



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YARASA ALGORİTMASI İLE TIBBI VERİ SETLERİNİN KÜMELENMESİ

Mina Arjamand Fadhil FADHIL

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

KONYA-2019

ÖZET

YÜKSEK LİSANS TEZİ

YARASA ALGORİTMASI İLE TIBBI VERİ SETLERİNİN KÜMELENMESİ

Mina Arjamand Fadhil FADHIL

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Gülay TEZEL

2019, 36 Sayfa

Jüri
Doç. Dr. Gülay TEZEL
Doç. Dr. Humar Kahramanlı ÖRNEK
Dr. Öğr. Üyesi Betül UZBAŞ

Kümeleme, nesneler arasındaki benzerliğe göre nesneleri gruplandıran danışmansız öğrenme algoritmasıdır. Aynı kümedeki nesneler birbirine çok benzerler. Ancak bu nesneler diğer küme nesneleri arasında farklılar göstermektedir. Kümeleme işlemi birçok kümeleme algoritmaları ile yapılabilir, önemli olan veriler arasında en iyi küme merkezlerinin bulunmasıdır. Bu tez çalışmasında, Yarasa Algoritması (YA) tabanlı bir kümeleme yöntemi önerilmiştir. Bu tezde önerilen YA tabanlı kümeleme yönteminin performansı İris, Wine, Tae, Glass ve Wbc veri setleri üzerinde literatürde kullanılan farklı kümeleme yöntemleri ile karşılaştırılıp değerlendirilmiştir. Sonuç olarak önerilen yarasa algoritması tabanlı kümeleme yöntemi özellikle Tae ve Glass veri setlerinde diğer algoritmalardan daha iyi sonuç vermiştir. Önerilen kümeleme yöntemi bu veri setleri ile değerlendirildikten sonra SPECT_kalp, Breast Cancer Wisconsin Diagnostic, SPECTF_kalp, Dermatology, Statlog_Heart ve Kalp_Hastalıkları tıbbı veri setlerinin kümelenmesinde de uygulanmıştır. 30 çalışmadan bazılarında Tıbbi veri setlerindeki veri örneklerinin yaklaşık %78'nin YA ile doğru kümelenebildiği görülmüştür.

Anahtar Kelimeler: Denetimsiz Sınıflandırma, Kümeleme, Optimizasyon, Yarasa Algoritması.

ABSTRACT

MS THESIS

CLUSTERING OF MEDICAL DATA SETS WITH BAT ALGORITHM

Mina Arjamand Fadhil FADHIL

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

Advisor: Assoc.Prof.Dr. Gülay TEZEL

2019, 36 Pages

Jury
Assoc.Prof.Dr. Gülay TEZEL
Assoc.Prof.Dr. Humar Kahramanlı ÖRNEK
Assist.Prof.Dr. Betul UZBAŞ

Clustering is an unsupervised learning algorithm that clusters group objects based on the similarity between objects. The objects in the same cluster are very similar. But they show differences from the other cluster objects. Clustering can be done with many clustering algorithms, and it is important to find the best cluster centers among the data. In this thesis, clustering method based on Bat Algorithm (BA) is proposed. The performance of the proposed clustering method based BA in this thesis was compared with different clustering methods in literature by using Iris, Wine, Tae, Glass and Wbc data sets. As a result, the proposed clustering method based BA yielded better results in the specially Tae and Glass data sets. After evaluating the proposed clustering method with these datasets, it was also applied to clustering SPECT_Heart, Breast Cancer Wisconsin Diagnostic, SPECTF_Heart, Dermatology, Statlog_Heart and Heart_Health medical data sets. It was seen that approximately 78% of data samples in the medical datasets were correctly clustered in the some of 30 runs.

Keywords: Bat Algorithm, Clustering, Optimization, Unsupervised Classification.

ÖNSÖZ

Bu tezi tamamlamamı sağlayan yüce Allah'a şükür ederim. Ayrıca sürekli rehberlik ve desteğinden dolayı danışmanım Sayın Dr. Gülay Tezel'e, akademisyenlerime, bana çok yardımcı olan arkadaşlarıma ve manevi desteğiyle beni hiçbir zaman yalnız bırakmayan sevgili Anne ve Babama, teşekkürü bir borç bilirim.

Mina Arjamand Fadhil FADHIL
KONYA-2019

İÇİNDEKİLER

ÖZET ..... **iv**
ABSTRACT ..... **v**
ÖNSÖZ ..... **vi**
İÇİNDEKİLER ..... **vii**
KISALTMALAR ..... **viii**
1. GİRİŞ **1**
2. KAYNAK ARAŞTIRMASI **3**
3. MATERYAL VE YÖNTEM **9**
3.1. Veri Setleri **9**
3.2. Kümeleme ve Kümeleme Yöntemleri **12**
3.2.1. K-Ortalama Kümeleme Algoritması 14
3.2.2. Bulanık C-Ortalamalar Yöntemi 18
3.3. Yarasa Algoritması (YA) **20**
3.3.1. Yarasaların Hareketi 21
3.3.2. Ses Şiddeti ve Sinyal Yayma Oranı 22
3.3.3. Yarasa Algoritması Tabanlı Kümeleme 22
4. UYGULAMA SONUÇLARI **25**
KAYNAKLAR **34**
ÖZGEÇMİŞ **36**

KISALTMALAR

YA	:	Yarasa Algoritması
SSE	:	Toplam Karesel Hata (Sum Squared Error)
RI	:	Rand Index
FCM	:	Bulanık C- Ortalamalar (Fuzzy C-Means)
PSO	:	Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
KIFCM	:	Kernel İntuitionistic Fuzzy C-Means
KFCM	:	Kernel Fuzzy C-Means
ABC	:	Yapay Arı Koloni Algoritması (Artificial Bee Colony)
GA	:	Genetik Algoritma
PM-FGCA	:	Partition-And-Merge Based Fuzzy Genetic Clustering Algorithm
BOA	:	Balina Optimizasyon Algoritması
LBG	:	Linde Buzo Gray
YAKA	:	Yapay Arı Koloni Algoritması
BOA	:	Balina Optimizasyon Algoritması
NMI	:	Normalleştirilmiş Karşılıklı Bilgi
DB	:	Davies Bouldin
EKMK	:	En Küçük Medyan Kareler
PSO- LBG	:	Particle Swarm Optimization Linde Buzo Gray
FA - LBG	:	Firefly Algorithm Linde Buzo Gray
HBMO	:	Honey Bee Mating Optimization
QPSO	:	Quantum Particle Swarm Algorithm
HMM	:	Hidden Markov Model
UCAV	:	Uninhabited Combat Air Vehicles
DE	:	Diferansiyel Evrim
GYA	:	Gelişmiş Yarasa Algoritması
UCAV	:	Unmanned Combat Air Vehicle
WSN	:	Wireless Sensor Network
LEACH	:	Low Energy Adaptive Clustering Hierarchal

1. GİRİŞ

Kümeleme sorunu, verileri benzer nesnelerin gruplarına bölmenin bir yolunu bulma sorunudur. Hayal edilebilecek birçok durum için verilerin bulunabileceği, çıkarılabileceği veya üretilebileceği ortamlar günlük yaşamda yaygındır. Verileri yönetmeyi veya bulmayı kolaylaştırmak için, bir şekilde benzer olan verileri gruplandırmak veya kümelemek yararlı bir uygulamadır. Hem veri noktalarının hem de değişkenlerin sayısı arttığı için veri kümelerinin sayısı da büyür. Bu nedenle kümeleme algoritmaları ile verileri gruplandırmak giderek daha önemli olmuştur. Son yıllarda, bu sorunun çözülebilmek amacıyla birçok farklı yaklaşım önerilmiştir (Zweije, 2017).

Kümeleme, çok değişkenli veri analizinde kullanılan yöntemlerden biridir. Bu nedenle de veri madenciliğinin en önemli dalları arasındadır. Kümeleme işlemi, aynı kümedeki nesnelerin mümkün olduğu kadar benzer ve farklı kümelerdeki nesnelerin mümkün oldukça farklı olmalarını hedefler. Kümeleme yöntemleri tıp verileri, psikoloji verileri, ekonomi verileri ve örüntü tanıma gibi birçok alanda uygulanmaktadır. Kümeleme işlemi bir sınıflandırma işlemi gibi düşünülebilir fakat sınıflandırma işleminde verilerin hangi sınıfa ait oldukları bellidir. Kümeleme işleminde ise verilerin hangi sınıfa ait olduklarını kümeleme algoritmaları belirler. Dolayısıyla kümeleme işlemi danışmansız bir öğrenme algoritması olarak tanımlanabilir. Kümeleme problemleri için bu güne kadar K-Ortalamalar, Bulanık C-Ortalamalar ve Hiyerarşik gibi birçok kümeleme algoritması geliştirilmiştir. Bu kümeleme algoritmaları birçok alanda çok başarılı bir şekilde kullanılmıştır. Bu kümeleme algoritmaları çeşitli sayıda değişkene göre nesneler arasındaki benzerlik ve uzaklıkları hesaplamak için Öklid Uzaklığı, Manhattan Uzaklığı, Mahalanobis Uzaklığı ve Minkowski Uzaklığı gibi değerlendirme ölçütlerini kullanmaktadır (Işık ve Çamurcu, 2007; Bilekdemir, 2010; Kuoa ve ark., 2018).

