-PSO algoritmasının performansını değerlendirmek için literatürde sıklıkla kullanılan UCI veri kümeleri kullanılmış ve elde edilen sonuçlar havai fişek, kril, genetik ve harmoni arama optimizasyon algoritmalarının sonuçları ile kıyaslanmıştır. Deney sonuçları değerlendirildiğinde, önerilen MM-PSO algoritması birçok deneyde havai fişek, kril, genetik ve harmoni arama optimizasyon algoritmalarından daha iyi performans sergilemiştir.

Tez kapsamında geliştirilen EF-Unique ayrıklaştırma yöntemi ve çoklu ortalama parçacık sürü optimizasyon algoritması literatüre bir yenilik getirmiştir. Geliştirilen her iki yaklaşım veri madenciliği ve makine öğrenmesi ile ilgili farklı alanlarda yapılacak farklı çalışmalarda kullanılabilecektir.

Anahtar Kelimeler: Ayrıklaştırma, Makine Öğrenmesi, Meta-Sezgisel Algoritmalar, Optimizasyon, Sınıflandırma Algoritmaları, Veri Madenciliği

ABSTRACT

Ph.D THESIS

IMPROVING THE PERFORMANCE OF CLASSIFICATION ALGORITHMS WITH DISCRETIZATION AND OPTIMIZATION APPROACHES

Mohammed Hussein Ibrahim IBRAHIM

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER ENGINEERING

Advisor: Asst. Prof. Dr. Mehmet HACIBEYOĞLU

2019, 100 Pages

Jury
Asst. Prof. Dr. Mehmet HACIBEYOĞLU
Prof. Dr. Ahmet ARSLAN
Assoc. Prof. Dr. Halife KODAZ
Assoc. Prof. Dr. Barış KOÇER
Asst. Prof. Dr. Ersin KAYA

Classification algorithms are one of the most commonly used algorithms for solving data mining and machine learning problems. The classification algorithms analyze the training data sets with statistical and mathematical equations to generate a classification model. Performance of the generated classification model is evaluated by test dataset and then this model is used to estimate class label of an unlabeled sample. Nowadays, classification algorithms are commonly used in numerous areas especially medical, finance, fraud detection, weather forecasts, banking, and social networks. Classification algorithms can be divided into three categories considering the method of generating classification model: rule-based, statistical based, and weighted based. While rule and statistical based algorithms are shown successful performance with categorical and discrete data sets, weighted based algorithms figured out successful performance with continuous data sets.

This thesis study is mainly discourse to improving the performance of classification algorithms. Firstly, a new method is proposed for the discretization process, which is a well-known data mining pre-processing technique, to improve the performance of rule and statistical based classification algorithms. The proposed method called as EF-Unique was compared with equal width, equal frequency, and entropy-based ID3 discretization methods. The proposed method has performed better results in many experiments than other methods. Besides, EF-Unique method was observed that the proposed method improved the performance of the frequently used machine learning classification algorithms such as naïve bayes, decision tree, support vector machine, and k-nearest neighbors. Secondly, the training process of an artificial neural network, which is a weighted based classification algorithm, was performed by a novel modified version of the particle swarm optimization algorithm. The proposed multi mean particle swarm optimization (MM-PSO) is increased classification accuracy of an artificial neural network. In experimental studies, frequently used UCI datasets were used to evaluate the performance of the proposed MM-PSO algorithm and the results obtained were compared with fireworks, krill, genetic, and harmony search optimization algorithms results. When the experimental results are evaluated, the proposed MM-PSO algorithm has performed better results in many experiments than the fireworks, krill, genetic, and harmony search optimization algorithms.

The most obvious finding to emerge from this study is that EF-Unique discretization method and multi mean particle swarm optimization algorithm provide a novel approach to literature. Additionally, the proposed method and algorithm can be used in various disciplines and future studies.

Keywords: Classification Algorithms, Data Mining, Discretization, Machine Learning, Meta-Heuristic Algorithms, Optimization

ÖNSÖZ

Bu tez çalışmamda bana yol gösteren ve hiçbir zaman yardımlarını benden esirgemeyen tez danışmanım Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU'na, değerli hocalarım Prof. Dr. Ahmet ARSLAN'a ve Prof. Dr. Şirzat KAHRAMANLI'ya, Necmettin Erbakan Üniversitesi Bilgisayar Mühendisliği ve Konya Teknik Üniversitesi Bilgisayar Mühendisliği Bölümleri'nin tüm öğretim elemanlarına teşekkür ederim.

Maddi ve manevi yönden beni her zaman destekleyen, üzerimde büyük hakları olan aileme ve eşime teşekkürü bir borç bilirim.

Mohammed Hussein Ibrahim IBRAHIM
KONYA-2019

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Simgeler

η	: Öğrenme faktörü
α	: Öğrenme kat sayısı
σ	: Standart Sapma

Kısaltmalar

AQ	: Algorithm Quasi-optima
RIPPER	: Repeated Incremental Pruning to Produce Error Reduction
ID3	: Iterative Dichotomiser 3
CART	: Classification and Regression Trees
K-NN	: K-Yakın Komşu (K-Nearest Neighbors)
KA	: Karar Ağaçları
NB	: Naïve Bayes
DVM	: Destek Vektör Makinesi (Support Vectors Machine)
YSA	: Yapay Sinir Ağları (Artificial Neural Network)
EG	: Eşit Genişlik
EF	: Eşit Frekans
ET_ID3	: Entropi Tabanlı ID3
MDLP	: Minimum Description Length Principle
Chi2	: Chi-Square
FFD	: Fixed Frequency Discretization
ITFP	: Information-Theoretic Fuzzy Partitioning
CAIM	: Class-Attribute Interdependence Maximization
CADD	: Class-Attribute Dependent Discretizer
MODL	: Minimum Optimal Description Length
MVD	: Multivariate Discretization
NCAIC	: Novel Class Attribute Interdependency Discretization Algorithm
MDLP	: Minimum Description Length Principle
GA	: Genetik Algoritması
PSO	: Parçacık Sürü Optimizasyon
W	: Eylemsizlik ağırlık (inertia weight)
MM-PSO	: Multi Mean - Parçacık Sürü Optimizasyon
DP	: Doğru Pozitif
DN	: Doğru Negatif
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
SB	: Sınıflandırma Başarısı

1. GİRİŞ

Veri madenciliği; makine öğrenmesi, istatistik ve veri tabanı birlikteliğindeki yöntemleri kullanarak büyük veri kümelerinden anlamlı bilgi çıkarma işlemidir. Başka bir deyişle, büyük ölçekli veri kümelerinden bilgi edinme/çıkarma ve elde edilen bilgileri daha anlamlı bir yapıya dönüştürme amacıyla kullanılan bilgisayar bilimi alt alanıdır (Han ve ark., 2011). Elde edilen anlamlı bilgiler ile hastalık teşhisi, alışveriş analizi, siber dolandırıcılık tespiti, müşteri edinme, üretim kontrolü ve bilgi keşfi gibi birçok problem başarılı bir şekilde analiz edilebilir ve çözülebilir. Veri madenciliği ve makine öğrenmesinin en önemli alt alanlarından biri olan sınıflandırma işlemi literatürde de sıklıkla kullanılmaktadır (Michalski ve ark., 1998). Sınıflandırma işleminde algoritmalar kendilerine verilen eğitim kümesinden öğrenerek, daha önce görülmemiş yeni örnekleri sınıflandırmak için kullanılan denetimli öğrenme yaklaşımlarıdır. Sınıflandırma algoritmasının eğitimi sonrasında, veri sınıflarını tanımlayan bir model elde edilir. Elde edilen model IF-THEN şeklindeki kurallardan, ağaç veri yapılarından, matematiksel denklemlerinden veya sinir ağları gibi farklı yapılardan oluşabilir (Han ve ark., 2011). Elde edilen bu sınıflandırma modellerinin hedefi sınıf etiketleri bilinmeyen örneklerin sınıflarını doğru bir şeklide tahmin etmektir (Witten ve ark., 2016). Literatürde bir çok araştırmacı tarafından kural tabanlı, olasılık tabanlı ve ağırlık tabanlı olmak üzere farklı özelliklerde sınıflandırma algoritmaları geliştirilmiştir (Michalski ve ark., 1998). Geliştirilen bütün bu sınıflandırma algoritmalarının ortak amacı gerçek dünya sınıflandırma problemlerine daha başarılı çözümler üretmektir.

Kural tabanlı sınıflandırma algoritmaları, veri madenciliği sınıflandırma problemlerinde kullanılan sınıflandırma yöntemlerinden bir tanesidir. Kural çıkarımı, eğitim verilerinden sınıflandırma işleminde kullanılacak kural kümelerin elde edilmesine denir. Elde edilen kural kümeleri karar mekanizması olarak sınıflandırma veya tahmin problemlerinde kullanılabilir (Witten ve ark., 2016). Günümüze kadar geliştirilen kural tabanlı sınıflandırma algoritmaları genellikle olasılık işlemlerini kullanarak sınıflandırma kuralları elde ettiğinden dolayı ayrık ve kategorik veri kümeleri ile birlikte daha iyi performans sergilemektedirler. (Hacibeyoglu ve ark., 2011). İstatistik tabanlı sınıflandırma algoritmaları olasılık teorilerine dayanarak sınıflandırma işlemini gerçekleştirir. Öncelikle eğitim kümesindeki örneklerin olasılıklarını hesaplar ve bu olasılık değerlerine göre yeni gelen örneğin sınıfını veya değerini tahmin eder. Ağırlık tabanlı sınıflandırma algoritmaları ise genellikle matematik fonksiyonları kullanarak

sınıflandırma modellerini tasarladığından dolayı sürekli veri kümeleri üzerinde daha iyi sınıflandırma performansı sergilemektedirler (Pyle, 1999). Dolayısıyla sınıflandırma algoritmaları sınıflandırılacak veriye göre ayrık sınıflandırma algoritmaları ve sürekli sınıflandırma algoritmaları olarak iki kategoriye ayrılabilir (Zaki ve ark., 2014).

Günümüzde birçok gerçek dünya uygulamasında sürekli veriler bulunur, ancak buna karşın CN2 (Clark ve Niblett, 1989), Algorithm Quasi-optima (AQ) (Michalski ve ark., 2013), Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (Cohen, 1995; Sasaki ve Kita, 1998), Iterative Dichotomiser 3 (ID3) (Loh, 2011), C4.5 (Loh, 2011), C5.0 (Loh, 2011), Classification and Regression Trees (CART) (Steinberg ve Colla, 2009), K-Yakın Komşu (K-YK) (Denoeux, 1995), Destek Vektör Makinesi (DVM) (Gunn, 1998), Karar Ağaçları (KA) ve Naïve Bayes (NB) (!!! INVALID CITATION !!! (Domingos ve Pazzani, 1997; Yang ve Webb, 2009; Zhang ve ark., 2011)) gibi birçok veri madenciliği sınıflandırma algoritmaları genellikle ayrık veya kategorik veriler ile daha kolay, daha hızlı ve daha başarılı olarak çalışırlar (Wu ve ark., 2008).

Veri madenciliği sınıflandırma algoritmalarının performansları genellikle veri önişleme teknikleri kullanarak veya farklı algoritmalar ile melez bir şekilde eğitilerek artırılır (Chlebus ve Nguyen, 1998). Veri önişleme ham verilerin anlaşılır bir formata dönüştürülmesini içeren bir veri madenciliği tekniğidir. Gerçek dünya verileri çoğu zaman eksik, tutarsız ve belirli davranışlardan yoksundur. Veri temizleme, veri birleştirme, veri dönüşümü, veri azaltma ve veri ayrıklaştırma literatürde sıklıkla uygulanan veri önişleme tekniklerindendir. Veri ayrıklaştırma, sürekli verileri ayrık verilere dönüştürmeyi sağlayan bir işlemdir ve ayrık veri kümeleri ile daha yüksek performans gösteren veri madenciliği sınıflandırma algoritmalarının uygulanması öncesinde sıklıkla kullanılmaktadır (Kotsiantis ve Kanellopoulos, 2006; Kumar ve Inbarani, 2018). Veri ayrıklaştırma işleminde sürekli veriler, bir aralık dizisi oluşturularak herhangi bir kesişimi olmayan ayrık değerlere dönüştürülür (Lustgarten ve ark., 2008). Veri ayrıklaştırma işleminin temel avantajları (!!! INVALID CITATION !!! (Hu ve ark., 2009; Dash ve ark., 2011; Zaki ve ark., 2014; Rahman ve Islam, 2016)):

- Veri boyutu azaltılır ve böylelikle veriler bellekte daha az yer kaplar.
- Ayrık verileri anlamak, kullanmak ve açıklamak çok daha kolaydır, çünkü bilgi düzeyine sürekli verilerden daha yakındır.
- Veri madenciliği sınıflandırma algoritmaları ayrık veriler ile daha hızlı çalışabilir ve daha iyi sınıflandırma başarıları elde edebilirler.

- Sürekli verilerde bulunan gürültülü veriler ve aykırı değerler veri ayrıklaştırma işlemi ile azaltılabilir.

Günümüze kadar literatürde farklı araştırmacılar tarafından farklı özelliklere sahip olan ve farklı alanlarda iyi performanslar sergileyen birçok ayrıklaştırma yöntemi geliştirilmiştir. Bunlardan önemli birkaçı şu şekilde verilebilir: Eşit Genişlik (EG), Eşit Frekans (EF) (Dougherty ve ark., 1995), Minimum Description Length Principle (MDLP) (Fayyad ve Irani, 1993), Entropi Tabanlı ID3 (ET-ID3) (Bertelsen ve Martinez, 1994; Quinlan, 2014), ChiMerge (Kerber, 1992), 1R (Holte, 1993), D2 (Catlett, 1991), Chi-Square (Chi2) (Liu ve Setiono, 1995; Su ve Hsu, 2005; Cebeci ve Yildiz, 2017), Fixed Frequency Discretization (FFD) (Yang ve Webb, 2009), Information-Theoretic Fuzzy Partitioning (ITFP) (Au ve ark., 2006), Class-Attribute Interdependence Maximization (CAIM) (Kurgan ve Cios, 2003; 2004), Class-Attribute Dependent Discretizer (CADD) (Ching ve ark., 1995), Minimum Optimal Description Length (MODL) (Boullé, 2006) ve Multivariate Discretization (MVD) (Bay, 2001). Geliştirilen bütün bu ayrıklaştırma yöntemlerinin ortak amacı verinin sınıflandırma algoritmaları tarafından daha iyi anlaşılmasını sağlamak ve sınıflandırma başarısını iyileştirmektir.

Bazı sınıflandırma algoritmaları genellikle sürekli veriler üzerinde daha başarılı sınıflandırma performansı sergilerler. Örnek olarak, ağırlık tabanlı bir sınıflandırma algoritması olan yapay sinir ağlarında (YSA) sınıflandırma modeli matematiksel denklemler ile tasarlandığından dolayı sürekli veri kümelerinin sınıflandırılmasında başarılı sonuçlar elde edilir. Ağırlık tabanlı sınıflandırma algoritmaları aynı zamanda parametrik sınıflandırma algoritmaları olarak da isimlendirilebilir. Bu tür sınıflandırma algoritmalarında sınıflandırma modelinin oluşturabilmesi için kullanıcı tarafından sınıflandırma performansını etkileyen bir veya birden fazla parametrenin belirlenmesi gerekmektedir. Ağırlık tabanlı sınıflandırma algoritmalarının eğitimi sınıflandırma modellinde kullanılan ağırlıkların uygun değerlere güncellenmesidir. Bu ağırlıkların en uygun değerlere güncellenmesi işleminde Geriye Yayılım (Lacher ve ark., 1992), Levenberge-Marquardt Algorithm (Moré, 1978), Scaled Gradient Conjugate Backpropagation (Møller, 1993), Resilient Backpropagation (Günther ve Fritsch, 2010), Conjugate Gradient Backpropagation with Powell-Beale Restarts (Saini ve Soni, 2002), Conjugate Gradient Backpropagation with Fletcher-Reeves Updates (Nawi ve ark., 2007) ve Conjugate Gradient Backpropagation with Polak-Riebre Updates (Liu ve ark., 2015) gibi öğrenme algoritmalarından faydalanılır. Fakat bu tür gradyan tabanlı öğrenme

algoritmaları genellikle yavaş çalışmakta ve yerel minimuma takılabilmektedir (Riedmiller ve Braun, 1993). Bu tür problemlerden kurtulmak için gradyan tabanlı öğrenme algoritmalarının yerine meta-sezgisel optimizasyon algoritmalar kullanılabilir. Bir ağırlık tabanlı sınıflandırma modelinin en uygun ağırlıklarını belirlemek, doğrusal olmayan bir optimizasyon problemidir ve meta-sezgisel optimizasyon algoritmalar ile bu tür problemlere çözüm bulunabilir.

1.1. Tez Çalışmasının Amacı

Veri madenciliği sınıflandırma problemleri ve bu problemlere çözüm olacak veri madenciliği ve makine öğrenmesi algoritmaları temelli yaklaşımların kullanımı günümüzde başta sağlık, bankacılık, ticaret, tarım ve endüstri olmak üzere hemen her alanda görülmektedir. Özellikle veri madenciliği ve makine öğrenme algoritmalarının kullanılmasıyla birlikte kendi kendine karar verebilen akıllı sistemlerin geliştirilmesi oldukça revaçta olan bir konudur. Özellikle sınıflandırma ve tahmin işlemleri için geliştirilen otonom sistemlerin başarıları direkt olarak sınıflandırma algoritmalarının başarıları ile doğru orantılıdır. Gerçek dünya problemlerinde veriler genellikle sürekli olarak bulunurlar. Fakat bazı sınıflandırma algoritmaları sürekli değere sahip veriler ile daha başarılı sonuçlar elde ederken bazı sınıflandırma algoritmaları ise ayrık değere sahip veriler ile daha başarılı çalışmaktadırlar. Ayrıklaştırma işlemi, veri madenciliği ve makine öğrenmesinin en önemli veri ön işleme yöntemlerinden bir tanesidir. (Han ve ark., 2011). Kural tabanlı sınıflandırma algoritmaları ayrık veri kümeleri üzerinde daha iyi bir sınıflandırma performansı sergileyebilirler (Hacibeyoglu ve ark., 2011). Burada veri ayrıklaştırma işleminin kural tabanlı sınıflandırma algoritmaları üzerindeki etkisi çok belirleyici olmaktadır. Sürekli verilerden iyi bir performansa sahip olan kural tabanlı sınıflandırma modeli oluşturmak için sürekli verilerin veri önişleme ile ayrıklaştırılması gerekmektedir. Literatürde bu güne kadar kural tabanlı sınıflandırma algoritmaları için birçok ayrıklaştırma yöntemleri geliştirilmiştir. Bu ayrıklaştırma yöntemleri farklı problemler için birçok sınıflandırma algoritmaları ile çok başarılı sınıflandırma işlemi gerçekleştirmişlerdir. Bunun yanında sürekli veriler üzerinde daha başarılı sınıflandırma işlemi yapan özellikle ağırlık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek için literatürde farklı yöntemler sunulmuştur. Sınıflandırma algoritmasının eğitiminin optimizasyon algoritmaları ile melez bir şekilde gerçekleştirilmesi, bu yöntemler içerisinde en sık kullanılanıdır.

Bu tezin amacı gerçek dünyada hemen hemen her alanda kullanılan sınıflandırma algoritmalarının performansını iyileştirmektir. Dolayısıyla kural ve olasılık tabanlı sınıflandırma algoritmaları için yeni bir ayrıklaştırma yöntemi ve ağılık tabanlı sınıflandırma algoritmaları için optimizasyon tabanlı yeni bir eğitim algoritması geliştirilmiştir. Ayrık veriler ile daha başarılı sonuçlar veren özellikle kural ve olasılık tabanlı sınıflandırma algoritmaları için literatürde Eşit Aralık, Eşit Frekans ve Entropi tabanlı ID3 ayrıklaştırma yöntemleri ayrıntılı olarak incelenmiştir. Denetimsiz Eşit Frekans ayrıklaştırma yöntemini iyileştirmek üzere EF_Unique adında yeni bir ayrıklaştırma yöntemi literatüre sunulmuştur. Sürekli veriler ile daha başarılı sonuçlar veren ağırlık tabanlı sınıflandırma algoritmalarının eğitimi için çoklu ortalama parçacık sürü optimizasyon algoritması geliştirilmiştir. Geliştirilen optimizasyon tabanlı eğitim algoritması çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının eğitiminde kullanılmıştır. Yapılan deneysel çalışmalar sonucunda ilk olarak, geliştirilen EF_Unique ayrıklaştırma yöntemi kural ve olasılık tabanlı sınıflandırma algoritmalarının performansını çok etkileyici olarak iyileştirmiştir. İkinci olarak, geliştirilen çoklu ortalama parçacık sürü optimizasyon algoritması çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının daha başarılı sınıflandırma sonuçları elde ettiği görülmüştür.

1.2. Tez Çalışmasının Önemi

Gelişen teknoloji ile birçok alanda insan gücü ve zekâsı ile yapılan birçok işlem veri madenciliği ve makine öğrenmesi algoritmaları ile yapılmaya başlanmıştır. Bu yüzden gerçek dünya problemlerinin çözümünde sıklıkla kullanılan sınıflandırma algoritmalarının performansı büyük önem kazanmaktadır. Sınıflandırma algoritmalarının sınıflandırma başarıları veri kümesinin hazırlanması, giriş parametrelerin belirlenmesi ve modelin uygun şekilde tasarlanması ve eğitim işleminin en iyi şekilde yapılmasıyla doğrudan ilgilidir. Kural tabanlı sınıflandırma algoritmalarında veri kümesinden çıkarılan sınıflandırma kurallarının basit, doğru ve verimli olması sınıflandırma modellerinin performansını iyileştirmektedir. Kural tabanlı sınıflandırma algoritmalarında sınıflandırma kuralları veri kümelerinden çıkarıldığından dolayı veri kümelerinin uygun bir formatta hazırlanması gerekmektedir. Kural tabanlı sınıflandırma algoritmalarının sürekli veriler üzerinde uygulanması için başarılı bir ayrıklaştırma önişleminin yapılması büyük önem arz etmektedir. Ağırlık tabanlı sınıflandırma algoritmaları ayrık ve sürekli veriler ile birlikte çalışabilmesine rağmen eğitim işleminin başarılı bir şekilde yapılması

gerekmektedir. Ağırlık tabanlı sınıflandırma algoritmalarının eğitimi ağırlıkların en uygun değerlere güncellenmesidir.

Bu tez çalışmasında, hem kural ve olasılık tabanlı hem de ağırlık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek için iki yeni yaklaşım geliştirilmiştir. Birinci geliştirilen yaklaşım, Eşit Frekans ayrıklaştırma yönteminin iyileştirilmesi ile daha yüksek performanslı EF_Unique isminde yeni bir ayrıklaştırma yöntemidir. Geliştirilen EF_Unique ayrıklaştırma yöntemi kural ve olasılık tabanlı sınıflandırma algoritmaları için bir veri önişleme tekniği olarak bütün dünya problemlerinin çözümünde kullanılabilir. İkinci geliştirilen yaklaşım ise, ağırlık tabanlı sınıflandırma algoritmalarının eğitiminde parçacık sürü optimizasyonu algoritmasının geliştirilmiş bir versiyonunun kullanılmasıdır. Her iki geliştirilen yaklaşım da literatüre bir yenilik getirmiş olup ileride bu konuda yapılacak çalışmalara katkı sağlayacağı düşünülmektedir.

2. KAYNAK ARAŞTIRMASI

Literatürdeki veri madenciliği sınıflandırma işlemi ile ilgili yapılan çalışmalar incelendiğinde bazı sınıflandırma algoritmalarının ayrık veri kümeler ve bazılarının ise sürekli veri kümeler ile daha iyi performans gösterdiği gözlenmiştir. Genellikle kural ve olasılık tabanlı sınıflandırma algoritmaları ayrık veri kümelerine daha uygun bir sınıflandırma modeli oluşturabilirler. Fakat gerçek dünyada birçok sınıflandırma probleminin veri kümesi sürekli değerlidir. Bundan dolayı bu tür sürekli veri kümelerinin kural ve olasılık tabanlı sınıflandırma algoritmaları ile kullanılması iyi bir sınıflandırma başarısı elde edemeyebilir. Bu yüzden sürekli veri kümeleri üzerinde kural ve olasılık tabanlı sınıflandırma algoritmaları ile sınıflandırma modeli oluşturulmadan önce, sürekli veri kümelerine bir veri önişleme tekniği olan ayrıklaştırma işleminin uygulanması gerekir. Günümüze kadar, araştırmacılar tarafından farklı uygulamalarda kullanılmak amacıyla EG, EF, ET-ID3, MDLP, Chi2, CAIM, CADD ve MODL gibi birçok ayrıklaştırma yöntemi geliştirilmiştir. Bu ayrıklaştırma yöntemlerinin kendilerine göre bazı avantaj ve dezavantajları bulunmaktadır. Literatürde bu ayrıklaştırma yöntemleri sınıflandırma ve kümeleme gibi makine öğrenmesi algoritmalarında veri ön işlemi teknikleri olarak kullanılmıştır (Kotsiantis ve Kanellopoulos, 2006).

Koçoğlu tarafından yapılan tez çalışmasında, veri madenciliğinde ChiMerge, Chi2, eşit genişlikli, eşit frekans, 1RD, ID3, CADD ve CAIM ayrıklaştırma yöntemleri birer örnek üzerinde detaylı olarak anlatmıştır. Deneysel çalışmalar kısmında, Wisconsin üniversitesi hastanesinden alınan hasta verileri üzerinde ayrıklaştırma yöntemlerinin sınıflandırma işlemine olan performans etkisinin karşılaştırılması yapılmıştır. Elde edilen deneysel sonuçlara göre denetimli ayrıklaştırma yöntemleri denetimsiz ayrıklaştırma yöntemlerinden daha iyi performans sergilemişlerdir (Koçoğlu, 2012).

Doğan tarafından yapılan tez çalışmasında, ayrıklaştırma yöntemleri ve çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması kullanılarak üç fazlı asenkron motorların arıza tespitine dayalı bir arıza tespit modeli geliştirilmiştir. Geliştirilen model asenkron motorların rulman, eksenden kaçıklık, rotor çubuk kırığı ve stator sargı kısa devresi arızalarının tespitinde kullanılmıştır. Sinyal analizinde frekans ve zaman-frekans yöntemleri yerine, zaman boyutunda ayrıklaştırma yöntemleri kullanılmıştır. Deneysel çalışmalar beş farklı arızaya sahip üç fazlı asenkron motorları üzerinde gerçekleştirilmiştir. Elde edilen deneysel sonuçlara göre önerilen ayrıklaştırma yöntemleri ve çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması tabanlı arıza

tespit modeli üç fazlı asenkron motorların arıza tespitini daha başarılı bir şekilde gerçekleştirmiştir (Doğan, 2012).

Sayın tarafından yapılan tez çalışmasında, ayrıklaştırma teknikleri, özellik çıkarımı algoritmaları ve sınıflandırma algoritmaları ile birlikte kullanılarak koroner arterlerde kalsifikasyon hastalığının tespitini yapan model geliştirilmiştir. Geliştirilen tespit modeli yardımıyla hastalardan alınan yaş, nakil süresi, diabet, fosfor, rose anjina testi, verici tipi ve hastanın hastalık geçmişi verileri ile hastada koroner arterlerde kalsifikasyon olup olmadığına karar verilebilmektedir. Çalışmanın deneysel sonuçlarına göre önerilen koroner arterlerde kalsifikasyon tespit modeli %75 başarı oranı ile hastalığı doğru tespit etmiştir (Sayın, 2013).

Özdemir tarafından yapılan tez çalışmasında, bilgisayar ağları için makine öğrenmesi tabanlı bir saldırı tespit sistemi önerilmiştir. Önerilen saldırı tespit sisteminde yerine geri koyarak örnekleme, ayrıklaştırma, öznitelik seçme veri önişleme teknikleri kullanmıştır. Önerilen saldırı tespit sisteminde önce veriler ayrıklaştırma yöntemi ile ayrıklaştırılmış ve daha sonra bu verilerden özellik çıkarım yöntemi ile belirleyici özellikler seçilmiştir. Son olarak, önerilen saldırı tespit sisteminin eğitimi için J48 sınıflandırma algoritması kullanılmıştır. Yaptığı deneysel çalışmaların sonuçlarına göre önerilen saldırı tespit sistemi KDD'99 veri kümesi üzerinde %97.6 sınıflandırma başarısı elde etmiştir (Özdemir, 2011).

Koç tarafından yapılan tez çalışmasında, veri kümesindeki önemli özelliklerin seçilme işlemi için sürekli optimizasyon tekniklerinden Guguk Kuşu (Yang ve Deb, 2009), Yapay Arı Kolonisi (Karaboga, 2005), Yerçekimsel Arama (Rashedi ve ark., 2009) ve Yarasa (Yang, 2010) optimizasyon algoritmaları kullanmıştır. Optimizasyon algoritmalarının sonuçları ayrıklaştırma işlemine tabi tutularak önemli özellikler seçilmiştir. Deneysel çalışmalarda literatürde sıklıkla kullanılan UCI makine öğrenmesi veri ambarından BCW, QSAR, DRD ve WPBC veri kümeleri kullanmıştır. Yazarın özellik seçme işlemi için önermiş olduğu optimizasyon tabanlı ayrıklaştırma yöntemi kullanılan veri kümeleri üzerinde başarılı sonuçlar elde etmiştir (Koç, 2016).

Türkiye'de yapılan yüksek lisans ve doktora tezleri çalışmalarına ek olarak uluslararası literatürde yapılan çalışmalar da incelenmiştir.

Dougherty ve arkadaşları tarafından yapılan çalışmada, denetimsiz ayrıklaştırma yöntemlerinden eşit genişlik, eşit frekans ve denetimli ayrıklaştırma yöntemi olan entropi tabanlı ID3 ayrıklaştırma yöntemleri çok detaylı olarak incelenip birbiriyle karşılaştırılmıştır. Karşılaştırma işlemi UCI makine öğrenmesi veri ambarından veri

kümeleri üzerinde NB ve C4.5 sınıflandırma algoritmaları kullanılarak yapılmıştır. Karşılaştırma sonucuna göre NB ve C4.5 sınıflandırma algoritmaları için entropi tabanlı ID3 ayrıklaştırma yönteminin sınıflandırma performansını artırdığı belirtilmiştir (Dougherty ve ark., 1995).

Boulle, yapmış olduğu çalışmada Khiops adında kesirli öznitelikler için istatistik tabanlı Chi-square ayrıklaştırma yöntemini önermiştir. Önerilen yöntem tüm ayrıklaştırma alanı üzerinde küresel bir şekilde Chi-square ölçütlerini optimize etmekte ve herhangi bir durma ölçütü gerektirmemektedir. Khiops ayrıklaştırma yöntemi NB sınıflandırma algoritmasının bir veri önişleme adımı olarak düşünülen diğer denetimli ve denetimsiz ayrıklaştırma yöntemleri ile karşılaştırılmıştır. Elde edilen sonuçlara göre Khiops ayrıklaştırma yöntemi NB sınıflandırıcısı için diğer ayrıklaştırma yöntemlerinden daha iyi bir performans sergilemiştir (Boullé, 2006). Boulle yapmış olduğu diğer bir çalışmada, MODL adında Bayesian yaklaşımına dayanan yeni bir ayrıklaştırma yöntemi önermiştir. Önerdiği ayrıklaştırma yönteminin performansını değerlendirmek için literatürdeki ayrıklaştırma yöntemleri ile karşılaştırmıştır. Karşılaştırmayı UCI makine öğrenmesi veri ambarındaki veri kümeleri üzerinde NB ve C4.5 sınıflandırma algoritmaları ile gerçekleştirmiştir. Karşılaştırma sonucuna göre yazarın önerdiği ayrıklaştırma yöntemi diğer ayrıklaştırma yöntemlerine göre daha yüksek performans göstermiştir (Boullé, 2006).

Yan ve arkadaşları, yapmış oldukları çalışmada yeni denetimli bir ayrıklaştırma yöntemi olan Novel Class Attribute Interdependency Discretization Algorithm (NCAIC) ayrıklaştırma yöntemi önermişlerdir. Önerilen NCAIC ayrıklaştırma yöntemi deneysel çalışmalarda CADD, CAIM, CACC, EF ve MDL olmak üzere beş farklı ayrıklaştırma yöntemi ile karşılaştırılmıştır. Deneysel çalışmalarda kullanılan ayrıklaştırma yöntemlerinin performansını değerlendirmek için C4.5 sınıflandırma algoritması kullanılmıştır. Deneysel sonuçlara göre önerilen NCAIC ayrıklaştırma yönteminin diğer ayrıklaştırma yöntemlerinden daha üstün performans gösterdiği söylenmiştir (Yan ve ark., 2014).

Gupta ve arkadaşları, kümeleme tabanlı bir ayrıklaştırma yöntemi önermişlerdir. Önerilen kümeleme tabanlı ayrıklaştırma yönteminde ayrım noktalarının sayıları kullanılan veri kümelerinin sınıf sayılarına eşitlenmiştir. Ayrıklaştırma yönteminin performansı UCI makine öğrenmesi veri ambarındaki veri kümeleri üzerinde DVM ve NB sınıflandırma algoritmaları kullanarak gerçekleştirilmiştir. Deneysel sonuçlara göre

önerilen ayrıklaştırma yönteminin sınıflandırma algoritmalarının performansını artırdığı belirtilmiştir (Gupta ve ark., 2010).

Jiang ve Sui, olasılık tabanlı NB algoritmasından türetilmiş denetimli ve çok değişkenli bir ayrıklaştırma algoritması olan supervised and multivariate discretization algorithm (SMDNS) algoritmasını önermişlerdir. Önerilen ayrıklaştırma yöntemi SMD, Entropi, CACC, NCAIC ve EF ayrıklaştırma yöntemleri ile karşılaştırılmıştır. Deneysel sonuçlara göre, SMDNS'nin sınıflandırma doğruluğunu artırdığı ve ayrıklaştırma işlemi için gerekli olan ayrım noktaların sayısı daha iyi bulduğu gözlemlenmiştir (Jiang ve Sui, 2015).

Wong, parametrik olmayan melez bir ayrıklaştırma yöntemi geliştirmiştir. Geliştirilen melez ayrıklaştırma yöntemi NB sınıflandırma algoritmasının sınıflandırma doğruluğunu artırmıştır. Önerilen ayrıklaştırma yöntemi 20 veri kümesi ile test edilmiş ve elde edilen test sonuçları, çeşitli ayrıklaştırma yöntemleri ile karşılaştırılmıştır. Önerilen ayrıklaştırma yöntemi ile NB sınıflandırma algoritmasının daha yüksek bir tahmin doğruluğuna sahip olduğu gözlemlenmiştir (Wong, 2012).

Rahman ve Islam, kullanıcı tarafından bir parametre girişi gerektirmeyen Düşük Frekanslı Ayrıklaştırıcı (Low Frequency Discretizer LFD) adında yeni bir ayrıklaştırma yöntemi önermişlerdir. Kullanıcı tarafından aralık sayısı girişi gerektiren ayrıklaştırma yöntemlerinde her aralıktaki eleman sayısı aynıdır. LFD, kesme noktalarını oluşturmak için düşük frekans değerlerini kullanır ve bunun sayesinde ayrıklaştırma işleminde ortaya çıkabilecek bilgi kaybını azaltır. Önerdikleri ayrıklaştırma yöntemini UCI makine öğrenmesi ambarından seçilen sekiz farklı veri kümesi üzerinde test etmişlerdir. Elde edilen test sonuçlarını literatürde sıkça kullanılan altı ayrıklaştırma yönteminin sonuçları ile karşılaştırmışlardır. Karşılaştırma sonuçlarının analizinde LFD ayrıklaştırma yönteminin kullanılan veri kümeleri üzerinde diğer yöntemlere göre daha iyi performans sağladığı sunulmuştur (Rahman ve Islam, 2016).

Ağırlık tabanlı sınıflandırma algoritmalarının performansını etkileyen en önemli etken sınıflandırma modelinde bulunan ağırlık değerleridir (Han ve ark., 2011). Çünkü bu ağırlıklar giriş değerleri ile birlikte matematiksel fonksiyonlardan geçerek sınıflandırma modelinin çıkışını üretmektedir. Günümüze kadar ağırlık tabanlı sınıflandırma algoritmalarının eğitimi için birçok ağırlık bulma yöntemi geliştirilmiştir. Bunlardan birkaçı şu şekildedir: Geriye Yayılım, Levenberge-Marquardt algorithm, Scaled Gradient Conjugate Backpropagation, Resilient Backpropagation, Conjugate gradient Backpropagation with Powell-Beale Restarts, Conjugate Gradient

Backpropagation with Fletcher-Reeves Updates ve Conjugate Gradient Backpropagation with Polak-Riebre Updates (Ekinci ve ark., 2015). Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması, ağırlık tabanlı sınıflandırma algoritmaları içerisinde başta medikal, bankacılık ve mühendislik gibi birçok alanda sıklıkla kullanılan etkin bir sınıflandırma algoritmasıdır. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının eğitiminde genellikle Geriye Yayılım öğrenme algoritması kullanılarak ağırlıklar problemin çözümü için en uygun değerlere güncellenir. Fakat geriye yayılım öğrenme algoritmasının yavaş çalışması ve yerel noktalara takılması gibi dezavantajları bulunmaktadır. Bu dezavantajların giderilmesi için literatürde yeni yaklaşımlar önerilmiştir. Türkiye'de yapılan çalışmalar incelendiğinde, Öztürk tarafından yapılan çalışmada, çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının eğitimi için sürü zekasına dayalı yeni bir teknik olan Yapay Arı Kolonisi (ABC) optimizasyon algoritması önerilmiştir. Araştırmacı tarafından önerilen yaklaşım MLP, LVQ, RBF gibi modern optimizasyon yöntemleriyle karşılaştırılmıştır. Deneysel çalışmalarda UCI makine öğrenmesi veri ambarından Kanser, Diyabet, Kalp, Kart, Gen, Cam, At, Soya ve Tiroit veri kümeleri kullanılmıştır. Deneysel sonuçların analizinde yazarın önerdiği ABC eğitimli YSA sınıflandırma algoritmasının kullanılan veri kümeleri üzerinde diğer yöntemlere göre daha iyi performans sağladığı belirtilmiştir (Öztürk, 2011).