Geliştirilen bu kümeleme algoritmaları farklı veri kümeleri üzerinde veri büyüklüğü, karmaşıklığı ve özellik sayısı gibi kriterlere bağlı olarak farklı performans gösterebilmektedir. Kümeleme işleminin başarısını algoritmanın yapısına bağlı olarak birden fazla faktör etkilemektedir. Bu faktörlerin arasında küme merkezlerinin güncellenmesi yöntemi çok büyük önem taşımaktadır. Dolayısıyla bu küme merkezlerinin iyi bir küme merkezi güncelleme algoritması ile güncellenmesi gerekmektedir. Bu amaçla geliştirilmiş farklı yöntemler vardır. Son dönemlerde optimizasyon algoritmaları da bu amaçla kullanılmaktadır. Özellikle Genetik Algoritma

(GA), Parçacık Sürü optimizasyonu (PSO), Yapay Arı Kolonisi (ABC) algoritmaları yaygın olarak literatürde tercih edilmiştir (Senthilnath ve ark., 2016). Yarasa Algoritması (YA), literatürde kümeleme problemlerinin çözümünde az sayıda çalışma olan algoritmalardan biridir. Genellikle hibrit yapılarda tercih edilmiştir (Senthilnath ve ark., 2016).

YA, yarasaların ekolokasyon adı verilen sesin yankılanmasından faydalanılarak bir cismin bulunduğu yön ve uzaklık saptama davranışından esinlenen bir popülasyon tabanlı optimizasyon algoritmasıdır. YA, 2010 yılında Yang (Yang, 2010b) tarafından sürü zekasına dayanan ve meta-sezgisel bir optimizasyon algoritması olarak önerilmiş ve optimizasyon çalışmalarının yanında kümeleme problemlerinde de kullanılmıştır (Tripathi ve ark., 2018). Kümeleme problemleri için ise YA kümeleme amacıyla K-ortalama kümeleme algoritması ile hibrit olarak çalıştırılmıştır (Senthilnath ve ark., 2016). Senthilnath ve ark. (2016) multispektral uydu görüntülerinin YA tabanlı kümeleme algoritması ile kümelenmesi üzerine çalışmış ve farklı hibrit yapılarla sonuçları karşılaştırmıştır. Yılmaza ve Küçüksille, (2014), yerel ve küresel arama özelliklerini üç farklı yöntemle uygulamış ve gerçek dünya problemlerinde denemiştir.

Bu tez çalışmasında YA kümeleme algoritması olarak tasarlanmış ve UCI Makine Öğrenim ambarından alınan 11 adet veri kümesi ile denemeler yapılmıştır. Sonuçlar Rand Index (RI) ve Toplam Karesel Hata (Sum Square Error-SSE) kriterlerine göre literatür ile karşılaştırılmıştır.

2. KAYNAK ARAŞTIRMASI

Kümeleme işlemi birçok kritik alanlarda kullanıldığından dolayı araştırmacılar tarafından çeşitli özeliklere sahip kümeleme algoritmaları geliştirilmiştir. Geliştirilen kümeleme algoritmaları çalışma mekanizmalarına göre belirli veri setlerinde iyi başarı gösterebilmektedir.

Bilekdemir (2010), kümeleme algoritmalarından biri olan K-ortalamalar kümeleme algoritmasının daha iyi sonuçlar vermesi için genetik algoritma operatörleri kullanmıştır. K-ortalamalar kümeleme algoritmasının en büyük dezavantajı olan yerel optimumlara takılma problemi, genetik operatörlerle (çaprazlama, mutasyon, seçilim ve seçkinlik) çözülmeye çalışılmıştır (Bilekdemir, 2010).

Bezdek ve ark. (1984), Bulanık C-Ortalamalar (FCM) kümeleme programını FORTRAN-IV ile kodlamıştır. Yapılan FCM programı, çok çeşitli coğrafi veri analizi problemlerine uygulanabilmektedir. Bu program, herhangi bir sayısal veri kümesi için bulanık bölümler ve prototipler üretir. Bu bölümler, bilinen altyapıları doğrulamak veya keşfedilmemiş verilerde alt yapı önermek için kullanışlıdır (Bezdek ve ark., 1984)

Işık ve Çamurcu (2007), başlıca bölünmeli kümeleme algoritmalarından K-Ortalamalar, K-Medoids ve Bulanık C-Ortalamalar algoritmalarının kümeleme yetenekleri ve performanslarını karşılaştırılmıştır. Bu algoritmalar merkez tabanlı kümeleri tespit etmede başarılıdır. Kümeleme algoritmalarında bulunan bölünmeli kümeleme tekniği, nesneleri giriş parametre sayısı kadar kümeye bölmektedir. Literatürde yer alan sentetik veri setleri kullanılmıştır (Işık ve Çamurcu, 2007)

Bir diğer çalışmada, Lamar ve ark., (2010) belge kümelemesi için bir Parçacık Sürü Optimizasyonu (PSO) algoritması sunmuştur. PSO kümeleme algoritmasında, arama alanı içinde, K tarafından yapılan yerel aramaya karşı kümeleme anlamına gelen global bir arama kullanılmıştır. Deneyler için, 4 metin belgesi veri setine PSO, K-ortalamalar ve hibrit bir PSO algoritması uygulanmıştır. Sonuçlar, PSO hibritinin K-ortalamalar daha iyi kümeleme yaptığı gösterilmiştir (Lamar ve ark., 2010).

Otomatik kümeleme problemini çözmek amacıyla Özturk ve ark., (2014) Yapay Arı Koloni Algoritması'nın (YAKA) küresel araştırma kabiliyeti geliştirmiş ve algoritmanın vektörle araştırma yapabilmesi sağlanmıştır. Önerilen yöntem en çok bilinen veri ve görüntü küme üzerinde test edilmiştir. Elde edilen sonuçlar neticesinde önerilen metodun diğer metotlara oranla daha iyi performans sağladığı ve otomatik

kümeleme problemlerinin çözümünde rahatlıkla kullanılabileceği görülmüştür (Özturk ve ark., 2014).

Yang tarafından 2010'da geliştirilen ve ilham verici bir algoritmadır ve YA'nın çok verimli olduğu bulunmuştur. Yang, (2013) bu makalede yarasa algoritmasının ve yeni varyantlarının gözden geçirilmesini sağlamaktadır. Çok çeşitli uygulamalar ve vaka çalışmaları da burada kısaca gözden geçirilmiş ve özetlenmiştir. Daha fazla araştırma konusu da tartışılmıştır (Yang, 2013) .

Gandomi ve ark. (2013) yeni bir meta-sezgisel optimizasyon algoritması olan YA, optimizasyon görevlerini kısıtlı olarak çözmek için kullanılmıştır. YA, birçok kıyaslama problemine başvurarak doğrulanmıştır ve bu görev için mevcut diğer çeşitli algoritmalardan daha iyi performans göstermiştir. Ayrıca yarasa algoritmasının benzersiz arama özellikleri de analiz edilmiştir (Gandomi ve ark., 2013).

Yeni bir YA tabanlı kümeleme yaklaşımı önerilmiştir. Önerilen yarasa tabanlı kümeleme algoritması multispektral uydu görüntüsü üzerinde test edilmiştir. Önerilen bölümleme kümeleme algoritması, eğitim örneklerinden optimal küme merkezleri şeklinde bilgi çıkarmak için kullanılmıştır. YA'nın performansı diğer iki doğadan esinlenen GA ve PSO ile karşılaştırılmıştır. Performans aynı zamanda K-Ortalamalar içeren YA gibi mevcut karma yaklaşımla karşılaştırılmıştır. Elde edilen sonuçlara göre, YA'nın ürün tipi sınıflandırma problemlerini çözmede başarıyla uygulanabileceği sonucuna varılmıştır (Senthilnath ve ark., 2016)

Canayaz ve Özdağ (2017), güncel meta-sezgisel algoritmalardan olan sürü tabanlı Balina Optimizasyon Algoritması (BOA)'nın veri kümelemedeki performansı incelemiştir. BOA kambur balinaların avlanma stratejisinden esinlenmiş olan meta-sezgisel algoritmalardan biridir. Kambur balinaların kendilerine özgü avlanma stratejileri bulunmaktadır. Veri kümeleme işlemi için kullanılan olan bu algoritmada verilen veri setleri içinde en iyi özellik vektörü çıkarılarak kümeleme işlemi için bu vektör kullanılmıştır (Canayaz ve Özdağ, 2017).

Kuo ve ark., (2018) kümeleme işlemi için Kernel Intuitionistic Bulanık C-Ortalamalar (KIFCM) adında bir kümeleme algoritmasını önermişlerdir. Önerilen KIFCM algoritması Bulanık C-Ortalamalar kümeleme algoritmasını iyileştirmiştir. Daha sonra PSO, genetik ve YAKA meta-sezgisel algoritmalar ile karşılaştırılarak önerilen KIFCM algoritmasının daha iyi küme merkezlerini bulduğu tespit edilmiştir. Önermiş oldukları PSO- KIFCM, GA- KIFCM ve ABC- KIFCM algoritmalarını altı

veri kümesi üzerinde test etmişlerdir. Deneysel sonuçlarına göre önermiş oldukları veri kümeleri üzerinde başarılı sonuçlar elde edilmiştir (Kuo ve ark., 2018).

Nguyen ve Kuo, (2019) kategorik veriler için Bölümleme ve Birleştirme Tabanlı Bulanık Genetik Kümeleme Algoritması (PM-FGCA) adında iki aşamalı bir yöntem önerilmişlerdir. Önerilen PM-FGCA'da, veri setini ilk aşamada maksimum sayıda kümeye bölmek için bulanık bir genetik kümeleme algoritması kullanılmıştır. Önerilen PM-FGCA, on kategorik veri setinde uygulanmıştır. PM-FGCA, k-mod algoritması, bulanık k-mod algoritması, genetik bulanık k-mod algoritması ve bulanık üyelik kromozomları kullanılarak domine edilmemiş sıralama genetik algoritması gibi bazı mevcut yöntemlerin karşılaştırılabilmesi için UCI makine öğrenim havuzundan veriler kullanılmıştır. Kümeleme performansını değerlendirmek için önerilen Düzeltilmiş Sıra İndeksi (DSI), Normalleştirilmiş Karşılıklı Bilgi (NKB) ve Davies-Bouldin (DB) endeksi, hem dış endekste (yani, DSI ve NKB) hem de iç indekste (yani, DB) temsil edilen üç kümeleme doğrulama endeksi giibi yöntemler kullanılmıştır. Sonuç olarak, önerilen PM-FGCA'nın test edilen endeksler açısından diğer kıyaslama yöntemlerinden daha iyi performans gösterdiği ifade edilmiştir (Nguyen ve Kuo, 2019).

Genlerin kümelenmesi, biyoinformatikte en önemli problemlerden biridir. Ardışık verilerin kümelemesinde, saklı Markov modelleri (Hidden Markov Models, HMM'ler), dizi uzunluklarını çeşitli uzunluklarda kullanma kabiliyetlerinden dolayı, sekanslar arasındaki benzerliği bulmak için yaygın olarak kullanılmaktadır. Soruri ve ark. (2018) yapılan çalışmada, parçacık sürüsü optimizasyon algoritması (POS) tarafından optimize edilmiş HMM'lere dayalı yeni bir gen kümeleme şeması tanıtılmıştır. Her gen dizisi belirli bir HMM ile tanımlanır ve daha sonra her model için, bireysel dizilim üretme olasılığı değerlendirilir. En iyi kümeleri bulmak için yeni bir uzaklık ölçüsünün tanımlanmasına dayalı hiyerarşik bir kümeleme algoritması uygulanmıştır. Akciğer kanseriyle ilişkili genler veri kümesi üzerinde yapılan deneyler sonucunda, önerilen yaklaşımın gen kümelemesi için başarıyla kullanılabileceğini gösterilmiştir (Soruri ve ark. 2018).

Tripathi ve ark. (2018) YA tabanlı yeni bir dinamik frekansı temel alan kümeleme algoritması sunmuştur. Ayrıca önerilen yarasa algoritması tabanlı kümeleme algoritması K-ortalama kümeleme algoritması ile hibrit olarak çalışmaktadır. Evrimsel hesaplamada, hesaplama işlemleri çok büyük veriler üzerinde olduğundan dolayı sezgisel algoritmalar makul bir süre içerisinde tatmin edici sonuçlar sağlamamaktadır. Bu sebepten dolayı büyük veri problemleri için meta-sezgisel algoritmalar iyi sonuca

yakın sonuçlar üretmektedir. Araştırmacılar tarafından önerilen kümeleme yaklaşımı k-ortalama ve PSO algoritmaları ile karşılaştırılmıştır. Elde edilen deneysel sonuçlarına göre, önerilen yarasa tabanlı kümeleme algoritmasının diğer kümeleme algoritmalarına göre daha iyi performans sergilediği gözlemlenmiştir (Tripathi ve ark, 2018).

Karri ve Jena (2016) LBG'nin (Linde–Buzo–Gray) ilk olarak YA'yı kullanan YA-LBG adlı yeni bir algoritma olarak önermişlerdir. Bu yöntem daha az hesaplama süresi olan verimli bir kod çizelgesi üretmektedir. Ayarlanabilir nabız emisyon oranı ve yarasaların yüksekliğini kullanan otomatik yakınlaştırma özelliği sayesinde çok iyi PSNR sağlar. Sonuçlar, Yarasa Algoritması-Linde Buzo Gray 'nin (BA-LBG), Particle Swarm Optimization-Linde Buzo Gray (PSO-LBG), Quantum PSO-LBG, Honey Bee Mating Optimization (HBMO-LBG) ve Firefly Algorithm-Linde–Buzo–Gray'ye (FA-LBG) kıyasla yüksek Sinyal-Gürültü Oranına (Peak signal-to-noise ratio PSNR) sahip olduğu ve ortalama yakınsama hızının HBMO-LBG ve FA-LBG'den 1,8 kat daha hızlı olduğunu göstermiştir. PSO ile arasında anlamlı bir fark yoktur (Karri ve Jena, 2016).

YA, mikrobatların eko-konumlandırma ya da biyolojik özelliklerine dayanan, 2010'da önerilen popülasyon tabanlı metaheuristik bir yaklaşımdır. Osaba ve ark., (2015) klasik YA temel yapısında bir gelişme önermiştir. Student t testi, Holm testi ve Friedman testi adında üç farklı istatistiksel test gerçekleştirilmiştir. Aynı zamanda Evrimsel Benzetilmiş Tavlama ve Ayrık Ateşböceği Algoritması ile elde edilen sonuçlarla karşılaştırılmıştır. Bu çalışma, sunulan gelişmiş yarasa algoritmasının, çoğu durumda diğer tüm alternatifleri önemli ölçüde yerine getirdiğini göstermiştir (Osaba ve ark., 2015).

Esnek atölye tipi çizelgeleme problemleri, işlerin makinelere atanması ve sıralanması gibi alt problemlerinden oluşmaktadır. Bu tür problemlerin çözümü için literatürde hiyerarşik ve bütünleşik olmak üzere iki tür çözüm yaklaşımı geliştirilmiştir. Bu tür problemlerin çözümünde sıklıkla meta sezgisel yöntemlere başvurulmaktadır. Serkan ve Fiğlali (2018) çalışmalarında çizelgeleme alanının en zor problem sınıfından olan esnek atölye tipi çizelgeleme problemlerinin meta sezgisel yöntemler ile çözümüne yönelik önerilerde bulunmuştur. Meta sezgisel yöntemlere en kısa işlem zamanı gibi basit sıralama kuralları ve yerel arama algoritmalarının hibritlenmesi uygulamalı olarak verilmiştir (Serkan ve Fiğlali, 2018)

Wanga ve ark. (2012) tarafından Uninhabited Combat Air Vehicles (UCAV) üç boyutlu yol planlama problemini ilk kez optimize etmek için, Diferansiyel Evrim (DE) ile birlikte YA kullanarak yeni IBA adında bir versiyonu önerilmiştir. IBA'da DE'nin

yarasa popülasyonunda en uygun kişiyi seçmesi gerekir. Seçilen düğümleri önerilen IBA'yı kullanarak, güvenli bir yol başarıyla elde edilir. Ek olarak, B-Spline eğrileri daha fazla elde edilen yolu basitleştirmek ve pratik olarak UCAV'a daha uygun hale getirmek için kullanılmıştır. IBA'nın performansı, bir 3-D UCAV yolu planlama problemindeki temel YA ile karşılaştırılmıştır. Deneysel sonuçlar, IBA'nın, UCAV üç boyutlu yol planlama problemleri için temel YA modeline kıyasla daha iyi bir teknik olduğunu göstermiştir (Wanga ve ark., 2012).

Yılmaza ve Kücüksille (2014), YA'nın yerel ve küresel arama özelliklerini üç farklı yöntemle test etmiştir. Gelişmiş Yarasa Algoritması'nın (GYA) performansının doğrulanması için standart test fonksiyonları ve kısıtlı gerçek dünya problemleri kullanılmıştır. Bu test setleri tarafından elde edilen sonuçlarla, EBA'nın standart olandan daha üstün olduğu kanıtlanmıştır. Ayrıca, bu çalışmada önerilen yöntem, literatürdeki gerçek dünya problemleri üzerine yapılan son araştırmalarla karşılaştırılmıştır. Sonuç olarak bu yöntemin bu tür problemler üzerine diğer literatüre ait çalışmalardan daha etkili olduğu kanıtlanmıştır (Yılmaza ve Küçüksille, 2014).

YA gibi modern metasezgisel algoritmalar karmaşık ve çözümü zaman alıcı zor optimizasyon problemlerinin çözümünde başarılı bir şekilde kullanılmaktadır. Yetkin ve Bilginer (2018) çalışmalarında, robust En Küçük Medyan Kareler (EKMK) yöntemi bir Nivelman Ağı'na global bir doğal optimizasyon algoritması olan YA kullanılarak uygulanmış ve kaba hatalı gözlem vektörü olması durumunda önerilen yaklaşımın klasik En Küçük Kareler (EKK) yönteminden daha iyi sonuç verdiği sayısal bir örnek ile gösterilmiştir (Yetkin ve Bilginer, 2018).

Sharawi ve ark., (2015) Kablosuz algılayıcı ağ (WSN) optimizasyonunun kapsamını, tek amaçlıdan çok amaçlı optimizasyona genişleten bir yöntem önermiştir. Bir WSN`nin kapsamını tanıtmış ve çok amaçlı yarasa sürüsü optimizasyon algoritmasına dayanan hiyerarşik kümeleme ve yönlendirme modelini korumuştur. Burada iki hedef dikkate alınır; önerilen optimizasyon modeli WSN'nin kapsamını ve yaşam boyu zorlukları aşmaktadır. Önerilen model, LEACH yönlendirme ve kümeleme protokolünden daha iyi performans göstermektedir (Sharawi ve ark., 2015) .