Yalçın tarafından yapılan çalışmada, elektroansefalogram (EEG) epilepsi teşhisi için sezgisel optimizasyon algoritması olan parçacık sürüsü optimizasyonu (PSO) eğitimli YSA sınıflandırma algoritması (PSONN) önerilmiştir. PSO ile eğitilen YSA sınıflandırma algoritması ve geriye yayılım yöntemi ile eğitilen YSA sınıflandırma algoritması EEG teşhisi üzerinde karşılaştırılmış ve sonuçlar analiz edilmiştir. Deneysel sonuçların analizine göre araştırmacının önerdiği PSONN sınıflandırma algoritması %98 sınıflandırma başarısı ile BPNN sınıflandırma algoritmasından daha iyi bir başarı elde etmiştir. Buna ek olarak yazar EEG teşhisi için PSONN1, PSONN2, PSONN3, PSONN4, PSONN5, PSONN6 ve PSONN7 isimlerinde PSO'nun yedi farklı sürümü ile eğitilmiş YSA (PSO-YSA) modeli önermiştir. Bu modeller arasından PSONN3 ve PSONN7 modellerinin EEG teşhisi için en uygun modeller olduğunu söylemiştir (Yalçın, 2012).

Kulluk tarafından yapılan çalışmada, sınıflandırma veri kümelerinden verimli bir şekilde sınıflandırma kuralları çıkarmak için meta-sezgisel tabanlı yeni bir kural çıkarma modeli önerilmiştir. Önerilen kural çıkarma modelinde Tur Atan Karınca Koloni Optimizasyon Algoritması (TAKKO) ile çok-katmanlı YSA sınıflandırma algoritması kullanılmıştır. Öncelikle çok-katmanlı YSA sınıflandırma algoritmasının eğitimi

sonucunda modele uygun ağırlıklar çıkarılmakta ve daha sonra TAKKO algoritması ile sınıflandırma kuralları elde edilmiştir. Kural çıkarma modeli farklı özelliklere sahip 12 UCI makine öğrenmesi veri ambarındaki veri kümesi ile test edilmiş ve deney sonuçları veri madenciliğinde sıklıkla kullanılan NBTree, DT, PART ve C4.5 sınıflandırma algoritmalarının sonuçları ile karşılaştırılmıştır. Elde edilen deney sonuçlarına göre önerilen kural çıkarma algoritması klasik NBTree, DT, PART ve C4.5 sınıflandırma algoritmalarına göre daha etkili ve özlü sınıflandırma kuralları elde etmiştir (Kulluk, 2009).

Özdemir tarafından yapılan çalışmada, çok-katmanlı YSA sınıflandırma algoritmasının parametrelerinin optimizasyonu ve ağırlık değerlerinin eğitimi genetik algoritma (GA) kullanılarak yapılmıştır. Önerilen sınıflandırma algoritmasına GYSA ismi verilmiştir ve doğrusal olmayan dinamik sistemlerde GYSA algoritmasının testleri yapılmıştır. Elde edilen test sonuçlarına göre doğrusal olmayan dinamik sistemler için tasarlanan GYSA algoritması iyi bir modelleme performansı ile çalıştığı belirtilmiştir (Özdemir, 2010).

Türkiye'de yapılan çalışmaların incelenmesine ek olarak literatürde uluslararası dergilerde yayınlanan ağırlık tabanlı sınıflandırma algoritmalarının eğitimi için yapılan ilgili çalışmalar da incelenmiştir. Mirjalili ve arkadaşları tarafından yapılan çalışmada, sosyal örümcek optimizasyon algoritması kullanılarak evrimsel ileriye dönük sinir ağları tasarlanmıştır. İleri YSA'nın yapısında bulunan ağırlıkların güncellenmesi sosyal örümcek optimizasyon algoritması ile yapılmıştır. Önerilen yöntemin matematiksel ve deterministik yöntemlere göre daha başarılı olduğu belirtilmiştir. Önerdikleri ileri dönük YSA'nın tasarımı UCI veri ambarında bulunan XOR lojik işlemi, Balloon, Iris, Breast cancer, ve Heart veri kümeleri ile test edilmiştir. Elde edilen deney sonuçları literatürde iyi bilinen PSO, GA, karınca kolonisi, evrim stratejisi ve geriye yayılım algoritmaları ile eğitilen YSA sonuçları ile karşılaştırılmıştır. Deney sonuçlarına göre, sosyal örümcek optimizasyon algoritması ile eğitilmiş ileriye dönük sinir ağları diğer algoritmalar ile eğitilen YSA'ya kıyasla umut verici sonuçlar sağlamıştır (Mirjalili ve ark., 2015).

Uzlu ve arkadaşları yapmış oldukları çalışmada, Türkiye'nin yıllık hidrolik enerji üretimini tahmin etmek için yapay arı kolonisi algoritması destekli YSA tahmin modeli önermişlerdir. Modelde brüt elektrik enerjisi talebi, ortalama yıllık sıcaklık ve enerji tüketimi bağımsız değişkenler olarak seçilmiştir. Geliştirdikleri tahmin modelinin performansını değerlendirmek için önce YSA-ABC tahmin modelinin sonuçları geri yayılım ile eğitilmiş YSA tahmin modelinden elde edilen sonuçlarla karşılaştırılmıştır.

İki modelin arasında ki hata oranlarında ciddi farklar olduğu ortaya çıkmıştır. Yazarlar en uygun parametreleri belirledikten sonra, Türkiye için gelecekteki hidroelektrik üretim değerlerini tahmin etmek için üç farklı model geliştirmişlerdir. Deney sonuçlarına göre, hidroelektrik enerji üretimi tahmininde YSA-ABC algoritmasının, geri yayılım algoritması ile eğitilmiş YSA'ya göre daha başarılı olduğu görülmüştür (Uzlu ve ark., 2014).

Salama ve Abdelbar tarafından yapılan çalışmada, ileriye dönük YSA sınıflandırma algoritmasının eğitimi karınca koloni algoritması ile gerçekleştirilmiştir. Önerilen karınca algoritması ile eğitimi gerçekleşen YSA sınıflandırma algoritmasının performansını ölçmek için, UCI makine öğrenmesi veri ambarındaki 40 veri kümesi üzerinde testler yapılmıştır. Önerdikleri karınca kolonisi algoritması ile eğitilmiş YSA sınıflandırma algoritmasından elde ettikleri test sonuçlarını geriye yayılım yöntemi ile eğitilmiş YSA sınıflandırma algoritması ve çeşitli evrimsel algoritmalar ile eğitilmiş YSA sınıflandırma algoritması karşılaştırılmıştır. Deney sonuçlarının analizine göre yazarlar tarafından önerilen karınca kolonisi ile eğitilmiş YSA sınıflandırma algoritması diğer sınıflandırma algoritmalarına göre üstün başarılar sergilemiştir (Salama ve Abdelbar, 2015).

Kankal ve Uzlu, Türkiye'deki elektrik enerjisi talebini modellemek için öğrenme-temelli optimizasyon tekniği (TLBO) ile eğitilmiş YSA sınıflandırma algoritmasının performansını incelemişlerdir. TLBO kullanılarak eğitilen YSA (ANN-TLBO) modeli, geri yayılımlı ANN (ANN-BP) ve yapay arı kolonisi algoritması ile eğitilen YSA (ANN-ABC) modelleriyle karşılaştırılmıştır. Brüt yerli üretim, nüfus, ithalat ve ihracat, modellerde bağımsız değişkenler olarak seçilmiştir. Karşılaştırma sonuçlarına göre, ANN-TLBO modelinin elektrik enerjisi talebinin tahmininde ANN-BP ve ANN-ABC modellerinden daha iyi performans gösterdiği ortaya konulmuştur (Kankal ve Uzlu, 2017).

Hu ve arkadaşları tarafından yapılan çalışmada, çok yollu ultrasonik debimetrenin karmaşık akış alanının akış hızını belirlerken ölçüm hatasını azaltmasına yardımcı olmak amacıyla bir YSA modeli tasarlanmıştır. Tasarlanan YSA modelinin yapısını ve ağırlık parametrelerini optimize etmek için GA kullanılmıştır. Önerilen GA-YSA modeli altı adet çok yollu ultrasonik debimetre yolları üzerinde test edilmiştir. Elde edilen deney sonuçları klasik gauss karesi ile modellenen YSA modeli ile karşılaştırılmıştır. Araştırmacıların önermiş oldukları GA-YSA modeli, daha az hata ile çok yollu ultrasonik

debimetrenin karmaşık akış alanının akış hızını daha doğru belirlemiştir (Hu ve ark., 2016).

Jaddi ve arkadaşları, dinamik YSA sınıflandırma algoritmasının modelini ve ağırlıklarının değerlerini yarasa algoritması kullanarak elde etmişlerdir. Buna ek olarak yarasa algoritmasının Modified Bat for Dynamic Neural Network (MBatDNN), Mean Bat for Dynamic Neural Network (MeanB atDNN), Piecewise Map for Dat Dynamic Neural Network (Piecewise-BatDNN), Logistic Map for Bat Dynamic Neural Network (LogisticBatDNN) ve Sinusoidal Map for Bat Dynamic Neural Network (SinBatDNN) farklı modellerini de aynı amaç için kullanmışlardır. Önerdikleri dinamik YSA sınıflandırma algoritması için yarasa algoritmasının ve diğer sınıflandırma modellerin testleri 8 veri kümesi üzerinde gerçekleştirilmiştir. Elde ettikleri deney sonuçlarına göre önerilen sınıflandırma modeli diğer sınıflandırma modellerinden daha iyi performans sergilemiştir (Jaddi ve ark., 2015).

Campos ve arkadaşları tarafından yapılan çalışmada, hybrid neuro-evolutive algorithm adında yeni bir YSA modeli geliştirilmiştir. Geliştirilen YSA modelinde, GA yardımı ile YSA sınıflandırma algoritmasının en uygun yapısı ve ağırlıklarının eğitimi gerçekleştirilmiştir. Geliştirilen melez neuro-evolutive algorithm sınıflandırma modeli Breast cancer detection, Iris flower, Heart disease, KDDCUP99 ve Breast cancer prediction veri kümeleri üzerinde dört farklı sınıflandırma modeli ile karşılaştırılmıştır. Karşılaştırma sonuçlarına göre, önerilen hybrid neuro-evolutive algorithm sınıflandırma modeli veri kümeleri üzerinde diğer sınıflandırma modellerinden daha üstün başarılar elde etmiştir. Bu üstün başarıların sebebinin karar alma sürecinde hybrid neuro-evolutive algorithm sınıflandırma modelinin artan etkinliği ve verimliliği olduğu söylenmiştir (de Campos ve ark., 2016).

Chatterjee ve arkadaşları, çok katlı RC yapılarının yapısal arıza tahmini için PSO'ya dayalı bir YSA tahmin modeli geliştirmişlerdir. Geliştirdikleri YSA sınıflandırma modelinin eğitimini PSO algoritması ile gerçekleştirmişlerdir. Amaçları çok katlı betonarme yapıların yapısal bozulmalarının tahmininde kullanılan ortalama-kare hatasının en aza indirilmesidir. Çok katlı RC bina yapısının gelecekteki başarısızlık ihtimalini tespit etmek için, deney çalışmalarında 150 katlı binadan oluşan RC yapısının bir veri tabanını kullanmışlardır. Yapısal tasarımdan on beş özellik çıkarılmış, bu özelliklerin dokuzu sınıflandırma işlemini gerçekleştirmek için seçilmiştir. Önerdikleri tahmin modelinin değerlendirilmesi için elde edilen sonuçlar standart YSA ve ileri beslemeli çok-katmanlı YSA sınıflandırıcı algoritmaları ile elde edilen sonuçlar

karşılaştırılmıştır. Deneysel sonuçlar, önerilen PSO eğitimli YSA tahmin modelinin diğer tahmin modellerinden daha iyi tahmin sonuçları verdiğini göstermiştir (Chatterjee ve ark., 2017).

Chau, YSA sınıflandırma algoritmasını PSO algoritması ile eğiterek bir tahmin modeli önermiştir. Önermiş olduğu tahmin modelini Hong Kong'daki Shing Mun Nehri'ndeki su seviyelerini tahmin etmek için kullanmıştır. Modelin değerlendirilmesi için veri kümesi olarak geçmişe dayalı su seviyelerinin değerleri kullanılmıştır. Değerlendirme sonuçlarına göre, PSO algoritması ile eğitilen tahmin modeli daha iyi performans sergilediği belirtilmiştir (Chau, 2006).

Faris ve arkadaşlarının yapmış olduğu çalışmada, multi-verse optimizasyon algoritması ile çok-katmanlı sinir ağı eğitilmiştir. Önerilen yeni eğitim yaklaşımı, UCI makine öğrenmesi ambarından seçilen dokuz farklı tıbbi veri kümesi üzerinde test edilmiştir. Elde edilen test sonuçları son zamanlarda sıkça kullanılan GA, PSO, diferansiyel evrim, ateş böceği algoritması ve guguk kuşu arama evrimsel meta-sezgisel optimizasyon algoritmalarının sonuçları ile karşılaştırılmıştır. Bu karşılaştırmaya ek olarak önerilen yeni eğitim yaklaşımı geri yayılım ve Levenberg-Marquardt gradyan tabanlı eğitim yöntemleriyle de karşılaştırılmıştır. Karşılaştırma sonuçlarına göre multi-verse optimizasyon algoritması ile eğitilen çok-katmanlı sinir ağı, veri kümelerinin eğitiminde yerel eğitimden kaçınma ve hızı yakınsama açısından diğer eğitim algoritmalarından daha iyi performans göstermiştir (Faris ve ark., 2016).

Liao ve arkadaşları tarafından yapılan çalışmada, YSA eğitimi için Nelder-Mead optimizasyon algoritması ile PSO melez olarak önerilmiştir. Önerilen melez eğitim algoritması diğer benzer melez PSO yöntemlerinden daha basittir, arama alanının keşfedilmesine daha fazla önem vermektedir ve hızlı bir arama yeteneğine sahiptir. Araştırmacılar önerdikleri melez PSO performansını bazı simülasyon problemleri üzerinde diğer benzer melez PSO algoritmaları ile karşılaştırmışlardır. Karşılaştırma sonuçlarına göre yazarların önerdikleri melez eğitim algoritması YSA'nın eğitiminde daha iyi bir performans sergilemiştir (Liao ve ark., 2015).

Kowalski ve Łukasik çalışmalarında, birçok uygulamada kullanılan meta-sezgisel krill sürü optimizasyon algoritmasını YSA'nın eğitimi için önermişlerdir. Önerdikleri yaklaşımın doğruluğunu UCI makine öğrenmesi veri ambarından alınan veri kümeleri üzerinde sınıflandırma hatası ve kare hataların toplamı kullanarak gerçekleştirmişlerdir. Yazarlar tarafından önerilen yaklaşımın performans değerlendirmesi farklı veri kümeleri kullanılarak Geriye Yayılım, GA ve Harmoni arama algoritmaları ile karşılaştırılarak

yapılmıştır. Elde edilen deney sonuçlarına göre önerilen eğitim algoritması diğer algoritmalardan hem yukarıda belirtilen ölçütler hem de YSA eğitimi için gereken zaman açısından umut verici bir performans sunduğu sonucuna varılmıştır (Kowalski ve Łukasik, 2016).

Bolaji ve arkadaşları tarafından yapılan çalışmada, YSA sınıflandırma algoritmasının eğitimi havai fişek algoritması ile gerçekleştirilmiştir. Önerilen havai fişek algoritması ile eğitimi gerçekleşen YSA sınıflandırma algoritmasının performansını ölçmek için UCI makine öğrenmesi veri tabanından farklı özelliklere sahip veri kümeleri üzerinde testler yapılmıştır. Önerilen havai fişek eğitimli YSA sınıflandırma algoritmasından elde ettikleri test sonuçları geriye yayılım, krill sürü algoritması, GA ve harmoni arama algoritmaları ile eğitilmiş YSA'nın sonuçları ile karşılaştırılmıştır. Deney sonuçlarının analizine göre yazarlar tarafından önerilen havai fişek algoritması ile eğitilen YSA sınıflandırma algoritması diğer sınıflandırma algoritmalarına göre sınıflandırma başarısını ve kare hataların toplamını daha uygun hale getirmiştir (Bolaji ve ark., 2018).

3. MATERYAL VE YÖNTEM

Bu bölümde, tez çalışmasında önerilen yaklaşımları değerlendirmek için kullanılan UCI makine öğrenmesi veri ambarından alınan veri kümelerinin özellikleri (Blake, 1998), veri madenciliğinde bir önişleme tekniği olarak kullanılan ayrıklaştırma yöntemleri, kural, olasılık ve ağırlık tabanlı sınıflandırma algoritmaları, sınıflandırma algoritmalarının ölçütleri ve son olarak PSO algoritması anlatılacaktır.

3.1. Veri Kümeleri

Bu tez çalışmasında makine öğrenmesi sınıflandırma algoritmaları için geliştirilen yaklaşımların performanslarını değerlendirmek için UCI makine öğrenmesi veri ambarından farklı özellik, sınıf ve örnek sayılarına sahip 22 adet veri kümesi kullanılmıştır. Bu veri kümeleri sürekli, kategorik ve karışık olmak üzere üç farklı veri türünde seçilmiştir. Seçilen veri kümeleri araştırmacılar tarafından veri önişleme, kümeleme ve sınıflandırma gibi birçok makine öğrenmesi algoritmaları ile kullanılmıştır. Kural ve olasılık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek için geliştirilen EF_Unique ayrıklaştırma yönteminin performansı Çizelge 3.1'de verilen veri kümeleri ile değerlendirilmiştir.

Çizelge 3.1. Geliştirilen EF_Unique ayrıklaştırma yönteminin performansının değerlendirildiği veri kümelerinin özellikleri

Veri kümesi	Özellik sayısı		Sınıf sayısı	Örnek sayısı
	Kategorik	Sürekli		
Iris	0	4	3	150
Wine	0	13	3	178
Glass Identification	0	9	6	214
New Thyroid	0	5	3	215
Heart	8	5	2	270
E.coli	1	7	8	336
Bupa	0	6	2	345
Australian Credit	8	6	2	690
Breast Cancer	0	9	2	699
Blood	0	4	2	748
Diabetes	0	8	2	768
Vehicle	0	18	4	846
German	13	7	2	1000
Wine Quality Red	0	11	11	1599
Spambase	0	57	2	4601
Magic Gamma Telescope	0	10	2	19020
Bank Marketing	9	7	2	45211

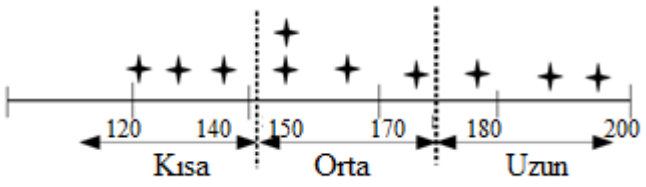
Ağırlık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek için geliştirilen MM-PSO meta-sezgisel algoritmasının performansı Çizelge 3.2'de verilen veri kümeleri ile değerlendirilmiştir.

Çizelge 3.2. Geliştirilen MM-PSO meta-sezgisel algoritmasının performansının değerlendirildiği veri kümelerinin özellikleri

Veri kümesi	Özellik sayısı		Sınıf sayısı	Örnek sayısı
	Kategorik	Sürekli		
Lymphography	18	0	4	148
Iris	0	4	3	150
Wine	0	13	3	178
Glass	0	9	6	214
Shuttle-landing	6	0	2	253
Ionosphere	0	33	2	351
Balance-scale	4	0	3	625
Breast cancer	0	9	2	699
Diabetes	0	8	2	768
Thyroid	0	21	3	7200

3.2. Ayrıklaştırma ve ayrıklaştırma yöntemleri

Ayrıklaştırma, sürekli değerleri ayrık değer uzayına eşleştirilme işlemidir. Aynı zamanda sürekli bir özniteliğin değer uzayını birbiriyle örtüşmeyen aralıklara bölerek değerleri gruplandırma işlemidir. Örneğin insan boyunun sürekli bir ölçü (örneğin, 170 cm) ile gösterildiğini farz edersek, ayrıklaştırma bu sürekli değeri uzun/kısa/orta gibi kategorik değerlere dönüştürme işlemidir. Bu şekilde insan boyu 3 kategoride toplanmaktadır. Örnek Şekil 3.1'de gösterilmektedir (Han ve ark., 2011).



Şekil 3.1. İnsan boyunun kategorik değerleri

Veri madenciliği sınıflandırma algoritmalarının birçoğu ayrık değerlerle daha iyi sınıflandırma performansı sergilerler ve etkili sınıflandırma modelleri oluşturabilir (Clark ve Niblett, 1989; Cohen, 1995). Sınıflandırma algoritmasının başarılı olabilmesi için eğitim verisinin, kullanılan sınıflandırma algoritmasına uygun bir şekilde hazırlanarak

verilmesi gerekmektedir. Kural ve olasılık tabanlı sınıflandırma algoritmaları için verilerin ayrık olması sınıflandırma başarısını artırmaktadır (Hacibeyoglu ve ark., 2011). Literatürde araştırmacılar tarafından farklı özelliklerde ayrıklaştırma yöntemleri önerilmiştir. Önerilen bu ayrıklaştırma yöntemleri özelliklerine göre aşağıdaki sınıflara ayrılabilir.

Denetimsiz ve denetimli ayrıklaştırma yöntemleri: Denetimsiz ayrıklaştırma yöntemleri sürekli verileri ayrık verilere dönüştürmek için sınıf bilgisinden yararlanmaz. EG, EF, FFD ve MVD en iyi bilinen denetimsiz ayrıklaştırma yöntemleridir. Denetimli ayrıklaştırma yöntemleri ise, sürekli özniteliklerin ayrıklaştırılması için sınıf bilgisinden yararlanır, yani verinin sınıf bilgisini kullanarak ayrıklaştırma işlemini gerçekleştirir. Hata tabanlı, entropi tabanlı ve istatistik tabanlı yöntemler denetimli ayrıklaştırma yöntemlerinin türleridir (Yan ve ark., 2014).

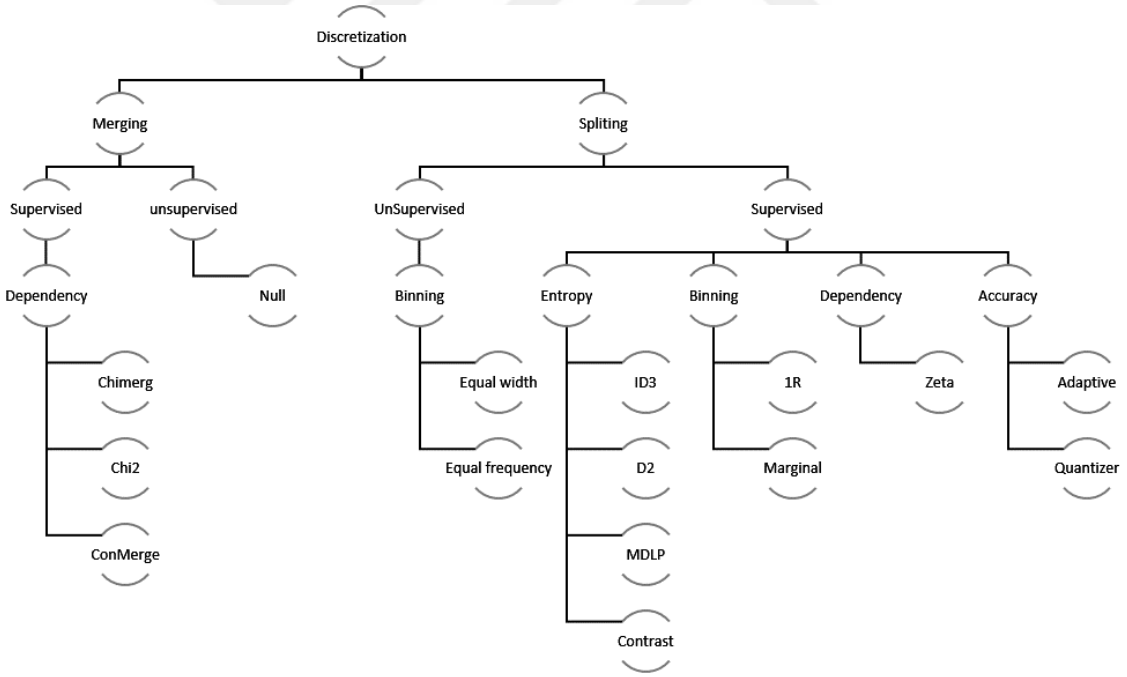
Ayrılmalı ve birleştirmeli ayrıklaştırma yöntemleri: Ayrılmalı ayrıklaştırma yöntemlerinde aralıklar boş bir aralık kümesi ile başlar ve tüm olası sınırlar arasından bir aralık oluşturur. Bu tür ayrıklaştırma yöntemlerinde, kullanıcı tarafından verilen bir eşik değerine ulaşana kadar en uygun aralıklar elde edilmeye çalışılır. Birleştirmeli ayrıklaştırma yöntemlerinde ise, ayrıklaştırma işlemi önceden tanımlanmış aralıklarla başlar daha sonra bu aralıklar birleştirilir. Bu yöntemde, birleştirme işlemi kullanıcı tarafından verilen bir eşik değere ulaşana kadar en uygun aralıkları elde etmeye devam eder (Garcia ve ark., 2013).

Ayrıklaştırma yöntemleri bir öğrenme modeliyle olan ilişkilerine göre statik ve dinamik olarak sınıflandırılabilir. Statik ayrıklaştırma yöntemleri, ayrıklaştırma işleminde herhangi bir öğrenme modelini kullanmadan ve öznitelikleri bir birinden bağımsız olarak ayrıklaştırır. Dinamik ayrıklaştırma yöntemleri ise ayrıklaştırma işlemini bir öğrenme modeli kullanarak gerçekleştirir ve tüm özellikler için aynı anda tüm olası aralıkları belirler. ET-ID3 ve ITFP, dinamik ayrıklaştırma yöntemlerinin örnekleridir (Dash ve ark., 2011).

Yerel ve küresel ayrıklaştırma yöntemleri: Yerel ayrıklaştırma yöntemlerinde veri kümesinin yerel bilgileri kullanılarak ayrıklaştırma işlemi gerçekleşir. Diğer taraftan, küresel ayrıklaştırma yöntemlerinde ise, ayrıklaştırma işleminde hem yerel bilgiler hem de dışarıdan gelen küresel bilgiler kullanılır. Bütün dinamik ayrıklaştırma yöntemleri yerel ayrıklaştırma sınıfına girmektedir. Yaygın olarak kullanılan yerel ayrıklaştırma yöntemlerine MDLP ve ET-ID3 örnek olarak verilebilir (Garcia ve ark., 2013).

Parametrik ve parametrik olmayan ayrıklaştırma yöntemleri: Ayrıklaştırma işleminin gerçekleştirilmesi için kullanıcıdan bir veya birden fazla parametre isteyen ayrıklaştırma yöntemleri parametrik, ayrıklaştırma işleminin gerçekleştirilmesi için kullanıcıdan hiçbir parametre istemeyen ayrıklaştırma yöntemleri ise parametrik olmayan yöntemler olarak ifade edilir.

Literatürde EG, EF ve ET-ID3 en sık kullanılan ayrıklaştırma yöntemleri olarak bilinir. EF ve EG, sürekli verileri ayrık veriye dönüştürürken sınıf bilgisi kullanmayan denetimsiz ayrıklaştırma yöntemleridir (Hacıbeyoğlu ve Ibrahim, 2016). ET-ID3, öğrenme sürecinin performansını iyileştirmek ve sonuçların anlaşılmasını geliştirmek için sınıf bilgilerini kullanan entropi tabanlı denetimli bir ayrıklaştırma yöntemidir (Zieliński ve Szmuc, 2005). Bu yöntemde, sınıf bilgisi kullanılarak birden fazla adımda en iyiye yakın aralık sayısı elde edilebilir (Witten ve ark., 2016). Günümüze kadar birçok ayrıklaştırma yöntemi geliştirilmiştir. Geliştirilen ayrıklaştırma yöntemlerinin hiyerarşik yapısı Şekil 3.2'de verilmiştir.



Şekil 3.2. Ayrıklaştırma yöntemlerinin hiyerarşik yapısı (Liu ve ark., 2002)

Şekil 3.2'de görüldüğü gibi birleştirmeli ve bölmeli olarak ikiye ayrılmaktadır ve genel olarak bu iki ayrımlar denetimli ve denetimsiz olarak sınıflandırılmıştır. Bu tez çalışmasında, denetimsiz ayrıklaştırma yöntemlerinden EG ve EF yöntemleri ve denetimli ayrıklaştırma yöntemlerinden ise ET-ID3 yöntemi alt bölümlerde açıklanmıştır.

3.2.1. Eşit Genişlikte ayrıklaştırma yöntemi

EG ayrıklaştırma yöntemi; denetimsiz, dinamik, küresel, bölünmeli, parametrik, doğrudan ve kolay uygulanabilir bir ayrıklaştırma yöntemidir (Hu ve ark., 2009). Bu ayrıklaştırma yönteminde, A dizisindeki elemanlar, kullanıcı tarafından belirlenen k sayısına göre eşit elemanlı parçalara bölünür. Bu parçaların genişliği Denklem 3.1'e göre ve aralıkların sınırları Denklem 3.2'ye göre hesaplanır (Rahman ve Islam, 2016),

$$Parça\ genişliği = (a_{max} - a_{min})/k \tag{3.1}$$

$$sınırlar = a_{min} + (i * parça\ genişliği);\ \ burada\ i = 1,2, \dots \dots, k-1 \tag{3.2}$$

Yukarıdaki Denklem 3.1'de sırasıyla a_{max} ve a_{min} *A* dizisindeki en büyük ve en küçük değerlerini temsil eder. Eşit genişlik ayrıklaştırma yönteminin algoritması Şekil 3.3'te detaylı olarak verilmiştir (Chlebus ve Nguyen, 1998; Hacıbeyoğlu ve Ibrahim, 2016).

Giriş: Dizideki sürekli öznitelik değerleri $\boldsymbol{A} = \{a_1, a_2, \dots, a_{n-1}, a_n\}$ ve aralık sayısı $\boldsymbol{k}$, $\boldsymbol{k} > 0$.
Adım 1: ***A*** diziyi küçükten büyüğe sırala daha sonra a_{max} ve a_{min} belirle,
Adım 2: Denklem 3.1'e göre aralıkların parça genişliğini hesapla,
Adım 3: Parça genişliğine göre parçaları oluştur,
Adım 4: Denklem 3.2'ye göre aralıkların sınırlarını belirle,
Adım 5: ***A*** dizisindeki sürekli değerler aralıklara göre ayrık değerlere dönüştürülür.
Çıkış: Ayrık değerli ***A*** dizisi.

Şekil 3.3. Eşit genişlik ayrıklaştırma yönteminin algoritması

Eşit genişlik ayrıklaştırma yöntemi bazı avantajlara ve dezavantajlara sahiptir. Avantaj olarak veriler üzerinde uygulanması basit ve kolaydır (Gupta ve ark., 2010). Fakat diğer yandan k sayısının tam olarak kaç olması gerektiğinin bilinememesi ve aykırı değerlere son derece hassas olması dezavantajlarıdır (Hacibeyoglu ve ark., 2011).

Eşit genişlik ayrıklaştırma yönteminin sürekli verilere uygulanması ve uygulama üzerinde eşit genişlik ayrıklaştırma yönteminin dezavantajı Örnek 3.1'de gösterilmektedir.

Örnek 3.1:

Giriş: Dizideki sürekli öznitelik değerleri $A = \{20, 30, 20, 10, 80, 100, 10, 20, 10\}$ ve aralık sayısı $k = 3$.

Adım 1: Sıralama işlemi $A = \{10, 10, 10, 20, 20, 20, 30, 80, 100\}$, $a_{max} = 100$ ve $a_{min} = 10$

Adım 2: $Parça\ genişliği = (a_{max} - a_{min})/k = (100 - 10)/3 = 30$

Adım 3: $Parça_1 = \{10, 10, 10, 20, 20, 20, 30\}$, $Parça_2 = \{\emptyset\}$, $Parça_3 = \{80, 100\}$

Adım 4: $Aralık_1 = [10{,}40)$, $Aralık_2 = [40{,}70)$, $Aralık_3 = [70{,}100]$

Adım 5: $A = \{[10, 40), [10, 40), [10, 40), [10, 40), [70, 100], [70, 100], [10, 40), [10, 40), [10, 40)\}$

Örnek 3.1'de görüldüğü gibi [40,70) aralığına hiçbir sayı atanmamış, dokuz değerin yedisi [10,40] aralığında ve dokuz değerin diğer ikisi ise [70,100] aralığına atanmaktadır. Böylelikle, EG ayrıklaştırma yöntemi dengesiz ve boş aralıklarla ayrıklaştırma işlemini sonuçlandırmıştır. Bu da EG ayrıklaştırma yönteminin aykırı değerlere duyarlı olduğunu göstermektedir.

3.2.2. Eşit frekans ayrıklaştırma yöntemi

EF ayrıklaştırma yöntemi; denetimsiz, dinamik, küresel, bölünmeli, parametrik, doğrudan ve kolay uygulanabilir bir ayrıklaştırma yöntemidir (Garcia ve ark., 2013). Bu ayrıklaştırma yönteminde, A dizisindeki eleman sayısı kullanıcı tarafından belirlenen k aralığına bölünür ve her bir parça yaklaşık olarak eşit sayıda eleman içerir. Bu ayrıklaştırma yöntemi, aykırı sürekli değerlerden etkilenmez (Rahman ve Islam, 2016). Öte yandan, bu ayrıklaştırma yönteminde en uygun k parça sayısını tahmin etmek zordur ve aynı değeri farklı aralıklara yerleştirebilir. Eşit frekans ayrıklaştırma yönteminin algoritması Şekil 3.4'te gösterilmektedir (Hu ve ark., 2009; Hacıbeyoğlu ve Ibrahim, 2016).

Giriş: Dizideki sürekli öznitelik değerleri $\boldsymbol{A} = \{a_1, a_2, \dots, a_{n-1}, a_n\}$ ve aralık sayısı $\boldsymbol{k}$, $\boldsymbol{k} > 0$.
Adım 1: $\boldsymbol{A}$ diziyi küçükten büyüğe sırala daha sonra a_{max} ve a_{min} belirle,
Adım 2: $\boldsymbol{A}$ dizideki eleman sayısını $\boldsymbol{k}$ aralık sayısına böl,
Adım 3: Her parçada bulunan eleman sayısına göre parçaları oluştur,
Adım 4: Geçerli parçanın maksimum değerinin ve bir sonraki parçanın minimum değerinin ortalama değerini hesaplayarak her bir aralığın sınırlarını belirle,
Adım 5: $\boldsymbol{A}$ dizisindeki sürekli değerler aralıklara göre ayrık değerlere dönüştürülür.
Çıkış: Ayrık değerli $\boldsymbol{A}$ dizisi.

Şekil 3.4. Eşit frekans ayrıklaştırma yönteminin algoritması

Eşit frekans ayrıklaştırma yöntemi sürekli veriler üzerinde basit ve kolay bir şekilde uygulanmaktadır (Gupta ve ark., 2010). Fakat diğer yandan k sayısının tam olarak kaç olması gerektiğinin bilinememesi ve benzer veya çok yakın değerleri iki farklı aralıklara atması dezavantajlarıdır (Hacibeyoglu ve ark., 2011).

Eşit frekans ayrıklaştırma yönteminin sürekli verilere uygulanması ve uygulama üzerinde eşit frekans ayrıklaştırma yönteminin dezavantajı Örnek 3.2'de gösterilmektedir.

Örnek 3.2:

Giriş: Dizideki sürekli öznitelik değerleri $A = \{20, 30, 20, 10, 80, 100, 10, 20, 10\}$, Aralık sayısı $k = 3$ ve dizideki eleman sayısı $n = 9$.