Rahman ve Islam (2014) GA'ya dayanan K-Ortalamalar ile entegre bir kümelenme yaklaşımı önererek kümelenme problemindeki doğru küme sayısını belirlemeye çalıştılar. Bu yaklaşımlarla daha fazla performans kümesi merkezi belirlediler. Çalışmalarında kullandıkları 20 veri setine kıyasla genellikle diğer beş yöntemden daha başarılı olduklarını gösterdiler (Rahman ve Islam, 2014).

Basu ve Murthy (2015), doküman kümeleme problemi üzerinde çalışmışlardır. Klasik K-Ortalama algoritmasına yeni bir yaklaşım getirerek, önerdikleri yeni bir uzaklık fonksiyonu kullanmışlardır. Bu yöntemlerinde kümeleme aşamasında, klasik K-Ortalama algoritmasının ihtiyaç duyduğu gibi, küme sayısına önceden ihtiyaç duyulmadığını belirtmişlerdir. Elde ettikleri deneysel sonuçlarda veri setlerindeki dokümanların yüksek başarı ile gruplandırıldığını belirtmişlerdir (Basu ve Murthy, 2015).

Tzortzis ve Likas (2014) klasik K-Ortalama algoritmasının, seçilen başlangıç konumundan dolayı yerel optimuma yakalanma durumunu bertaraf etmek için MinMax K-Ortalama algoritması olarak isimlendirdikleri yöntemi geliştirmişlerdir. Önerdikleri yöntemlerinde kümelere, varyans değerlerine göre ağırlıklar atanmakta ve bu ağırlık değerleri hedefe göre optimize edilmektedir. Elde ettikleri deneysel sonuçlar ile önerilen yöntemin doğruluğunu ve etkinliğini göstermişlerdir (Tzortzis ve Likas, 2014).

3. MATERYAL VE YÖNTEM

Bu bölümde genel olarak kümeleme problemlerinde kullanılan kümeleme yöntemleri, optimizasyon ve tez çalışmasında kullanılan veri setleri ve yarasa algoritması detaylı olarak açıklanmıştır

3.1. Veri Setleri

Tez çalışmasında önerilen YA kümeleme yöntemi 11 adet veri seti üzerinde test edilmiştir. Bu veri setleri UCI veri ambarından alınmıştır, veri setlerinin özelikleri Çizelge 3.1'de verilmiştir.

Çizelge 3.1 Veri setlerinin özelikleri

Veri Setinin Adı	Özellik Sayısı	Örnek Sayısı	Sınıf Sayısı
İris	4	150	3
Wine	13	178	3
Tae	5	151	3
Glass	9	214	7
SPECT_kalp	22	267	2
SPECTF_kalp	44	267	2
Statlog_Heart	13	270	2
Dermatology	33	366	6
Kalp_Hastalıkları	13	303	2
Breast Cancer Wisconsin Diagnostic (BCWD)	32	569	2
Wisconsin Breast Cancer (WBC)	9	683	2

İris veri seti: 3 farklı türden 150 iris çiçeği örneği içerir. Türler Setosa, Versicolour ve Virginica'dır. Her tür için 50 gözlem vardır. Her türdeki nitelikler sepal uzunluk, sepal genişlik, taç uzunluğu ve taç genişliğidir (UCI, 2019).

Wine veri seti: Aynı bölgede yetişen üç farklı çeşitten elde edilen şarabın kimyasal analizini içerir. Şarap çeşitleri, alkol, malik asit, kül, magnezyum, toplam fenoller, flavanoitler, flavanoid olmayan fenoller, proantosiyaninler, renk yoğunluğu, renk tonu, OD280 / OD315 seyreltilmiş şaraplar, prolin olmak üzere 13 özellikten ayırt edilir (UCI, 2019).

Tae veri seti: Wisconsin-Madison Üniversitesi İstatistik Bölümü'ndeki 151 öğretmen yardımcısının öğretim performansının bir değerlendirmesidir. Performans, sınıf büyüklüğü, dönem türü, kurs, kurs eğitmeni ve öğretim asistanının anadili İngilizce olup olmadığına göre ölçülür. Performans, "düşük", "orta" ve "yüksek" olmak üzere üç sınıfa ayrılır (UCU, 2019).

Glass veri seti: Kriminolojik yatırımların incelenmesi ile ilgili bir çalışmada uygulanmıştır. Suç mahallinde bırakılan bardağı delil olarak kullanabilmek için araştırmacı doğru şekilde tanımlamalıdır. Veri seti altı bardak tipinden oluşmaktadır. Bunlar şamandıralı işlenmiş camlar, şamandıralı olmayan işlenmiş camlar, şamandıralı araç camları, şamandıralı olmayan taşıtlar, kaplar, sofra takımı ve farlardır. Bu çalışmada yatırımcı, kırıcılık indeksi, sodyum, magnezyum, alüminyum, silikon, potasyum, kalsiyum, baryum, andiron olan camların bileşimine dayanarak cam türünü ayırt edebilmelidir (UCI, 2019).

SPECT-Kalp veri seti: Veri kümesi, kardiyak Tek Proton Emisyonlu Bilgisayarlı Tomografi (SPECT) görüntülerinin teşhisini açıklar. Hastaların her biri iki kategoriye ayrılır: normal ve anormal. 267 SPECT görüntü setinin (hasta) veri tabanı, orijinal SPECT görüntülerini özetleyen özellikleri çıkarmak için işlenmiştir. Sonuç olarak, her hasta için 44 sürekli özellik deseni oluşturulmuştur. Desen, 22 ikili özellik deseni elde etmek için ayrıca işlenmiştir. CLIP3 algoritması, bu modellerden kümeleme kuralları oluşturmak için kullanılmıştır. CLIP3 algoritması, kardiyologların tanılarına göre % 84.0 oranında kesin kurallar üretmiştir. SPECT, ML algoritmalarını test etmek için iyi bir veri setidir; 22 ikili özelliği tarafından azalan 267 örneğe sahiptir (UCI, 2019).

SPECTF-Kalp veri seti: Veri kümesi, kardiyak Tek Proton Emisyonlu Bilgisayarlı Tomografi (SPECT) görüntülerinin teşhisini açıklamaktadır. Hastaların her biri iki kategoriye ayrılır: normal ve anormal. 267 SPECT görüntü setinin (hasta) veri tabanı, orijinal SPECT görüntülerini özetleyen özellikleri çıkarmak için işlenmiştir. Sonuç olarak, her hasta için 44 sürekli özellik deseni oluşturulmuştur. CLIP3 algoritması, bu modellerden sınıflandırma kuralları oluşturmak için kullanılmıştır. CLIP3 algoritması, kardiyologların tanılarına göre % 77.0 oranında kesin olan kurallar oluşturmaktadır. SPECTF, ML algoritmalarını test etmek için iyi bir veri setidir; 44 özellik tarafından azalan 267 örneğe sahiptir (UCI, 2019).

Statlog (Heart) veri seti: Bu veri setinde örnek sayısı 270 olup, özelik sayısı 13'tür. Burada satırlar gerçek değerleri ve öngörülen sütunları temsil eder (UCI, 2019).

Dermatoloji veri seti: En büyük sınıfa sahip veri setidir. Veri setinde 34 nitelik ve 6 küme vardır (dermatolojik hastalıklar: Psoriasis, Seboreik Dermatit, Liken Planus, Pityriasis Rosea Kronik Dermatit ve Pityriasis Rubra Pilaris). Veri seti 366 örnek sayısına sahiptir (UCI, 2019).

Kalp Hastalıkları veri seti: Normalde 75 niteliğe sahiptir ancak 13 tanesi bu çalışmada kullanılmıştır. Küme özelliği, hastadaki kalp hastalığının varlığını ifade eder

ve 0'dan (hiç yok) 4'e kadar bir tamsayıdır. Bu çalışmada, hastanın hastalığı olan veya olmayan şeklinde 2 sınıf olarak kullanılmıştır. Veri setinde 303 örnek var (UCI, 2019).

Breast Cancer Wisconsin Diagnostic (BCWD) veri seti: Veri setinde yer alan özellikler bir göğüs kütlesinin ince iğne aspiratının sayısallaştırılmış görüntüsünden hesaplanır. Görüntüde mevcut olan hücre çekirdeğinin özellikleri tarif edilir. Bu veri seti 32 özelliğe, 2 küme ve 569 örneğe sahiptir (UCI, 2019).

Wisconsin Breast Cancer (WBC) *veri seti:* Meme kanseri vakalarında yapılan ölçümlerden oluşur. Bu veri seti, Dr. Wolberg'den Madison Üniversitesi'nden Wisconsin Üniversitesi Hastanelerinden alınmıştır. İyi ve kötü huylu olmak üzere iki tür meme kanseri vakası vardır. Bu veri setinde, kanser türü yığın kalınlığı, hücre büyüklüğü ve şekli düzgünlüğü, marjinal adezyon, tek epitel hücre büyüklüğü, çıplak çekirdekler, blandchromatin, normal nükleoller ve mitozlara göre tanımlanmıştır. Bu veri seti 9 özelliğe, 2 sınıfa ve 683 örneğe sahiptir (UCI, 2019).