Adım 1: sıralama işlemi $A = \{10, 10, 10, 20, 20, 20, 30, 80, 100\}$

Adım 2: $Her\ parçadaki\ eleman\ sayısı = n/k = 9/3 = 3$

Adım 3: $\text{Parça}_1 = \{10, 10, 10\}$, $\text{Parça}_2 = \{20, 20, 20\}$, $\text{Parça}_3 = \{30, 80, 100\}$

Adım 4: $Aralık_1 = [10, (10 + 20)/2) = [10{,}15)$

$Aralık_2 = [15, (20 + 30)/2) = [15{,}25)$

$Aralık_3 = [25{,}100]$

Adım 5: $A = \{[15, 25), [25, 100], [15, 25), [10, 15), [25, 100], [25{,}100], [10, 15), [15, 25), [10, 15)\}$

Örnek 3.2'de görüldüğü gibi, tüm parçaların aynı sayıda elemanı vardır. Fakat bu yöntemin en büyük dezavantajı, eğer her hangi bir parçanın bir veya daha fazla değeri tekrarlanırsa, bu tekrarlanan değerler iki farklı parçaya atanabilir (Dash ve ark., 2011; Cebeci ve Yildiz, 2017).

3.2.3. Entropi tabanlı ID3 ayrıklaştırma yöntemi

ET-ID3 ayrıklaştırma yöntemi; denetimli, statik, yerel, ayrılmalı, artırmalı, parametriktir ve EG ve EF ayrıklaştırma yöntemlerinden daha karmaşıktır (Peng ve ark., 2009). Bu ayrıklaştırma yönteminde, belirli bir ölçütle birden fazla adımda aralık sayısı elde edilir (Witten ve ark., 2016). Başlangıçta, bu ayrıklaştırma yöntemi, özniteliğin tüm değerlerini içeren bir aralığı belirler ve ardından durma ölçütlerine ulaşana kadar bu aralığı alt parçalara tekrar tekrar böler. Durdurma ölçütleri, kullanıcı tarafından belirlenen aralık sayısı veya eşik değeridir. Algoritma kullanıcı tarafından belirlenen aralık sayısına ulaşıncaya kadar veya entropi kazancı kullanıcı tarafından belirlenen eşik değerinin altına düşene kadar özyinelemeli olarak çalışmaya devam eder. ET-ID3 entropi ve bilgi kazancı kullanarak aynı sınıfa karşılık gelen en uygun aralığı elde eder (Kohavi ve Sahami, 1996). Entropi, bilginin belirsizliğini ve düzensizliğini ifade etmektedir. ET-ID3'te, entropi Denklem 3.3'e göre hesaplanır (Han ve ark., 2011).

$$H(S) = -\sum_{i=1}^{n} P_i \ log_2(P_i) \tag{3.3}$$

Denklem 3.3'te S veri kümesi, n veri kümesindeki sınıfların sayısıdır. P_i veri kümesindeki i'nci sınıfının olasılığıdır ve H(S) veri kümesinin entropisini ifade etmektedir. Bilgi kazancı ise, kesme noktalarının verimliliğini ölçer ve Denklem 3.4'e göre hesaplanır.

$$Bilgi\ kazancı\ \ (S,T) = H(S) - \frac{|S_{left}|}{|S|} H(S_{left}) - \frac{|S_{right}|}{|S|} H(S_{right}) \tag{3.4}$$

T kesme noktası ve S_{left} ve S_{right}, T'ye göre S alt kümesinin sırasıyla sol ve sağ taraf aralıklarıdır. En yüksek bilgi kazancına sahip olan kesme noktası, en iyi kesme noktası olarak seçilir (Hacıbeyoğlu ve Ibrahim, 2016). ET-ID3 ayrıklaştırma yönteminin algoritması Şekil 3.5'te verilmektedir.

Giriş: Dizideki sürekli öznitelik değerleri $\boldsymbol{A} = \{a_1, a_2, \dots, a_{n-1}, a_n\}$ ve özniteliğin sınıf bilgisi $\boldsymbol{C} = \{c_1, c_2, \dots, c_{n-1}, c_n\}$, durdurma ölçütü.
Adım 1: $\boldsymbol{A}$ diziyi küçükten büyüğe sırala,
Adım 2: Veri kümesinin belirsizliğini hesapla,
Adım 3: Her kesme noktası için,
- Belirsizlik ve bilgi kazancı hesapla,

Adım 4: En yüksek bilgi kazancına sahip olan kesme noktasına göre aralıkları belirle,
Adım 5: Adım 3 ve 4 özyinelemeli olarak aşağıda durma ölçütlerine ulaşana kadar,
- Kullanıcı tarafından verilen $\boldsymbol{k}$ sayısına ulaşana kadar
- Kullanıcı tarafından girilen eşik değerine varana kadar

Adım 6: $\boldsymbol{A}$ dizisindeki sürekli değerler aralıklara göre ayrık değerlere dönüştürülür.
Çıkış: Ayrık değerli $\boldsymbol{A}$ dizisi.

Şekil 3.5. Entropi tabanlı ID3 ayrıklaştırma yönteminin algoritması

ET-ID3 ayrıklaştırma yönteminin avantajı, verideki bilgi kazancına göre yaklaşık olarak en iyi aralıkları elde etmesidir. Fakat belirsizliği yüksek olan veri kümelerinde ET-ID3 ayrıklaştırma yöntemi çok başarılı değildir. ET-ID3 ayrıklaştırma yönteminin zaman karmaşıklığı veri kümesinin büyüklüğüne bağlıdır, dolayısıyla ET-ID3 ayrıklaştırma yöntemi büyük veri kümelerinde daha fazla çalışma süresi ile ayrıklaştırma işlemini gerçekleştirmektedir. ET-ID3 ayrıklaştırma yönteminin uygulaması Örnek 3.3'te verilmektedir.

Örnek 3.3:

Giriş: Dizideki sürekli veriler $A = \{20, 30, 20, 10, 80, 100, 10, 20, 10\}$ ve sınıf bilgisi $C = \{F, F, F, F, T, T, \mathrm{F}, F, F\}$, durdurma ölçütü $k = 2$.

Adım 1: Sıralama işlemi $A = \{10, 10, 10, 20, 20, 20, 30, 80, 100\}$,

Adım 2: A dizisinin belirsizliğini hesapla $H(S) = 0.7642$

Adım 3: kesme noktaları: 20, 30

- 20 noktası için $bilgi\ kazancı\ \ (S, 20) = 0.4582$
- 30 noktası için b$ilgi\ kazancı\ \ (S, 30) = 0.7642$

Adım 4: Yüksek bilgi kazancına sahip olan noktayı seç = 30.

30 $noktası\ için\ Aralık_1 = [10{,}30]\ ve\ Aralık_2 = (30{,}100]$

Adım 5: Durdurma ölçütü $k = 2$

Adım 6: $A = \{[10, 30], [10, 30], [10, 30], [10, 30], (30{,}100], (30{,}100], [10, 30], [10, 30], [10, 30]\}$

Örnek 3.3'te ayrıklaştırma işlemi ET-ID3 ayrıklaştırma yöntemi tarafından verilerin sınıf bilgilerine göre yapılmıştır. Sınıf bilgilerine göre yapılan ayrıklaştırma işlemi, genelde entropi tabanında çalışan sınıflandırma algoritmalarında daha iyi bir sınıflandırma başarısı sağlamaktadır.

3.3. Sınıflandırma ve Sınıflandırma Algoritmaları

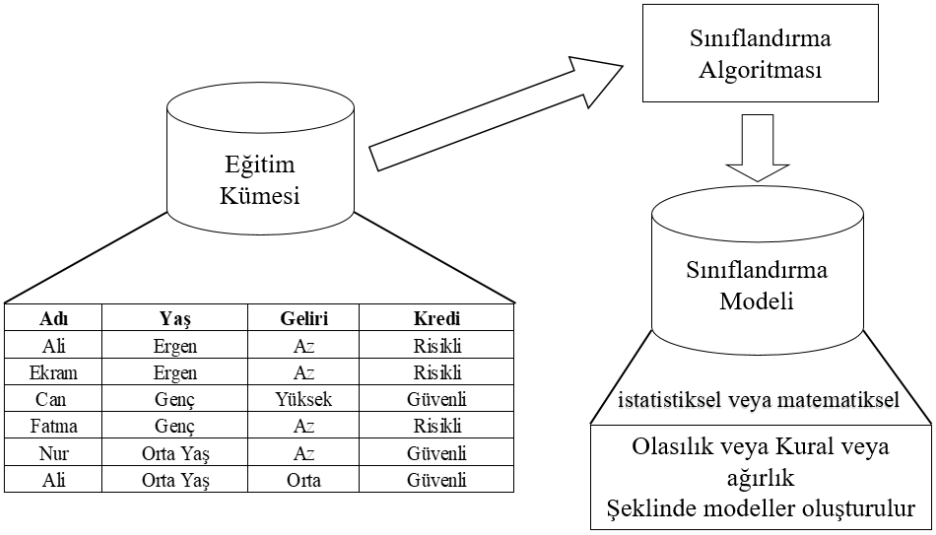
Veri tabanları keşfedilmemiş bilgiler içerir ve bu bilgiler kullanılarak zeki bir karar verme sistemi oluşturmak mümkündür (Witten ve ark., 2016). Karar destek sistemleri genelde sınıflandırma ve tahmin algoritmalarına dayanmaktadır. Sınıflandırma ve tahmin; önemli veri sınıflarını tanımlamak, veriden modeller çıkarmak veya gelecekteki veri eğilimlerini tahmin etmek için kullanılabilen iki veri analiz biçimidir. Bu tür analizler, bizim geniş kapsamlı verileri daha iyi anlamamıza yardımcı olabilir. Sınıflandırma modelleri kategorik etiketleri tahmin etmektedir. Tahmin modelleri ise sürekli değerli fonksiyonları belirtir. Örneğin, banka kredi başvurularını güvenli veya riskli olarak sınıflandırmak için bir sınıflandırma modeli oluşturabilir ve oluşturulan sınıflandırma modeli yeni başvurular için güvenli veya riskli olarak kategorik şeklinde bir çıkış üretebilir. Banka müşterilerinin gelirleri ve meslekleri göz önünde bulundurulduğunda, banka müşterilerin günlük veya aylık harcamalarını tahmin etmek için bir tahmin modeli oluşturabilir (Witten ve ark., 2016). Araştırmacılar tarafından makine öğrenimi, tanıma, teşhis ve istatistik konularında kullanılan birçok sınıflandırma ve tahmin algoritmaları geliştirilmiştir. Bu algoritmalar veri madenciliği araştırmalarında ve büyük verilerden desen, anlam ve bilgi çıkarmada kullanılmaktadır.

Günümüzde sınıflandırma algoritmaları çok önemli işlerde kullanılmaktadırlar. Örneğin, bir firmanın pazarlama müdürü, geçmiş alışverişlerine göre belirli bir profili olan bir müşterinin yeni bir elektronik cihazı satın alıp almayacağını tahmin etmeye yardımcı olması için veri analizine ihtiyaç duyabilir. Bir tıbbi araştırmacı, bir meme kanser hastasının alması gereken üç tedaviden hangisini alması gerektiğini tahmin etmek için meme kanserinin verilerini analiz edebilir. Bu örneklerin her birinde, veri analizi işlemi sınıflandırma algoritmaları ile yapılmaktadır. Sınıflandırma modeli pazarlama verileri için "evet" veya "hayır" ve meme kanseri için "tedavi A", "tedavi B" veya "tedavi C" gibi sınıf etiketlerini tahmin etmek için inşa edilmiştir.

Sınıflandırma modelinin yanında sayısal tahmin de birçok alanda kullanılır ve önemli bir konudur. Örneğin, meme kanseri hastalığının tedavisinde kullanılacak ilacın

doz miktarı sayısal bir veridir ve hastanın daha iyi bir şekilde iyileşmesini sağlayabilir. Sınıflandırma algoritmaları kullanılarak kategorik bir sınıf etiketi tahmin edilebileceği gibi sayısal bir değerde tahmin edilebilir. Regresyon analizi, sayısal tahmin için en sık kullanılan istatistiksel bir metodolojidir. Dolayısıyla iki terim sıklıkla eşanlamlı olarak kullanılır.

Kredi başvurusu için veri sınıflandırma işlemi, iki aşamadan oluşmaktadır. İlk aşamada, önceden belirlenmiş sınıf etiketli eğitim veri kümesi kullanılarak eğitim işlemini yapacak bir sınıflandırma modeli oluşturulur. Bu sınıflandırma modeli sınıf etiketli veri kümesinden sınıf ilişkili eğitim kümesini analiz ederek veya öğrenerek bir öğrenme veya eğitim aşamasından geçer. Her eğitim örneği sınıf etiketi ile etiketlendiğinden dolayı bu adım aynı zamanda denetimli öğrenme olarak da bilinir. Kredi başvurusu için eğitim veri kümesinden oluşturulan sınıflandırma modeli aşağıda Şekil 3.6'da gösterilmiştir.

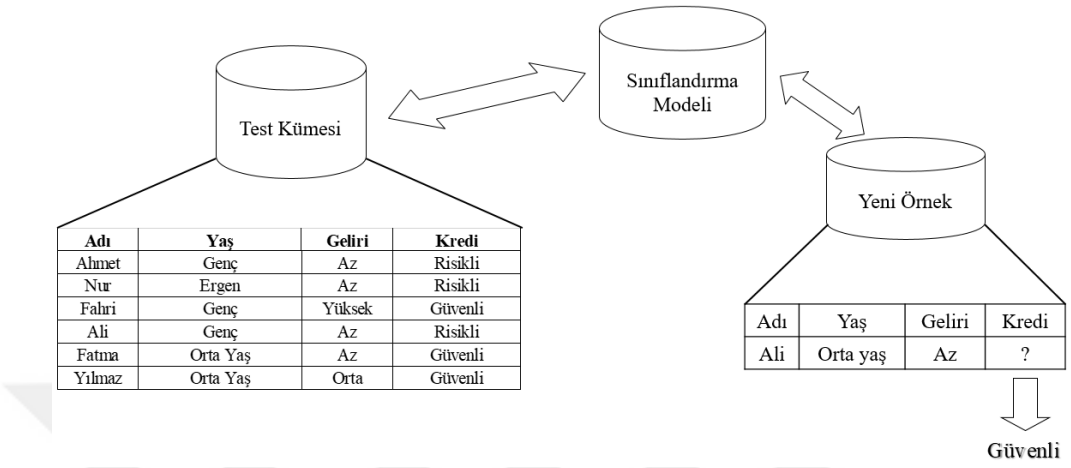


Adı	Yaş	Geliri	Kredi
Ali	Ergen	Az	Risikli
Ekram	Ergen	Az	Risikli
Can	Genç	Yüksek	Güvenli
Fatma	Genç	Az	Risikli
Nur	Orta Yaş	Az	Güvenli
Ali	Orta Yaş	Orta	Güvenli

Şekil 3.6. Kredi başvuru verileri için sınıflandırma modeli oluşturma

Şekilde görülen veri kümesi $V = \{I_i\}_{i=1}^{m}$ şeklinde gösterilir. Burada $I_1, I_2, \ldots I_m$ veri kümesindeki örnekleri temsil eder ve her örnek $I_j = \{A, C\}$ şeklinde temsil edilir. Burada $A = \{a_j\}_{j=1}^{n}$ öznitelik kümesini ve $C = \{c_k\}_{k=1}^{d}$ sınıf kümesini temsil eder. Öznitelikler ve sınıf etiketleri ayrık değerli veya sürekli değerli olabilir. Eğitim kümesini oluşturan örnekler eğitim örnekleri olarak bilinir ve bu eğitim örnekleri sınıflandırma modelini eğitmek için kullanılır.

İkinci aşamada ise, eğitim kümesi ile oluşturulan sınıflandırma modeli test veri kümesi veya yeni gelen örnek ile test edilmektedir. Kredi başvurusu için oluşturulan sınıflandırma modelinin testi aşağıda Şekil 3.7'de gösterilmiştir.



Adı	Yaş	Geliri	Kredi
Ahmet	Genç	Az	Risikli
Nur	Ergen	Az	Risikli
Fahri	Genç	Yüksek	Güvenli
Ali	Genç	Az	Risikli
Fatma	Orta Yaş	Az	Güvenli
Yılmaz	Orta Yaş	Orta	Güvenli

Adı	Yaş	Geliri	Kredi
Ali	Orta yaş	Az	?

Şekil 3.7. Kredi başvuru verileri için sınıflandırma modeli

Şekil 3.7'de görüldüğü gibi, eğitim verisinden oluşturulan model sınıflandırma ve tahmin işlemleri için kullanılır. Eğitilmiş sınıflandırma modelinden beklenen eğitim kümesinde olmayan ve sınıf değeri belli olmayan örnekleri yüksek oranda doğru sınıflandırmasıdır. Sınıflandırma modelinin başarısını ölçmek için eğitim kümesini kullanacak olursak muhtemelen çok iyi bir başarı elde edebiliriz. Çünkü sınıflandırma modeli oluşturulurken bu eğitim örnekleri üzerinde eğitilmiştir. Bu nedenle, sınıflandırma modelinin başarısını doğru bir şekilde ölçmek için test örnekleri ve bu örneklerle ilişkili sınıf etiketlerinden oluşan bir test kümesi kullanılmalıdır. Bu test örnekleri, genel veri kümesinden rasgele seçilir ve bu test örnekleri eğitim örneklerinden bağımsız olmalıdır. Yani sınıflandırma modelinin eğitim aşamasında kullanılmazlar. Sınıflandırma modelinin belli bir test kümesindeki başarısı, sınıflandırma modeli tarafından doğru sınıflandırılan test kümesi örneklerinin yüzdesidir (örneğin, %90). Sınıflandırma modeli tarafından tahmin edilen test örneklerinin sınıfları, test örneklerin gerçek sınıf etiketleri ile karşılaştırılır. Sınıflandırma modelinin başarısını hesaplamak için kullanılabilecek yöntemler Sınıflandırma Algoritmaların Performansını Değerlendirme Ölçütleri bölümünde açıklanmaktadır.

Sayısal tahmin modeli, sınıflandırma modeline benzer ve iki aşamadan oluşmaktadır. Fakat bu tahmin modelinde tahmin değerleri kategorik olarak değil sayısal

tahmin olduğundan dolayı sınıf etiketi terminolojisini kaybetmektedir. Örneğin, yukarıda vermiş olduğumuz örnekte sınıflandırma modeli bir müşterinin güvenli olduğuna karar verdiğini varsayalım. Tahmin modeli ise bu müşteriye kredi miktarı ile ilgili bir tahmini sayısal değer üretir. Ayrıca sınıflandırma ve tahmin kendi modellerini oluşturmak için kullanılan yöntemlerde de farklılık gösterir. Sınıflandırmada olduğu gibi, bir tahmini oluşturmak için kullanılan eğitim kümesi tahmin modelinin başarısını değerlendirmek için kullanılmamalıdır. Bunun yerine bağımsız bir test kümesi kullanılmalıdır. Bir tahmin modelinin başarısı, tahmin modelinden tahmin edilen değer ile test örneğinin her biri için bilinen gerçek değeri arasındaki farka dayalı bir hata hesaplanarak tahmin edilir. Ortalama hata, ortalama yüzde hata, ortalama mutlak hata, ortalama kare hata, kök ortalama kare hata ve ortalama mutlak yüzde hata gibi çeşitli tahmin hata ölçütleri bulunmaktadır.

Sınıflandırma problemlerinde kullanılan sınıflandırma veri kümelerinin örnek, öznitelik ve sınıf etiketlerinin sayıları artıkça veri analizi işlemi de oldukça zorlaşır. Buna ek olarak sınıflandırma işlemi ses tanıma, el yazısı tanıma, hastalık teşhisi ve belge sınıflandırması vb. gibi alanlarda kullanılabilir (Han ve ark., 2011). Dolayısıyla, araştırmacılar tarafından daha karmaşık veriler üzerinde analiz yeteneğine sahip olan kural tabanlı (ID3, C4.5, CART, REPPER, vb.), olasılık tabanlı (NB, vb.) ve ağırlık tabanlı (YSA, Logistic, vb.) gibi sınıflandırma algoritmaları tasarlanmıştır (Berry ve Linoff, 1997; Wu ve ark., 2008). Tasarlanan makine öğrenmesi kural tabanlı ve olasılık tabanlı sınıflandırma algoritmaları ayrık ve kategorik veri kümeleri üzerinde çok iyi bir performansa sahip sınıflandırma modeli oluşturabilir. Ancak gerçek dünyadaki uygulamalarından elde edilen veriler sürekli veriler olduğundan dolayı kural ve olasılık tabanlı sınıflandırma algoritmalarında iyi bir performansa sahip sınıflandırma modelleri oluşturmayabilir. Dolayısıyla uygulamalardan elde edilen sürekli verilerin kural ve olasılık tabanlı sınıflandırma algoritmalarına verilmeden önce ayrıklaştırma yöntemleri ile ayrıklaştırılmaları gerekmektedir.

Makine öğrenmesi ağırlık tabanlı sınıflandırma algoritmalarında ise oluşturulan sınıflandırma modelinin başarısı hem veri kümesinin hazırlanması hem de ağırlık tabanlı sınıflandırma algoritmasında kullanılan ağırlık parametrelerine bağlıdır. Dolayısıyla ağırlık tabanlı sınıflandırma algoritmasında iyi bir performans elde etmek için veri kümesinin iyi bir şekilde hazırlanması ve iyi bir eğitim (ağırlık güncelleme) algoritmasının seçilmesi gerekmektedir.

Bu bölümün bir alt bölümlerinde makine öğrenmesinin öne çıkan ve birçok sınıflandırma modellerinde sıkça kullanılan kural tabanlı, olasılık tabanlı ve ağırlık tabanlı sınıflandırma algoritmaları ayrıntılı olarak anlatılmıştır.

3.3.1. Kural tabanlı sınıflandırma algoritmaları

Kural tabanlı sınıflandırma algoritmaları, sınıf tahmini için eğitim örneklerinden oluşturulan *IF-THEN* sınıflandırma kurallarını kullanan sınıflandırma algoritmalarıdır. Kural tabanlı sınıflandırma algoritmaları tipik olarak aşağıdaki bileşenlerden oluşur (Clark ve Niblett, 1989):

- Kural İndüksiyon Algoritması: Eğitim verilerinden ilgili *IF-THEN* sınıflandırma kurallarının çıkarılması sürecini ifade eder. Kural çıkarma işlemi doğrudan sıralı örtü algoritması (Sequential Covering Algorithms) veya dolaylı olarak veri madenciliğinin karar ağacı (decision tree) veya birleştirme kuralı madenciliği (Association Rule Mining) gibi veri madenciliği algoritmaları ile gerçekleşebilir.
- Kural Sıralaması Ölçütleri: Sınıflandırma kurallarının doğru tahmin değerini ölçerek bu kuralları sıralayarak sınıflandırma modeline olan faydasını sağlamaktadır. Kural sıralama ölçütleri, gereksiz kuralların belirlenmesini sağlamak ve kural tabanlı sınıflandırma modelinin verimliliğini artırmak için kural indüksiyon algoritmasında sıklıkla kullanılır.

Kural tabanlı sınıflandırma algoritmalarında sınıflandırma kurallarının basit, doğru ve verimli olması sınıflandırma modellerinin performansını iyileştirmektedir ve aşağıdaki avantajları sağlamaktadır (Yin ve Han, 2003).

- Kurallar bilgi sunumu için çok doğaldır, çünkü insanlar sınıflandırma kurallarını kolayca anlayabilir ve yorumlayabilir.
- Sınıflandırma sonuçlarını açıklamak kolaydır, yeni gelen örneğin giriş değerlerini sınıflandırmak için kurallar tablosundan yeni gelen örnek için hangi kuralların kullanıldığı belirlenir.

- Kural tabanlı sınıflandırma modelleri, alan bilgisine dayanarak alan uzmanları tarafından yeni kurallar eklenerek kolayca geliştirilebilir. Dolayısıyla kural tabanlı sınıflandırma modelleri birçok uzman sistemde başarıyla uygulanmıştır.
- Kurallar öğrenilip bir kural veri tabanına depolandıktan sonra, bu kurallar yeni gelen örnekleri hızlı bir şekilde sınıflandırmak için kullanılır ve ilgili kurallar verimli bir şekilde bulunabilir.
- Kural tabanlı sınıflandırma algoritmaları, diğer sınıflandırma algoritmalarıyla rekabetçidir ve çoğu durumda diğer sınıflandırma algoritmalarından daha iyi performans sergilerler.

Kısaca kural tabanlı sınıflandırma algoritmalarının sınıflandırma kuralları veri veya bilgiyi çok basit, insanın anlayabileceği formatta ve etkili bir şekilde temsil edebilir. Sınıflandırma kuralları, mantık formunda *IF-THEN* ifadeleri kullanılarak temsil edilir. Örneğin, yaygın olarak kullanılan bir kural aşağıdaki gibi ifade edilebilir:

$$IF \quad şart \quad THEN \quad sonuç$$

Yukarıdaki kural analiz edildiğinde şart veya şartlar veri kümesinin özniteliklerinin değerlerini, sonuç ise veri kümesinin sınıf etiketini belirlemektedir. Kuralın şartı sağlandığı takdirde, bir sonuç elde edilebilir anlamına gelir. Bu şartlar bir veya birden fazla olabilir. Her hangi bir kuralın şartları birden fazla olursa bu şartlar mantıksal operatörler ile birbirine bağlanır (örneğin, hava = “güneşli” ve nem = “orta” ve sıcaklık = “orta” şartları sağlandığında tenis oynanabilir, oyun = “Evet”)

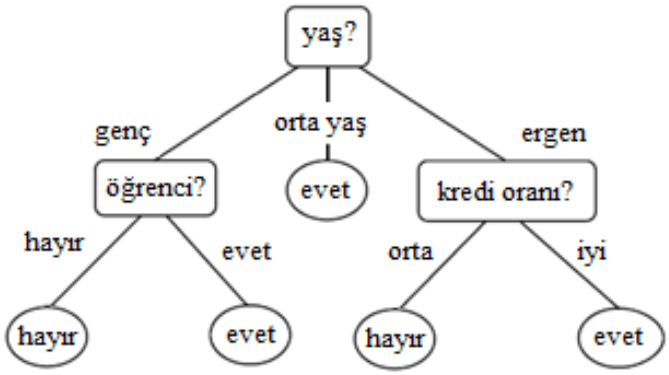
Veri kümesinden kuralları otomatik olarak oluşturmak için birçok makine öğrenmesi ve veri madenciliği algoritmaları önerilmiştir. Bilgisayar bilimi alanında, kural tabanlı sınıflandırma sistemleri bilgiyi depolamak ve mantık çıkarımı yapmak için etkili bir yol olarak yaygın bir şekilde kullanılmaktadır. Özellikle, kurallara dayalı sınıflandırma sistemleri, havacılık, arıza teşhisi, hastalık teşhisi, dolandırıcılık teşhisi vb. gibi çeşitli uzman sistemlerde yaygın olarak uygulanmaktadır (Li ve ark., 2001). AQ, CN2, REPPER, CART ve C4.5 makine öğrenmesi kural tabanlı sınıflandırma algoritmalarının türleridir. Teşhis ve tahmin problemlerinde sıklıkla ve çok başarılı olarak kullanılan C4.5 karar ağacı kural tabanlı sınıflandırma algoritması aşağıda detaylı olarak anlatılmıştır.

3.3.1.2. Karar ağaçları sınıflandırma algoritması

1970'lerin sonları ve 1980'lerin başında, makine öğrenmesi konularında araştırmacı olan J. Ross Quinlan, ID3 (Iterative Dichotomiser) olarak bilinen bir karar ağacı algoritması önermiştir. Bu algoritma, E. B. Hunt, J. Marin ve P. T. Stone tarafından daha önceki çalışmalarda tanımlanan öğrenme sistemlerinin geliştirilmiş halidir. Quinlan, daha sonra, denetimli öğrenme algoritmaları içerisinde sıkça kullanılacak bir algoritma olan C4.5 (ID3'ün benzeri) algoritmasını sunmuştur (Quinlan, 2014). C4.5 karar ağacı sınıflandırma algoritmasını anlatmadan, aşağıda önce karar ağaçlarının yapısını incelenecektir.

Karar ağacı, sınıf etiketli eğitim kümelerinden ağaç yapısı şeklinde sınıflandırma kuralları elde eden bir öğrenme algoritmasıdır. Karar ağacı algoritmaları, ağaç yapısını eğitim kümelerindeki özniteliklerin belirsizliğine göre oluşturmaktadır. Oluşturmuş olduğu karar ağacına göre yeni gelen örneğin sınıf bilgisini tahmin eder.

Bir karar ağacının yapısı genel olarak düğümlerden (yapraklardan) ve dallardan oluşmaktadır. Her iç düğüm (yaprak düğümü) sınıflandırma kuralının bağımsız özniteliği gösterir ve her bir terminal düğüm yani son yaprak bağımsız özniteliğin sınıf etiketini gösterir. Genelde karar ağacı yapısında her iç düğüm dikdörtgenler şeklinde ve terminal düğüm ise yuvarlak şeklinde temsil edilir. Ağacın en üstteki düğümü ise kök düğümü olarak bilinir. Şekil 3.8'de bir müşterinin bilgisayarı alıp alamayacağı ile ilgili bir karar ağacı yapısı örnek olarak verilmiştir. Her iç düğüm bir özniteliği temsil eder ve her yaprak düğüm ise bir sınıfı temsil eder (ya bilgisayar alır: evet ya da bilgisayar alamaz = hayır).



Şekil 3.8. Karar ağacı yapısı

Bir müşterinin bilgisayar satın alıp almayacağını temsil eden, tipik bir karar ağacı Şekil 3.8'de gösterilmektedir. Bu şekilde dikdörtgenlerle temsil edilen iç düğümler müşterinin özniteliklerini ve düğümlerden çıkan dallar ise özniteliğin değerini göstermektedir. Yuvarlaklarla temsil edilen terminal düğümleri ise müşterinin bilgisayar alıp almadığını yani Evet veya Hayır bilgisini göstermektedir. Bazı karar ağacı algoritmalarında her iç düğüm sadece tam iki düğüme dallanır, bu tür karar ağaçları ikili ağaçlar olarak bilinir.

Karar ağaçlarının çalışma mantığı, sınıf etiketi bilinmeyen bir X örneği verildiğinde, X örneğinin öznitelik değerleri karar ağacı ile karşılaştırılır ve son olarak X örneğinin sınıfını tahmin etmek için kök düğümünden terminal düğüm noktasına kadar bir yol izlenir. Dolayısıyla karar ağacındaki izlenen yol kolayca bir sınıflandırma kuralına dönüştürülebilir, bir karar ağacı bir veya birden fazla sınıflandırma kuralları içerebilir. Karar ağacı sınıflandırma algoritmasının ağaç yapısını oluşturmak için eğitim kümesi dışında herhangi bir bilgiye veya parametreye ihtiyaç duyulmaz ve bu nedenle veri kümelerinde bilgi keşfi için çok uygundur ve kullanışlıdır.

Karar ağaçları yüksek boyutlu veri kümelerini işleyebilir ve elde edilen sınıflandırma kurallarını ağaç yapısında gösterilmesi genellikle araştırmacılar tarafından kolaylıkla anlaşılır. Karar ağaçlarının öğrenme aşaması ve sınıflandırma işlemi basit ve hızlıdır. Genel olarak, karar ağacı sınıflandırma algoritmalarının sınıflandırma başarıları yüksektir ve tıp, imalat ve üretim, finansal analiz ve mühendislik alanları gibi birçok alanda kullanılmaya devam edilmektedir. Ayrıca birçok ticari kural çıkarımı sisteminin temelini oluşturmaktadır. Bunlarla birlikte, karar ağacı sınıflandırma algoritmalarının performansı eğitim veri kümesine bağlı olabilir.

C4.5 sınıflandırma algoritması, sınıflandırma işlemi için öznitelik seçimi için bilgi kazancını kullanır. Bilgi kazancı, Claude Shannon'un bilgi teorisi üzerinde yaptığı çalışmaya dayanmaktadır. Diyelim ki, sınıf etiketli D eğitim kümesi ve bu eğitim kümesinin öznitelikleri olsun, $C_i = \{c_1, c_2, \dots, c_m\}$ ise öznitelik değerlerinden oluşan bağımsız örneklerin sınıf etiketleri olsun. D eğitim kümesindeki C_i sınıf etiketine ait örnekler $|C_i, D|$ ile temsil edilmektedir ve sırasıyla D veri kümesindeki ve C_i sınıfındaki örnek sayıları $|D|$ ve $|C_i, D|$ ile temsil edilir. D veri kümesindeki C_i sınıfına ait tüm öznitelikler için bilgi kazancı hesaplanır. En yüksek bilgi kazancına sahip olan öznitelik kök özniteliği olarak seçilir. Bu öznitelik, sonuç bölümlerindeki grupları sınıflandırmak için gerekli olan bilgiyi en aza indirir. Böyle bir yaklaşım belirli bir örneği sınıflandırmak için gereken test sayısını en aza indirir ve basit bir sınıflandırma ağacın oluşturulmasını

sağlar. D veri kümesinde bir kümeyi sınıflandırmak için gereken entropi (belirsizlik) H Denklem 3.5 ile hesaplanmaktadır (Quinlan, 2014).

$$H(D) = -\sum_{i=1}^{m} P_i \log_2(P_i) \tag{3.5}$$

Burada P_i, D veri kümesindeki C_i sınıf bilgilerine ait olan grup örnekleridir, $|C_i, D|/D$ formülü ile hesaplanır. Bilgiler bitler şeklinde kodlandığı için logaritma iki tabanında olmaktadır. Entropi (D), sadece D veri kümesinde bir grubun sınıf etiketini tanımlamak için gerekli olan ortalama bilgidir. Bu noktada elde edilen bilgiler, her sınıfın örneklerinin oranlarına dayanır. Bu işlemleri basit bir örnek üzerinde açıklamak istersek, 10 elemanlı bir veri kümesi için $\mathrm{S} = \{\mathrm{E}, \mathrm{E}, \mathrm{F}, \mathrm{F}, \mathrm{E}, \mathrm{F}, \mathrm{E}, \mathrm{F}, \mathrm{F}, \mathrm{F}\}$ önce olasılıklar aşağıdaki gibi hesaplanır.

E için olasılık $(P_E) = \frac{4}{10} = 0.25$ ve F için olasılık $(P_F) = \frac{6}{10} = 0.75$

Bu durumda olasılık = (E olasılık, F olasılık) = (0.25, 0.75) şeklinde yazılır ve bu sonuç değerlerine bağlı olarak E ve F olasılıkları için toplam belirsizlik entropi aşağıdaki gibi elde edilir.

$$H(S) = -(P_E \log_2(P_E) + P_F \log_2(P_F))$$

$$= -(0.25 \log_2(0.25) + 0.75 \log_2(0.75)) = 0.81128$$

Veri kümesinde her hangi bir özniteliğin entropi değeri sıfır olması durumunda, özniteliğin tüm örneklerinin aynı sınıfta olduğu anlamına gelir.

Son olarak dallanma için özniteliklerin seçim işlemi, özniteliklerin bilgi kazançlarına göre belirlenir. Bilgi kazancı veri kümesindeki ayırt edici özniteliği belirlemek için kullanılır ve en yüksek bilgi kazancına sahip olan öznitelik, en iyi bölünmeyi sağlayacak öznitelik olarak belirlenir ve bu öznitelik ağacın kökünde yer alır. Veri kümesindeki her özniteliğin bilgi kazancı Denklem 3.6 ile hesaplanır (Quinlan, 2014).

$$\text{Bilgi kazancı (T, X)} = H(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i) \quad (3.6)$$

Burada, $H(T)$ veri kümesini X özniteliğine göre belirsizliğini, n veri kümesindeki sınıf bilgilerinin sayısını, T veri kümesinde örnek sayısını, T_i ise i'inci sınıfın örnek kümesini ve son olarak $H(T_i)$ i'inci sınıfına ait olan örneklerin belirsizliğini temsil etmektedir. C4.5 karar ağacı sınıflandırma algoritmasının sözde kodu Şekil 3.9'da verilmektedir (Han ve ark., 2011).

Giriş: ***D*** Eğitim veri kümesi.
Adım 1: Sonlandırma ölçütünü kontrol et,
Adım 2: Tüm öznitelikler için bilgi kazancı ölçütü hesapla,
Adım 3: Bilgi kazancı ölçütlerine göre en yüksek bilgi kazancına sahip olan özniteliği seç,
Adım 4: 3. adımdaki en yüksek özniteliği temel alarak bir karar düğümü oluştur,
Adım 5: 4. adımda yeni oluşturulan karar düğümünü temel alarak veri kümesini alt veri kümelere böl,
Adım 6: 5. adımdaki tüm alt veri kümeleri için, bir alt karar düğümlerini oluşturmak için özyinelemeli olarak adım 1'e git.
Adım 7: 6. adımda elde edilen ağacı 4. adımdaki karar düğümüne ekle,
Çıkış: Ağaç yapısı.

Şekil 3.9. C4.5 karar ağacı sınıflandırma algoritmasının sözde kodu

Şekil 3.9'da anlatılan C4.5 karar ağacı sınıflandırma algoritmasının çalışma prensibi Çizelge 3.3'te verilen örnek veri kümesi üzerinde ayrıntılı olarak gösterilmiştir.