Normalizasyon, istatistiksel veri işleme alanlarında, veriler arasındaki farklılığın büyük ölçülerde olduğu durumlarda, verileri birbirine yaklaştırmak ve tek bir düzen içerisine almak amacıyla uygulanan bir yöntemdir. Bu şekilde veriler birbirine yaklaştırılarak aykırı olan değerlerin, çalışmanın sonucunu etkilemesinin önüne geçilmesi amaçlanır. Sınıflandırma ve kümeleme işlemlerinde veriler genellikle belli bir aralıkta normalize edilir. Verilerin normalize edilmesi için kullanılan farklı yöntemler vardır. Bu çalışmada kullanılan verilerin normalize edilmesi aşamasında Min-Max Normalizasyonu yöntemi kullanılmıştır. Bu normalizasyon yönteminde veriler belirlenen aralıklar arasında yeni değerlerini almaktadır. Denklem (3.1) bir verinin normalize edilmiş değerinin nasıl hesaplandığını göstermektedir (Karakoyun, 2015).

$$nX = \frac{X - minX}{maxX - minX}(nmax - nmin) + nmin \tag{3.1}$$

Burada X normalize edilecek değer, nX normalizasyon işleminden sonra X'in alacağı değer, min$X$ normalize edilecek değerlerin en küçüğü (minimum), max$X$ normalize edilecek değerlerin en büyüğü (maksimum), *nMax* yeni maksimum değer yani verilerin normalize edileceği üst sınır, *nMin* yeni minimum değerdir. Bu çalışmada veriler [0, 1] aralığında normalize edilmiştir. Yani *nMin=0* ve *nMax=1* olarak belirlenmiştir. Bu durumda Denklem (3.1) güncellenerek Denklem (3.2) şeklinde özetlenerek kullanılmıştır (Karakoyun, 2015).

$$nX = \frac{X - minX}{maxX - minX} \quad (3.2)$$

3.2. Kümeleme ve Kümeleme Yöntemleri

Kümeleme; veri nesnelerini, kümeler adı verilen, bir grup sınıftan oluşan gruplara ayırma işlemidir. Böylece "aynı kümedeki nesneler birbirleriyle yüksek benzerlik gösterirken, ayrı kümelerdeki nesneler daha farklıdır. Kümeleme denetimsiz öğrenme örneğidir. "Sınıflandırma", veri nesnelerini önceden tanımlanmış bir dizi sınıfa atayan bir prosedürü ifade eder (Karakoyun, 2015).

Kümeleme yöntemleri nesneleri gruplara ayırır, böylece bir gruptaki nesneler birbirlerine benzer ve diğer gruplardaki nesnelerden mümkün olduğunca farklı olur. Kümeleme denetimsiz bir süreçtir ve kümeleme yöntemleri sadece veri noktalarının benzerliklerini kullanarak verilerin içindeki ilişkileri bulur.

Verilerin özelliklerini tanımlamak için bir uzman tarafından etiketlenebilirler veya benzer veri nesnelerini tanımlamak için kullanılırlar. Bu nesneler genellikle özellikler veya özellikler olarak da bilinen çok boyutlu değişkenlerle temsil edilir. İki veri nesnesi arasındaki benzerlik veya farklılık, tipik olarak nesneleri temsil eden çok boyutlu özellik vektörleri arasındaki mesafe olarak ölçülür (Karakoyun, 2015).

Kümeleme, nesnelerin aynı grupta ise birbiri ile benzer, farklı gruplarda ise birbirlerinden farklı olması durumunu esas almaktadır. Nesnelerin birbiri olan ilişkisini daha iyi anlatabilmek için benzerlik ve benzersizlik (farklılık) kavramları kullanılmaktadır. İki nesnenin benzerliği ne kadar fazla olursa, o iki nesne arasındaki ilişki o kadar kuvvetlidir demektir. Bu durumda ilişkinin zayıf olduğu durumlarda nesneler arasındaki farklılık yüksektir denilebilir. Nesneler arasındaki benzerlik ve farklılık değerleri farklı metotlarla ölçülebilir. Benzerlik ve farklılık değerleri kullanılarak nesnelerin ayırt ediciliği daha net sınırlarla belirlenip gruplara daha rahat ayrıştırılabilir. Nesneler arasındaki benzerlik değerinin ölçülmesinde kullanılan metodun seçilmesinde, nesnelere ait özelliklerin veri tipleri etkili olur. Bir nesneye ait özelliğin aldığı değerlerin sayısı sonlu ise bu veri tipi kesikli (discrete) veri kategorisine dâhil olur. Ancak nesnenin özelliği birden çok aralıkta, o aralıklardaki her değeri alabiliyorsa sürekli (continuous) veri kategorisine girer (Karakoyun, 2015).

Kümeleme işlemlerinde kullanılan bazı uzaklık ölçüleri, Kümeleme işlemleri, veri setlerini homojen bir şekilde sınıflara ya da kümelere bölme işlemini yaparken benzer gözlemleri ya da nesneleri aynı sınıfa (kümeye), farklı olanları ise farklı sınıfa

(kümeye) yerleştirmeye çalışır. Her küme iki önemli özelliği barındırmalıdır: kümeler arası düşük benzerlik derecesi ve küme içi yüksek benzerlik derecesi. Veri setine ve uygulamaya bağlı olarak ilişkileri belirlemek için iki tip ölçü kullanılır: uzaklık ölçüleri ve benzerlik ölçüleri. Öklid uzaklığı, Manhattan Uzaklığı, Mahalanobis uzaklığı ve Minkowski Uzaklığı (Erilli, 2009).

Öklid Uzaklığı, en sık kullanılan uzaklık ölçüsüdür. Çok boyutlu uzayda geometrik uzaklıktır. Denklem 3.3. ile hesaplanır.

$$d_{ij} = \sqrt{\sum_{k=1}^{K} (X_{ik} - X_{jk})^2} \tag{3.3}$$

Burada *d(i, j)*: *i.* Nesnenin *j.* Küme merkezine uzaklığı

X_{ik}:*i.* nesnenin *k.* niteliğinin değeri

X_{jk}:*j.* merkezin *k.* niteliğinin değeri

i = 1, 2, …, *N*

j = 1, 2, …, *C*

k=1,2,….. ,K

N: Veri setindeki toplam nesne sayısı

C: Küme merkezi sayısı.

K: Nitelik sayısı

Öklid uzaklıkları standartlaştırılmış değil, ham veriden hesaplanır. Bu metot kesin avantajlar sağlar. Herhangi iki nesne arasındaki uzaklık, gözlem olabilecek ve analize katılacak yeni nesnelerden etkilenmez (Erilli, 2009).

Minkowski metodu kullanılarak iki nesne arasındaki uzaklık Denklem 3.4'e göre hesaplanır (Erilli, 2009):

$$d(i,j) = \left[\left| X_{i1} - X_{j1} \right|^m + \left| X_{i2} - X_{j2} \right|^m \dots \dots \left| X_{in} - X_{jn} \right|^m \right]^{1/m} \tag{3.4}$$

Biçiminde hesaplanır burada

d(i, j): *i.* Nesnenin *j.* Küme merkezine uzaklığı

X_i: Veri setindeki *i.* nesne

X_j: Veri setindeki *j.* küme merkezi

Minkowski yönteminde *m* değeri 1 seçilirse formül, Manhattan uzaklık metodu formülüne, *m* değeri 2 seçilirse Öklid uzaklık metodu formülüne dönüşür. Bu açıdan değerlendirildiğinde Minkowski metodunun, Manhattan ve Öklid metotlarının genel formu olduğu söylenebilir.

Mahalanobis uzaklığı, iki birim değeri veya noktası arasındaki uzaklığı ölçmede iki değişken arasındaki kovaryans veya korelasyon katsayısını da dikkate alan bir uzaklık ölçüsüdür. Denklem 3.5 ile hesaplanır.

$$Md_{ik} = (X_i - X_k)S^{-1}(X_i - X_k) \tag{3.5}$$

Burada S kovaryans matrisini göstermektedir (Erilli, 2009).

Kümeleme analizinin amacı, verilerin belirli bir model üzerinden geçirilerek farklı gruplara ayrılmasıdır. Bu gruplar küme olarak adlandırılmıştır. Kümeleme işlemleri verilerin içerdiği nesnelerin özelliklerine göre yapılır. Benzer özellikli nesneler aynı kümede yer alır. Farklı özellikli nesneler ise farklı kümelerde yer alır. Yukarıdaki bahsi geçen değişken türleri kümeleme analizinden geçirilerek kümelere dağılımı yapılır. Bu veri değişkenlerinin kümelenmesi için kümeleme algoritmaları kullanılır. Kümeleme algoritmaları belirlenen bir model üzerinde çalışan ve verilerin içerdiği nesne özelliklerine göre kümelere ayırır. Kümeleme algoritmalarının birbirinden farklı çalışma yöntemleri vardır; bazı yoğun bölgelere göre, bazı merkez noktalara göre ve bazıları ise veriyi hiyerarşik şekilde gruplarlar (Erilli, 2009).