Çizelge 3.3. Örnek veri kümesi

Örnekler	Öznitelikler				Sınıf
	A_1	A_2	A_3	A_4	C
$ö_1$	2	1	Hayır	0	1
$ö_2$	3	2	Hayır	0	1
$ö_3$	3	0	Evet	0	1
$ö_4$	2	0	Evet	1	1
$ö_5$	1	0	Evet	0	1
$ö_6$	3	2	Evet	0	1
$ö_7$	1	2	Evet	1	1
$ö_8$	2	2	Hayır	1	1
$ö_9$	2	1	Evet	0	1
$ö_{10}$	1	1	Hayır	1	0
$ö_{11}$	3	0	Evet	1	0
$ö_{12}$	1	2	Hayır	0	0
$ö_{13}$	3	2	Hayır	1	0
$ö_{14}$	1	1	Hayır	0	0

Veri kümesi A_1, A_2, A_3, A_4 özniteliklerinden ve C sınıf bilgisinden oluşmaktadır, C özniteliği için bilgi kazancı aşağıdaki gibi hesaplanmaktadır. C_1 sınıf değerinin sayısı = 9 ve C_2 sınıf değerinin sayısı = 5, bu durumda $|C_1| = 9\ ve\ |C_2| = 5$, bu bilgilere göre C_1 ve C_2 sınıflarının olasılıkları ve belirsizliği aşağıdaki gibi hesaplanır.

$$P_{C_1} = 9/14 = 0.64\ ve\ P_{C_2} = 5/14 = 0.36$$

C sınıf etiketinin belirsizliği Denklem 3.5'e göre elde edilir.

$$H(C) = -\left(P_{C_1}\log_2\left(P_{C_1}\right) + P_{C_2}\log_2\left(P_{C_2}\right)\right)$$

$$= (-0.64\ \log_2(0.64\ ) - 0.36\log_2(0.36)) = 0.940$$

Sınıf etiketinin belirsizliği hesaplandıktan sonra veri kümesindeki her öznitelik için sınıf bilgilerine göre bilgi kazançları Denklem 3.6'ya göre hesaplanır. En yüksek bilgi kazancına sahip olan öznitelik en iyi bölünmeyi sağlayacak nitelik olarak belirlenir ve ağaç yapısının kökünde yer alır. Aşağıda veri kümesinin her bir özniteliği için ayrı ayrı bilgi kazancı hesaplanmaktadır.

Birinci öznitelik (A_1) için bilgi kazancı $\left|A_{1_1}\right| = 5, \left|A_{1_2}\right| = 4\ ve\ \left|A_{1_3}\right| = 5$

$$H(A_1, C) = \left(\frac{5}{14} * H\left(A_{1_1}\right) + \frac{4}{14} * H\left(A_{1_2}\right) + \frac{5}{14} * H\left(A_{1_2}\right)\right)$$

$$H(A_1, C) = \frac{5}{14} * \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14} * \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) + \frac{5}{14} * \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694$$

$$Bilgi\ kazancı = H(C) - H(A_1, C) = 0.940 - 0.694\ =\ 0.246$$

İkinci öznitelik (A_2) için bilgi kazancı $\left|A_{2_0}\right| = 4,\ \left|A_{2_1}\right| = 4\ ve\ \left|A_{2_2}\right| = 6$

$$H(A_2, C) = \left(\frac{4}{14} * H\left(A_{2_0}\right) + \frac{4}{14} * H\left(A_{2_1}\right) + \frac{6}{14} * H\left(A_{2_2}\right)\right)$$

$$H(A_2, C) = \frac{4}{14} * \left(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right) + \frac{4}{14} * \left(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right) + \frac{6}{14} * \left(-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}\right) = 0.911$$

$$Bilgi\ kazancı = H(C) - H(A_2, C) = 0.029$$

Üçüncü öznitelik (A_3) için bilgi kazancı $\left|A_{3_{Evet}}\right| = 7\ ve\ \left|A_{3_{Hayır}}\right| = 7$

$$H(A_3, C) = \left(\frac{7}{14} * H\left(A_{3_{Evet}}\right) + \frac{4}{14} * H\left(A_{3_{Hayır}}\right)\right)$$

$$H(A_3, C) = \frac{7}{14} * \left(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}\right) + \frac{7}{14} * \left(-\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}\right) = 0.788$$

$$Bilgi\ kazancı = H(C) - H(A_3, C) = 0.151$$

Dördüncü öznitelik (A_4) için bilgi kazancı $\left|A_{4_0}\right| = 8\ ve\ \left|A_{4_1}\right| = 6$

$$H(A_4, C) = \left(\frac{8}{14} * H\left(A_{4_0}\right) + \frac{6}{14} * H\left(A_{4_1}\right)\right)$$

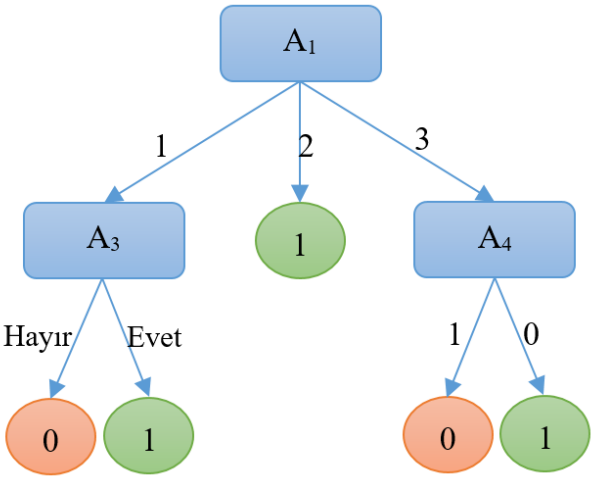
$$H(A_2, C) = \frac{8}{14} * \left(-\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}\right) + \frac{6}{14} * \left(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}\right) = 0.892$$

$$Bilgi\ kazancı = H(C) - H(A_4, C) = 0.048$$

Yukarıda veri kümesi ile ilgili yapılan işlemler sonucu $A_1, A_2, A_3\ ve\ A_4$ öznitelikleri sırasıyla 0.246, 0.029, 0.151 ve 0.048 bilgi kazançları elde etmişlerdir. Veri kümesinin A_1 özniteliği en yüksek bilgi kazancına sahip olduğundan dolayı veri kümesi bu özniteliğe göre bölünmektedir ve bu öznitelik ağaç yapısının kökünde yer almaktadır. Karar ağacının ilk aşaması Şekil 3.10'da görülmektedir.

Şekil 3.10. Karar ağacının ilk aşaması

Karar ağacının iç düğümlerini ve terminal düğümlerini belirlemek için yukarıdaki bilgi kazancı hesaplamaları veri kümesinin geriye kalan özniteliklerine de uygulanmaktadır. Uygulama sonucunda kullanılan eğitim veri kümesinden ağaç yapısında sınıflandırma kuralları elde edilir. Son olarak yeni gelen test örneği bu ağacın yapısına göre yol izleyerek bir sınıfa atanmaktadır. Örneğin son ağaç yapısı Şekil 3.11'de verilmiştir.



Şekil 3.11. Örneğin ağaç yapısı

3.3.2. Olasılık tabanlı sınıflandırma algoritmaları

Olasılık yöntemleri oyun geliştirme, bilgi keşfi ve tahmin gibi birçok alanda kullanılmaktadır. Olasılık tabanlı sınıflandırma algoritmalarından beklenen eksik verilere sahip olan veri kümelerinden veya küçük veri kümelerden çok daha fazla bilgi çıkarmasıdır. Olasılık yöntemleri ayrıca tıp gibi alanlar için gerçekten istenen bir özellik olan hastalık teşhislerindeki belirsizliği tahmin etmemize yardımcı olmaktadır. Olasılık yöntemleri makine öğrenmesi algoritmalarına uygulandığında, bu algoritmaların model oluşturma hızını ve verimliliğini artırmaktadır (John ve Langley, 1995).

Olasılık tabanlı sınıflandırma algoritmaları olasılık teorisine dayanmaktadır. NB sınıflandırma algoritması birçok alanda kullanılan olasılık tabanlı sınıflandırma algoritmalarından biri olarak bilinmektedir. NB olasılık tabanlı sınıflandırma algoritmasının kural tabanlı ve ağırlık tabanlı sınıflandırma algoritmalarına göre en büyük avantajı, hızlı ve iyi bir sınıflandırma başarısı ile büyük veri kümelerini analiz etmesidir

(McCallum ve Nigam, 1998; Rish, 2001). Aşağıda NB olasılık tabanlı sınıflandırma algoritması ayrıntılı olarak açıklanmıştır.

3.3.2.1. Naive bayes sınıflandırma algoritması

NB sınıflandırma algoritmasının çalışma prensibine geçmeden önce, NB sınıflandırma algoritmasında kullanılan terimlerin açıklamasında fayda vardır. NB sınıflandırma algoritması olasılıklı sınıflandırıcı kategorisi altında yer almaktadır ve istatistiksel sınıflandırıcılardır. Belirli bir örneğin belirli bir sınıfa ait olma olasılığı gibi, sınıf aitliği olasılıklarını tahmin edebilirler. Olasılıklı sınıflandırıcılar olasılık teorisine dayanmaktadır, NB sınıflandırma algoritması olasılıklı sınıflandırıcıların en basit, hızlı ve başarılı algoritmasıdır. Araştırmacılar tarafından yapılan çalışmalarda performans açısından NB sınıflandırma algoritması karar ağacı ve sinir ağları gibi sınıflandırma algoritmaları ile karşılaştırılmıştır. Karşılaştırma sonucunda, NB sınıflandırma algoritması, büyük veri kümeleri üzerinde uygulandığında hızlı bir şekilde yüksek sınıflandırma doğruluğu sergilemiştir (McCallum ve Nigam, 1998; Rish, 2001). NB sınıflandırma algoritması olasılık teorisi ile sınıflandırma işlemini gerçekleştirmektedir. Bayes teorisi, 18. yüzyılda olasılık ve karar teorisinde çalışmış olan Thomas Bayes'in adını taşımaktadır. Bayesian terimlerinde X, bir ifade olarak kabul edilir, bu ifade bir n nitelik seti üzerinde yapılan ölçümlerle açıklanmaktadır. H ise bir hipotezi ifade eder, yani X veri örneğini belirli bir C sınıfına ait olması gibi. Sınıflandırma problemlerinde sınıf etiketi belli olmayan bir örneği sınıflandırmak için bu örneğin koşullu olasılık $P(H|X)$ değeri belirlenmelidir. Sınıfı belli olmayan örneğin $P(H|X)$'i Denklem 3.7 ile hesaplanır (Wu ve ark., 2008).

$$P(H|X) = P(X|H) * P(H)/P(X) \tag{3.7}$$

$P(H|X)$: Hava durumu, nem oranı ve sıcaklık değeri giriş bilgilerini bildiğimiz için oyuncunun oyun oynama ihtimalidir (Evet veya Hayır). H'nin artçıl (posterior) olasılığı olarak bilinir. $P(H)$: Hava durumu, nem oranı ve sıcaklık değeri ne olursa olsun bir oyuncunun oyun oynama ihtimalidir. Yani sınıf etiketlerinin olasılıkları, H'nin olasılığı olarak bilinir. $P(X|H)$: Hava durumu = güneşli, nem oranı = orta ve sıcaklık değeri = orta, oyuncunun oyun oynama olasılığıdır. Yani girdilerin sınıf etiketine göre olasılığı ve X'nin posterior olasılığı olarak bilinir. $P(X)$: hava durumu = güneşli, nem

oranı = orta ve sıcaklık değeri = orta girdilerin her birinin olasılığıdır. Yani girdilerin olasılıkları ve X'in olasılığı olarak bilinir.

NB sınıflandırma algoritması kolay uygulanabilir olduğu kadar üstün performansıyla da birçok sınıflandırma problemlerinde en çok kullanılan sınıflandırma algoritmalarından biri olarak bilinir. Bu sınıflandırma algoritmasında önce eğitim veri kümesinden her özellik değerinin sınıf etiketine göre olasılıkları hesaplanır. Daha sonra bu girişlerin C sınıfına ait olma olasılıkları çarpılarak yeni gelen örnek bir sınıf etiketine atanır. NB sınıflandırma algoritmasının çalışma mekanizması aşağıda D eğitim veri kümesi üzerinde açıklanabilir. Örneklerden oluşan bir eğitim veri kümesi D, eğitim kümesinde her örnek n boyutlu özniteliklerden $X = (x_1, x_2, \dots, x_n)$ ve x_k her A_k özniteliğinin değerini temsil etmektedir. Veri kümesindeki sınıf boyutu m ile temsil edilmektedir ve $C = (c_1, c_2, \dots, c_m)$ şeklinde gösterilir. Yeni gelen X örneği C_i sınıf etiketine ait olursa: $P(C_i|X) > P(C_j|X)\ her\ i, j \geq 1\ ve\ i \neq j$. Maksimum posterior olasılığı $P(C_i|X) = P(X|C_i) * P(C_i)/P(X)$ ile belirlenir. Burada $P(C_i) = P(C_i)/P(C)$ ve $P(X|C_i)$, C_i sınıfına ait olan X örneğinin olasılığıdır (Rish, 2001).

$$P(X_k|C_i) = \frac{C_i\ sınıfında\ bulunan\ X_k\ örnek\ sayısı}{C_i\ sınıfının\ örnek\ sayısı} \tag{3.8}$$

Daha sonra Denklem 3.9'a göre yeni gelen X örneğinin $P(X|C_i)$'i yani sınıf bilgisi tahmin edilir.

$$P(X|C_i) = \prod_{k=1}^{n} P(X_k|C_i) = P(X_1|C_i) * P(X_2|C_i) * \cdots * P(X_n|C_i) \tag{3.9}$$

$P(X)$ tüm sınıflar için sabit olduğuna göre $P(X|C_i) * P(C_i)$ Posterior olasılığının maksimum değeri bulunmalıdır. Yeni bir örnek X, maksimum $P(X|C_i) * P(C_i)$ değerine sahip olan sınıfa atanır.

NB sınıflandırma algoritması metin sınıflandırma, ses tanıma sistemleri, şifre kontrolü uygulamaları, hastalık teşhisi gibi birçok uygulama alanlarında kullanılmaktadır. Her sınıflandırma algoritması gibi NB sınıflandırma algoritmasının da avantajları ve dezavantajları bulunmaktadır. Hızlı ve kolay uygulanabilmesi, üstün bir performansa sahip olması ve büyük veri kümelerinde iyi sınıflandırma başarısı

gösterebilmesi NB sınıflandırma algoritmasının avantajlarıdır. Her bir özelliğe eşit değer verilmesi, sistem dinamik olduğunda her defasında eğitimin tekrardan yapılması, kategorik olasılık değeri sıfır ise laplace tahmincisi ile bir eklenerek sıfır değeri ortadan kaldırılıyor olsa bile ve sürekli değerler üzerinde bazı zaman iyi sonuç vermemesi NB sınıflandırma algoritmasının dezavantajıdır (John ve Langley, 1995).

Yukarıda anlatılan NB sınıflandırma algoritmasının çalışma mantığı aşağıda basit bir örnek üzerinde uygulanmıştır. Örneğin Çizelge 3.4'te verilen veri kümesi yedi adet örnekten ve her örnek (A1, A2, A3, A4) dört adet öznitelik ve bir adet C sınıf bilgisinden oluşmaktadır. A1, A2, A3 öznitelikleri ayrık ve A4 özniteliği ise kategorik değerlerden oluşmaktadır. Sınıf bilgisi ise C_1 ve C_2 olarak iki adet sınıf etiketinden oluşmaktadır.

Çizelge 3.4. Örnek veri kümesi

Örnekler	Öznitelikler				Sınıf
	A_1	A_2	A_3	A_4	C
$ö_1$	1	2	0	Küçük	1
$ö_2$	1	3	2	Orta	1
$ö_3$	1	2	2	Küçük	1
$ö_4$	0	3	4	Büyük	0
$ö_5$	0	2	3	Orta	0
$ö_6$	1	3	0	Küçük	0
$ö_7$	0	1	3	Büyük	0

Veri kümesinde görülmeyen ve sınıf etiketi belli olmayan bir örnek geldiğinde NB sınıflanırma algoritması bu örneğin hangi sınıfa ait olduğunu olasılık teorisini kullanarak tahmin edebilir.

Örnek olarak, $X = (A_1 = 1, A_2 = 2, A_3 = 0, A_4 = küçük, C = ?)$ sınıf etiketi belli olmayan örneğin, sınıf etiketi aşağıdaki adımları ile tahmin edilmektedir.

Adım 1: Her sınıf bilgisinin olasılığı hesaplanır.

$P(C_1) = p(C = 1) = 3/7 = 0.43$

$P(C_2) = p(C = 0) = 4/7 = 0.57$

Adım 2: Tahmin edilecek örneğin öznitelik değerlerinin sınıf etiketlerine göre olasılıkları hesaplanır.

$P(A_1|C_1) = P(A_1 = 1|C_1 = 1) = 3/4 = 0.75$

$P(A_1|C_2) = P(A_1 = 1|C_2 = 0) = 1/4 = 0.25$

$P(A_2|C_1) = P(A_2 = 2|C_1 = 1) = 2/3 = 0.66$

$P(A_2|C_2) = P(A_2 = 2|C_2 = 0) = 1/3 = 0.33$

$P(A_3|C_1) = P(A_3 = 0|C_1 = 1) = 1/2 = 0.5$

$P(A_3|C_2) = P(A_3 = 0|C_2 = 0) = 1/2 = 0.5$

$P(A_4|C_1) = P(A_4 = küçük|C_1 = 1) = 2/3 = 0.66$

$P(A_4|C_2) = P(A_4 = küçük|C_2 = 0) = 1/3 = 0.33$

$P(X|C_1 = 1) = P(A_1|C_1) * P(A_2|C_1) * P(A_3|C_1) * P(A_4|C_1)$

$P(X|C_1 = 1) = 0.75 * 0.66 * 0.5 * 0.66 = 0.163$

$P(X|C_2 = 0) = P(A_1|C_2) * P(A_2|C_2) * P(A_3|C_2) * P(A_4|C_2)$

$P(X|C_2 = 0) = 0.25 * 0.33 * 0.25 * 0.33 = 0.006$

Adım 3. Son olarak $P(H|X) = P(X|H) * P(H)$ maksimum *posterior* olasılığı örneğin hangi sınıfa ait olduğunu tahmin eder.

$P(C_i|X) = P(X|C_i) * P(C_i)$

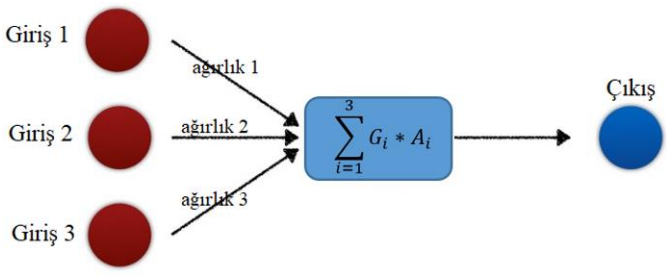
$P(C_1|X) = P(X|C_1 = 1) * P(C_1 = 1) = 0.163 * 0.43 = 0.07$

$P(C_2|X) = P(X|C_2 = 0) * P(C_2 = 0) = 0.006 * 0.57 = 0.003$

Birinci sınıfın C_1 posterior olasılığı ikinci sınıfın C_2 Posterior olasılığından büyük olduğundan dolayı yeni gelen X örneği NB sınıflandırma algoritması tarafından C_1 sınıfına atanmaktadır.

3.3.3. Ağırlık tabanlı sınıflandırma algoritmaları

Ağırlık tabanlı sınıflandırma algoritmaları, ağlar ile temsil edilen ve bu ağları oluşturan hücreleri ilişkilendiren bağlantıların üzerinde bulunan ağırlıkları içeren sınıflandırma algoritmalarıdır. Ağırlık tabanlı sınıflandırma algoritmalarının birçoğu, ağ ile temsil edildiğinden dolayı genelde bu tür sınıflandırma algoritmalarının yapıları Şekil 3.12 gibi olmaktadır (Dreiseitl ve Ohno-Machado, 2002).



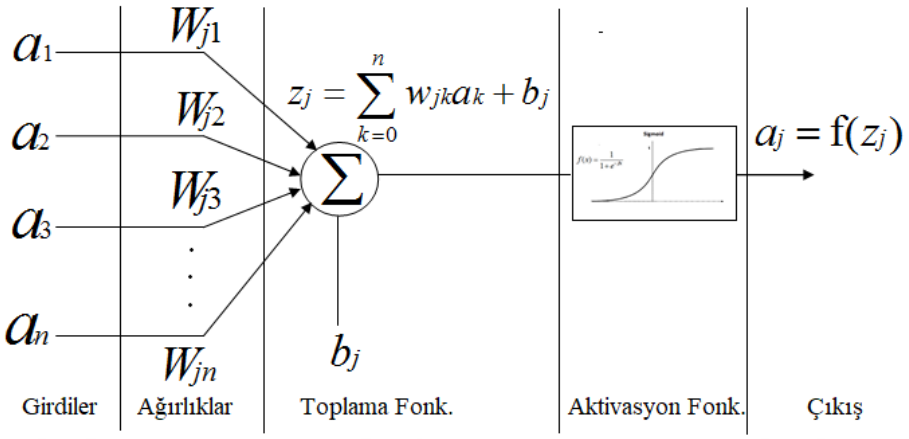
Şekil 3.12. Ağırlık tabanlı sınıflandırma algoritmasının genel yapısı

Ağırlık tabanlı sınıflandırma algoritmalarının eğitilmesi literatürde de bahsedildiği gibi zor bir problem olarak görülmektedir. Bu problem, maksimum sınıflandırma başarısı veya minimum tahmin hatalarını elde etmek için ağırlık tabanlı sınıflandırma algoritmalarının yapısal parametrelerini yani ağırlıklarını en uygun değerlere güncellemektir. Aslında öğrenme, her türlü makine öğrenmesi sınıflandırma algoritmalarının başarısını etkileyen önemli bir süreçtir. Bu problemin yüksek boyutsallığı ve verilen eğitim veri kümesine göre değişen arama alanı nedeniyle, ağırlık tabanlı sınıflandırma algoritmalarında öğrenme geliştirme problemi zor bir görev olarak kabul edilmektedir (Mirjalili ve ark., 2015). Ağırlık tabanlı sınıflandırma algoritmalarının eğitimi ağ yapısının üzerinde bulunan ağırlıkların değerleri ile ilgilidir. Ağ yapısı üzerinde bulunan ağırlıkların en uygun değerlerini elde etmek için araştırmacılar tarafından birçok öğrenme algoritmaları geliştirilmiştir. Birçok tahmin ve sınıflandırma problemlerinde başarılı bir şekilde kullanılan YSA sınıflandırma algoritması ve eğitim işlemi için kullanılan geriye yayılım öğrenme algoritması aşağıdaki bölümlerde ayrıntılı olarak açıklanmıştır.

3.3.3.1. Yapay sinir ağları

YSA sınıflandırma algoritması, insan beyninden esinlenerek ortaya çıkan bir makine öğrenmesi sınıflandırma algoritması olarak bilinir. Başka bir deyişle, YSA sınıflandırma algoritması biyolojik sinir ağlarının yapısını ve işlevselliğini taklit etmeye çalışan matematiksel bir modeldir. YSA sınıflandırma algoritması üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik üniteler olan nöronların modellenmesi ve daha sonra bilgisayar sistemlerinde uygulanması ile başlamıştır. Bilgisayar sistemlerinin gelişimi ile paralel olarak YSA sınıflandırma algoritması havacılık, tıp, robotik, finans ve mühendislik gibi birçok alanda kullanılır hale gelmiştir. YSA sınıflandırma algoritmasının temel yapısı, basit bir matematik fonksiyonu olan yapay nöron ve bu nöronları birbirleri ile ilişkilendiren ağlardan oluşmaktadır (Gardner ve Dorling, 1998). Genel olarak YSA sınıflandırma modeli çarpma, toplama ve aktivasyon fonksiyonları gibi üç basit işlem kümesine sahiptir. Çarpma fonksiyonu, her girdi değerinin bireysel ağırlığı ile çarpılması anlamına gelir, yapay nöronun orta bölümünde tüm ağırlıklı girdileri ve biasları toplayan toplama fonksiyon bulunur. Yapay nöronun çıkışında ise, daha önce ağırlıklı girdilerin ve biasların toplamı, aktivasyon fonksiyonundan geçerek bir çıkış üretmektedir, aktivasyon fonksiyonları aynı zamanda transfer fonksiyonu olarak da

adlandırılabilir. Basit bir YSA sınıflandırma algoritmasının çalışma modeli Şekil 3.13'te verilmiştir (Hippert ve ark., 2001).



Şekil 3.13. Basit bir YSA sınıflandırma algoritmasının çalışma modeli

Yukarıdaki Şekil 3.13 incelendiğinde, *a* değerleri ağın girdi değerleri olarak bilinir ve bu değerler yapay sinir hücrelerine dış dünyadan veya bir önceki hücreden gelmektedir. Bu girdilerin her biri kendi ağırlığı ile çarpılarak bir sonraki bölüme gönderilir. Hücrelerin bağlantıları üzerinde bulunan ağırlıklar girdilerin ürettiği çıkışların üzerinde olan etkisini ayarlayabilmektir. Ağda bulunan ağırlık değerleri negatif, pozitif veya sıfır olabilir. Fakat bir girdinin ağırlığı sıfır olursa, o girdinin çıktı üzerinde bir etkisi kalmamaktadır. Daha sonra toplanmak üzere gelen ağırlıklı girdiler bir toplama fonksiyonu işlemi sonucu, hücrenin net girdisini hesaplamaktadır (Haykin ve ark., 2009). Bu güne kadar, belli girdiler için sabit veya uygun bir toplama fonksiyonu belirleme yöntemi geliştirilmemiştir, genelde deneme yanılma yoluyla toplama fonksiyonları belirlenmektedir. Aşağıda toplama fonksiyonları türleri verilmiştir (Jain ve ark., 1996).

Toplam: Hücreye gelen girdilerin değerleri kendi ağırlıkları ile çarpılır daha sonra bu ağırlıklı girdiler birbirleri ile toplanarak net girdi değeri hesaplanır, toplam fonksiyonun denklemi Denklem 3.10'da verilmektedir.

$$Net\ girdi\ değeri = \sum_{i=1}^{m} I_i * W_i \tag{3.10}$$

Çarpım: Hücreye gelen girdilerin değerleri kendi ağırlıkları ile çarpılır daha sonra bu ağırlıklı girdiler birbirleri ile çarpılarak net girdi değeri hesaplanır, çarpım fonksiyonun denklemi Denklem 3.11'de verilmektedir.

$$Net\ girdi\ değeri = \prod_{i=1}^{m} I_i * W_i \tag{3.11}$$

Maksimum: Hücreye gelen girdilerin değerleri kendi ağırlıkları ile çarpıldıktan sonra ağırlıklı girdilerin değerlerinden en büyük değer net girdi değerini temsil etmektedir. Maksimum fonksiyonun denklemi Denklem 3.12'de verilmektedir.

$$Net\ girdi\ değeri = Max(I_i * W_i) \tag{3.12}$$

Minimum: Hücreye gelen girdilerin değerleri kendi ağırlıkları ile çarpıldıktan sonra ağırlıklı girdilerin değerlerinden en küçük değer net girdi değerini temsil etmektedir. Minimum fonksiyonun denklemi Denklem 3.13'te verilmektedir.

$$Net\ girdi\ değeri = Min(I_i * W_i) \tag{3.13}$$

Çoğunluk: Hücreye gelen girdilerin değerleri kendi ağırlıkları ile çarpıldıktan sonra ağırlıklı girdilerin negatif ile pozitif değerlerinin sayıları hesaplanır ve en büyük değere sahip olanlar toplanarak net girdi değeri belirlenir. Çoğunluk fonksiyonun denklemi Denklem 3.14'te verilmektedir.

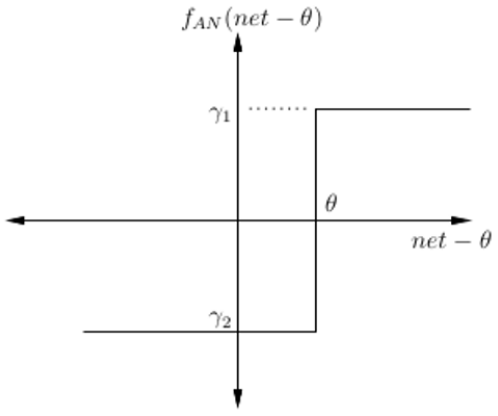
$$Net\ girdi\ değeri = \sum_{i=1}^{m} sgin(I_i * W_i) \tag{3.14}$$

Her hücrenin toplam fonksiyonuna göre net girdisi hesaplandıktan sonra, hücreye gelen net girdi değerine bir aktivasyon fonksiyonu uygulanır ve bir çıktı üretilir. Aktivasyon fonksiyonları transfer fonksiyonu olarak da bilinir. Ayrıca, giriş katmanı haricinde diğer katmanların hücrelerinde kullanılır. YSA sınıflandırma algoritmasının eğitim aşamasında, sezgisel öğrenme algoritmaları aktivasyon fonksiyonunun türevini alarak ağırlık güncelleme işlemini gerçekleştirmektedir. Dolayısıyla YSA sınıflandırma

katmanlarında türevi kolay hesaplanabilir aktivasyon fonksiyonlarının seçilmesi YSA sınıflandırma algoritmasının eğitim aşamasını hızlandırır. Sinir ağının çıkışını evet veya hayır gibi belirlemek için kullanılır, Elde edilen değerleri 0 ila 1 veya -1 ila 1 arasında işleve bağlı olarak eşleştirir. Aktivasyon fonksiyonları temel olarak doğrusal ve doğrusal olmayan aktivasyon fonksiyonları olarak iki türe ayrılabilir. Aşağıda aktivasyon fonksiyonlarının türleri açıklanmıştır (Hashem, 1992; Debes ve ark., 2005; Karlik ve Olgac, 2011).

Adım aktivasyon fonksiyonu, eşik temelli bir aktivasyon fonksiyonudur. Toplama fonksiyonundan elde edilen net değer belirli bir eşik değerin üzerindeyse, aktif olduğunu beyan eder, eğer belirli bir eşik değerin altındaysa aktif olmadığını beyan etmektedir. Adım aktivasyon fonksiyonu ikili çıktı sınıflandırma problemleri için ideal bir aktivasyon fonksiyonu olabilir, fakat çok çıktılı sınıflandırma problemlerinde adım fonksiyonu yerine sofmax gibi fonksiyonlardan yardım alınarak hücre çıkışları elde edilir. Adım aktivasyon fonksiyonunun denklemi Denklem 3.15'te ve şekli Şekil 3.14'te verilmiştir.

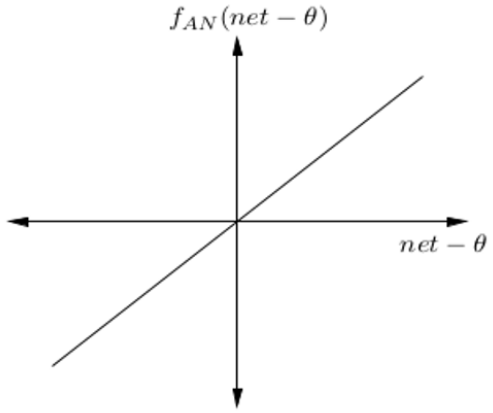
$$F(Net) = \begin{cases} 1\ Eğer\ Net\ değeri > Eşik\ değer \\ 0\ Eğer\ Net\ değeri < Eşik\ değer \end{cases} \tag{3.15}$$



Şekil 3.14. Adım aktivasyon fonksiyonu

Doğrusal aktivasyon fonksiyonu, doğrusal problemlerin çözümünde kullanılan aktivasyon fonksiyonudur. Toplama fonksiyonundan elde edilen değer, belli bir sabit sayı ile çarpılarak hücrenin çıktısını belirler. Doğrusal fonksiyonunun denklemi Denklem 3.16'da ve şekli Şekil 3.15'te verilmiştir.

$$F(Net) = Sabit\ değer * Net \tag{3.16}$$



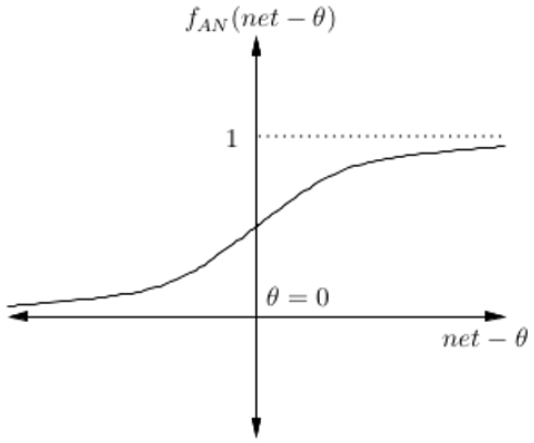
Şekil 3.15. Doğrusal aktivasyon fonksiyonu

Doğrusal aktivasyon fonksiyonunun türevi sabit bir değer olduğundan dolayı x hücresinin ağırlık güncellemesinde bir etkisi olmayabilir. Çünkü tahminde bir hata varsa, geri yayılım tarafından yapılan değişiklikler sabittir ve delta girişindeki değişime bağlı değildir. Eğer YSA sınıflandırma algoritması birden fazla katmandan oluşuyorsa ve bu katmanlar doğrusal fonksiyonu kullanıyorsa, bu aktivasyon fonksiyonu çıkışı sırayla bir sonraki katmana aktarılır, ikinci katman o girdi üzerindeki ağırlıklı toplamı hesaplar ve bir doğrusal aktivasyon fonksiyonundan geçerek başka katmana aktarılır. YSA sınıflandırma algoritması ne kadar katmana sahip olursa olsun, eğer bu katmanlar doğrusal fonksiyonu kullanıyorsa, son katmanın son aktivasyon fonksiyonu sadece birinci katmanın girişinin doğrusal bir fonksiyonundan başka bir şey değildir.

Sigmoid aktivasyon fonksiyonu, kombinasyonları doğrusal olmayan bir aktivasyon fonksiyonudur. Sigmoid aktivasyon fonksiyonu türevi alınabilir bir fonksiyon olduğundan dolayı geriye yayılım algoritmasında sıkça kullanılan aktivasyon fonksiyonlarından biri olarak bilinir. Sigmoid aktivasyon fonksiyonunun girdi değeri ne olursa olsun, bu girdi değeri için 0 ile 1 arasında bir değer üretmektedir.

Sigmoid aktivasyon fonksiyonu adım aktivasyon fonksiyonundan farklı olarak analog bir çıkış vermektedir. Sigmoid aktivasyon fonksiyonunda, X değerleri -2 ila 2 arasında olduğundan Y değerleri çok diktir. Bu değişim ise, o bölgedeki X değerlerinde yapılan küçük değişikliklerin Y değerlerinin önemli ölçüde değişmesine neden olacağı anlamına gelmektedir. Sigmoid aktivasyon fonksiyonunun denklemi Denklem 3.17'de ve şekli Şekil 3.16'da verilmiştir.

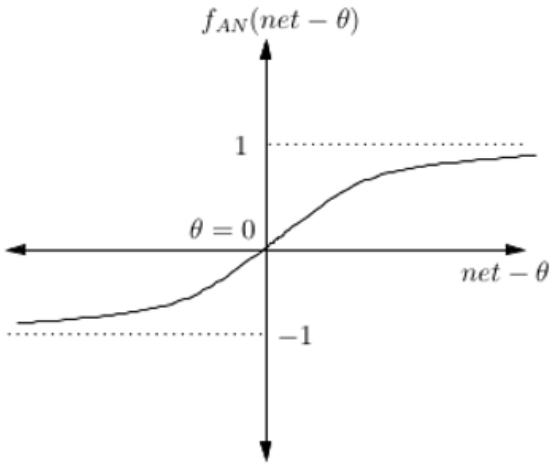
$$F(Net) = \frac{1}{1 + e^{-Net}} \tag{3.17}$$



Şekil 3.16. Sigmoid aktivasyon fonksiyonu

Tanjant hiperbolik aktivasyon fonksiyonu, sigmoid aktivasyon fonksiyonu gibi doğrusal olmayan bir aktivasyon fonksiyonudur. YSA sınıflandırma algoritmalarında sıkça kullanılan bir aktivasyon fonksiyonu olarak bilinir. Fakat tanjant hiperbolik aktivasyon fonksiyonuna hangi aralıkta bir net değeri gelirse gelsin tanjant aktivasyon fonksiyonu -1 ile 1 arasında bir çıkış değeri üretir. Tanjant hiperbolik aktivasyon fonksiyonunun denklemi Denklem 3.18'de ve şekli Şekil 3.17'de verilmiştir.