3.2.1. K-Ortalama Kümeleme Algoritması

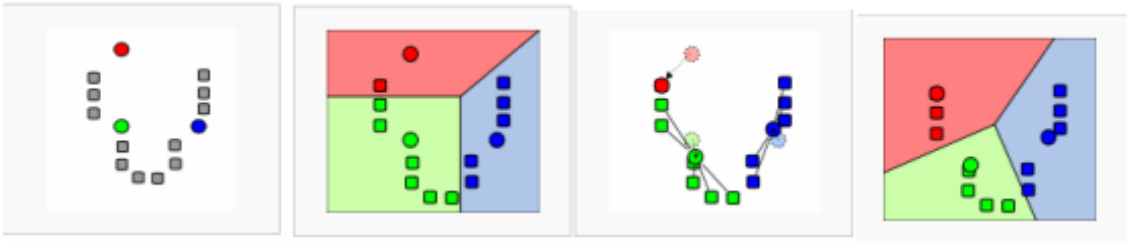
En eski kümeleme yöntemlerinden biri olan K- Ortalamalar 1957 yılında ilk kez Hugo Steinhaus'un öne sürdüğü bir fikir olmasına rağmen 1967 yılında J.B. Mac Queen tarafından geliştirilmiştir. K-ortalama, bir merkez tabanlı kümeleme algoritmasıdır. "K" küme sayısını gösterir ve aynı zamanda bir giriş parametresidir. Veri kümesindeki her eleman, en küçük mesafeye sahip bir küme merkezine atanır. Zweije (2017)' ye göre bu algoritma, çok doğru olabilen veya bulunamayan merkezleri bulur ve her bir veri öğesini mesafeye bağlı olarak kümeye atar (Zweije, 2017).

K-ortalama, popüler bir algoritmadır çünkü uygulanması çok kolaydır. Uygun bir mesafe tanımı ve makul bir K değeri ayarı olması koşuluyla, genellikle makul sonuçlar elde eder. Şekil 3.1, K = 3 için standart K-ortalama kümeleme algoritması sürecini göstermektedir (Zweije, 2017)

Bir dizi n öğesi (x1, x2, ..., xn) için her öğe bir d-boyutlu vektördür (d bir tamsayı). k adet farklı kümelere veya gruplara ayırmak için kullanılacak algoritmanın detayları aşağıdaki gibidir:

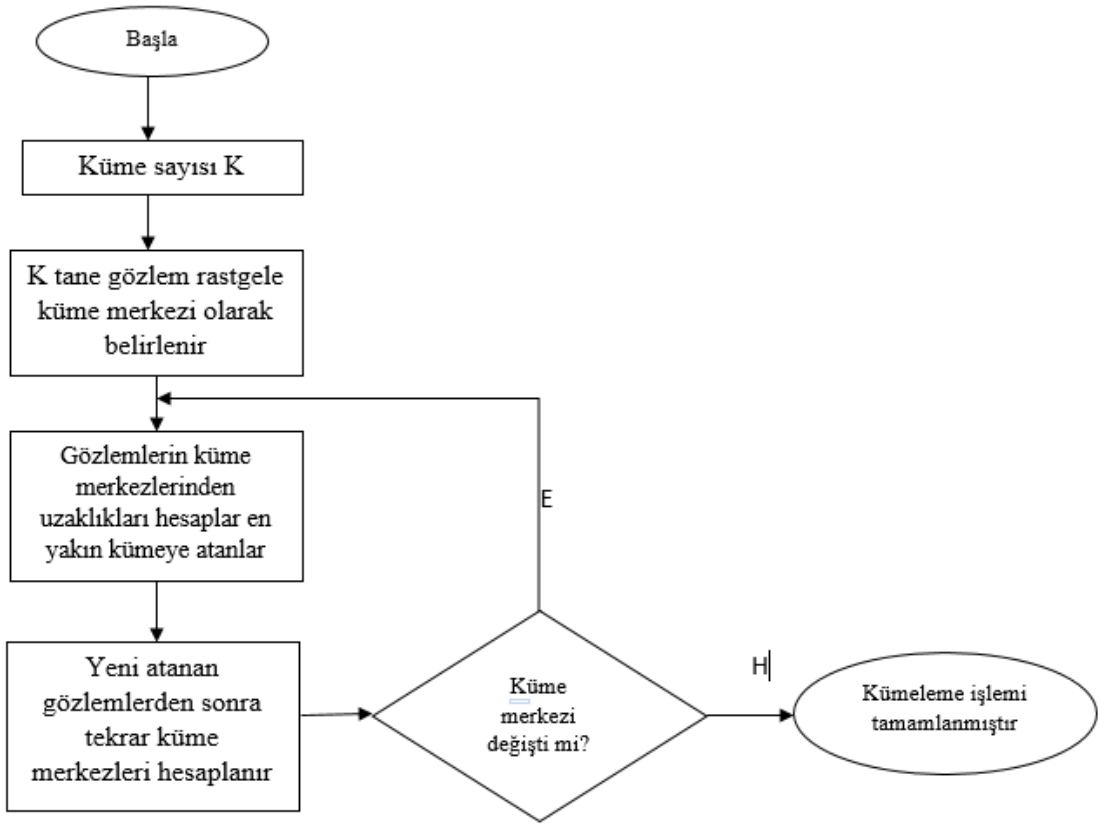
1. Küme sayısı (*k*) okunur. Bu değer algoritmaya dışarıdan verilir.
2. *k* adet rastgele veya belirli bir yöntemle küme merkezleri belirlenir.
3. Tüm elemanların merkezlere olan uzaklıkları hesaplanır.
4. Elemanlar yakın oldukları merkezlere göre kümelenir.
5. Dördüncü adımda oluşan kümelerin ortalamaları hesaplanarak yeni küme merkezleri belirlenir.
6. Son bulunan küme merkezleri bir önceki küme merkezlerine eşit oluncaya ya da belirlenen döngü sayısına ulaşılana kadar işlem tekrarlanır.

Algoritmanın akış diyagramı Şekil 3.2'de görülmektedir



1.k başlangıç aracı (bu durumda k = 3) veri alanında rastgele oluşturulur (renkli olarak gösterilir)	k, küme, her gözlemi en yakın ortalama ile İlişkilendirerek oluşturulur.	K kümelerinin her birinin merkezi yeni ortalama olur.	2. ve 3. adımlar yakınsamaya ulaşılana kadar tekrarlanır.

Şekil 3.1 K ortalama genel çalışma mantığı (Zweije, 2017).



Şekil 3.2 K- Ortalamalar algoritmasının adımları (Ray ve Turi, 1999)

K-Ortalamalar algoritmasının nasıl hesaplandığını göstermek için aşağıda bir örnek verilmiştir. Bu örnek için Çizelge 3.1 de belirtilen veri kümesi kullanılmıştır.

1. Adım

Çizelge 3.2 de verilmiş örnek veri kümesini 2 kümeye ayırmak için başlangıç küme merkezleri (*v1* ve *v2*) olarak 2 ve 5. gözlemler rasgele seçilsin. Bu durumda *v1*=4 ve *v2*=10 olur (Çizelge 3.3)

Çizelge 3.2. K-ortalamalar algoritması için örnek veri kümesi

Gözlem Numarası	Öznitelik
1	**2**
2	**4**
3	**5**
4	**8**
5	**10**
6	**12**

Çizelge 3.3. K-ortalamalar algoritması başlangıç küme merkezleri

Küme	Gözlem Numarası	Küme Merkezi
1. Küme	**2**	**4**
2. Küme	**5**	**10**

Bütün gözlemler ve küme merkezleri arasındaki Öklid uzaklığı hesaplanır. Sonuçlar aşağıdaki tabloda gösterilmiştir (Çizelge 3.1).

Çizelge 3.4.K-ortalamalar algoritması Örneği 2. adım

Gözlem Numarası	1. Merkeze uzaklık	2.Merkeze Uzaklık	Küme No
1	8	64	1
2	0	36	1
3	1	25	1
4	16	4	2
5	36	0	2
6	64	4	2

1. Kümeye ait gözlemlerin aritmetik ortalaması tekrar hesaplanıp, yeni küme merkezi bulunmuştur

$$v1 = \frac{2+4+5}{3} = \frac{11}{3} = 3.6$$

$$v2 = \frac{8+10+12}{3} = \frac{30}{3} = 10$$

Yeni küme merkezleri, ***v1***= 3.6 ve ***v2*****=** 10 olarak hesaplanmıştır.

2. Adım

Hesaplanan yeni küme merkezleri ile gözlemlerin küme merkezine olan uzaklıkları aşağıdaki tabloda gösterilmiştir (Çizelge 3.5).

Çizelge 3.5 K-ortalamalar algoritması Örneği 3. Adım

Gözlem Numarası	1. Merkeze Uzaklık	2.Merkeze Uzaklık	Küme No
1	2.56	64	1
2	0.16	36	1
3	1.96	25	1
4	19.36	4	2
5	40.96	0	2
6	70.56	4	2

1. Adım ve 2.adımdaki küme durumları değişmediği için, algoritma sonlandırılır. Bu durumda 1., 2. ve 3. gözlemler 1. kümeye ve 4., 5., ve 6. gözlemler 2. kümeye ait olur.

3.2.2. Bulanık C-Ortalamalar Yöntemi

Kümeleme yöntemleri klasik ve bulanık kümeleme olarak iki şekilde incelenebilir. Klasik kümelemede küme elemanları bir tek kümeye ait olabilirken, bulanık kümelemede ise veri elemanları farklı üyelik düzeyleriyle birden çok kümeye ait olabilmektedir. Bulanık kümeleme yöntemleri, sistemi daha iyi taklit ederek klasik kümeleme yöntemlerinden daha güçlü sonuçlar verebilmektedir.

Son dönemlerde bulanık kümeleme yaklaşımları geliştirilmiş, bulanık C-ortalamalar yöntemi temel alınarak çeşitli durumlar için formüllerde modifikasyonlar uygulanmıştır. Bu algoritma uygulama kolaylığı da göz önünde bulundurulduğunda en sık kullanılan yöntemlerden biridir (Şen ve ark., 2010)

Sık kullanılan Bulanık C-Ortalamalar Algoritması amaç fonksiyonuna dayalı olan tüm kümeleme tekniklerinin temelini oluşturmaktadır. Bu algoritma ilk olarak Bezdek tarafından 1974 yılında geliştirilmiştir. Daha sonra bir ağırlıklara dayalı bir fonksiyon tanımlanarak bu bulanık amaç fonksiyonu genelleştirmiştir. Algoritmanın son sürümü m boyutlu uzayda noktaların küresel şeklini tanır (Bezdek, 1981). Nesneler ve küme merkezleri arasındaki uzaklık, Öklid uzaklığı ile ölçümlenir (Bezdek, 1981).
Bulanık kümeleme algoritmaları, Denklem 3.6 ile formüle edilen C-Ortalamalar amaç fonksiyonunun küçültülmesine dayanmaktadır (Bezdek, 1981) (Bezdek, 1974).

$$JM = \sum_{i=1}^{n}\sum_{j=1}^{v} U_{ij}^{m} \left\| X_i - v_j \right\|^2, 1 \leq m \leq \infty \qquad (3.6)$$

Burada;

X: Herhangi bir etikete tabi tutulmamış veri seti {x1, x2,⋯, xn}

N: veri sayısı,

c: küme sayısı,

$U= [uik]$, X matrisinin bulanık üyelik dereceleri matrisi,

$V = [v1, v2, \ldots, vc]$, $vi \in Rn$ belirlenmesi gereken küme merkezi (prototip) vektörü,

$\|X_k - V_i\|^2$, xk değeri ile vi merkezi arasındaki Öklid uzaklığı normu,

$m \in [1,\infty]$, oluşan kümelerin bulanıklığını belirleyen bir parametredir

Bu amaç fonksiyonunun tabi olduğu bir kısıt vardır. Bulanık mantık prensibi gereği her veri, kümelerin her birine [**0**, **1**] arasında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı "1" olmalıdır (Bezdek, 1974)

$$\sum_{i=1}^{c} U_{ik} = 1, \qquad 1 \leq K \leq N \tag{3.7}$$

Denklem 3.7, ağırlıklandırılmış kare hatalarının toplamının bir ölçüsüdür (Bezdek, 1974). C- Ortalamalar fonksiyonunun minimizasyonu, çeşitli yöntemlerle çözülebilen doğrusal olmayan bir optimizasyon problemini temsil eder. En bilinen yöntem durağan noktalar için birinci dereceden koşullar vasıtasıyla basit bir Picard iterasyonudur. Amaç fonksiyonu Lagrange çarpanları aracılığıyla yeniden yazılırsa Denklem 3.8'daki amaç fonksiyonunda u, r, s ve vi parametrelerine göre kısmi türev alınıp sıfıra eşitlenirse şu eşitlikler elde edilir;

$$U_{rs} = \frac{\lambda_r^{1/m-1}}{[m\|X_r - V_s\|^2]^{1/m-1}} \tag{3.8}$$

Aynı denklemde λk' Lara göre türev alınıp sıfıra eşitlenirse;

$$\sum_{i=1}^{c} \frac{\lambda_r^{1/m-1}}{\left[m\|X_r - V_s\|^2\right]^{1/m-1}} \tag{3.9}$$

$$\lambda_r^{1/m-1} = \frac{1}{\sum_{i=1}^{c}[m\|X_r - V_s\|^2]^{1/m-1}} \tag{3.10}$$

Elde edilir. (3.13) denklemi (3.8) denkleminde yerine yazılırsa üyelik değerlerini hesaplama da kullanılan formül aşağıdaki gibi elde edilir;

$$U_{rs} = \frac{1}{\sum_{i=1}^{c}\left[\frac{\|X_r - V_s\|}{\|X_r - V_k\|}\right]^{2/m-1}} \tag{3.11}$$

Aynı şekilde küme merkezleri için hesaplanacak olan ifadeyi bulmak için (3.9) denkleminde vi'lere göre kısmi türev alınıp sıfıra eşitlenirse;

$$\frac{\partial \mathcal{L}(U_{ik}, V_i, \lambda_k)}{\partial V_i} = 0 \Longrightarrow \sum_{i=1}^{c} U_{ik}^m \frac{\partial \|X_k - V_i\|^2}{\partial V_i} = 0 \tag{3.12}$$

$$Vi = \frac{\sum_{k=1}^{n}(U_{ik})^m X_k}{\sum_{k=1}^{n}(U_{ik})^m}, 1 \leq i \leq c \tag{3.13}$$

şeklinde bulunur. Bu denklemler yardımı ile Bulanık C-Ortalamalar Algoritması adımları oluşturulur;

Adım1: Başlangıç değerler belirlenir. *c* küme sayısı, *m* bulanıklık katsayısı, *i* iterasyon sayısı, ε işlem sonlanma kriteri, uij 'lerden oluşan *U* üyelik matrisine rasgele değerler atanır.

Adım2: (3.8) eşitliği kullanılarak küme merkezleri hesaplanır.

Adım3: (3.11) denkleminden faydalanılarak başlangıç üyelik değerleri yenilenir.

Adım4: Yeni üyelik değerleri ile eski üyelik değerleri karşılaştırılır:

$$\|U_{yeni} - U_{eski}\| \leq \varepsilon \tag{3.14}$$

Denklem 3.14 ile verilen şart sağlanırsa iterasyon durdurulur. Değilse Adım 2'ye geri dönülür.

3.3. Yarasa Algoritması (YA)

Optimizasyon, bir problemde belirli şartlar altında en iyi çözümü elde etme işlemidir. Optimizasyon, bir fonksiyonun maksimumunu veya minimumunu veren koşulları bulma işlemi olarak tanımlanabilir. En basit anlamda, optimizasyon bir minimizasyon veya maksimizasyon probleminin çözümüdür. Araştırmacılar tarafından geliştirilen PSO, ABC ve GA gibi bir çok optimizasyon algoritması bulunmaktadır ve bu optimizasyon algoritmaları farklı problem çözümlerinde başarılı bir şekilde kullanılmıştır (Karaboga ve Ozturk, 2011). Bu çalışmada YA kümeleme işlemi için kullanılmıştır. YA'nın çalışma prensibi detaylı olarak aşağıda açıklanmıştır (Yang, 2010).

Xin-She Yang tarafından 2010 yılında temel YA yarasa türlerinin avlarını bulmak için yankı bulma davranışına dayanarak geliştirilmiştir. Yarasa, ses darbesinin üretimini ve yankı tanıma zaman farkını, ses darbesinin değişken yoğunluğunu ve yarasanın kulakları arasındaki zaman gecikmesini birleştirerek üç boyutlu bir çevre oluşturur. Bu şekilde, yarasa avın tipini, hareket hızını, mesafesini ve yönünü tanımlayabilir (Yang, 2010).

Basitleştirmek için, algoritma aşağıdaki temel kurallara dayanır:

1. Tüm yarasalar mesafeyi hissetmek için, ayrıca yiyecek/av ve arka plan engellerini algılayabilmek için ekolokasyon kullanır.
2. Yarasalar yiyeceklerini veya avlarını bulmak için V_i hızında, X_i konumunda, sabit frekansta f_{min} dalga boyu λ ve ses şiddeti A_0 ile rasgele uçarlar. Yaydıkları sinyallerin frekansını otomatik olarak ayarlayabilir ve hedeflerinin yakınlığına bağlı olarak sinyal yayma oranını ∈ [0, 1] ayarlayabilirler.

3. Ses şiddeti değişiklik göstermektedir, ses şiddetinin büyük değer A_0'dan minimum değer A_{min}'e değişir. Orijinal YA adımları Şekil 3.3'de verilmiştir

YA'nın temel adımları **Şekil 3.3**'deki gibi özetlenebilir (Yang, 2013):

Her bir yarasaya ait başlangıç frekansı f_i belirle
Ses şiddeti A_i ve sinyal yayılım oranı r_i değerlerini belirle
While $(t <$ *maksimum iterasyon sayısı*$)$
Frekansı ayarlayarak konumları, hızları güncelle
ve yeni uygunluk değeri hesapla
if $(rand\ (0,1) > r_i)$
En iyi çözümler arasında bir çözüm seç
Seçilen en iyi çözümü etrafında yerel bir çözüm üret
end if
Rasgele uçarak yeni bir çözüm üret
if $\left(rand < A_i\ \&f(x_i) < f(x_*)\right)$
Yeni çözümleri kabul et
r_i arttır ve A_i azalt
end if
Yarasaları sırala ve en iyi olanını bul
end while
En iyi olanın uygunluk değerini göster

Şekil 3.3. YA'nın Adımları

3.3.1. Yarasaların Hareketi

Her yarasa, arama alanı içinde bir frekans (f_i) ve popülasyondaki tüm yarasalar arasında mevcut en iyi bireyin sahip olduğu çözüm değerine (x_*) bağlı olarak üretilen hız değeri (v_i^t) ile sonraki adımda konumlarını belirlemek ister (Yang, 2013).