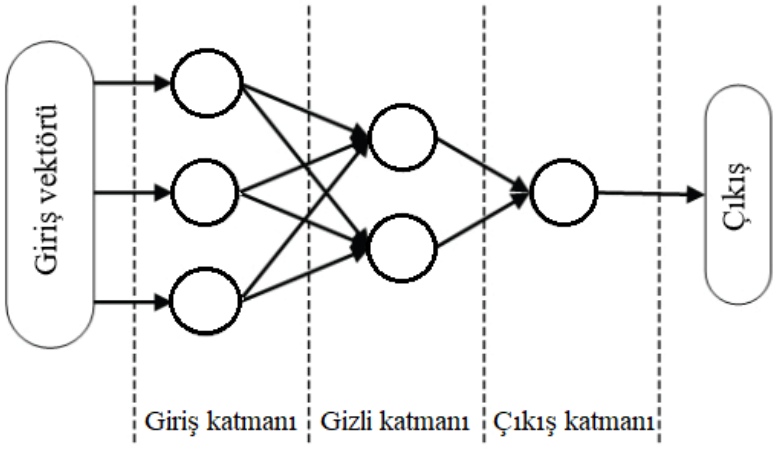
$$F(Net) = \frac{2}{1 + e^{-2Net}} - 1 \tag{3.18}$$



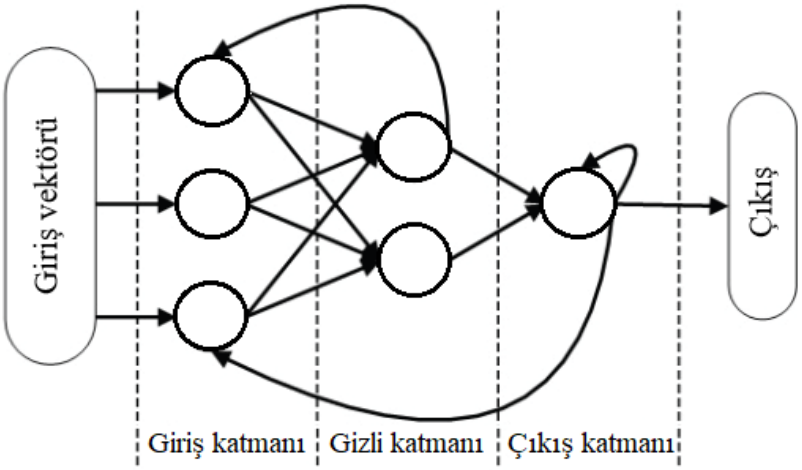
Şekil 3.17. Tanjant hiperbolik aktivasyon fonksiyonu

Tanjant hiperbolik aktivasyon fonksiyonunda, aralıklar -1 ile 1 arasında olduğundan dolayı aktivasyonların kayma endişesi ortadan kaldırılmış olur. Tanjant hiperbolik aktivasyon fonksiyonunun eğimi sigmoid aktivasyon fonksiyonuna göre daha güçlüdür. Sigmoid aktivasyon fonksiyonunda olduğu gibi tanjant hiperbolik aktivasyon fonksiyonunda da eğilim kayıpları yaşanabilir. Ayrıca, tanjant hiperbolik aktivasyon fonksiyonu çok popüler ve yaygın olarak kullanılan bir aktivasyon fonksiyonudur.

Son olarak, YSA sınıflandırma algoritmasındaki bir hücre yukarıdaki aktivasyon fonksiyonlarından birini kullanarak bir çıkış elde etmektedir. Tek yapay nöron kullanılması gerçek yaşam problemlerini nerdeyse çözemez, fakat çok daha fazla yapay nöronu birleştirirken yapay bir sinir ağı elde edilebilir. YSA sınıflandırma algoritması, temel yapı taşlarındaki bilgisini doğrusal olmayan bir şekilde işleyerek karmaşık gerçek yaşam problemlerini çözebilir. Bireysel yapay nöronların birbirine bağlı olmalarına YSA sınıflandırma algoritmasının topolojisi, mimarisi veya yapısı denir. YSA sınıflandırma algoritması genelde topolojisine göre çok-katmanlı ileri Beslemeli YSA ve çok-katmanlı geriye beslemeli YSA olarak iki kategoriye ayrılır. Aşağıda çok-katmanlı ileri beslemeli YSA ve çok-katmanlı geriye beslemeli YSA sınıflandırma algoritmasının topolojileri sırasıyla Şekil 3.18 ve Şekil 3.19'da verilmektedir.



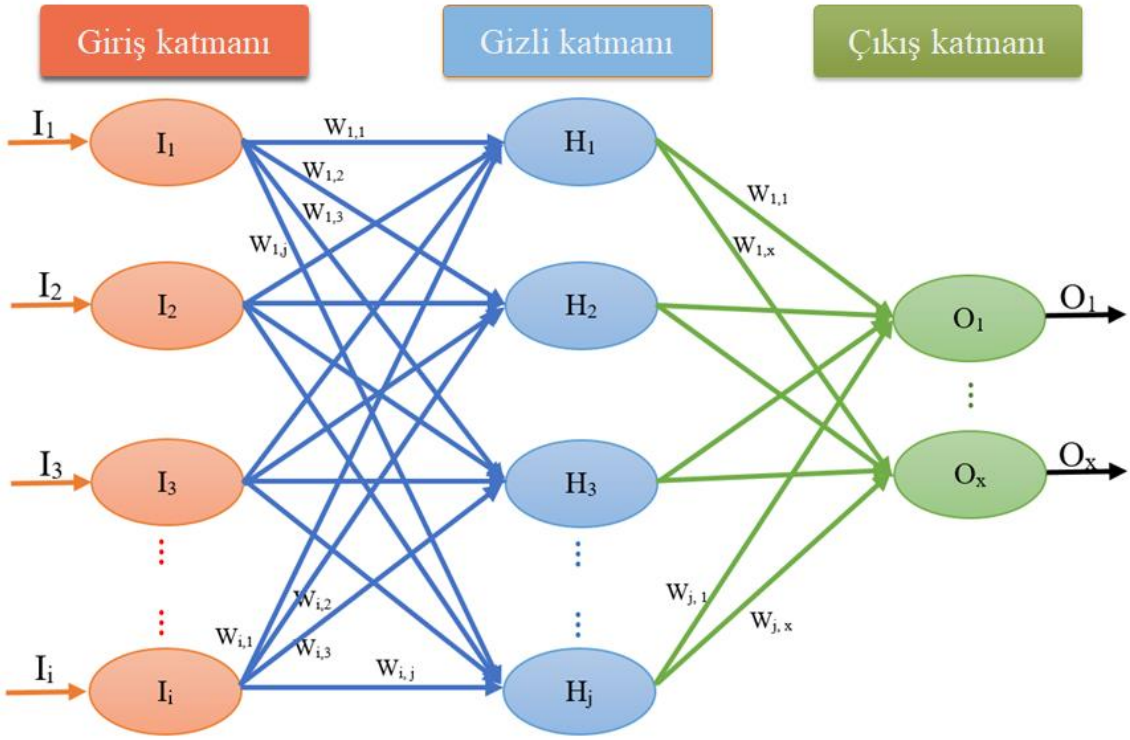
Şekil 3.18. Çok-katmanlı ileri beslemeli YSA (Yegnanarayana, 2009)



Şekil 3.19. Çok-katmanlı geriye beslemeli YSA (Yegnanarayana, 2009)

YSA'yı katmanlara ayırmanın en büyük avantajları, YSA'nın daha kolay bir şekilde anlaşılabilmesi, kullanılabilmesi ve matematiksel olarak tanımlanmasıdır. Bundan dolayı birçok doğrusal olamayan problemlerin çözümünde çok-katmanlı ileri YSA kullanılmaktadır. Şekil 3.18'de gösterilen çok-katmanlı ileri beslemeli YSA, çeşitli biçimlerde birbirine bağlanmış yapay sinir hücrelerini içerir ve genellikle katmanlar halinde tasarlanır. Donanımsal olarak elektronik devrelerde veya bilgisayarlarda yazılım olarak kolay bir şekilde uygulanabilir. Beyin bilgi işlem metoduna uygun olarak, çok-katmanlı ileri beslemeli YSA bir öğrenme sürecinden sonra bilgiyi saklama ve genelleştirme yeteneğine sahiptir. Bazı başarılı ağlar tek bir katmanla oluşturulabilir, fakat tek bir katmandan oluşan bir ağ sadece doğrusal fonksiyonlarının çözümünde kullanılır (Hertz ve ark., 1991). Ancak doğrusal olmayan gerçek dünya problemlerinin uygulamasında en az üç katmanlı ağlar gereklidir. Yukarıda Şekil 3.18'de görüldüğü gibi bu katmanlar giriş katmanı, gizli katman ve çıkış katmanı olarak bilinir.

Çok-katmanlı ileri beslemeli YSA, giriş ve çıkış katmanları arasında yer alan gizli katmanlar ile tek katmanlı sistemlerin tutarsızlığını ortadan kaldırmaktadır (Ilonen ve ark., 2003). Çok-katmanlı ileri beslemeli YSA yapısında, giriş katmanındaki nöronlar dış dünyadan veya bir sistemden girdi almak için kullanılırken, çıkış katmanındaki nöronlar çıktıları taşımak için ve gizli katmanlardaki tüm nöronlar sistem eğitimine yardımcı olmak için kullanılır. Temel olarak, çok-katmanlı ileri beslemeli YSA'nın ağırlıklı yapısı Şekil 3.20'de gösterilmiştir.



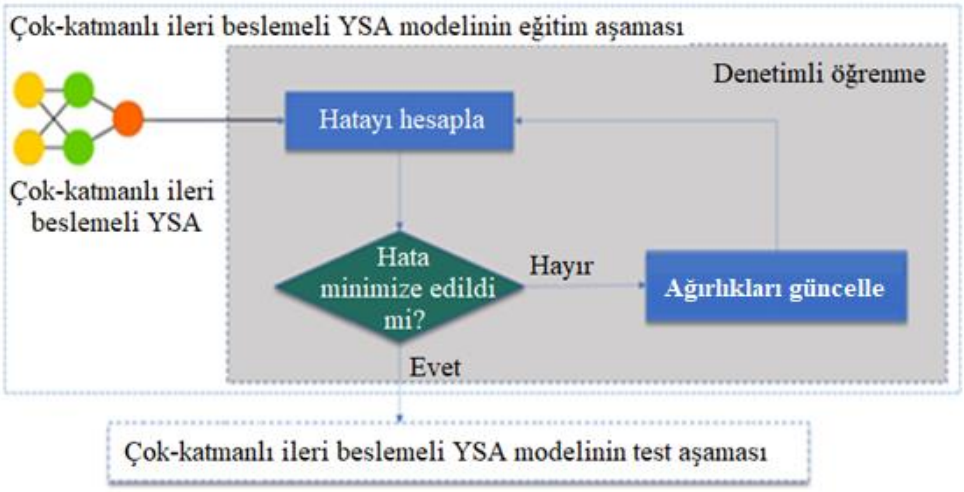
Şekil 3.20. Çok-katmanlı ileri beslemeli YSA'nın ağırlıklı yapısı

Şekil 3.20'ye baktığımızda, $I_1, I_2, I_3, \dots, I_i$ giriş katmanındaki girdi nöronları, $H_1, H_2, H_3, \dots, H_j$ gizli katmandaki gizli nöronları ve $O_1, O_2, O_3, \dots, O_x$ çıkış katmanındaki nöronları temsil etmektedir. $W_{1,1}, W_{1,2}, W_{1,3}, \dots, W_{1,j}$ ağırlıklar olarak tanımlanır ve bir nöron tarafından alınan bilgilerin etkisini gösterir. Çok-katmanlı ileri beslemeli YSA'lardan elde edilen çıkışı hesaplamak için ilk olarak, toplama işlevi fonksiyonu (NET) bir sinir hücresine gelen net bilgiyi hesaplar. Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Bu net değerini bulmak için toplama, çarpma, maksimum, minimum ve çoğunluk gibi toplam fonksiyonları kullanılabilir. Bu tez çalışmasında, ağa uygun olarak her katmanda toplam fonksiyonu toplama fonksiyonu ve sigmoid fonksiyonu aktivasyon fonksiyonu olarak kullanılmıştır.

Bir problem için YSA'mızın topolojisini seçip oluşturduğumuzda, problemin ilk aşamasını çözmüş olmaktayız. Bir sonraki aşama ise seçilen YSA topolojisi biyolojik sinir ağları gibi, ortamdan verilen girdilere verilen tepkilerini öğrenmelidir. YSA'nın doğru cevabı öğrenmesi, denetimli ve denetimsiz öğrenme yoluyla elde edilebilir. Öğrenme aşamasında hangi öğrenme yöntemini kullanırsak kullanalım, öğrenme yöntemlerinin görevi, seçilen maliyet fonksiyonunu en aza indirmesi için ağ yapısında bulunan ağırlık değerlerini öğrenme verisine göre ayarlamaktır. Çok-katmanlı ileri

beslemeli YSA'da, üç ana öğrenme yöntemi bulunmaktadır; denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve takviyeli öğrenme (reinforcement learning) (Haykin ve ark., 2009). Bu öğrenme şekilleri, herhangi bir çok-katmanlı ileri beslemeli YSA topoloji türü tarafından kullanılabilirler. Her öğrenme şeklinde birçok eğitim algoritması vardır. Çok-katmanlı ileri beslemeli YSA yapılarında en sık kullanılan denetimli öğrenme algoritmalarıdır.

Denetimli öğrenme, çok-katmanlı ileri beslemeli YSA'ya verilen eğitim setine göre, ağ yapısındaki bulunan ağırlık parametrelerini ayarlayan bir makine öğrenmesi tekniğidir. Genel olarak eğitim verileri, veri faktörlerinde temsil edilen girdiler ve istenen çıktı değerlerinden oluşmaktadır. Çok-katmanlı ileri beslemeli YSA'da öğrenmenin görevi, çıktı değerine göre geçerli herhangi bir giriş değeri için ağırlık parametrelerinin değerini ayarlamaktır. Verilen bir denetimli öğrenme problemini çözmek için çeşitli adımlar göz önünde bulundurulmalıdır (Borgersen ve Karlsson, 2008). İlk adımda, veri kümesinin türünü belirlemekteyiz. İkinci adımda, verilen bir problemi tatmin edici şekilde tanımlayan bir eğitim veri kümesi toplamamız gerekmektedir. Üçüncü adımda, seçilmiş bir çok-katmanlı ileri beslemeli YSA'ya göre anlaşılabilir formda toplanan eğitim verilerini tanımlamamız gerekmektedir. Dördüncü adımda ise, çok-katmanlı ileri beslemeli YSA'nın öğrenmesi ve öğrendikten sonra öğrenilen çok-katmanlı ileri beslemeli YSA'nın performansını test veri kümesi ile test etmekteyiz. Test veri kümesi, çok-katmanlı ileri beslemeli YSA'nın öğrenme aşamasında kullanılmayan veri örneklerinden oluşmaktadır (Borgersen ve Karlsson, 2008). Genel olarak çok-katmanlı ileri beslemeli bir YSA'nın denetimli öğrenme modeli Şekil 3.21'de verilmektedir.



Şekil 3.21. Çok-katmanlı ileri beslemeli YSA'da denetimli öğrenme modeli

Şekil 3.21'de olan Çok-katmanlı ileri beslemeli YSA'da denetimli öğrenme şeklinin adımları şu şekilde özetlenebilir: "hatayı hesapla" kısmı, çok-katmanlı ileri beslemeli YSA modelinin çıktısının gerçek çıktı ile ne kadar farklı olduğunu hesaplar. Minimum hata, hatanın minimize edilip edilmediği kontrol edilir. Ağırlıkları güncelle; hata çok büyükse, çok-katmanlı ileri beslemeli YSA modelinde bulunan ağırlıkları günceller. Bundan sonra hata tekrar kontrol edilir. Hata minimum hale gelinceye kadar denetimli öğrenme işlemi tekrarlanır. Çok-katmanlı ileri beslemeli YSA modelinin öğrenme aşaması gerçekleştirildikten sonra yani hata minimum hale geldiğinde, çok-katmanlı ileri beslemeli YSA modeli test veri kümesi ile test edilmektedir.

Çok-katmanlı ileri beslemeli YSA'da hatanın minimum olması geliştirilen sınıflandırma modelin iyi bir şekilde öğrendiğini ifade etmektedir (Baba, 1989). Araştırmacılar tarafından hataların ortalaması (Mean Error ME), hataların yüzde ortalaması (Mean Percentage Error MPE), hataların mutlak ortalaması (Mean Absolute Error MAE), hataların kare ortalaması (Mean Squared Error MSE), hataların kare ortalamasının karekökü (Root Mean Square Error RMSE), hataların mutlak ortalama yüzdesi (Mean Absolute Percentage Error MAPE) ve hataların karesinin toplamı (Sum of Squared Error SSE) gibi birçok hata hesapla fonksiyonları önerilmiştir. MAE, MSE, RMSE ve SSE çok-katmanlı ileri beslemeli YSA modellerinde sık kullanılan hata hesaplama fonksiyonlarıdır ve formülleri sırasıyla Denklem 3.19, 3.20, 3.21 ve 3.22'de gösterilmektedir (Yao, 1999).

$$\mathrm{MAE} = \frac{1}{n}\sum_{x=1}^{n}|O_x - T_x| \tag{3.19}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{x=1}^{n}(O_x - T_x)^2 \tag{3.20}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{x=1}^{n}(O_x - T_x)^2} \tag{3.21}$$

$$\mathrm{SSE} = \sum_{x=1}^{n}(O_x - T_x)^2 \tag{3.22}$$

Bu hata hesaplama denklemlerinde n, veri kümesindeki örnek sayısını temsil etmektedir. O_x, çok-katmanlı ileri beslemeli YSA modelinden x örneği için elde edilen çıktıyı ve T_x ise x örneğinin veri kümesindeki gerçek çıkış değerini temsil etmektedir. Genel olarak, tüm hata hesaplama fonksiyonları tasarlanan ağın çıkışı ile örneğin gerçek çıkış arasındaki farkı hesaplamaktır ve bu fark değerine göre öğrenme algoritması ağın ağırlıklarını güncellemektedir. Yukarıda bahsi geçen öğrenme algoritmaları arasında genelde çok-katmanlı ileri beslemeli YSA'lar için geriye yayılım algoritması kullanılmaktadır.

Geriye yayılım algoritması, kolaylıkla kullanılabilmesinden dolayı ağ eğitiminde tercih edilen en popüler algoritmalardan biridir (Basheer ve Hajmeer, 2000). Bu algoritmanın geriye yayılım ismini almasının sebebi hataları geriye doğru diğer bir deyişle çıkıştan girişe doğru azaltmaya çalışmasıdır. Aynı zamanda, hata hesaplama fonksiyonlarının türevini alarak ağın parametrelerini güncellemesinden dolayı türev tabanlı öğrenme algoritması olarak da bilinmektedir. Geriye yayılım öğrenme algoritması tasarlanan ağ modelinin mevcut çıkış hatasına göre katmanlar arasında bulunan ağırlıkları yeniden hesaplamak için kullanılır. Geriye yayılım öğrenme algoritmasını kullanan ağ modeli, geriye yayılımlı ağ modeli olarak bilinmektedir. Geriye yayılımlı ağ modelleri giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç katmandan oluşmaktadır. Problemin çözüm planına göre oluşturulan ağda gizli katman sayısı ve gizli katmanda bulunan nöronların sayıları artırılabilir. Geriye yayılım öğrenme algoritması, çok-katmanlı ileri beslemeli YSA'nın yapısında bulunan ağırlıkların en uygun ağırlık değerlerini bulmayı hedeflemektedir. Bir geriye yayılımlı çok-katmanlı ileri beslemeli YSA'da öğrenme aşaması genel olarak dört adımdan oluşmaktadır (Kriesel, 2007).

1. İleriye beslemeli hesaplama işlemi,
2. Çıkış katmanı için gradyan hesaplama işlemi,
3. Gizli katman için gradyan hesaplama işlemi,
4. Ağda bulunan ağırlıkların güncellenmesi.

Ağda gradyan hesaplamaları sağdan sola olmak zorundadır. Fakat ağırlıkların güncellenmesi için her hangi bir yön zorunluluğu yoktur. Yukarıda sıralanmış olan adımlar ya belirli bir iterasyon sayısına kadar ya da ağın hata oranı belirli bir hata oranına eşit veya altına düşene kadar devam etmektedir. Yukarıdaki geriye yayılım öğrenme adımlarını sigmoid aktivasyon fonksiyonuna göre işlediğimizde (Gershenson, 2003),

Birinci adımda, çok-katmanlı ileri beslemeli YSA'da bulunan nöronların çıkışları ileriye beslemeli topolojisine göre elde edilir. Aynı zamanda her nöronda aktivasyon fonksiyonun türevi alınarak kayıt edilir.

İkinci adımda, çok-katmanlı ileri beslemeli YSA'nın çıkış katmanında bulunan nöronların gradyan hataları hesaplanır. Gradyan denklemleri aktivasyon fonksiyonların türevine göre elde edilmektedir. Denklem 3.23'te sigmoid aktivasyon fonksiyonun gradyan denklemi verilmiştir.

$$\delta_x = O_x(1 - O_x)(O_x - T_x) \tag{3.23}$$

Üçüncü adımda, çok-katmanlı ileri beslemeli YSA'nın gizli katmanında bulunan nöronların gradyan hataları hesaplanır. Denklem 3.24'te gizli katman için sigmoid aktivasyon fonksiyonun gradyan denklemi verilmiştir.

$$\delta_j = OH_j\left(1 - OH_j\right)\sum_{x=1}^{n} \delta_x * W_{j,x} \tag{3.24}$$

Dördüncü adımda ise, çok-katmanlı ileri beslemeli YSA'da bulunan tüm ağırlıkların güncellemesi işlemi gerçekleşir. Çok-katmanlı ileri beslemeli YSA'da bulunan tüm ağırlıklar Denklem 3.25 ve 3.26'ya göre güncellenir.

$$\Delta W = -\eta\delta^t O^{t-1} + \alpha \tag{3.25}$$

$$W_{yeni} = \Delta W + W_{eski} \tag{3.26}$$

Adımlar sonucu, belirli bir eşik hata değerine veya iterasyon sayısına ulaşana kadar, çok-katmanlı ileri beslemeli YSA'da elde edilen yeni ağırlık değerlerine göre adımlar tekrar baştan sona doğru yapılmaktadır. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasında, geriye yayılım öğrenme algoritması sıklıkla kullanılmasına rağmen sınıflandırma performansını etkileyen iki önemli dezavantaja sahiptir. Çok-katmanlı ileri beslemeli YSA katmanlarında eğer karmaşık bir aktivasyon fonksiyonu kullanılırsa, bu aktivasyon fonksiyonunun türevi karmaşık bir işlem olduğundan dolayı ağın öğrenme hızını yavaşlatabilir. Aynı zamanda arama uzayında ağırlıkların en uygun değerleri aranırken yerel minimumlara takılabilir (Leung ve Haykin, 1991).

3.4. Sınıflandırma Algoritmalarının Performanslarını Değerlendirme Ölçütleri

Bu bölümde sınıflandırma algoritmalarının performanslarının değerlendirilme yöntemleri açıklanmıştır. Performans değerlendirmesi, sınıflandırma modeli geliştirme sürecinin ayrılmaz bir parçasıdır. Verilerimizi temsil eden en iyi sınıflandırma modelini bulmaya ve seçilen modelin gelecekte ne kadar iyi çalışacağına yardımcı olur (Wong, 2015). Eğitimde kullanılan verilerle sınıflandırma modelinin performansının değerlendirilmesi, veri madenciliğinde uygun değildir. Çünkü modelin aşırı öğrenmesine sebep olabilir. Veri madenciliğinde Hold-Out ve Çapraz Doğrulama (Cross-Validation) gibi birçok model değerlendirme yöntemleri bulunmaktadır. Bu yöntemlerden K-Katlamalı Çapraz Doğrulama (K-Fold Cross-Validation) yöntemi sıklıkla kullanılmaktadır (Arlot ve Celisse, 2010). Aşırı öğrenmeyi önlemek için, model değerlendirme yöntemleri ile modelin performansını değerlendirmek için bir test seti (model tarafından görülmeyen örnekler) kullanır. Oluşturulan modellerin sınıflandırma başarılarını birbiri ile karşılaştırmak çok önemlidir ve sonuçta en iyi performans başarısı elde eden sınıflandırma modeli seçilir. Veri madenciliğinde farklı sınıflandırma modellerinin değerlendirilmesinde karmaşıklık matrisi (Confusion Matris) adında bir matris kullanılır. Bu matrisin içinde sınıflandırma modelinin doğru ve yanlış sınıflandırdığı adetler bulunmaktadır (Wong, 2015).

3.4.1. K-Katlamalı çapraz doğrulama

K-katlamalı çapraz doğrulama yöntemi model değerlendirmesi için çok önemli bir yöntemdir. K-katlamalı çapraz doğrulama yöntemi modelin geliştirilme aşamasında aşırı öğrenmeyi (overfiting) ve eksik öğrenmeyi (underfiting) tespit eder (Singh ve ark., 2018) ve modelin test edilme aşamasında en iyi modeli oluşturmayı hedefler (Arlot ve Celisse, 2010). Aşırı öğrenmede, modelin eğitim kümesindeki örüntüler yerine gözlemleri öğrenmesidir. Bu durumda eğitim aşaması için kullanılmış olduğunuz veri kümesini öğrenirsiniz, ancak oluşturulan model yeni gelen gözlemler ile karşılaştığında başarılı bir tahmin yapmayabilir. Genelde aşırı öğrenme modelleri eğitim aşamasını küçük hata oranı değeri ile tamalar, fakat test aşamasında ise büyük bir hata oranı ile tahmin işini yapar (Arlot ve Celisse, 2010). Eksik öğrenmede, modelin gözlemlerdeki örüntüyü eksik bir şekilde öğrenmesidir. Bu durumda eğitim aşaması için kullanılan veri kümesini başarılı bir şekilde öğrenemez. Dolayısıyla eksik öğrenme modelleri, eğitim ve test aşamalarını

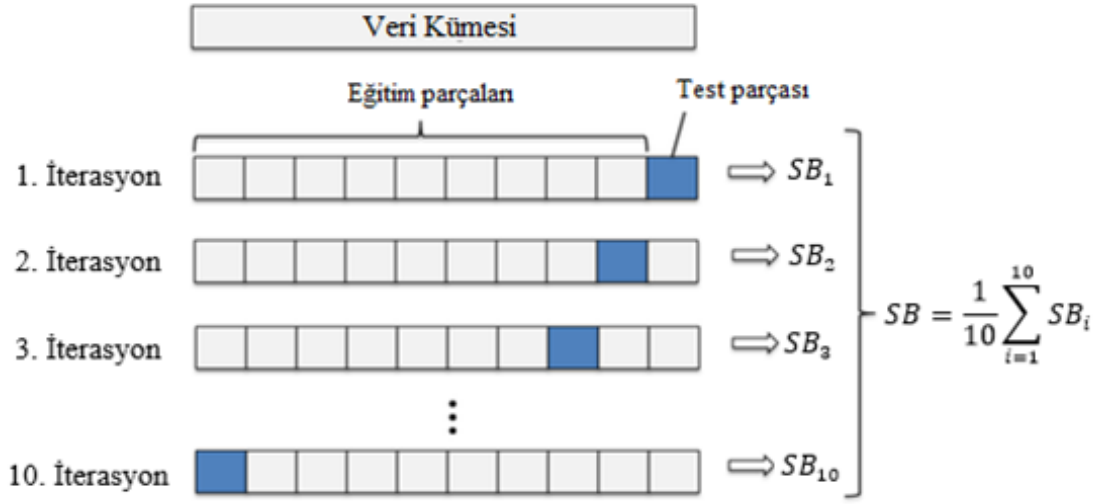
küçük hata oranı değeri ile tamalar. Sınıflandırma modellerinde aşırı öğrenmeyi ve eksik öğrenmeyi önlemek için k-katlamalı çapraz doğrulama, erken durdurma ve budama gibi birçok yöntem geliştirilmiştir. Bu yöntemlerin arasında genelde k-katlamalı çapraz doğrulama yöntemi kullanılmaktadır.

K-katlamalı çapraz doğrulama yönteminde, eğitim sürecinde kullanılan eğitim kümesi önce karıştırılır daha sonra eşit büyüklükteki k alt kümelere bölünür. Her iterasyonda sıradaki alt küme eğitim veri kümesinden çıkarılır ve test kümesi olarak kullanılır. K-katlamalı çapraz doğrulama yönteminde eğer k sayısı örnek büyüklüğüne eşitse, buna "leave-one-out" denir (Wong, 2015).

K-katlamalı çapraz doğrulama, modele gelen verileri rastgele parçalara böler, bu parçalara parça (fold) adı verilir. Model, test edilmek üzere 1. parçadaki verileri bir kenara bırakır (buna bazen “holdout fold” denir) ve kalan k-1 tane parçayı eğitim için kullanır. Örneğin, eğer k değeri 5 olursa veriler rastgele 5 parçaya bölünecektir, verilerin 4/5’i kullanılarak model eğitilir ve kalan 1/5 üzerinde model test edilir. Her parça için modelin testi sırasında doğruluk istatistikleri değerlendirilir. Hangi istatistik ölçütlerin kullanılması gerekliliği değerlendirmekte olan modelin türüne bağlıdır. Örneğin sınıflandırma modelleri için sınıflandırma başarıları, karmaşıklık matrisi ve hata oranları gibi ölçütler kullanılır. Tüm parçalar için değerlendirme süreci tamamlandığında, çapraz doğrulama modeli tüm veriler için bir performans ölçütü oluşturur ve sonuçlar üretir (Wiens ve ark., 2008). Sınıflandırma algoritmalarında sınıflandırma başarıları önemli olduğundan dolayı genelde her test veri kümesi parçası için Sınıflandırma Başarısı (SB) ölçütü hesaplanmaktadır. Test veri kümesi için SB denklemi Denklem 3.27’de verilmektedir.

$$SB_i = \frac{Doğru\ sınıflandırılmış\ örnek\ sayısı}{Test\ veri\ kümesindeki\ örnek\ sayısı} * 100 \qquad ve\ i = 1, 2, \dots, k \qquad (3.27)$$

Sınıflandırma algoritmalarının değerlendirme aşamasında aşırı öğrenmeyi (overfiting) ve eksik öğrenmeyi (underfiting) engellemek ve daha iyi bir sınıflandırma modeli oluşturmak için genelde 10-katlamalı çapraz doğrulama yöntemi kullanılmaktadır. 10-katlamalı çapraz doğrulama yöntemi Şekil 3.22’de ayrıntılı olarak gösterilmiştir.



Şekil 3.22. 10-katlamalı çapraz doğrulama

Bir sınıflandırma algoritması birden çok çalıştırıldığında ve her çalışmada sınıflandırma algoritmasının SB'si hesaplandığında, bu SB değerlerinin birbirleri ile tutarlı olması sınıflandırma algoritması için büyük önem taşımaktadır. Birden çok çalıştırdıktan sonra elde edilen SB değerlerinin tutarlılığı veri analizi ölçütleri merkezi dağılım standart sapma yöntemi ile hesaplanmaktadır. Standart sapma, birden fazla kez çalışmalardan elde edilen sınıflandırma başarılarının birbirine ne kadar benzer veya yakın olup olmadığını göstermektedir. Merkezi dağılım ölçütü olan standart sapma Denklem 3.28'e göre hesaplanmaktadır.

$$\sigma = \sqrt[2]{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.28}$$

Denklem 3.28'de, σ standart sapmayı, n kaç kez çalıştırıldığını, $\bar{x}$ SB değerlerinin aritmetik ortalamasını ve x_i i'inci SB değerini ifade etmektedir. Standart sapma n kez çalıştırma sonucu elde edilen SB değerlerinin birbirine ne kadar uzak veya yakın olduğunu göstermektedir.

3.4.2. Karmaşıklık matrisi (Confusion Matrix)

Bu bölümde veri madenciliğinde bir sınıflandırma modelinin performansının değerlendirilmesini sağlayan karmaşıklık matrisi ve karmaşıklık matrisinden elde edilen doğruluk (Accuracy), duyarlılık (Sensitivity veya Recall), özgüllük, hassasiyet (Precision), ROC eğrisi altında kalan alan ve F1-Score ölçütlerinin kısa bir açıklaması sunulmaktadır (García ve ark., 2017). Sınıflandırma algoritmaları eğitim veri kümesini kullanarak sınıflandırma modelini oluşturduktan sonra, bu sınıflandırma modelinin ne kadar iyi olup olmadığı test edilmelidir. Dolayısıyla, sınıflandırma modellerinin değerlendirilmesi, sınıflandırma modellerinden elde edilen tahmin değerlerinin ne kadar iyi olduğunu belirleyen en önemli görevdir. Bir sınıflandırma modelinin değerlendirme ölçütleri karmaşıklık matrisi üzerinden elde edilir. Karmaşıklık matrisi, gerçek değerlerin bilindiği bir test veri kümesinde bir sınıflandırma modelinin performansını tanımlamak için sıklıkla kullanılan bir tablodur. İki sınıflı bir sınıflandırma modelinin karmaşıklık matrisi Çizelge 3.5'te verilmiştir.

Çizelge 3.5. Karmaşıklık matrisi

		Tahmini sınıf	
		Evet	Hayır
Gerçek sınıf	Evet	Doğru Pozitif	Yanlış Negatif
	Hayır	Yanlış Pozitif	Doğru Negatif

Doğru pozitif ve doğru negatifler sınıflandırma modeli tarafından doğru tahmin edilen pozitif ve negatif test örnekleridir. Yanlış negatif ve yanlış pozitif test örnekleri ise sınıflandırma modeli tarafından yanlış tahmin edilen pozitif ve negatif test örnekleridir. Yanlış pozitif ve yanlış negatif değerler gerçek sınıfın öngörülen sınıfla çeliştiğinde ortaya çıkar. Sınıflandırma modellerinde en önemli amaç yanlış negatif ve yanlış pozitif değerlerini en aza indirgemektir. Bu terimler biraz kafa karıştırıcıdır ve detaylı açıklaması aşağıda verilmektedir (García ve ark., 2017).

Doğru Pozitifler (DP): DP, verilen örneğin gerçek sınıf değeri evet olduğunda ve sınıflandırma modeli tarafından tahmin edilen sınıf değeri de evet olduğu anlamına gelen doğru tahmin edilen pozitif değerlerdir.

Doğru Negatifler (DN): DN, doğru tahmin edilen negatif değerlerdir, verilen örneğin gerçek sınıf değeri hayır olduğunda ve sınıflandırma modeli tarafından öngörülen sınıf değeri de hayır olduğu anlamına gelir.

Yanlış Pozitifler (YP): Verilen örneğin gerçek sınıf değeri hayır olduğunda ve sınıflandırma modelinde tahmin edilen sınıf değeri evet olduğunda ortaya çıkar. Örnek olarak eğer gerçek sınıf, bu oyuncunun oyun oynamadığını ancak sınıflandırma modeli tarafından tahmin edilen sınıf bilgisi size bu oyuncunun oyun oynayabilir olduğu sonucuna varırsa yanlış pozitif bir tahminde bulunulmuş olur.

Yanlış Negatifler (YN): Verilen örneğin gerçek sınıf değeri evet olduğunda ve sınıflandırma modelinde tahmin edilen sınıf değeri hayır olduğunda ortaya çıkar. Örnek olarak eğer gerçek sınıf, bu oyuncunun oyun oynayabilir ancak sınıflandırma modeli tarafından tahmin edilen sınıf bilgisi size bu oyuncunun oyun oynamadığının sonucuna varırsa. Sınıflandırma modeli yanlış negatif bir tahminde bulunulmuş olmaktadır.

İki sınıflı karmaşıklık matrisinde bulunan DP, DN, YP ve YN değerleri elde edildikten sonra bir sınıflandırma modelinin doğruluk, duyarlılık, özgüllük, hassasiyet, ROC eğrisi altında kalan alan ve F1-Score olan değerlendirme ölçütlerini hesaplayabiliriz (Rong ve ark., 2014).

Doğruluk (ACC): En sezgisel (intuitive) performans ölçüsüdür ve doğruca tahmin edilen örnek sayılarının toplam örnek sayılarına oranıdır. Yüksek doğruluk değerine sahip olan sınıflandırma modelinin iyi bir model olduğunu düşünebilir. Doğruluk önemli bir ölçüt olabilir, ancak yanlış pozitif ve yanlış negatif değerlerinin neredeyse aynı olduğu simetrik veri kümeleri olduğunda modelin performansını değerlendirmek için diğer parametrelere de bakmakta fayda vardır. Doğruluk denklemi Denklem 3.29'da verilmiştir.

$$Doğruluk\ (ACC) = \frac{DP\ +\ DN}{DP\ +\ YP\ +\ YN\ +\ DN} \tag{3.29}$$

Duyarlılık (SEN): Doğru tahmin edilen pozitif örnek sayısının gerçek sınıftaki tüm örnek sayılarına oranıdır. Duyarlılık değeri düşük oranda yanlış negatif değeri ile yükselir. Duyarlılık, doğru pozitif oranı olarak bilinmektedir. Duyarlılık denklemi Denklem 3.30'da verilmiştir.

$$Duyarlılık\ (SEN)\ = \frac{DP}{DP\ +\ YN} \tag{3.30}$$

Özgüllük (SPE): Doğru tahmin edilen negatif örneklerin sayısının toplam tahmini negatif örnek sayısına oranıdır. Yüksek Özgüllük değerine sahip olan sınıflandırma modellerinde yanlış pozitif oranı düşük olmaktadır. Özgüllük, doğru negatif oranı olarak bilinmektedir. Özgüllük denklemi Denklem 3.31'da verilmiştir.

$$\text{Özg}\ddot{u}llük\ (SPE)\ = \frac{DN}{DN\ +\ YP} \tag{3.31}$$

F-Measure veya F1-Skoru: F1-M, duyarlılık ve özgüllük değerlerinin ağırlıklı ortalamasıdır. Dolayısıyla, bu değer hem yanlış pozitif hem de yanlış negatifleri dikkate alır. Sezgisel olarak doğruluk olarak anlaşılması kolay değildir, ancak özellikle düzensiz bir sınıf dağılımınız varsa, F1-Skor genellikle doğruluktan daha yararlıdır. Eğer doğru pozitif ve yanlış negatif değerler benzer maliyetlere sahipse, doğruluk en iyi şekilde çalışır. Yanlış pozitif ve yanlış negatiflerin maliyeti çok farklıysa, hem duyarlılık hem de özgüllük değerlerine bakmak daha iyidir. Dolayısıyla, bir model oluşturduğumuzda, bu parametrelerin ne anlama geldiğini ve modelimizin ne kadar iyi performans gösterdiğini anlamamıza yardımcı olabilir. F1-M denklemi Denklem 3.32'de verilmiştir.

$$F1-M = 2\ *\frac{\text{Özg}\ddot{u}llük\ *\ Duyarlılık}{\text{Özg}\ddot{u}llük\ +\ Duyarlılık} \tag{3.32}$$

3.5. Meta-Sezgisel Optimizasyon Algoritmalar

Meta-sezgisel optimizasyon algoritmalar, optimizasyon problemlerinin çözümünde sıklıkla kullanılır. Optimizasyon temel olarak ekonomi ve mühendislik alanları başta olmak üzere birçok alanda yer almaktadır. Para ve zaman faktörleri her zaman sınırlı olduğundan dolayı bu faktörlerin en uygun kullanımı çok önemlidir (Gandomi ve ark., 2013).

Gerçek dünyadaki optimizasyon problemlerinin birçoğu karmaşık kısıtlar altında çok modelli ve doğrusal olmayan problemlerdir. Bu tür optimizasyon problemlerinin hedefleri farklı olduğundan dolayı, bazen en uygun çözümler bulunamayabilir veya bulmak çok uzun sürebilir. Genel olarak belirli bir optimizasyon problemi için optimal

çözümü bulmak veya hatta optimale yakın çözümleri bulmak kolay bir görev değildir (Beheshti ve Shamsuddin, 2013). Optimizasyon en basit anlamıyla, minimizasyon veya maksimizasyon problemi olarak düşünülebilir. Bu tür problemlerde belirli veya belirsiz bir arama uzayında en uygun çözümler aranmaktadır. Bu arama işlemi ise meta-sezgisel veya optimizasyon algoritmalar ile gerçekleştirilebilir. Meta-sezgisel optimizasyon algoritmalar, yineleme sayısı veya hata oranı gibi durma ölçütleriyle karşılaşıncaya kadar yinelemeli olarak optimum çözümü bulmaya çalışırlar (Lee ve Geem, 2005). Genel olarak, meta-sezgisel algoritmalar rastgele bir popülasyon ile başlatılır ve her adımda popülasyondaki bireyler meta-sezgisel algoritmaya özgü matematiksel denklemler ile geliştirilmeye çalışılır.

Ağırlık tabanlı sınıflandırma modellerinde en uygun ağırlık değerlerinin bulunması da bir optimizasyon problemleridir. En uygun ağırlıkların bulunması için araştırmacılar tarafından literatürde farklı sezgisel ve meta-sezgisel optimizasyon algoritmalar geliştirilmiştir (Mirjalili ve ark., 2015). Bu sezgisel ve meta-sezgisel optimizasyon algoritmalarının kendilerine göre avantajlar ve dezavantajları bulunmaktadır. Bu tez çalışmasında bir meta-sezgisel optimizasyon algoritması olan PSO algoritmasının geliştirilmiş versiyonu, çok-katmanlı ileri beslemeli YSA'nın en uygun ağırlık değerleri bulmak için kullanılmıştır.

3.5.1. Parçacık sürü optimizasyon algoritması

PSO algoritması, kuşların doğal davranışından esinlenerek Kennedy ve Eberhart tarafından geliştirilen bir meta-sezgisel optimizasyon algoritmasıdır (Eberhart ve Kennedy, 1995; Shi, 2001). Global arama özelliği sayesinde doğrusal olmayan problemlerin çözümünde çok başarılı sonuçlar vermektedir. PSO, çok değişkenli fonksiyonların optimizasyonu, görüntü işleme, çizelgeleme, kümeleme, çok-katmanlı ileri beslemeli YSA'nın eğitimi, bulanık mantık sistemleri gibi birçok alanda yaygın olarak kullanılmaktadır. PSO algoritması belli sayıda parçacıklardan oluşmaktadır $X = x_1, x_2, x_3, \dots x_N$ burada N parçacık sayısını ifade etmektedir, her bir parçacık belirli bir çözüm sunar. Sürüdeki her bir parçacık hız (velocity) ve konum (position) bilgilerine sahiptir, d-boyutlu bir problem için parçacıkların hızları $V_i = v_{i1}, v_{i2}, v_{i3}, \dots v_{id}$ ve $i = 1,2,3, \dots, N$ şeklinde ifade edilir, parçacıkların konumları ise $X_i = x_{i1}, x_{i2}, x_{i3}, \dots x_{id}$ ve $i = 1,2,3, \dots, N$ şeklinde ifade edilmektedir. Buna göre N parçacıktan oluşan bir sürüdeki parçacıkların hızları ve konumları aşağıdaki matrisler şeklinde ifade edilir.

$$V = \begin{bmatrix} V_{11} & \cdots & V_{1D} \\ \vdots & \ddots & \vdots \\ V_{N1} & \cdots & V_{ND} \end{bmatrix}$$

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1D} \\ \vdots & \ddots & \vdots \\ X_{N1} & \cdots & X_{ND} \end{bmatrix}$$

Sürüdeki her bir parçacığın elde ettiği en iyi çözüm pbest olarak adlandırılır ve $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \dots, pbest_{id})$ ve $i = 1,2,3, \dots, N$ şeklinde ifade edilir. D-boyutlu bir problem için parçacıkların en iyi çözümleri aşağıdaki matris ile gösterilebilir.

$$pbest = \begin{bmatrix} pbest_{11} & \cdots & pbest_{1D} \\ \vdots & \ddots & \vdots \\ pbest_{N1} & \cdots & pbest_{ND} \end{bmatrix}$$

Tüm parçacıkların arasında en iyi çözüm ise gbest olarak adlandırılır ve $gbest = (gbest_1, gbest_2, gbest_3, \dots, gbest_N)$ şeklinde ifade edilir. Her bir parçacığın arama uzayına göre çözümleri, sahip oldukları hızlarına ve konumlarına göre elde edilmektedir. PSO rastgele üretilmiş belirli sayıda çözümle (parçacıkla) başlar ve parçacıkların konumları güncellenerek en uygun çözüm değeri araştırılır. Her bir iterasyon adımında parçacıkların konumları güncellenerek en uygun çözüm aranır. Hız bilgilerine göre parçacıkların hızları ve konumları sırasıyla Denklem 3.33 ve Denklem 3.34'e göre güncelenmektedir (Eberhart ve Kennedy, 1995).

$$v_{id}^{t} = w * v_{id}^{t-1} * c_1 * r_1(pbest_{id} - x_{id}^{t-1}) + c_2 * r_2(gbest_i - x_{id}^{t-1}) \quad (3.33)$$

$$x_{id}^{t} = x_{id}^{t-1} + v_{id}^{t} \quad (3.34)$$

t zaman adımında, v_{id}^{t} çözüm uzayındaki i'inci parçacığın yeni hızını, v_{id}^{t-1} i'inci parçacığın bir önceki hızını göstermektedir ve $V_i = v_{i1}, v_{i2}, v_{i3}, \dots v_{id}$, d ise problemin boyutunu ifade eder. w eylemsizlik ağırlığını (inertia weight), x_{id}^{t} i'inci parçacığın yeni konumunu, x_{id}^{t-1} i'inci parçacığın bir önceki konumunu göstermektedir ve $X_i = x_{i1}, x_{i2}, x_{i3}, \dots x_{id}$ ile ifade edilir. c_1 ve c_2 öğrenme faktörleridir ve genelde 2 ile 4

değerleri arasında bir değer almaktadır. r_1 ve r_2 değerleri [0,1] arasında üretilen rastgele sayılardır. $pbest_{id}$ i'inci parçacığın pbest değerini ve $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \dots, pbest_{id})$ yani o parçacığın o ana kadar ki en iyi uygunluk değerine sahip pozisyonu ifade eder. $gbest$ tüm parçacıklar içinde o ana kadar ki en iyi uygunluk değerine sahip parçacığın pozisyonu ifade eder ve $gbest = (gbest_1, gbest_2, gbest_3, \dots, gbest_N)$ ile ifade edilir. c_1, c_2 ve w değerleri PSO algoritmasının performansı üzerinde önemli rol oynamaktadır. c_1 değeri parçacıkların kendi lokal en iyilerine yönlendirirken, c_2 değeri ise global en iyiye doğru yönlendirmektedir ve w değeri ise lokal ve global arama arasında dengeyi sağlamaktadır.

PSO algoritmasının performansını etkileyen birden fazla giriş parametresi bulunmaktadır. Bunlar parçacık sayısı, parçacık boyutu, parçacık aralığı, eylemsizlik ağırlığı ve öğrenme faktörleridir.

Parçacık Sayısı: Sürüdeki parçacıkların sayıları probleme göre değişmektedir. PSO algoritması tarafından çözülen problemin karmaşıklığı arttıkça gerekli olan parçacık sayısı da artmaktadır. Genel olarak, birçok problemin çözümünde yaklaşık 20 veya 30 parçacık kullanılabilir.

Parçacık Boyutu: Parçacık boyutu optimize edilecek problemdeki amaç fonksiyonun değişken sayısını temsil etmektedir. Parçacık boyutları problemin karmaşıklığı artıkça artmaktadır ve bu artış en iyi çözümü elde etmeyi zorlaştırmaktadır.

Parçacık Aralığı: Optimize edilecek amaç fonksiyonundaki bulunan değişkenlerin aralıklarını ifade etmektedir ve aynı zamanda problemin arama uzayını temsil etmektedir. Bu değişkenlerin aralıkları probleme göre değişmektedir. Değişkenlerin aralıkları çok iyi bir şekilde seçilmelidir. Geniş bir arama uzayına sahip olan bir problemin meta-sezgisel algoritmalar tarafından çözümü uzun sürebilir, buna karşın arama uzayının dar olması meta-sezgisel algoritmaların en iyi çözümü bulacağını garanti etmez.

PSO algoritmasında parçacıkların iyi bir çözüm sunmaları aslında parçacıklarının hızlarına bağlıdır. PSO algoritmasında hız güncelleme fonksiyonuna bakıldığı zaman fonksiyonda birden fazla önemli faktör bulunmaktadır.

Eylemsizlik Ağırlığı (Inertia Weight w): Bu parametre sayısında eski hız bilgisinin yeni hız bilgisine olan etkisi ayarlanmaktadır. Bu etki ayarlama sonucunda parçacıkların yerel ve küresel arama arasındaki denge sağlanır ve sürüdeki diğer tüm parçacıkların tecrübelerinden yararlanarak daha az iterasyonla en uygun sonuca varılır.

Öğrenme Faktörleri (c_1, c_2): Sürüdeki her bir parçacığı $pbest$ ve $gbest$ değerlerine göre hareket ettirmesini sağlamaktadır. Bu öğrenme faktörlerinin değerleri 2 ile 4 arasında pozitif sayılardır. c_1 değeri parçacığın kendi tecrübesine göre hareketini sağlarken c_2 değeri ise parçacığın tüm sürüdeki en iyi parçacığa göre hareket etmesini sağlamaktadır. PSO algoritmasının sözde kodu Şekil 3.23'te gösterilmiştir (Shi, 2001).

```
Begin
Parametreleri belirle (Parçacık sayıs = p, boyut = b, iterasyon sayısı = t,
arama uzayı aralığı, max iterasyon, w, c1 ve c2 değerleri)
Rastgele başlangıç popülasyon oluştur,
Reapet
        t = t + 1
        For i = 1 to p
            Amaç fonksiyonunun değerini hesapla
            Yerel en iyi konumu güncelle
        End For
        Küresel en iyi konumu güncelle
        For j = 1 to p
            3.33 ve 3.34 denklemlerine göre parçacıkların hızlarını ve
            konumlarını güncelle
        End For
Until (t > max iterasyon)
End
```

Şekil 3.23. PSO algoritmasının sözde kodu

4. ÖNERİLEN YAKLAŞIMLAR

Sınıflandırma algoritmalarının performansını etkileyen birden fazla faktör bulunmaktadır ve bu faktörler sınıflandırma algoritmalarının çalışma prensiplerine bağlıdır. Örneğin kural tabanlı sınıflandırma algoritmalarının birçoğu parametrik olmayan sınıflandırma algoritmaları kategorisi altında olduğundan dolayı performansı daha çok veri kümesinin ayrık veya sürekli olmasına göre değişebilir. Diğer yandan ağırlık tabanlı sınıflandırma algoritmaları parametrik sınıflandırma algoritmaları kategorisi altında olduğundan dolayı hem veri kümesi hem de sınıflandırma algoritmasında kullanılan parametreler sınıflandırma performansını etkilemektedir. Bu tezde ayrık tabanlı sınıflandırma algoritmaları için önerilen ayrıklaştırma yöntemi ve ağırlık tabanlı sınıflandırma algoritmaları için önerilen ağırlık güncelleme yöntemi aşağıdaki alt bölümlerde ayrıntılı olarak anlatılmıştır.

4.1. Önerilen Ayrıklaştırma Yöntemi: EF_Unique

Bu bölümde, kural tabanlı sınıflandırma algoritmalarının sürekli veriler üzerinde iyi bir performansla çalışması için EF_Unique adında yeni bir ayrıklaştırma yöntemi önerilmiştir. EF_Unique aralık sınırlarını parametrik olmayan yeni bir teknik kullanarak hesaplayarak veri madenciliği ayrıklaştırma yöntemlerinden olan EF ayrıklaştırma yönteminin performansını iyileştirmiştir. Denetimsiz ayrıklaştırma yöntemlerinde, k aralıklarının en uygun sayısının bulunması NP-zor bir problemdir (Garcia ve ark., 2013). Bu güne kadar, en iyi k aralık sayısı tahmini için birçok kurallar geliştirilmiştir. Geliştirilen kurallar kullanımlarına göre farklılık gösterebilir, yani tüm sürekli veri kümeleri için uygun bir kural yoktur. Önerilen EF_Unique ayrıklaştırma yöntemi iki ana adım ile sürekli verileri ayrıklaştırır. Birinci ana adımda, önce sürekli dizi değerleri artan veya azalan düzende sıralanır, daha sonra sürekli dizi elemanlarından tekrarlanan değerler diziden silinerek A' tekil dizisi elde edilir. Daha sonra, elde edilen A' tekil dizisinin eleman sayısının karekökü alınır ve bu sayı en yakın tam sayıya ayarlanarak k aralıklarının sayısı belirlenir. Bu nedenle, önerilen EF_Unique ayrıklaştırma yöntemi, her özniteliği farklı aralıklarla ayrıklaştırmasını sağlayabilir. İkinci ana adımda ise, birinci adımda elde edilen k sayısına göre A' tekil dizisinin parçaları ve her parçada kaç elemanın olduğu belirlenir. Ardından, aralıkların sınırlarını belirlemek için parçaların aritmetik ortalamaları hesaplanır. Son olarak, özniteliğin sürekli değerleri, ait oldukları aralığı

belirleyerek ayrık değerlere dönüştürülür. Önerilen EF_Unique ayrıklaştırma yönteminin algoritması Şekil 4.1'de verilmektedir.

Giriş: Dizideki sürekli öznitelik değerleri $\boldsymbol{A} = \{a_1, a_2, \dots, a_{n-1}, a_n\}$.
Adım 1: $\boldsymbol{A}$ diziyi küçükten büyüğe veya büyükten küçüğe sırala,
Adım 2: $\boldsymbol{A}$ dizisinden tekrarlanmamış $\boldsymbol{A'} = \{a'_1, a'_2, \dots, a'_{m-1}, a'_m\}$ dizisini oluştur,
Adım 3: $\boldsymbol{k}$ sayısını $\boldsymbol{A'}$ dizisindeki eleman sayısının karekökünü alarak hesapla,
Adım 4: EF ayrıklaştırma yöntemine göre parçaları belirle ve her parçanın aritmetik ortalamasını hesapla,
Adım 5: Geçerli parçanın maksimum değerinin ve bir sonraki parçanın minimum değerinin ortalama değerini hesaplayarak her bir aralığın sınırlarını belirle,
Adım 6: $\boldsymbol{A}$ dizisindeki sürekli değerler aralıklara göre ayrık değerlere dönüştürülür.
Çıkış: Ayrık değerli $\boldsymbol{A}$ dizisi.

Şekil 4.1. Önerilen EF_Unique ayrıklaştırma yönteminin algoritması

Önerilen EF_Unique ayrıklaştırma yönteminde, ikinci, dördüncü ve beşinci adımlarının en kötü zaman karmaşıklığı *n*'dir, üçüncü adımının en kötü zaman karmaşıklığı ise sabit c_1'dir, n ise sürekli dizinin eleman sayısı ve k aralıkların sayısıdır. Bu nedenle, önerilen EF_Unique ayrıklaştırma yönteminin toplam zaman karmaşıklığı $3n + k + c_1$, büyük-O notasyon zaman karmaşıklığına göre önerilen ayrıklaştırma yönteminin zaman karmaşıklığı $O(n)$'dir (Stein ve ark., 2001; Alsuwaiyel, 2016). Birinci adımdaki dizinin sıralama işlemi algoritmanın çalışma zamanına dâhil edilmemiştir. Literatürdeki EG ve EF ayrıklaştırma yöntemlerinin zaman karmaşıklıklarının hesaplanmasında da sıralama işlemi çalışma zamanına dâhil edilmemektedir. Önerilen EF_Unique ayrıklaştırma yöntemi basit şekilde uygulanabilir ve her öznitelik için tekrarsız eleman sayısının karekökünü alarak farklı sayıda aralık sayısı belirleyebilir.

Önerilen EF_Unique ayrıklaştırma yönteminin geliştirme aşamalarında, EG, EF ve ET_ID3 ayrıklaştırma yöntemleri çok ayrıntılı olarak 20 UCI sınıflandırma veri kümeleri üzerine uygulanmış ve incelenmiştir. İnceleme aşamasında veri kümelerinin özniteliklerinin ET_ID3 ayrıklaştırma yöntemi tarafından kaç tane aralığa bölündüğü analiz edilmiştir. Analiz sonucunda EG ve EF ayrıklaştırma yöntemlerinin kullanıcı tarafından sağlanan aralık sayısı, aykırı değerlere duyarlı olması ve aynı veriyi iki farklı aralığa yerleştirmesi gibi dezavantajları önerilen EF_Unique ayrıklaştırma yönteminde ortadan kaldırılmıştır. Diğer yandan ET_ID3 ayrıklaştırma yönteminin veri kümesindeki her özniteliği farklı aralıklara bölme avantajı, önerilen EF_Unique ayrıklaştırma

yöntemine eklenmiştir. EF_Unique ayrıklaştırma yönteminin örnek uygulaması Örnek 4.1'de verilmektedir.

Örnek 4.1:

Giriş: Dizideki sürekli öznitelik değerleri $A = \{20, 30, 20, 10, 80, 100, 10, 20, 10\}$

Adım 1: Sıralama işlemi $A = \{10, 10, 10, 20, 20, 20, 30, 80, 100\}$

Adım 2: $A' = \{10, 20, 30, 80, 100\}$

Adım 3: $k = round\left(\sqrt{length(A')}\right) = round\left(\sqrt{5}\right) = round(2.236) = 2$

Adım 4: EF ayrıklaştırma yöntemine göre parçaları belirle ve her parçanın aritmetik ortalamasını hesapla,

$\text{Parça}_1 = \{10,20,30\}$

$Aritmetik\ ortalama = (10 + 20 + 30)/3 = 20$

$\text{Parça}_2 = \{80, 100\}$

$Aritmetik\ ortalama = (80 + 100)/2 = 90$

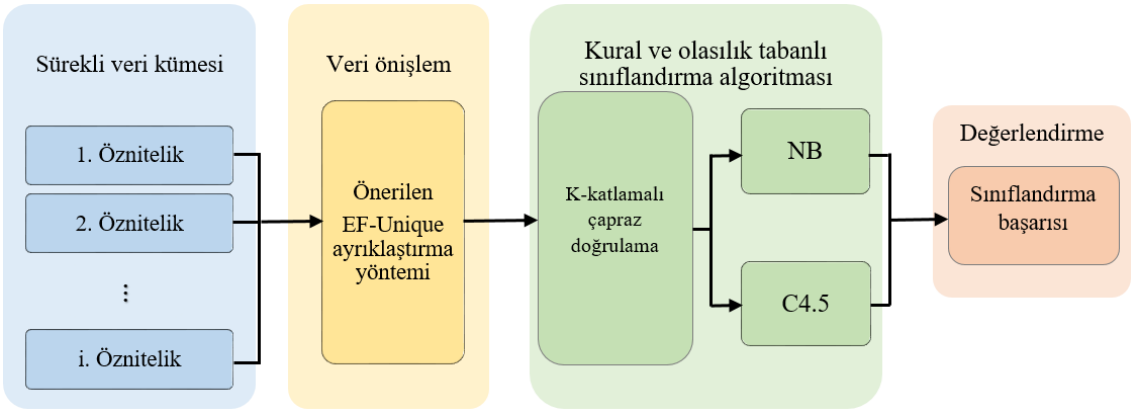
Adım 5: $Aralık_1 = [10, (20 + 90)/2) = [10,55)$

$Aralık_2 = [55,100]$

Adım 6: $A = \{[10, 55), [10, 55), [10, 55), [10, 55), [55, 100], [55, 100], [10, 55), [10, 55), [10, 55)\}$

Örnek 4.1'de görüldüğü gibi, önerilen EF_Unique ayrıklaştırma yöntemi kullanıcı tarafından hiçbir parametre istemeden sürekli özniteliklerin ayrıklaştırma işlemini gerçekleştirmiştir. Aralıkların sayısı ve sınırları özniteliğin tekrarsız dizi elemanlarının sayısının karekökü alınarak EG ve EF denetimsiz ayrıklaştırma yöntemlerinin en büyük dezavantajını ortadan kaldırmış olmaktadır. Önerilen EF_Unique ayrıklaştırma yöntemi aykırı değerlere hassa değildir, aynı değerleri veya birbirlerine yakın olan değerleri iki farklı aralıklara yerleştirmez ve sürekli veriler üzerinde basit ve kolay bir şekilde uygulanmaktadır.

Böylelikle, önerilen EF_Unique ayrıklaştırma yöntemi, diğer denetimsiz ayrıklaştırma yöntemlerinin dezavantajlarını ortadan kaldırmış olmaktadır. Önerilen EF_Unique ayrıklaştırma yönteminin kural tabanlı sınıflandırma algoritmalarına uygulanmasının grafiksel gösterimi Şekil 4.2'de verilmiştir.



Şekil 4.2. Önerilen EF_Unique ayrıklaştırma yönteminin kural ve olasılık tabanlı sınıflandırma algoritmalarına uygulanması

Şekil 4.2'de gösterilen önerilen EF_Unique ayrıklaştırma yönteminin kural tabanlı sınıflandırma algoritmalarına uygulanması aşağıda adım adım açıklanmaktadır.

Adım 1. Veri kümesinde ön işleme: Kural tabanlı sınıflandırma algoritmaları sürekli veri kümesi üzerinde iyi bir performans sergilemediğinden dolayı sürekli veri kümesi önerilen EF_Unique ayrıklaştırma yöntemi ile ayrıklaştırılmaktadır.

Adım 2. Sınıflandırma için veri kümesinin organizasyonu: Bu tez çalışmasında, veri kümeleri iki farklı deney için iki farklı şekilde düzenlenmiştir. Deneyde, önerilen EF_Unique ayrıklaştırma yöntemini veri madenciliğinde sıkça kullanılan ayrıklaştırma yöntemleri ile karşılaştırmak için veri kümeleri 10-katlamalı çapraz doğrulama kullanılarak düzenlenmiştir.

Adım 3. Kural tabanlı sınıflandırma algoritmasının modellenmesi: Bu adımda kural tabanlı sınıflandırma algoritması ayrık veri kümesinden farklı yapılarda kurallar oluşturur. Daha sonra bu kuralları sınıf bilgileri belli olmayan örneklerin sınıflandırılması için kullanır.

Adım 4. Kural tabanlı sınıflandırma algoritmasının test edilmesi: Önerilen kural tabanlı sınıflandırma modeli eğitim veri kümesinden sınıflandırma kurallarını elde ettikten sonra sınıflandırma modelinin performansını belirlemek için, sınıflandırma modelinin performans değerlendirme ölçütleri hesaplanır.

4.1.1. Önerilen EF_Unique ve Sık kullanılan ayrıklaştırma yöntemlerinin karakteristik analizi

Her ayrıklaştırma yöntemi çalışma mekanizmasına göre farklı özelliklere sahiptir ve bu özellikler kullanıcının bir alan için uygun ayrıklaştırma yöntemini seçmesine yardımcı olur (Dash ve ark., 2011; Garcia ve ark., 2013). Ayrıklaştırma yöntemlerinin taksonomisinde, ayrıklaştırma yöntemlerinin özelliklerini birbirleriyle karşılaştırmak için çeşitli sınıflandırma ölçütleri bulunmaktadır. Bu sınıflandırma ölçütleri, denetimli/denetimsiz, bölmeli (yukarıdan aşağıya)/birleştirmeli (aşağıdan yukarıya), statik/dinamik, yerel/küresel ve doğrudan/artımlı gibi kategorilere ayrılabilir. Önerilen EF_Unique ayrıklaştırma yönteminin özelliklerini göstermek için, Çizelge 4.1'de verilen sınıflandırma ölçütlerine göre sık kullanılan EW, EF ve ID3 ayrıklaştırma yöntemleri ile karşılaştırılmıştır.

Çizelge 4.1. Önerilen EF_Unique ve sık kullanılan ayrıklaştırma yöntemlerinin karakteristik analizi

Sınıflandırma ölçütleri	**EW**	**EF**	**ET-ID3**	**EF_Unique**
Denetimli / Denetimsiz	Denetimsiz	Denetimsiz	Denetimli	Denetimsiz
Küresel / Yerel	Küresel	Küresel	Yerel	Yerel
Dinamik / Statik	Statik	Statik	Dinamik	Statik
Doğrudan / Artırmalı	Doğrudan	Doğrudan	Artırmalı	Artırmalı
Parametrik / Parametrik olmayan	Parametrik	Parametrik	Parametrik	Parametrik olmayan
Durdurma ölçütü	Kullanıcı tarafından verilen aralık sayısı	Kullanıcı tarafından verilen aralık sayısı	Eşik değer	EF_Unique tarafından elde edilen aralık sayısı
Farklı aralıklarda aynı değerler	Hayır	Evet	Hayır	Hayır
Aykırı değere hassasiyet	Evet	Hayır	Hayır	Hayır
Zaman karmaşıklığı	$O(n)$	$O(n)$	$O(n \log n)$	$O(n)$

Çizelge 4.1'e göre, önerilen EF_Unique ayrıklaştırma yöntemi aşağıdaki özellikleri taşımaktadır:

- Denetimsiz, önerilen ayrıklaştırma yöntemi ayrıklaştırma işlemi gerçekleştirirken veri kümesinin sınıf bilgisini kullanmaz.
- Veri kümesinin bilgilerini kullanarak k sayısının elde edilmesi önerilen ayrıklaştırma yöntemine yerel özelliği atamıştır.
- Statik, çünkü ayrıklaştırma işlemi bir öğrenme modeline dayanmamaktadır.

- Durma ölçütüne ulaşana kadar aralık sayısını artırdığı için önerilen ayrıklaştırma yöntemi artırmalı özeliğini almaktadır.
- Ayrıklaştırma işlemini gerçekleştirmesi için kullanıcıdan hiçbir parametre talebinde bulunmadığından parametrik olmayan bir ayrıklaştırma yöntemidir.
- Durdurma ölçütü bir özniteliğin benzersiz elemanlarının sayısının karekköküne göre belirler.
- Aynı değerleri farklı aralıklarda yerleştirmez.
- Aykırı değerlere duyarlı değildir.
- Önerilen EF_Unique ayrıklaştırma yönteminin zaman karmaşıklığı ise $O(n)$'dir.

4.2. Önerilen Çoklu Ortalama-PSO Meta-Sezgisel Optimizasyon Algoritması

Bu bölümde, ağırlık tabanlı sınıflandırma algoritmalarının performansını doğrudan etkileyen eğitim işlemi için önerilen çoklu sürü PSO meta-sezgisel optimizasyon algoritması anlatılmaktadır. Çoklu sürü optimizasyon (Multi-swarm optimization MSO), doğrusal olmayan sürekli optimizasyon problemlerine en uygun çözümü elde etmek için kullanılan bir meta-sezgisel optimizasyon tekniğidir. Problemlerin boyutları artıkça pek çok meta-sezgisel optimizasyon algoritmasının performansı ve verimliliği kötüleşir (Gülcü ve Kodaz, 2015). Bu problemin üstesinden gelmek için, büyük boyutlu problemlerin kısa sürede en uygun çözümleri elde eden MM-PSO adında bir meta-sezgisel optimizasyon algoritması geliştirilmiştir. Önerilen MM-PSO algoritması, PSO algoritmasından önerilen çok popülasyonlu bir meta-sezgisel optimizasyon algoritmasıdır. PSO algoritmasında, her parçacığın yeni konumu parçacığın güncellenen hızı ve parçacığın şu anki konumunun toplanması ile elde edilir. Hız yukarıda gösterilen Denklem 3.33'e göre her parçacığın en iyi çözümü ($pbest$) ve tüm parçacıkların en iyi çözümü ($gbest$) ile güncellenir (Shi, 2001). Önerilen MM-PSO algoritmasında ise parçacıklar çoklu sürüler şeklinde gruplara ayrılmaktadır. Her bir sürüdeki her parçacığın hızı, Denklem 4.1'e göre güncellenmektedir. Bu denklemde, her parçacığın hızı tüm sürüdeki en iyi çözümlerin ortalaması ($mpbest$), sürüdeki en iyi çözüm ($gbest$) ve tüm sürüdeki en iyi çözüm ($gsbest$) ile güncellenir. Hız denklemindeki bu değişiklik PSO algoritmasına iki avantaj getirmektedir. Birincisi,

$mpbest$ kullanmak, parçacıkların arama uzayı alanından çıkmasını önler ve her parçacığın yerel aramasını güçlendirir. İkincisi ise, her bir parçacık yalnızca kendi kümesinin en iyi çözümüne göre güncellenmez, aynı zamanda tüm kümelerin en iyi çözümünü de dikkate alarak güncellenir. Böylece, önerilen MM-PSO algoritması en iyi çözüme daha hızlı yaklaşabilir. d boyutlu bir problemde i'inci parçacığın konumu aşağıdaki Denklem 4.1'e göre güncellenir.

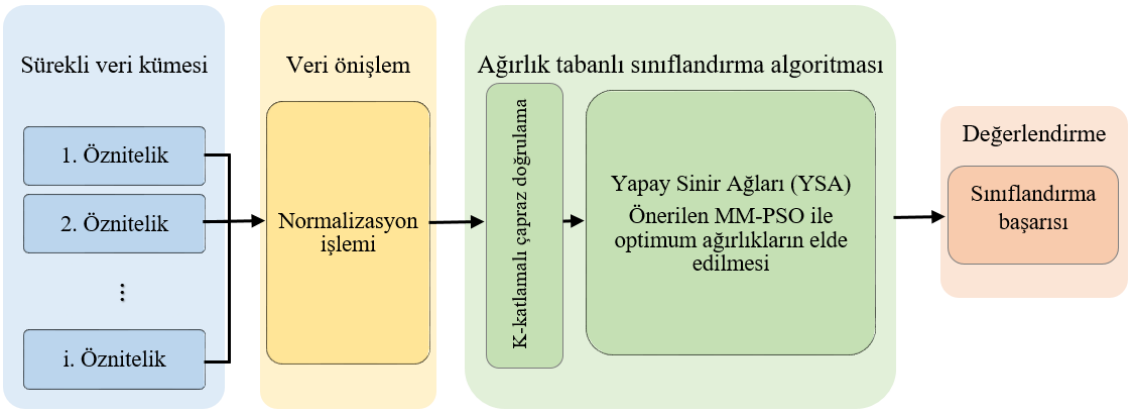
$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 r_1 (mpbest_i - x_{id}^{t}) + c_2 r_2 (gbest - x_{id}^{t}) + (gsbest - x_{id}^{t}) \qquad (4.1)$$

Burada, t ve w sırasıyla iterasyon sayısını ve atalet ağırlığını, v_{id}^{t+1} i'inci parçacığın şu anki iterasyondaki hızını, v_{id}^{t} i'inci parçacığın bir önceki iterasyondaki hızını, x_{id}^{t} i'inci parçacığın bir önceki iterasyondaki konumunu, c_1 ve c_2 0 ile 2 arasında hızlanma faktörünü temsil eden iki pozitif sabit. r_1 ve r_2 0 ile 1 arasında rastgele değerleri, $mpbest_i$ sürülerdeki en iyi çözümlerin ortalamasını, $gbest$ sürüdeki en iyi çözüm ve $gsbest$ tüm sürüdeki en iyi çözümü temsil etmektedir. Önerilen MM-PSO algoritmasının sözde kodu Şekil 4.3'te verilmiştir.

```
Begin
Parametreleri belirle (Parçacık sayıs = p, sürü sayısı = s boyut = b,
 iterasyon sayısı = t, arama uzayı aralığı, max iterasyon, w, c1 ve c2 değerleri)
Rastgele başlangıç popülasyon oluştur,
Reapet
        t = t + 1
        For i = 1 to s
           For j = 1 to p
             Amaç fonksiyonunun değerini hesapla
             Yerel en iyi konumu güncelle
             Küresel en iyi konumu güncelle
           End For
        End For
        Tüm sürünün en iyi konumunu güncelle
        For i = 1 to s
           For j = 1 to p
             Parçacıkların hızlarını ve konumlarını sırasıyla denklem
             4.1'e ve 3.34'e göre güncelle
           End For
        End For
Until (t > max iterasyon)
End
```

Şekil 4.3. Önerilen MM-PSO algoritmasının sözde kodu

Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmalarında en uygun ağırlık değerlerinin belirlenmesi doğrusal olmayan bir optimizasyon problemidir. Bu nedenle meta-sezgisel optimizasyon algoritmalar çok-katmanlı ileri beslemeli YSA'nın en uygun ağırlık değerlerini bulmak için kullanılabilir. Önerilen MM-PSO meta-sezgisel algoritmasın çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasına uygulanması Şekil 4.4'te verilmiştir.



Şekil 4.4. Önerilen MM-PSO meta-sezgisel algoritmasının çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasına uygulanması

Şekil 4.4'te önerilen MM-PSO meta-sezgisel algoritmasının çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasına uygulanması aşağıda adım adım açıklanmaktadır.

Adım 1. Veri kümesinde ön işleme: Veri ön işleme tekniği olan normalizasyon işlemi, sınıflandırılacak veri kümesine uygulanır. Böylece, veri kümesi çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması üzerinde daha düzenli ve uygun bir şekilde çalışır. Literatürde, birden fazla normalizasyon teknikleri bulunmaktadır. Bu tez çalışmasında normalizasyon işlemi, Denklem 4.2'de gösterilen min-max normalizasyon fonksiyonu kullanılarak yapılmıştır (Jayalakshmi ve Santhakumaran, 2011).

$$x' = \frac{x_i - Min(x_i)}{Max(x_i) - Min(x_i)} \tag{4.2}$$

Adım 2. Sınıflandırma için veri kümesinin organizasyonu: Bu tez çalışmasında, veri kümeleri iki farklı deney için iki farklı şekilde düzenlenmiştir. İlk deneyde, önerilen

çoklu-ortalama algoritmasını standart PSO algoritmasıyla karşılaştırmak için veri kümeleri 5-katlamalı çapraz doğrulama kullanarak düzenlenmiştir. İkinci deneyde ise, önerilen MM-PSO algoritmasını literatürdeki önceki çalışmalarla karşılaştırmak için veri kümesinin % 80 eğitim ve % 20'si test için kullanılmıştır.

Adım 3. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının yapısının modellenmesi: Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının yapısı giriş katmandaki nöron sayıları ve çıkış katmandaki nöron sayıları veri kümesinin özelliklerine göre belirlenir. Giriş katmandaki nöron sayısı, veri kümesinin öznitelik sayısına eşittir. Benzer şekilde, çıkış katmandaki nöron sayısı veri kümesinin sınıflarının sayısına eşittir. Gizli katmanın nöron sayıları belirli aralıkta denenerek uygun sayılar belirlenmiştir.

Adım 4. Önerilen MM-PSO algoritması ile çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının eğitimi: İyi eğitilmiş bir çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının en uygun ağırlıkları olmalıdır ve en uygun ağırlıkların belirlenmesi doğrusal olmayan bir optimizasyon problemidir. Meta-sezgisel optimizasyon algoritmasının yapısı gereği bu problemi çözmek için kullanılabilir. Genellikle meta-sezgisel optimizasyon algoritmaların başlangıç çözümleri rastgele bir popülasyonla başlar. Popülasyondaki her bir bireyin uygunluğu amaç fonksiyonuna göre hesaplanır. Buradaki meta-sezgisel optimizasyon algoritmasının amacı, sınıflandırma hatalarını en aza indirmektir. Bu nedenle, meta-sezgisel optimizasyon algoritması problemin çözüm uzayını yerel ve küresel olarak araştırır ve en iyi çözümü günceller.

Adım 5. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının test edilmesi: Önerilen MM-PSO algoritması ile eğitimi gerçekleşen çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının performansını belirlemek için, sınıflandırma modelinin performans değerlendirme ölçütleri hesaplanır.

5. DENEYSEL SONUÇLARI VE TARTIŞMA

Bu bölümde, tez çalışmasında kural tabanlı sınıflandırma algoritmaları için önerilen EF_Unique ayrıklaştırma yönteminin ve ağırlık tabanlı sınıflandırma algoritmaları için önerilen MM-PSO meta-sezgisel algoritmasının sınıflandırma algoritmaları üzerinde performanslarını gösteren deney sonuçları analiz edilmiştir.

5.1. Önerilen EF-Unique Ayrıklaştırma Yönteminin Deneysel Sonuçları ve Tartışmalar

Önerilen EF_Unique ayrıklaştırma yönteminin performansını değerlendirmek için, EG, EF ve ET-ID3 gibi veri madenciliğinde sıkça kullanılan ayrıklaştırma yöntemleri önerilen EF_Unique ayrıklaştırma yöntemiyle karşılaştırılmıştır. UCI makine öğrenme veri ambarından, Bölüm 3.1'de gösterildiği gibi farklı özelliklere sahip 17 veri kümesi kullanılmıştır. Önerilen ve EG, EF ve ET-ID3 ayrıklaştırma yöntemlerinin değerlendirilmesi NB, DVM, K-NN ve C4.5 sınıflandırma algoritmaları üzerinde sınıflandırma başarılarına göre gerçekleştirilmiştir. Önerilen EF_Unique ve EG, EF ve ET-ID3 ayrıklaştırma yöntemlerinin kodlaması Microsoft Visual Studio Ultimate 2017 C#.NET kullanılarak gerçekleştirilmiştir. Ayrık veri kümelerinin sınıflandırma işlemleri WEKA veri madenciliği aracı kullanılarak gerçekleştirilmiştir (Hall ve ark., 2009). Tüm deneyler, 8 GB belleğe sahip Intel Core i7 3840QM 2.80 GHz işlemcili bir bilgisayar kullanılarak gerçekleştirilmiştir.

Önerilen EF_Unique ayrıklaştırma yönteminin başarısını belirlemek için, denetimsiz EG ve EF yöntemleri için aralık sayısı optimize edilmiştir. Optimizasyon işleminde aralık sayısı, 3 ile 21 arasında ikişer artırılarak en yüksek sınıflandırma başarısına sahip olan aralık sayısı seçilmiştir. Optimizasyon işleminin sonucu olarak, EG için dokuz aralık sayısı ve EF için beş aralık sayısı en yüksek sınıflandırma başarısı ile seçilmiştir. NB, DVM, K-NN ve C4.5 sınıflandırma algoritmaları için elde edilen deney sonuçları, sırasıyla Çizelge 5.1, 5.2, 5.3 ve 5.4'te gösterilmiştir. İlk sütun veri kümelerini gösterirken, diğer sütunlarda EG, EF ve ET-ID3 ve önerilen EF_Unique ayrıklaştırma yöntemleri ile ayrıklaştırılan veri kümelerinin sınıflandırma başarılarını verilmektedir.

Çizelge 5.1. NB sınıflandırma algoritmasının sınıflandırma başarısı

Veri kümesi	NB sınıflandırma başarıları %			
	EG	**EF**	**ET-ID3**	**EF_Unique**
Iris	**96.00**	94.00	94.00	93.33
Wine	96.07	97.75	**98.88**	97.75
Glass Identification	68.22	64.49	**74.30**	68.69
Thyroid	93.02	94.42	96.28	**96.74**
Heart	84.44	**84.81**	83.33	84.44
E.coli	80.36	82.74	**87.46**	86.17
Bupa	**65.80**	64.06	63.19	64.06
Australian Credit	85.36	85.36	**85.51**	85.36
Breast Cancer	**97.14**	97.00	97.00	97.07
Blood	76.20	**76.74**	75.40	76.34
Pima Indians Diabetes	75.13	74.22	**77.86**	74.74
Vehicle	**63.00**	57.80	62.65	**63.00**
German	75.30	75.00	75.80	**76.50**
Wine Quality Red	**59.16**	55.16	58.41	58.22
Spambase	83.83	89.09	**90.20**	89.95
Magic Gamma Telescope	75.18	75.30	**78.30**	77.90
Bank Marketing	88.47	87.62	88.45	**89.16**

Çizelge 5.1'deki sınıflandırma sonuçları incelendiğinde, NB sınıflandırma algoritmasında önerilen EF_Unique ayrıklaştırma yöntemiyle Wine, Glass Identification, Thyroid, Heart, E.coli, Blood, German, Spambase, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 97.75, 68.69, 96.74, 84.44, 86.17, 76.34, 76.34, 76.50, 89.95, 77.90 ve 89.16 sonuçları elde edilmiş ve bu sonuçlarla EG ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı elde ettiği görülmüştür.

Önerilen EF_Unique ayrıklaştırma yöntemi Glass Identification, Thyroid, Heart, E.coli, Bupa, Breast Cancer, Pima Indians Diabetes, Vehicle, German, Wine Quality Red, Spambase, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 68.69, 96.74, 84.44, 64.06, 97.07, 74.74, 63.00, 76.50, 58.22, 89.95, 77.90 ve 89.16 sonuçları ile EF ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Önerilen EF_Unique ayrıklaştırma yöntemi Thyroid, Heart, Bupa, Breast Cancer, Vehicle, German ve Bank Marketing veri kümelerinde sırasıyla 96.74, 84.44, 64.06, 97.07, 63.00, 76.50 ve 89.16 sonuçları ile ET-ID3 ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Genel olarak, önerilen EF_Unique ayrıklaştırma yöntemi Thyroid veri kümesinde 96.74, Vehicle veri kümesinde 63.00, German veri kümesinde 76.50 ve Bank Marketing veri kümesinde ise 89.16 NB sınıflandırma algoritmasında EG, EF ve ET-ID3 ayrıklaştırma yöntemlerinde en iyi sınıflandırma başarısı elde etmiştir.

Çizelge 5.2. DVM sınıflandırma algoritmasının sınıflandırma başarısı

Veri kümesi	DVM sınıflandırma başarıları %			
	EG	**EF**	**ET-ID3**	**EF_Unique**
Iris	**96.00**	94.00	94.00	93.33
Wine	96.07	96.63	**98.31**	96.07
Glass Identification	66.36	67.29	76.17	**76.64**
Thyroid	90.70	93.95	**96.74**	95.81
Heart	83.33	80.37	**84.07**	82.96
E.coli	82.14	85.42	85.93	**87.32**
Bupa	68.41	68.70	63.19	**70.14**
Australian Credit	85.07	**85.65**	85.36	84.93
Breast Cancer	95.42	96.28	**96.42**	96.34
Blood	75.94	**76.20**	**76.20**	76.07
Pima Indians Diabetes	77.21	75.26	**77.47**	75.39
Vehicle	72.46	70.69	73.52	**75.77**
German	75.50	75.50	76.00	**77.50**
Wine Quality Red	59.35	57.60	56.91	**62.04**
Spambase	84.72	**93.87**	**93.87**	93.10
Magic Gamma Telescope	82.19	83.55	84.34	**84.89**
Bank Marketing	89.60	89.50	**89.90**	**89.90**

Çizelge 5.2'deki DVM sınıflandırma algoritmasının sınıflandırma sonuçları incelendiğinde, önerilen EF_Unique ayrıklaştırma yöntemiyle Glass Identification, Thyroid, E.coli, Bupa, Breast Cancer, Blood, Vehicle, German, Wine Quality Red, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 76.64, 95.81, 87.32, 70.14, 96.34, 76.07, 75.77, 77.50, 62.04, 84.89 ve 89.90 sonuçları elde edilmiş ve bu sonuçlarla EG ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı elde ettiği görülmüştür.

Önerilen EF_Unique ayrıklaştırma yöntemi Glass Identification, Thyroid, Heart, E.coli, Bupa, Breast Cancer, Pima Indians Diabetes, Vehicle, German, Wine Quality Red, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 76.64, 95.81, 82.96 87.32, 70.14, 96.34, 75.39, 75.77, 77.50, 62.04, 84.89 ve 89.90 sonuçları ile EF ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Önerilen EF_Unique ayrıklaştırma yöntemi Glass Identification, E.coli, Bupa, Vehicle, German, Wine Quality Red, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 76.64, 87.32, 70.14, 75.77, 77.50, 62.04, 84.89 ve 89.90 sonuçları ile ET-ID3 ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Genel olarak, önerilen EF_Unique ayrıklaştırma yöntemi Glass Identification, E.coli, Bupa, Vehicle, German, Wine Quality Red, Magic Gamma Telescope ve Bank Marketing veri kümelerinde sırasıyla 76.64, 87.32, 70.14, 75.77, 77.50, 62.04, 84.89 ve 89.90 değerleri ile en iyi sınıflandırma başarısı elde etmiştir.

Çizelge 5.3. K-NN sınıflandırma algoritmasının sınıflandırma başarısı

Veri kümesi	K-NN sınıflandırma başarısı %			
	EG	**EF**	**ET-ID3**	**EF_Unique**
Iris	93.33	**94.67**	**94.67**	**94.67**
Wine	94.38	93.82	**96.07**	94.38
Glass Identification	65.42	70.09	66.36	**73.83**
Thyroid	95.28	**97.21**	93.95	92.56
Heart	80.74	81.85	81.85	**82.96**
E.coli	77.37	81.96	82.74	**85.78**
Bupa	64.93	60.87	63.19	**66.09**
Australian Credit	**85.94**	85.80	85.51	85.36
Breast Cancer	95.28	95.28	**97.00**	95.46
Blood	75.94	77.25	77.74	**78.48**
Pima Indians Diabetes	70.05	**71.88**	69.66	70.05
Vehicle	65.96	69.74	**73.52**	69.15
German	72.70	73.10	**73.50**	73.00
Wine Quality Red	56.97	53.72	57.04	**57.16**
Spambase	81.59	92.05	**93.48**	91.25
Magic Gamma Telescope	78.05	81.70	82.03	**83.09**
Bank Marketing	89.02	88.31	**89.55**	89.33

Çizelge 5.3'teki K-NN sınıflandırma algoritmasının sınıflandırma sonuçları incelendiğinde, önerilen EF_Unique ayrıklaştırma yöntemiyle Iris, Glass Identification, Heart, E.coli, Bupa, Breast Cancer, Blood, German, Wine Quality Red, Spambase, Magic Gamma Telescope ve veri kümelerinde sırasıyla 94.67, 73.83, 82.96, 85.78, 66.09, 95.46, 78.48, 73.00, 57.16, 91.25, 83.09 ve sonuçları elde edilmiş ve bu sonuçlarla EG ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı elde ettiği görülmüştür.

Önerilen EF_Unique ayrıklaştırma yöntemi Wine, Glass Identification, Heart, E.coli, Bupa, Breast Cancer, Blood, Wine Quality Red, Magic Gamma Telescope ve veri kümelerinde sırasıyla 94.38, 73.83, 82.96, 85.78, 66.09, 95.46, 78.48, 57.16, 83.09 ve sonuçları ile EF ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Önerilen EF_Unique ayrıklaştırma yöntemi Glass Identification, Heart, E.coli, Bupa, Blood, Pima Indians Diabetes, Wine Quality Red ve Magic Gamma Telescope veri kümelerinde sırasıyla 73.83, 82.96, 85.78, 66.09, 78.48, 70.05, 57.16 ve 83.09 sonuçları ile ET-ID3 ayrıklaştırma yönteminden daha iyi sınıflandırma başarısı vermiştir.

Genel olarak, önerilen EF_Unique ayrıklaştırma yöntemi Iris, Glass Identification, Heart, E.coli, Bupa, Blood, German, Wine Quality Red ve Magic Gamma Telescope veri kümelerinde sırasıyla 94.67, 73.83, 82.96, 85.87, 66.09,78.48, 57.16 ve 83.09 değerleri ile en iyi sınıflandırma başarısı elde etmiştir.

Çizelge 5.4. C4.5 sınıflandırma algoritmasının sınıflandırma başarısı

Veri kümesi	C4.5 sınıflandırma başarısı %			
	EG	**EF**	**ET-ID3**	**EF_Unique**
Iris	**96.00**	94.67	94.00	94.00
Wine	89.33	84.27	**93.82**	81.46
Glass Identification	64.02	63.08	**73.83**	72.43
Thyroid	91.16	92.09	95.35	**96.74**
Heart	77.78	80.37	**81.85**	79.26
E.coli	72.62	76.79	**86.24**	76.95
Bupa	63.48	66.38	63.19	57.68
Australian Credit	84.78	87.10	**85.65**	84.78
Breast Cancer	94.99	**95.85**	94.99	95.31
Blood	**76.20**	**76.20**	**76.20**	75.80
Pima Indians Diabetes	73.44	74.87	74.35	**75.13**
Vehicle	71.04	66.08	71.99	70.57
German	72.60	71.20	**72.10**	71.50
Wine Quality Red	59.04	58.72	**59.10**	57.47
Spambase	80.59	87.61	**89.11**	85.85
Magic Gamma Telescope	77.83	76.32	**79.01**	77.86
Bank Marketing	83.92	83.16	84.79	**85.04**

Çizelge 5.4'teki sınıflandırma sonuçları incelendiğinde önerilen EF_Unique ayrıklaştırma yöntemi Thyroid veri kümesinde 96.74, Pima Indians Diabetes veri kümesinde 75.13 ve Bank Marketing veri kümelerinde ise 85.04 değerleri ile C4.5 sınıflandırma algoritmasında en iyi sınıflandırma başarısı elde etmiştir. ET-ID3 ayrıklaştırma yöntemi veri ayrıklaştırmayı C4.5 eğitimine dayalı olduğundan dolayı ET-ID3 ayrıklaştırma yöntemi C4.5 sınıflandırma algoritmasında 17 veri kümesinin 10 tanesinde iyi sınıflandırma başarısı elde etmiştir.

Deneysel sonuçlara ayrıntılı bir şekilde bakıldığında, önerilen denetimsiz EF_Unique ayrıklaştırma yöntemi denetimsiz EG ve EF yöntemlerinden 68 deneyden sırasıyla 43 ve 41 tanesinde NB, DVM, K-NN ve C4.5 sınıflandırma algoritmalarında daha iyi bir sınıflandırma başarısı sonucu elde etmiştir. Önerilen EF_Unique ayrıklaştırma yönteminin sınıflandırma sonuçları denetimli ET-ID3 ayrıklaştırma yöntemi ile karşılaştırıldığında, önerilen EF_Unique ayrıklaştırma yöntemi 68 deneyden 27 tanesinde daha iyi performans göstermiştir. Deneysel sonuçlar, önerilen EF_Unique ayrıklaştırma yönteminin, EG, EF ve ET-ID3 ayrıklaştırma yöntemlerine göre iyi bir performans sergilediğini göstermektedir.

Önerilen EF_Unique ile EG, EF ve ET-ID3 ayrıklaştırma yöntemlerinin deney sonuçları arasında istatistiksel olarak bir fark olup olmadığını belirlemek için Wilcoxon işaretli sıralar testi kullanılmıştır (Rosner ve ark., 2006; Johnson, 2009). Test NB, DVM, K-NN ve C4.5 sınıflandırma algoritmalarının sınıflandırma başarılarına göre ve p değeri 0.05 olarak ayarlanmıştır, sonuçlar Çizelge 5.5'te verilmektedir.

Çizelge 5.5. Önerilen EF_Unique ve EG, EF ve ET-ID3 ayrıklaştırma yöntemleri arasında Wilcoxon işaretli sıralama testinin sonuçları

Sınıflandırma algoritmaları	EG- EF_Unique	EF- EF_Unique	ET-ID3- EF_Unique
NB	0.140	0.008	0.687
DVM	0.017	0.025	0.758
K-NN	0.011	0.163	0.959
C4.5	0.642	0.831	0.007

Wilcoxon işaretli sıralama testinin sonuçlarına göre, önerilen EF_Unique ayrıklaştırma yöntemi, DVM ve K-NN sınıflandırma algoritmaları için EG ayrıklaştırma yönteminden istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiş, NB ve C4.5 için istatistiksel olarak anlamlı bir farkın olmadığı görülmüştür. Ayrıca, önerilen EF_Unique ayrıklaştırma yöntemi NB ve DVM sınıflandırma algoritmaları için EF ayrıklaştırma yönteminden istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiş, K-NN ve C4.5 için istatistiksel olarak anlamlı bir fark görülememiştir. Ek olarak, önerilen EF_Unique ayrıklaştırma yöntemi NB, DVM ve K-NN sınıflandırma algoritmaları için ET-ID3 ayrıklaştırma yöntemi ile istatistiksel olarak anlamlı bir fark yoktur, fakat ET-ID3 ayrıklaştırma yöntemi C4.5 sınıflandırma algoritması için EF_Unique ayrıklaştırma yönteminden istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiştir.

Önerilen EF_Unique ve EG, EF ve ET-ID3 ayrıklaştırma yöntemlerinin zaman karmaşıklıklarını analiz etmek için, ayrıklaştırma yöntemlerinin CPU çalışma zamanları Microsoft Process Explorer programı ile saniye cinsinden ölçülmüş ve Çizelge 5.6'da verilmiştir.

Çizelge 5.6. Ayrıklaştırma yöntemlerinin saniye cinsinden çalışma süreleri

Veri kümesi	EG	EF	ET-ID3	EF_Unique
Iris	0.001	0.001	0.018	0.002
Wine	0.003	0.002	0.022	0.005
Glass Identification	0.002	0.002	0.021	0.005
Thyroid	0.002	0.001	0.020	0.003
Heart	0.003	0.003	0.021	0.005
E.coli	0.002	0.002	0.021	0.005
Bupa	0.002	0.002	0.019	0.004
Australian Credit	0.006	0.005	0.025	0.009
Breast Cancer	0.005	0.004	0.029	0.006
Blood	0.003	0.002	0.022	0.004
Pima Indians Diabetes	0.005	0.004	0.028	0.007
Vehicle	0.015	0.013	0.043	0.019
German	0.006	0.004	0.027	0.012
Wine Quality Red	0.015	0.010	0.043	0.019
Spambase	0.069	0.067	0.345	0.220
Magic Gamma Telescope	0.085	0.083	0.735	0.196
Bank Marketing	0.236	0.236	0.858	0.487

Çizelge 5.6'da görüldüğü gibi, denetimli ET-ID3 ayrıklaştırma yöntemi zaman karmaşıklığı $O(n\,log\,n)$ olduğundan dolayı EG, EF ve önerilen EF_Unique denetimsiz ayrıklaştırma yöntemlerinden daha uzun CPU çalışma süresi gerektirmiştir. Önerilen denetimsiz EF_Unique ayrıklaştırma yöntemi ve denetimsiz EG ve EF ayrıklaştırma yöntemleri ile aynı zaman karmaşıklığına sahiptir, fakat önerilen EF_Unique ayrıklaştırma yöntemi CPU çalışma süresi EG ve EF ayrıklaştırma yöntemlerine göre biraz daha uzundur. Bunun nedeni, önerilen denetimsiz EF_Unique ayrıklaştırma yönteminin, EG ve EF denetimsiz ayrıklaştırma yöntemlerinden farklı olarak, aralıkların sınırlarını hesaplamak için özniteliğin benzersiz değerlerinin sayısını kullanmasıdır.

Genel olarak deneylerin sonuçları, önerilen EF_Unique ayrıklaştırma yönteminin veri madenciliği sınıflandırma problemleri için üstün bir performans sağladığını göstermektedir. Önerilen EF_Unique ayrıklaştırma yöntemi medikal, mühendislik, tarım ve biyoloji alanlarından elde edilen sürekli verileri ayrıklaştırılması için kullanılabilir. Buna ek olarak kural ve olasılık tabanlı sınıflandırma algoritmalarının sınıflandırma performanslarını iyileştirmek için çok iyi bir veri ön işleme tekniği olarak da kullanılabilir.

5.2. Önerilen MM-PSO Meta-Sezgisel Optimizasyon Algoritmasının Deneysel Sonuçları ve Tartışmalar

Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasına uygulanmasının kodlaması Microsoft Visual Studio Ultimate C#.Net 2017 kullanılarak gerçekleştirilmiştir. Tüm deneyler, Microsoft Windows 10 işletim sistemi, 8 GB belleğe sahip Intel Core i7 3840QM@2.00 GHz işlemcili bir bilgisayar kullanılarak gerçekleştirilmiştir. Orijinal PSO ve önerilen MM-PSO meta-sezgisel optimizasyon algoritmalarının performansları UCI makine öğrenmesi veri ambarından farklı özelliklere sahip 10 farklı veri kümesi üzerinde değerlendirilmiştir ve literatürdeki önceki çalışmalar ile karşılaştırılmıştır. Bu veri kümelerinin özellikleri bölüm 3.1'de gösterilmiştir.

Genel olarak, çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmalarının yapısı I-H-O ile temsil edilmektedir; burada I, giriş katmanındaki düğümlerin sayısını, H, gizli katmandaki düğümlerin sayısını ve O ise, çıkış katmanındaki düğümlerin sayısını ifade etmektedir. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmalarında

bulunan ağırlıkların sayısı Denklem 5.1 kullanılarak hesaplanır. Bu sayı aynı zamanda optimizasyon probleminin boyutunu temsil eder.

$$Ağırlıkların\ sayısı = (I * H) + (H * O) + H + O \tag{5.1}$$

Ayrıca, çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmalarının uygun yapısını belirlemek bir optimizasyon problemidir ve çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmalarının performansını etkileyen en önemli faktörlerden biri olarak bilinir (İbrahim, Cihad ve Kamal, 2017). Kullanılan sınıflandırma veri kümelerine göre belirlenen en uygun çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının yapısı Çizelge 5.7'de verilmiştir (Bolaji ve ark., 2018).

Çizelge 5.7. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının yapısı

Veri kümesi	I	H	O	Ağırlıkların sayısı
Lymphography	18	15	4	349
Iris	4	5	3	43
Wine	13	10	3	173
Glass	9	12	6	198
Shuttle-landing	6	8	2	74
Ionosphere	33	4	2	146
Balance-scale	4	5	3	43
Breast cancer	9	8	2	98
Diabetes	8	6	2	68
Thyroid	21	12	3	303

Gizli katman sayısı ve gizli katmanda bulunan nöron sayıları çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının sınıflandırma performansını etkilemektedir. Önerilen MM-PSO meta-sezgisel algoritması ve literatürde önerilen meta-sezgisel algoritmaları ile karşılaştırma aynı şartlar altında olması için çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının gizli katmandaki nöron sayıları literatürden alınmıştır (Bolaji ve ark., 2018).

Meta-sezgisel algoritmalarında bulunan giriş parametreleri, problemin çözüm hızını ve en uygun çözümün aranmasını etkilemektedir. Dolayısıyla, bu giriş parametrelerin iyi bir şekilde seçilmesi gerekmektedir. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmaları için parametre ayarlamaları Çizelge 5.8'de verilmektedir.

Çizelge 5.8. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının parametreleri

Parametre	**Değer**
Parçacık sayısı	20
Sürü sayısı	3
İterasyon sayısı	1000
Ağırlık w	0.74
c_1 ve c_2	1.49
r_1 ve r_2	0 ile 1 arasında rastgele sayı
Arama uzayı	-10 ile 10 arasında

Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının gizli ve çıkış katmanlarında eğitim ve test aşamaları için sigmoid aktivasyon fonksiyonu kullanılmıştır. Durdurma ölçütü olarak maksimum iterasyon sayısı kullanılmış ve sınıflandırma işlemi, 5-katlamalı çapraz doğrulama ile gerçekleştirilmiştir. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının 5-katlamalı çapraz doğrulamadan elde edilen sınıflandırma performansları Çizelge 5.9'da verilmiştir.

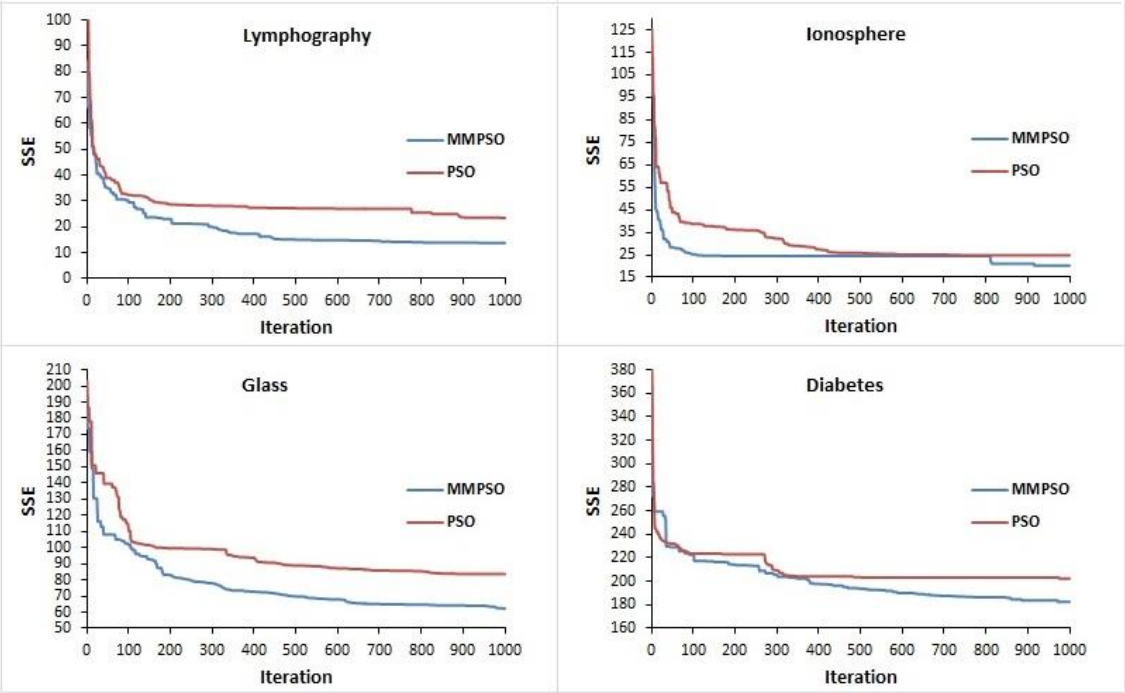
Çizelge 5.9. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının sınıflandırma performansları

Veri kümesi	**Algoritma**	**SSE**	**Eğitim SB (%)**	**Test SB (%)**
Lymphography	PSO	23.46	89.32	78.67
	MM-PSO	**13.87**	**91.49**	**83.67**
Iris	PSO	6.72	91.00	90.67
	MM-PSO	**1.35**	**97.33**	**91.67**
Wine	PSO	3.86	92.25	87.22
	MM-PSO	**0.39**	**99.58**	**95.56**
Glass	PSO	83.60	66.35	55.84
	MM-PSO	**62.36**	**69.21**	**58.14**
Shuttle-landing	PSO	2.07	**99.51**	94.12
	MM-PSO	**1.79**	**99.51**	**96.08**
Ionosphere	PSO	24.77	93.07	85.35
	MM-PSO	**20.16**	**95.79**	**88.73**
Balance-scale	PSO	58.21	90.20	86.56
	MM-PSO	**35.85**	**90.24**	**89.12**
Breast cancer	PSO	73.80	98.38	95.14
	MM-PSO	**59.43**	**98.57**	**96.29**
Diabetes	PSO	202.39	76.78	72.08
	MM-PSO	**182.54**	**77.55**	**78.05**
Thyroid	PSO	528.55	92.95	92.83
	MM-PSO	**373.63**	**93.66**	**94.01**

Çizelge 5.9'daki 5-katlamalı çapraz doğrulama sonuçları incelendiğinde, önerilen MM-PSO meta-sezgisel algoritması ile eğitilmiş çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının tüm veri kümelerinde çok daha iyi SSE, eğitim SB ve test SB sonuçları elde ettiği gözlenmiştir. Orijinal PSO algoritması ile eğitilmiş çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması önerilen yaklaşıma göre iyi SSE, eğitim SB

ve test SB sonuçları elde etmemiştir. Sonuç olarak, önerilen MM-PSO meta-sezgisel algoritması PSO algoritmasına göre çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması için daha uygun ağırlıklar elde etmiş ve çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının sınıflandırma performansını iyi bir şekilde iyileştirmiştir.

Önerilen MM-PSO meta-sezgisel algoritması küresel uzayı daha verimli araştırması ve en uygun sonuçları daha hızlı birleştirmesi önerilen algoritmanın avantajlarıdır. Bu avantajları sağladığından dolayı, önerilen MM-PSO meta-sezgisel algoritması minimum uygunluk değerine (SSE) PSO algoritmasından daha hızlı bir süreçte yaklaşmaya çalışır. Eğitim sürecindeki iterasyon sayısına göre SSE değerlerinin en aza indirilmesi Lymphography, Ionosphere, Glass ve Diabetes veri kümeleri üzerinde Şekil 5.1'de verilmiştir. Lymphography, Ionosphere ve Glass veri kümeleri üzerinde önerilen MM-PSO meta-sezgisel algoritması, SSE değerini başlangıç iterasyondan bitiş iterasyona kadar çok eğimli bir şekilde en aza indirmiştir. Diyabet veri kümesinde ise, önerilen MM-PSO meta-sezgisel algoritması, 90. iterasyondan sonra küresel alanı daha verimli biçimde araştırmıştır. Ayrıca, önerilen MM-PSO meta-sezgisel algoritması, tüm veri kümelerinde başlangıç iterasyonlarda daha iyi SSE sağlamıştır.



Şekil 5.1. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarından eğitim aşamasında iterasyon sayısına göre elde edilen SSE

Yukarıdaki Şekil 5.1'deki sınıflandırma veri kümelerinin sınıflandırma SSE hata oranı incelendiğinde, önerilen MM-PSO meta-sezgisel algoritması daha hızlı bir şekilde en uygun ağırlık değerlerine yaklaşmaktadır. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasında SSE hata değeri ağırlıkların değerlerine bağlıdır. En uygun ağırlık değerlerine sahip olan çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması daha az bir SSE hata değerine sahiptir.

Önerilen MM-PSO meta-sezgisel algoritmasının yanı sıra çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması üzerinde etkisini değerlendirmek için önerilen algoritmanın 10-katlamalı çapraz doğrulama deneyinin karmaşıklık matrisinden elde edilen istatistiksel değerlendirme ölçütleri yapılmıştır. Karmaşıklık matrisinden elde edilen ACU, SEN, SPE, PRE, FPR ve F-M istatistiksel ölçütleri, sınıflandırma modelinin genel bir performansını değerlendirmektedir (García ve ark., 2017). Orijinal PSO ve MM-PSO meta-sezgisel algoritmalarının istatistik değerlendirme ölçütleri Çizelge 5.10'da verilmiştir.

Çizelge 5.10. Orijinal PSO ve MM-PSO meta-sezgisel algoritmalarının istatistik değerlendirme ölçütleri

Veri kümesi	**Algoritma**	ACU	SEN	SPE	PRE	FPR	F-M
Lymphography	PSO	89.19	78.38	92.79	78.38	0.07	78.38
	MM-PSO	**91.89**	**83.78**	**94.60**	**83.78**	**0.05**	**83.78**
Iris	PSO	93.78	90.67	95.33	90.67	0.05	90.67
	MM-PSO	**94.67**	**92.00**	**96.00**	**92.00**	**0.04**	**92.00**
Wine	PSO	91.76	87.64	93.82	87.64	0.06	87.64
	MM-PSO	**97.00**	**95.51**	**97.75**	**95.51**	**0.02**	**95.51**
Glass	PSO	85.36	56.08	91.22	56.08	**0.08**	56.08
	MM-PSO	**86.14**	**58.41**	**91.68**	**58.41**	**0.08**	**58.41**
Shuttle-landing	PSO	94.07	94.07	94.07	94.07	0.06	94.07
	MM-PSO	**96.44**	**96.44**	**96.44**	**96.44**	**0.04**	**96.44**
Ionosphere	PSO	85.47	85.47	85.47	85.47	0.15	85.47
	MM-PSO	**88.60**	**88.60**	**88.60**	**88.60**	**0.11**	**88.60**
Balance-scale	PSO	91.15	86.72	93.36	86.72	0.07	86.72
	MM-PSO	**92.75**	**89.12**	**94.56**	**89.12**	**0.05**	**89.12**
Breast cancer	PSO	95.14	95.14	95.14	95.14	0.05	95.14
	MM-PSO	**96.42**	**96.42**	**96.42**	**96.42**	**0.04**	**96.42**
Diabetes	PSO	72.01	72.01	72.01	72.01	0.28	72.01
	MM-PSO	**78.13**	**78.13**	**78.13**	**78.13**	**0.22**	**78.13**
Thyroid	PSO	95.15	92.72	96.36	92.72	0.04	92.72
	MM-PSO	**96.11**	**94.17**	**97.08**	**94.17**	**0.03**	**94.17**

Çizelge 5.10'daki istatistiksel değerlendirme ölçütlerinin sonuçları analiz edildiğinde, karmaşıklık matrisinden elde edilen ACU, SEN, SPE, PRE, FPR ve F-M istatistiksel ölçütlerinde önerilen MM-PSO tabanlı çok-katmanlı ileri sınıflandırma algoritması Lymphography, Iris, Wine, Glass, Shuttle-landing, Ionosphere, Balance-scale, Breast Cancer, Diabetes ve Thyroid veri kümelerinde orijinal PSO tabanlı çok-

katmanlı ileri sınıflandırma algoritmasından daha iyi ACU, SEN, SPE, PRE, FPR ve F-M değerleri elde etmiştir.

Genel olarak, tüm istatistik değerlendirme ölçütleri değerlerinde deneyde kullanılan tüm veri kümelerinde önerilen MM-PSO orijinal PSO algoritmasından daha iyi sonuçlar vermiştir. Önerilen MM-PSO algoritmasının sonuçları çok-katmanlı ileri beslemeli YSA sınıflandırma modelindeki ağırlıkların çok iyi bir şekilde güncellediğini dolayısıyla çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının çok iyi bir şekilde eğitildiğini göstermektedir.

Ayrıca, orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının zaman karmaşıklıklarını analiz etmek için, her bir algoritmanın CPU çalışma zamanı Microsoft Process Explorer programı tarafından saniyeler cinsinde ölçülmüştür. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının CPU çalışma zamanları Çizelge 5.11'de verilmiştir.

Çizelge 5.11. Orijinal PSO ve önerilen MM-PSO meta-sezgisel algoritmalarının CPU çalışma zamanları

Veri kümesi	PSO	MM-PSO
Lymphography	6.48	**5.16**
Iris	2.16	**1.37**
Wine	4.37	**3.37**
Glass	7.59	**5.17**
Shuttle-landing	4.34	**3.19**
Ionosphere	6.52	**6.26**
Balance-scale	8.03	**6.32**
Breast cancer	14.44	**11.05**
Diabetes	11.29	**9.36**
Thyroid	479.08	**396.35**

Çizelge 5.11'de gösterildiği gibi, önerilen MM-PSO meta-sezgisel algoritmasının çalışma süresi, tüm veri kümeleri için PSO algoritmasından daha kısadır. Ek olarak, önerilen MM-PSO meta-sezgisel algoritması paralel uygulama için uygundur ve MM-PSO algoritmasının çalışma süresi paralel programlama ile çok daha kısa bir süreye düşürülebilir.

Önerilen MM-PSO meta-sezgisel algoritmasının performansı çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması üzerinde değerlendirmek için, literatürden seçilen Harmony Search Algoritması (HSA) (Kattan ve ark., 2010), Krill Herd Algoritması (KHA), Genetik Algoritması (GA) (Kowalski ve Łukasik, 2016) ve Havai Fişek Algoritması (FWA) (Bolaji ve ark., 2018) ile karşılaştırılmıştır. Literatürdeki çalışmalarda kullanılan altı adet veri kümeleri %80 eğitim kümesi ve %20 test kümesi

şeklinde ayarlanmıştır. Bu karşılaştırmayı aynı şartlar altında yapabilmek için önerilen çalışmada da aynı veri kümeleri %80 eğitim kümesi ve %20 test kümesi şeklinde ayarlanmıştır. Aynı zamanda önerilen MM-PSO meta-sezgisel algoritması FWA, KHA, HS ve GA algoritmalar gibi on kez çalıştırılarak en iyi SSE, eğitim SB ve test SB sonuçları seçilmiştir. Meta-sezgisel algoritmalarında bulunan iterasyon sayısı ve popülasyon sayısı ortak giriş parametreleri sırasıyla 1000 ve 20 olarak ayarlanmıştır (Bolaji ve ark., 2018).

Önerilen MM-PSO meta-sezgisel ve literatürde önerilen FWA, KHA, HS ve GA meta-sezgisel algoritmalarının %80 eğitim kümesi ve %20 test kümesi deneyinin sonuçları Çizelge 5.12'de gösterilmektedir.

Çizelge 5.12. Önerilen MM-PSO ve literatürde önerilen FWA, KHA, HS ve GA meta-sezgisel algoritmalarının %80 eğitim kümesi ve %20 test kümesi deney sonuçları

Veri kümesi	Algoritma	SSE	Eğitim SB (%)	Test SB (%)
Iris	MM-PSO	**0.31**	**100**	**100**
	FWA	0.52	**100**	**100**
	KHA	21.28	99.59	**100**
	HS	18.00	98.33	96.67
	GA	96.00	90.00	90.00
Glass	MM-PSO	47.56	**78.36**	62.79
	FWA	94.33	61.99	60.47
	KHA	**41.21**	58.79	58.14
	HS	355.85	70.12	**72.09**
	GA	544.00	57.89	67.44
Ionosphere	MM-PSO	**18.91**	**99.28**	91.54
	FWA	25.28	95.71	90.14
	KHA	31.0	89.00	91.43
	HS	106.4	95.00	**94.37**
	GA	152	93.21	**94.37**
Breast cancer	MM-PSO	**47.00**	**99.46**	97.85
	FWA	66.11	93.92	96.43
	KHA	-	-	-
	HS	126.37	-	**100**
	GA	172	-	98.57
Diabetes	MM-PSO	**166.21**	**81.27**	**79.87**
	FWA	267.20	65.96	66.88
	KHA	-	-	-
	HS	856	-	77.27
	GA	1108	-	**79.87**
Thyroid	MM-PSO	**237.07**	**95.37**	**94.19**
	FWA	749.11	93.21	93.82
	KHA	320.3	94.81	92.90
	HS	3146.4	93.06	92.78
	GA	3416.0	92.58	92.57

Çizelge 5.12'deki %80 eğitim kümesi ve %20 test kümesi deneyinin sonuçları analiz edildiğinde, önerilen MM-PSO meta-sezgisel algoritmasının Iris, Diabetes ve

Thyroid veri kümelerinde FWA, KHA, HS ve GA meta-sezgisel optimizasyon algoritmalarından daha iyi SSE, eğitim SB ve test SB değerleri sunmuştur. Önerilen MM-PSO meta-sezgisel algoritması, Ionosphere ve Breast cancer veri kümelerinde SSE ve eğitim SB FWA, KHA, HS ve GA meta-sezgisel optimizasyon algoritmalardan daha iyi değer elde etmesine rağmen, bu veri kümeleri için en iyi test SB değeri elde edememiştir. Son olarak Glass veri kümesi için, önerilen MM-PSO meta-sezgisel algoritması FWA, KHA, HS ve GA meta-sezgisel optimizasyon algoritmalarından yalnızca iyi bir eğitim SB değeri elde etmiştir. Özet olarak, Çizelge 5.12'deki karşılaştırma sonuçlarına bakıldığında, önerilen MM-PSO meta-sezgisel algoritması genel olarak diğer algoritmalardan daha iyi sınıflandırma sonuçları vermiştir.

Önerilen MM-PSO ve orijinal PSO meta-sezgisel algoritmalarından on kez çalıştırma sonucu elde edilen eğitim ve test aşamaları için SB değerlerinin standart sapma değerleri hesaplanmıştır. Önerilen MM-PSO ve orijinal PSO meta-sezgisel algoritmalarının SB'nin standart sapma değerleri Çizelge 5.13'te verilmektedir.

Çizelge 5.13. Önerilen MM-PSO ve orijinal PSO meta-sezgisel algoritmalarının SB'nin standart sapma değerleri

Veri kümesi	Algoritma	Standart sapma eğitim	Standart sapma test
Lymphography	PSO	0.2294	0.0384
	MM-PSO	**0.1604**	**0.0223**
Iris	PSO	0.0564	**0.0242**
	MM-PSO	**0.0426**	0.0286
Wine	PSO	0.0940	0.0455
	MM-PSO	**0.0347**	**0.0268**
Glass	PSO	0.1306	0.0453
	MM-PSO	**0.0602**	**0.0239**
Shuttle-landing	PSO	0.0553	0.0193
	MM-PSO	**0.0193**	**0.0107**
Ionosphere	PSO	0.0494	**0.0227**
	MM-PSO	**0.0432**	0.0247
Balance-scale	PSO	0.0206	0.0083
	MM-PSO	**0.0079**	**0.0072**
Breast cancer	PSO	0.0679	0.0662
	MM-PSO	**0.0209**	**0.0108**
Diabetes	PSO	0.0436	0.0109
	MM-PSO	**0.0116**	**0.0042**
Thyroid	PSO	0.0020	0.0013
	MM-PSO	**0.0013**	**0.0011**

Çizelge 5.13'teki standart sapma değeri analiz edildiğinde, önerilen MM-PSO algoritmasından elde edilen 10 kez çalışma SB değerlerinin birbirine çok yakın değerler olduğu gözlenmiştir. Sonuç olarak, önerilen MM-PSO algoritmasının 10 kez çalışma SB

değerleri PSO algoritmasından elde edilen SB değerlerinden daha tutarlı olduğu görülmektedir.

Önerilen MM-PSO meta-sezgisel algoritması ile FWA, KHA, HS ve GA meta-sezgisel algoritmalarının deney sonuçları arasında istatistiksel olarak bir fark olup olmadığını belirlemek için Wilcoxon işaretli sıralar testi kullanılmıştır (Rosner ve ark., 2006; Johnson, 2009). Test SSE, eğitim SB ve test SB sınıflandırma ölçütlerine göre ve p değeri 0.05 olarak ayarlanmıştır. Önerilen MM-PSO meta-sezgisel algoritması ile FWA, KHA, HS ve GA meta-sezgisel algoritmalarının sonuçlar Çizelge 5.14'te verilmektedir.

Çizelge 5.14. Önerilen MM-PSO ve FWA, KHA, HS ve GA algoritmaları arasında Wilcoxon işaretli sıralama testinin sonuçları

Sınıflandırma ölçütü	**FWA ile MM-PSO**	**KHA ile MM-PSO**	**HS ile MM-PSO**	**GA ile MM-PSO**
SSE	0.028	0.145	0.028	0.028
Eğitim SB	0.034	0.028	0.028	0.028
Test SB	0.043	0.043	0.753	0.893

Wilcoxon işaretli sıralama testinin sonuçlarına göre, önerilen MM-PSO meta-sezgisel algoritması SSE değerlerinde FWA, KHA, HS ve GA meta-sezgisel algoritmalarından istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiştir. Eğitim SB sonuçlarına göre önerilen MM-PSO meta-sezgisel algoritması FWA, KHA, HS ve GA meta-sezgisel algoritmalarından istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiştir. Test SB sonuçlarına göre ise önerilen MM-PSO meta-sezgisel algoritması FWA ve KHA meta-sezgisel algoritmalarından istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiştir. HS ve GA meta-sezgisel algoritmaları önerilen MM-PSO meta-sezgisel algoritmasından istatistiksel olarak anlamlı derecede daha iyi sonuçlar vermiştir.

6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Veri madenciliği ve makine öğrenmesi algoritmalarından daha iyi bir performans elde etmek için veri kümesinin önişleme ile hazırlanması ve uygun giriş parametrelerinin verilmesi gerekmektedir. Kural ve olasılık tabanlı sınıflandırma algoritmaları parametrik olmayan sınıflandırma algoritmaları olduğundan dolayı iyi bir performans elde etmek için sınıflandırma algoritmasına uygun biçimde veri kümesinin hazırlanması gerekmektedir. Veri kümeleri farklı kaynaklardan toplandığında, bu veriler analize uygun değildir ve ham biçimdedir. Veri madenciliğinde veri ön işleme aşaması, makine öğrenmesi algoritmalarının veri kümeleri üzerinde işlem yapmadan önce veri kümelerine uygulanan dönüşümleri ifade etmektedir. Ayrıklaştırma, sürekli öznitelikleri içeren veri kümelerini ayrık veri kümelerine dönüştüren veri madenciliğinin önemli bir veri ön işleme tekniğidir. Bazı makine öğrenmesi algoritmaları belirli bir biçimde veri kümesine ihtiyaç duymakta ve doğrudan uygulanabilmektedir. Kural ve olasılık tabanlı sınıflandırma algoritmaları ayrık veri kümeleri ile daha iyi bir performans sergilemektedir. Bu tez çalışmasının birinci kısmında, kural ve olasılık tabanlı sınıflandırma algoritmalarının performansını iyileştirmek için EF_Unique adında yeni bir ayrıklaştırma yöntemi önerilmiştir. Denetimsiz ayrıklaştırma yöntemleri, kullanıcı tarafından sağlanan k aralıklarının sayısını gerektirmektedir, ancak en uygun k'nın belirlenmesi veri madenciliğinde zor bir problem olarak bilinmektedir. Önerilen EF_Unique ayrıklaştırma yönteminde, k sayısı, benzersiz değer kümesinin uzunluğunun karekökünü hesaplayarak elde etmektedir. Önerilen EF_Unique ayrıklaştırma yönteminin performansını değerlendirmek için, farklı özelliklere sahip UCI makine öğrenmesi veri ambarından 17 veri kümesi kullanılmıştır. Veri kümeleri, ayrık tabanlı sınıflandırma algoritmalarında uygulamak için EF_Unique, EG, EF ve ET-ID3 ayrıklaştırma yöntemleri ile ayrıklaştırılmıştır. Sınıflandırma sürecinde NB, DVM, K-NN ve C4.5 makine öğrenmesi sınıflandırma algoritmaları kullanılmıştır. Deneysel sonuçlar, önerilen EF_Unique ayrıklaştırma yöntemi, denetimsiz EG, EF ve denetimli ET-ID3'ü, 68 deneyden sırasıyla 43, 41 ve 27 tanesinde daha iyi performans göstermiştir. Ayrıca, Wilcoxon sıra testi EF_Unique ve EG, EF ve ET-ID3 ayrıklaştırma yöntemleri arasında karşılaştırma yapmak için yapılmıştır. Önerilen EF_Unique ayrıklaştırma yönteminin, NB ve DVM sınıflandırma algoritmaları için EF ayrıklaştırma yönteminden istatistiksel olarak anlamlı derecede farklı olduğu

görülmüştür. Ayrıca EF_Unique'in, DVM ve K-NN sınıflandırma algoritmalarında EG ayrıklaştırma yönteminden istatistiksel olarak anlamlı derecede farklı olduğu gözlemlenmiştir. Deneysel sonuçlara göre, önerilen denetimsiz EF_Unique ayrıklaştırma yöntemi, makine öğrenmesi ve istatistikler dâhil olmak üzere birçok alanda kullanılabilir.

Diğer yandan ağırlık tabanlı makine öğrenmesi algoritmalarının performansı giriş parametrelerine ve veri kümesinin hazırlanmasına bağlıdır. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritmasının performansını önemli bir şekilde etkileyen, çok-katmanlı ileri beslemeli YSA'da bulunan ağırlık parametreleridir. Çok-katmanlı ileri beslemeli YSA sınıflandırma algoritması için en uygun ağırlık değerlerinin belirlenmesi literatürde zor bir optimizasyon problemi olarak bilinmektedir. Bu tez çalışmasının ikinci kısmında ise, ağırlık tabanlı sınıflandırma algoritması olan çok-katmanlı ileri beslemeli YSA'da en uygun ağırlıkları elde etmek için MM-PSO adında yeni bir PSO tabanlı meta-sezgisel optimizasyon algoritması önerilmiştir. MSO tekniğine dayanan önerilen MM-PSO meta-sezgisel algoritmasının, orijinal PSO algoritmasına göre iki avantajı vardır. İlk olarak, önerilen MM-PSO meta-sezgisel algoritması, arama uzayında yerel bir arama yapmak için parçacıkları güçlendirir. İkincisi, önerilen MM-PSO meta-sezgisel algoritması birden fazla kümeye sahiptir ve her kümenin en iyi çözümünü ve tüm kümelerin en iyi çözümünü hesaba katar. Böylece daha az süre ile en uygun çözüme yaklaşır. Önerilen öğrenme algoritması UCI makine öğrenmesi veri ambarından alınan veri kümeleri üzerinde önceki çalışmalarda bulunan meta-sezgisel optimizasyon algoritmaları ile karşılaştırılmıştır. Bu karşılaştırma analizine göre, önerilen MM-PSO meta-sezgisel algoritması literatürdeki FWA, KHA, HS ve GA meta-sezgisel algoritmalara göre rekabetçi bir avantaj göstermiştir. Sonuç olarak, önerilen MM-PSO meta-sezgisel algoritması iyi performans göstermiştir ve çok-katmanlı ileri beslemeli YSA eğitimi için yeni bir yöntem olarak kabul edilebilir.

6.2 Öneriler

Veri madenciliği ve makine öğrenmesi algoritmaları birçok gerçek dünya problemlerinde uygulanmaktadır. Makine öğrenmesi, bilgisayar sistemlerinin belirli bir görevindeki performanslarını kademeli olarak geliştirmek için kullandıkları algoritmaların ve istatistiksel modellerin bilimsel çalışmasıdır. Makine öğrenmesi algoritmaları, görevi gerçekleştirmek için eğitim veri kümesi olarak bilinen matematiksel bir örnek veri modeli oluşturur, bu modeli karar vermek ve tahmin etmek için

kullanmaktadır. Makine öğrenmesi algoritmaları havacılık, medikal ve saldırı gibi çok önemli alanlarda kullanılması nedeniyle, bu makine öğrenmesi algoritmalarının performansları önem arz etmektedir. Makine öğrenmesi sınıflandırma algoritmalarının modelleri genel olarak istatistik veya matematik hesaplamaları ile oluşturulmaktadır. İstatistik hesaplamaları ile oluşturulan makine öğrenmesi sınıflandırma algoritmaları kural ve olasılık tabanlı olarak dallanabilir, matematiksel hesaplamaları ile oluşturulan makine öğrenmesi sınıflandırma algoritmaları ise ağırlık tabanlı olarak sınıflandırılabilir.

Kural ve olasılık tabanlı makine öğrenmesi sınıflandırma algoritmalarının performansı eğitim veri kümesinin hazırlanması ile ilgilidir. Örneğin, karar ağacı sınıflandırma algoritmaları sürekli veri kümeleri ile iyi bir performans göstermemektedir. Bu performansı iyileştirmek ve daha verimli sınıflandırma ağacı oluşturmak için eğitim veri kümesinin ayrık veya kategorik veriler olması gerekmektedir.

Ağırlık tabanlı makine öğrenmesi algoritmalarının performansı ise hem eğitim veri kümesinin hazırlanmasına hem de algoritmanın giriş parametreleri ile ilgilidir. Örneğin, YSA sınıflandırma algoritmalarının performansları, oluşturulan sınıflandırma modeli üzerinde bulunan ağırlık parametrelerinin değerlerine bağlıdır. En uygun ağırlık değerlerine sahip olan YSA sınıflandırma modelleri çok iyi bir performans ile sınıflandırma veya tahmin işlevini yerine getirmektedir. YSA modelinde en uygun ağırlık değerlerinin belirlenmesi literatürde bir optimizasyon problemi olarak bilinmektedir.

Bu tez çalışması kapsamında kural ve olasılık tabanlı makine öğrenmesi sınıflandırma algoritmalarının performansını iyileştirmek için EF_Unique adında yeni bir ayrıklaştırma yöntemi önerilmiştir. Ağırlık tabanlı makine öğrenmesi sınıflandırma algoritmalarının performansını iyileştirmek için MM-PSO adında yeni bir meta-sezgisel optimizasyon algoritması geliştirilmiştir.

Gelecekteki çalışmalar için, önerilen EF_Unique ayrıklaştırma yöntemi, medikal, finansal bankacılık, tarım ve havacılık gibi birçok alanda sürekli verilere dayanan gerçek dünyadaki veri madenciliği problemlerini çözmek için bir veri ön işleme tekniği olarak kullanılabilir. Önerilen MM-PSO tabanlı YSA sınıflandırma algoritması ile akıllı sistemler, tasarım, tanımlama, teşhis, planlama ve çizelgeleme gibi çeşitli alanlarda karmaşık gerçek dünya sınıflandırma ve optimizasyon problemleri çözülebilir.

KAYNAKLAR

!!! INVALID CITATION !!! (Domingos ve Pazzani, 1997; Yang ve Webb, 2009; Zhang ve ark., 2011).

!!! INVALID CITATION !!! (Hu ve ark., 2009; Dash ve ark., 2011; Zaki ve ark., 2014; Rahman ve Islam, 2016).

Alsuwaiyel, M. H., 2016, Algorithms: Design Techniques And Analysis (Revised Edition), World Scientific, p.

Arlot, S. ve Celisse, A., 2010, A survey of cross-validation procedures for model selection, *Statistics surveys*, 4, 40-79.

Au, W.-H., Chan, K. C. ve Wong, A. K., 2006, A fuzzy approach to partitioning continuous attributes for classification, *IEEE Transactions on knowledge and data engineering*, 18 (5), 715-719.

Baba, N., 1989, A new approach for finding the global minimum of error function of neural networks, *Neural networks*, 2 (5), 367-373.

Basheer, I. A. ve Hajmeer, M., 2000, Artificial neural networks: fundamentals, computing, design, and application, *Journal of microbiological methods*, 43 (1), 3-31.

Bay, S. D., 2001, Multivariate discretization for set mining, *Knowledge and information systems*, 3 (4), 491-512.

Beheshti, Z. ve Shamsuddin, S. M. H., 2013, A review of population-based meta-heuristic algorithms, *Int. J. Adv. Soft Comput. Appl*, 5 (1), 1-35.

Berry, M. J. ve Linoff, G., 1997, Data mining techniques: for marketing, sales, and customer support, John Wiley & Sons, Inc., p.

Bertelsen, R. ve Martinez, T. R., 1994, Extending ID3 through discretization of continuous inputs, *Proceedings of the 7th Florida Artificial Intelligence Research Symposium*, 122-125.

Blake, C., 1998, UCI repository of machine learning databases, *http://www. ics. uci. edu/~ mlearn/MLRepository. html*.

Bolaji, A. L. a., Ahmad, A. A. ve Shola, P. B., 2018, Training of neural network for pattern classification using fireworks algorithm, *International Journal of System Assurance Engineering and Management*, 9 (1), 208-215.

Borgersen, G. ve Karlsson, L., 2008, Supervised learning in artificial neural networks, *IRCSE, Västerås, Sweden*, 1-6.

Boullé, M., 2006, MODL: a Bayes optimal discretization method for continuous attributes, *Machine learning*, 65 (1), 131-165.

Catlett, J., 1991, On changing continuous attributes into ordered discrete attributes, *European working session on learning*, 164-178.

Cebeci, Z. ve Yildiz, F., 2017, Comparison of Chi-square based algorithms for discretization of continuous chicken egg quality traits, *J. Agric. Inform*, 8 (1), 13-22.

Chatterjee, S., Sarkar, S., Hore, S., Dey, N., Ashour, A. S. ve Balas, V. E., 2017, Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings, *Neural Computing and Applications*, 28 (8), 2005-2016.

Chau, K., 2006, Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River, *Journal of hydrology*, 329 (3-4), 363-367.

Ching, J. Y., Wong, A. K. C. ve Chan, K. C. C., 1995, Class-dependent discretization for inductive learning from continuous and mixed-mode data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (7), 641-651.

Chlebus, B. S. ve Nguyen, S. H., 1998, On finding optimal discretizations for two attributes, *International Conference on Rough Sets and Current Trends in Computing*, 537-544.

Clark, P. ve Niblett, T., 1989, The CN2 induction algorithm, *Machine learning*, 3 (4), 261-283.

Cohen, W. W., 1995, Fast effective rule induction, In: Machine Learning Proceedings 1995, Eds: Elsevier, p. 115-123.

Dash, R., Paramguru, R. L. ve Dash, R., 2011, Comparative analysis of supervised and unsupervised discretization techniques, *International Journal of Advances in Science and Technology*, 2 (3), 29-37.

de Campos, L. M. L., de Oliveira, R. C. L. ve Roisenberg, M., 2016, Optimization of neural networks through grammatical evolution and a genetic algorithm, *Expert Systems with Applications*, 56, 368-384.

Debes, K., Koenig, A. ve Gross, H.-M., 2005, Transfer Functions in Artificial Neural Networks A Simulation-Based Tutorial, *Brains, Minds and Media*, 2005 (1).

Denoeux, T., 1995, A k-nearest neighbor classification rule based on Dempster-Shafer theory, *IEEE transactions on systems, man, and cybernetics*, 25 (5), 804-813.

Doğan, Z., 2012, Ayrıklaştırma Yöntemleri ve Yapay Sinir Ağı Kullanarak Asenkron Motorlarda Arıza Teşhisi. Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 168s.

Dougherty, J., Kohavi, R. ve Sahami, M., 1995, Supervised and unsupervised discretization of continuous features, In: Machine Learning Proceedings 1995, Eds: Elsevier, p. 194-202.

Dreiseitl, S. ve Ohno-Machado, L., 2002, Logistic regression and artificial neural network classification models: a methodology review, *Journal of biomedical informatics*, 35 (5-6), 352-359.

Eberhart, R. ve Kennedy, J., 1995, A new optimizer using particle swarm theory, *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, 39-43.

Ekinci, Ş., Çarman, K. ve Kahramanlı, H., 2015, Investigation and modeling of the tractive performance of radial tires using off-road vehicles, *Energy*, 93, 1953-1963.

Faris, H., Aljarah, I. ve Mirjalili, S., 2016, Training feedforward neural networks using multi-verse optimizer for binary classification problems, *Applied Intelligence*, 45 (2), 322-332.

Fayyad, U. ve Irani, K., 1993, Multi-interval discretization of continuous-valued attributes for classification learning.

Gandomi, A. H., Yang, X.-S., Talatahari, S. ve Alavi, A. H., 2013, Metaheuristic algorithms in modeling and optimization, In: Metaheuristic applications in structures and infrastructures, Eds: Elsevier, p. 1-24.

García, D. L., Nebot, À. ve Vellido, A., 2017, Intelligent data analysis approaches to churn as a business problem: a survey, *Knowledge and information systems*, 51 (3), 719-774.

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V. ve Herrera, F., 2013, A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning, *IEEE Transactions on knowledge and data engineering*, 25 (4), 734-750.

Gardner, M. W. ve Dorling, S., 1998, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, *Atmospheric environment*, 32 (14-15), 2627-2636.

Gershenson, C., 2003, Artificial neural networks for beginners, *arXiv preprint cs/0308031*.

Gunn, S. R., 1998, Support vector machines for classification and regression, *ISIS technical report*, 14 (1), 5-16.

Gupta, A., Mehrotra, K. G. ve Mohan, C., 2010, A clustering-based discretization for supervised learning, *Statistics & probability letters*, 80 (9-10), 816-824.

Gülcü, Ş. ve Kodaz, H., 2015, A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization, *Engineering Applications of Artificial Intelligence*, 45, 33-45.

Günther, F. ve Fritsch, S., 2010, neuralnet: Training of neural networks, *The R journal*, 2 (1), 30-38.

Hacibeyoglu, M., Arslan, A. ve Kahramanli, S., 2011, Improving classification accuracy with discretization on data sets including continuous valued features, *Ionosphere*, 34 (351), 2.

Hacıbeyoğlu, M. ve Ibrahim, M. H., 2016, Comparison of the effect of unsupervised and supervised discretization methods on classification process, *International Journal of Intelligent Systems and Applications in Engineering*, 105-108.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. ve Witten, I. H., 2009, The WEKA data mining software: an update, *ACM SIGKDD explorations newsletter*, 11 (1), 10-18.

Han, J., Pei, J. ve Kamber, M., 2011, Data mining: concepts and techniques, Elsevier, p.

Hashem, S., 1992, Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions, *Neural Networks, 1992. IJCNN., International Joint Conference on*, 419-424.

Haykin, S. S., Haykin, S. S., Haykin, S. S. ve Haykin, S. S., 2009, Neural networks and learning machines, Pearson Upper Saddle River, p.

Hertz, J., Krogh, A. ve Palmer, R. G., 1991, Introduction to the theory of neural computation, Addison-Wesley/Addison Wesley Longman, p.

Hippert, H. S., Pedreira, C. E. ve Souza, R. C., 2001, Neural networks for short-term load forecasting: A review and evaluation, *IEEE Transactions on Power Systems*, 16 (1), 44-55.

Holte, R. C., 1993, Very simple classification rules perform well on most commonly used datasets, *Machine learning*, 11 (1), 63-90.

Hu, H.-W., Chen, Y.-L. ve Tang, K., 2009, A dynamic discretization approach for constructing decision trees with a continuous label, *IEEE Transactions on knowledge and data engineering*, 21 (11), 1505.

Hu, L., Qin, L., Mao, K., Chen, W. ve Fu, X., 2016, Optimization of neural network by genetic algorithm for flowrate determination in multipath ultrasonic gas flowmeter, *IEEE Sensors Journal*, 16 (5), 1158-1167.

Ilonen, J., Kamarainen, J.-K. ve Lampinen, J., 2003, Differential evolution training algorithm for feed-forward neural networks, *Neural Processing Letters*, 17 (1), 93-105.

Jaddi, N. S., Abdullah, S. ve Hamdan, A. R., 2015, Optimization of neural network model using modified bat-inspired algorithm, *Applied Soft Computing*, 37, 71-86.

Jain, A. K., Mao, J. ve Mohiuddin, K. M., 1996, Artificial neural networks: A tutorial, *Computer*, 29 (3), 31-44.

Jayalakshmi, T. ve Santhakumaran, A., 2011, Statistical normalization and back propagation for classification, *International Journal of Computer Theory and Engineering*, 3 (1), 1793-8201.

Jiang, F. ve Sui, Y., 2015, A novel approach for discretization of continuous attributes in rough set theory, *Knowledge-Based Systems*, 73, 324-334.

John, G. H. ve Langley, P., 1995, Estimating continuous distributions in Bayesian classifiers, *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 338-345.

Johnson, R. A., 2009, Statistics: principles and methods, John Wiley & Sons, p.

Kankal, M. ve Uzlu, E., 2017, Neural network approach with teaching–learning-based optimization for modeling and forecasting long-term electric energy demand in Turkey, *Neural Computing and Applications*, 28 (1), 737-747.

Karaboga, D., 2005, An idea based on honey bee swarm for numerical optimization, *Technical report-tr06, Erciyes university, engineering faculty, computer …*.

Karlik, B. ve Olgac, A. V., 2011, Performance analysis of various activation functions in generalized MLP architectures of neural networks, *International Journal of Artificial Intelligence and Expert Systems*, 1 (4), 111-122.

Kattan, A., Abdullah, R. ve Salam, R. A., 2010, Harmony search based supervised training of artificial neural networks, *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, 105-110.

Kerber, R., 1992, Chimerge: Discretization of numeric attributes, *Proceedings of the tenth national conference on Artificial intelligence*, 123-128.

Koç, İ., 2016, Sınıflandırma problemlerinde meta-sezgisel optimizasyon yöntemlerinin özellik seçimi ve ayrıklaştırma amacıyla kullanımı, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Koçoğlu, F. Ö., 2012, Veri madenciliği sürecinde veri ayrıklaştırma yöntemlerinin karşılaştırılması ve bir uygulama.

Kohavi, R. ve Sahami, M., 1996, Error-based and entropy-based discretization of continuous features, *KDD*, 114-119.

Kotsiantis, S. ve Kanellopoulos, D., 2006, Discretization techniques: A recent survey, *GESTS International Transactions on Computer Science and Engineering*, 32 (1), 47-58.

Kowalski, P. A. ve Łukasik, S., 2016, Training neural networks with krill herd algorithm, *Neural Processing Letters*, 44 (1), 5-17.

Kriesel, D., 2007, A brief introduction on neural networks.

Kulluk, S., 2009, Karınca koloni optimizasyonu ile yapay sinir ağlarından kural çıkarımı, *Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı, Kayseri*.

Kumar, S. S. ve Inbarani, H. H., 2018, Cardiac arrhythmia classification using multi-granulation rough set approaches, *International Journal of Machine Learning and Cybernetics*, 9 (4), 651-666.

Kurgan, L. A. ve Cios, K. J., 2003, Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm, *ICMLA*, 30-36.

Kurgan, L. A. ve Cios, K. J., 2004, CAIM discretization algorithm, *IEEE Transactions on knowledge and data engineering*, 16 (2), 145-153.

Lacher, R., Hruska, S. I. ve Kuncicky, D. C., 1992, Back-propagation learning in expert networks, *IEEE Transactions on Neural Networks*, 3 (1), 62-72.

Lee, K. S. ve Geem, Z. W., 2005, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer methods in applied mechanics and engineering*, 194 (36-38), 3902-3933.

Leung, H. ve Haykin, S., 1991, The complex backpropagation algorithm, *IEEE Transactions on signal processing*, 39 (9), 2101-2104.

Li, W., Han, J. ve Pei, J., 2001, CMAR: Accurate and efficient classification based on multiple class-association rules, *icdm*, 369.

Liao, S.-H., Hsieh, J.-G., Chang, J.-Y. ve Lin, C.-T., 2015, Training neural networks via simplified hybrid algorithm mixing Nelder–Mead and particle swarm optimization methods, *Soft Computing*, 19 (3), 679-689.

Liu, H. ve Setiono, R., 1995, Chi2: Feature selection and discretization of numeric attributes, *Tools with artificial intelligence, 1995. proceedings., seventh international conference on*, 388-391.

Liu, H., Hussain, F., Tan, C. L. ve Dash, M., 2002, Discretization: An enabling technique, *Data mining and knowledge discovery*, 6 (4), 393-423.

Liu, H., Tian, H.-q., Liang, X.-f. ve Li, Y.-f., 2015, Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks, *Applied Energy*, 157, 183-194.

Loh, W. Y., 2011, Classification and regression trees, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1 (1), 14-23.

Lustgarten, J. L., Gopalakrishnan, V., Grover, H. ve Visweswaran, S., 2008, Improving classification performance with discretization on biomedical datasets, *AMIA annual symposium proceedings*, 445.

McCallum, A. ve Nigam, K., 1998, A comparison of event models for naive bayes text classification, *AAAI-98 workshop on learning for text categorization*, 41-48.

Michalski, R. S., Bratko, I. ve Bratko, A., 1998, Machine learning and data mining; methods and applications, John Wiley & Sons, Inc., p.

Michalski, R. S., Carbonell, J. G. ve Mitchell, T. M., 2013, Machine learning: An artificial intelligence approach, Springer Science & Business Media, p.

Mirjalili, S. Z., Saremi, S. ve Mirjalili, S. M., 2015, Designing evolutionary feedforward neural networks using social spider optimization algorithm, *Neural Computing and Applications*, 26 (8), 1919-1928.

Møller, M. F., 1993, A scaled conjugate gradient algorithm for fast supervised learning, *Neural networks*, 6 (4), 525-533.

Moré, J. J., 1978, The Levenberg-Marquardt algorithm: implementation and theory, In: Numerical analysis, Eds: Springer, p. 105-116.

Nawi, N. M., Ransing, R. ve Ransing, M., 2007, An improved conjugate gradient based learning algorithm for back propagation neural networks, *International Journal of Computational Intelligence*, 4 (1), 46-55.

Özdemir, A., 2010, Genetik algoritma ile yapay sinir ağlarında yapı ve parametre optimizasyonu, Fırat Üniversitesi Fen bilimleri enstitüsü, 94s.

Özdemir, S., 2011, A Decision Tree Based Intrusion Detection System With Bootstrap Aggregating, Discretization, and Feature Selection, Boğaziçi University, Graduate Program in Electrical and Electronics Engineering, 80s.

Öztürk, C., 2011, Yapay sinir ağlarının yapay arı kolonisi algoritması ile eğitilmesi, *Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı*, 205.

Peng, L., Qing, W. ve Yujia, G., 2009, Study on comparison of discretization methods, *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, 380-384.

Pyle, D., 1999, Data preparation for data mining, morgan kaufmann, p.

Quinlan, J. R., 2014, C4. 5: programs for machine learning, Elsevier, p.

Rahman, M. G. ve Islam, M. Z., 2016, Discretization of continuous attributes through low frequency numerical values and attribute interdependency, *Expert Systems with Applications*, 45, 410-423.

Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S., 2009, GSA: a gravitational search algorithm, *Information sciences*, 179 (13), 2232-2248.

Riedmiller, M. ve Braun, H., 1993, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Neural Networks, 1993., IEEE International Conference on*, 586-591.

Rish, I., 2001, An empirical study of the naive Bayes classifier, *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 41-46.

Rong, T., Gong, H. ve Ng, W. W., 2014, Stochastic sensitivity oversampling technique for imbalanced data, *International Conference on Machine Learning and Cybernetics*, 161-171.

Rosner, B., Glynn, R. J. ve Lee, M. L. T., 2006, The Wilcoxon signed rank test for paired comparisons of clustered data, *Biometrics*, 62 (1), 185-192.

Saini, L. M. ve Soni, M. K., 2002, Artificial neural network-based peak load forecasting using conjugate gradient methods, *IEEE Transactions on Power Systems*, 17 (3), 907-912.

Salama, K. M. ve Abdelbar, A. M., 2015, Learning neural network structures with ant colony algorithms, *Swarm Intelligence*, 9 (4), 229-265.

Sasaki, M. ve Kita, K., 1998, Rule-based text categorization using hierarchical categories, *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, 2827-2830.

Sayın, K., 2013, Feature Selection and Discretization For Improving Classification Performance on CAC Data Set, Kadir Has University Graduate School Of Science And Engineering, 45s.

Shi, Y., 2001, Particle swarm optimization: developments, applications and resources, *evolutionary computation, 2001. Proceedings of the 2001 Congress on*, 81-86.

Singh, A., Tiwari, V. ve Tentu, A. N., 2018, A Machine Vision Attack Model on Image Based CAPTCHAs Challenge: Large Scale Evaluation, *International Conference on Security, Privacy, and Applied Cryptography Engineering*, 52-64.

Stein, C., Cormen, T., Rivest, R. ve Leiserson, C., 2001, Introduction to algorithms, *The MIT Press*, 31 (77), 13.

Steinberg, D. ve Colla, P., 2009, CART: classification and regression trees, *The top ten algorithms in data mining*, 9, 179.

Su, C.-T. ve Hsu, J.-H., 2005, An extended chi2 algorithm for discretization of real value attributes, *IEEE Transactions on knowledge and data engineering*, 17 (3), 437-441.

Uzlu, E., Akpınar, A., Özturk, H. T., Nacar, S. ve Kankal, M., 2014, Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for Turkey, *Energy*, 69, 638-647.

Wiens, T. S., Dale, B. C., Boyce, M. S. ve Kershaw, G. P., 2008, Three way k-fold cross-validation of resource selection functions, *Ecological Modelling*, 212 (3-4), 244-255.

Witten, I. H., Frank, E., Hall, M. A. ve Pal, C. J., 2016, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, p.

Wong, T.-T., 2012, A hybrid discretization method for naïve Bayesian classifiers, *Pattern Recognition*, 45 (6), 2321-2325.

Wong, T.-T., 2015, Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation, *Pattern Recognition*, 48 (9), 2839-2846.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B. ve Philip, S. Y., 2008, Top 10 algorithms in data mining, *Knowledge and information systems*, 14 (1), 1-37.

Yalçın, N., 2012, Sezgisel algoritma öğrenmeli yapay sinir ağları ile epilepsi hastalığının teşhisi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.

Yan, D., Liu, D. ve Sang, Y., 2014, A new approach for discretizing continuous attributes in learning systems, *Neurocomputing*, 133, 507-511.

Yang, X.-S. ve Deb, S., 2009, Cuckoo search via Lévy flights, *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 210-214.

Yang, X.-S., 2010, A new metaheuristic bat-inspired algorithm, In: Nature inspired cooperative strategies for optimization (NICSO 2010), Eds: Springer, p. 65-74.

Yang, Y. ve Webb, G. I., 2009, Discretization for naive-Bayes learning: managing discretization bias and variance, *Machine learning*, 74 (1), 39-74.

Yao, X., 1999, Evolving artificial neural networks, *Proceedings of the IEEE*, 87 (9), 1423-1447.

Yegnanarayana, B., 2009, Artificial neural networks, PHI Learning Pvt. Ltd., p.

Yin, X. ve Han, J., 2003, CPAR: Classification based on predictive association rules, *Proceedings of the 2003 SIAM International Conference on Data Mining*, 331-335.

Zaki, M. J., Meira Jr, W. ve Meira, W., 2014, Data mining and analysis: fundamental concepts and algorithms, Cambridge University Press, p.

Zieliński, K. ve Szmuc, T., 2005, Software engineering: Evolution and emerging technologies, IOS Press, p.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	MOHAMMED HUSSEIN IBRAHIM
Uyruğu	**:**	IRAK
Doğum Yeri ve Tarihi	**:**	KERKÜK / 04.07.1980
Telefon	**:**	05375666879
Faks	**:**	
e-mail	**:**	mohammedkbc@gmail.com / mibrahim@erbakan.edu.tr

EĞİTİM

Derece		Adı, İlçe, İl	Bitirme Yılı
Lise	:	**EL-HİKME, KERKÜK**	**1998**
Üniversite	:	**TEKNİK FAKÜLTESİ, KERKÜK Bilgisayar Yazılım Mühendisliği Bölümü**	**2003**
Yüksek Lisans	:	**SELÇUK ÜNİVERSİTESİ, KONYA Bilgisayar Mühendisliği Bölümü**	**2011**
Doktora	:	**SELÇUK ÜNİVERSİTESİ, KONYA Bilgisayar Mühendisliği Bölümü**	**2019**

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2015-Devam ediyor	**Necmettin Erbakan Üniversitesi Bilgisayar Mühendisliği Bölümü**	**Öğretim Görevlisi**

UZMANLIK ALANI

Makine öğrenmesi, Veri madenciliği, Gömülü sistemleri, Lojik tasarım, Optimizasyon algoritmaları.

YABANCI DİLLER

İngilizce, Arapça

YAYINLAR

Hacibeyoglu, M., & Ibrahim, M. H. (2018). EF_Unique: An Improved Version of Unsupervised Equal Frequency Discretization Method. Arabian Journal for Science and Engineering, 1-10.

Hacibeyoglu, M., & Ibrahim, M. H. (2018). A Novel Multimean Particle Swarm Optimization Algorithm for Nonlinear Continuous Optimization: Application to Feed-Forward Neural Network Training. Scientific Programming, 2018.