YA konum (x_i^t) ve hız (v_i^t) için güncellenme işlemleri Denklem 3.15, Denklem 3.16 ve Denklem 3.17 kullanılarak gerçekleştirilir

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{3.15}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \tag{3.16}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3.17}$$

Burada β 0 ile 1 aralığında rasgele bir değerdir. x_* n adet yarasa arasında şu anki en iyi konumu göstermektedir. Yani oluşan yarasalar için tüm yarasaların sonuçlarının en iyi sonucunu göstermektedir. Algoritmada başka en iyiler çıktıkça bu değer güncellenmektedir. f_i ise her problem için değişebilir çünkü f_i'nin değerini doğrudan

etkileyen $(f_{min} - f_{max})$ değerleri vardır ve bu değerler problemdeki boyutların arama uzayını temsil etmektedir. Her yarasa popülasyon içerisinden uygunluk değerinin kalitesine bağlı olarak bir başka çözüm seçmekte ve seçtiği çözümün etrafında yeni kaynağa yönelmektedir. Konum güncelleme için Denklem 3.18 kullanılmaktadır.

$$X_{yeni} = X_{eski} + \epsilon A^t \tag{3.18}$$

Burada ε değeri −1 ile 1 aralığında rastgele bir sayıdır, A^t ise, t anında tüm yarasaların ortalama ses şiddetini ifade etmektedir.

3.3.2. Ses Şiddeti ve Sinyal Yayma Oranı

Ayrıca, A_i ses şiddeti yüksekliği ve darbe emisyonunun sinyal yayma oranı (r_i), yinelemeler ilerledikçe buna göre güncellenmelidir. Ses yüksekliği genellikle bir yarasa avını bulduğunda azalır. Bu sırada Darbe emisyon oranı arttıkça ses yüksekliği herhangi bir değerde belirlenebilir. Ses şiddeti A_{min} ve A_{max} arasında herhangi bir değer olarak seçilebilir, $A_{min} = 0$ varsayılarak bir yarasa avını bulduğu ve geçici olarak herhangi bir sesi yaymayı bıraktığı anlamına gelmektedir. Bu varsayımlarla ses şiddeti güncelleme işlemi Denklem 3.19 ve Denklem 3.20'deki gibi gerçekleşmektedir

$$A_i^{t+1} = \alpha A_i^t \, , r_i^{t+1} = r_i^0 [1 - exp(-\gamma t)] \tag{3.19}$$

Burada α ve γ [0,1] aralığında değerlerdir. α parametresi 0 ile 1 aralığında olduğunda ve $\gamma > 0$ olduğunda Denklem 3.20'deki durum elde edilir.

$A_i^t = 0$, $r_i^t = r_i^0$, gibi $t \to \infty$ (3.20)

3.3.3. Yarasa Algoritması Tabanlı Kümeleme

YA elde ettiği başarıdan dolayı birçok problem çözümünde kullanılmaktadır. Yarasa algoritması global çözümleri daha geniş bir arama uzayından elde edebilir. Bu tez çalışmasında kümeleme probleminde en uygun küme merkezlerinin bulunması için kullanılmıştır. En uygun küme merkezlerinin bulunması için küme merkezlerinin iyi bir şekilde güncellenmesi gerekmektedir. Bu tez çalışmasında veri setlerinin küme merkezleri, YA'nın Denklem 3.15 - Denklem 3.20 arasındaki denklemlerini kullanarak güncellenmektedir. Güncelleme işleminde YA, toplam karesel hatayı (SSE) uygunluk

fonksiyonu olarak kullanmaktadır. En az SSE değerine sahip olan yarasa en iyi çözümü ifade etmektedir, en iyi SSE değerine göre diğer yarasaların konumları dolayısıyla küme merkezleri güncellenmektedir. Güncelleme işlemi sonucunda YA, veri seti için optimum küme merkezlerini bulmayı hedeflemektedir. Önerilen yarasa algoritması tabanlı kümeleme yönteminin adımları aşağıda sıralı olarak verilmiştir.

Adım 1: Veri setini oku

Adım 2: Yarasa algoritmasının parametrelerini ayarla (r, A, yarasa sayısı, iterasyon sayısı, küme sayısı).

Adım 3: Küme sayısı kadar rastgele başlangıç küme merkezleri üret ve bu küme merkezlerine göre veri setini kümele ve SSE'yi hesapla.

Adım 4: En iyi SSE değerine sahip olan yarasanın küme merkezlerine göre diğer yarasaların küme merkezlerini güncelle. Güncellenme işlemi Denklem 3.15, Denklem 3.16 ve Denklem 3.17 kullanılarak gerçekleştirilir.

Adım 5: Elde edilen güncellenmiş küme merkezlerine göre veri setini kümele ve SSE'yi hesapla.

Adım 6: İterasyon sayısı kadar 4. ve 5. Adımlar tekrar et.

Yarasa kümeleme algoritmasında kümeleme sonuçlarının değerlendirilmesinde kümelenme yapısı arasındaki benzerliği hesaplamak için Rand İndex (RI) ve SSE uygunluk fonksiyonu değerlendirme ölçütü olarak kullanılmıştır.

Kümeleme başarısı (RI), kümeleme yöntemlerinin performansını değerlendirmek için sıkça kullanılan formüldür. Kümeleme başarısı formülü elde edilen çözüm ile bilinen çözümün benzerliğini hesaplamaktadır ve elde edilen çözümün ne kadar doğru olduğunu ölçmektedir. N adet nesneden oluşan ve k tane kümeye bölünecek olan bir veri setinde, gerçek küme bilgisi bilinen nesneler E (Expected) ve kümeleme işlemine tabi tutulduktan sonra tahmin edilen nesneler O (Obtained) ile temsil edilsin. Veri setindeki her nesne, kendisi hariç geriye kalan tüm nesneler için bu kıyaslamaya tabi tutulur. $i, j = 1,2,...,N$ olmak üzere veri setindeki her xi ve xj nesne çiftinin, E ve O durumlarında birbirine göre 4 farklı durumu vardır.

1. xi ve xj nesne çiftinin O durumunda aynı kümeye dâhil olması ve E durumunda da bu nesne çiftinin aynı kümede olması.
2. xi ve xj nesne çiftinin O durumunda aynı kümeye dâhil olması ve E durumunda da bu nesne çiftinin farklı kümelerde olması.

3. *xi* ve *xj* nesne çiftinin *O* durumunda farklı kümelere dâhil olması ve *E* durumunda da bu nesne çiftinin aynı kümede olması.
4. *xi* ve *xj* nesne çiftinin *O* durumunda farklı kümelere dâhil olması ve *E* durumunda da bu nesne çiftinin farklı kümelerde olması.

Veri setindeki *xi* ve *xj* nesne çiftlerinin yukarıdaki 4 durum için elde edilen eşleşme sayıları sırasıyla *a, b, c* ve *d* ile gösterilsin. Yukarıdaki 4 durumun elde edilen toplam sayıları *M* ile gösterilsin. Bu durumda $M = a + b + c + d$ ile elde edilir. *RI,* kümeleme sonucu elde edilen Rand Index değeri Denklem 3.21'de verilmiştir.

$$RI = \frac{(a+b)}{M} \tag{3.21}$$

Denklem 3.21 Rand Index değeri [0, 1] aralığında değerler alır. *RI* değeri bire yaklaştıkça *E* ve *O* durumlarının birbirine benzerliği artmaktadır. Bu durum kümelemenin iyi yapıldığını göstermektedir. (Zweije, 2017).

SSE denetimsiz bir ölçü olarak kullanılır, bu nedenle mükemmel çözüm hakkında bilgi sahibi olmadan bir çözümün kalitesini değerlendirir. Aşağıdaki gibi hesaplanır:

$$SSE = \sum_{i=0}^{k} \sum_{x \in c_i} dist^2(mi, x) \tag{3.22}$$

Burada dist, uzayındaki iki nesne arasındaki standart mesafesidir, *k* adet kümeye ayrılacak bir veri seti olduğu kabul edildiğinde; *mi*, *Ci* kümesinin merkez noktası ve *x*, *Ci* kümesine ait bir nesne olmak üzere tüm kümelerin karesel hata toplamı değeri ile hesaplanır (Zweije, 2017).

4. UYGULAMA SONUÇLARI

Bu tez çalışmasında Yarasa Algoritmasının kümeleme problemlerinde performansını değerlendirmek amacıyla UCI KDD veri tabanından alınan 11 adet farklı veri seti (İris, Wine, Tae, Glass, SPECT-kalp, SPECTF-kalp, Statlog_Heart, Dermatology, Kalp_Hastalıkları, WBC ve BCWD) kullanılarak denemeler yapılmıştır. Bunun için YA farklı yarasa sayılarında her veri seti için 30 defa çalıştırılmıştır. 30 çalıştırma sonucunda RI, SSE ve doğru kümelenmiş örnek sayısı kriterlerine göre sonuçlar değerlendirilmiştir. 30 çalıştırma sonucunda elde edilen değerlerin ortalaması, en küçük ve en yüksek değerler karşılaştırma kriteri olarak kullanılmıştır. Aynı zamanda literatürde aynı veri setleri kullanılarak yapılmış farklı algoritmalarla da sonuçlar karşılaştırılmıştır.

Önerilen YA tabanlı kümeleme algoritmasının kodlaması Matlab programında yapılmıştır. Yarasa algoritmasının giriş parametrelerinin değerleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1. Önerilen yarasa algoritmasının giriş parametreleri

Parametre	Açıklama	Parametre Değeri
Ai	Ses şiddeti	0.5
ri	Sinyal yayılım oranı	0.5
N_iter	Algoritma iterasyon sayısı	500
Fmax	Frekans maksimum	0
Fmin	Frekans minimum	2
N	Yarasa sayısı	10, 20, 30, 40

Bu çalışmada öncelikle kümeleme problemlerini çözmek için önerilen YA'nın performansını ölçmek amacıyla yaygın olarak kullanılan İris, Wine, Tae, Glass ve Wbc ortak veri setleri tercih edilmiştir. Elde edilen sonuçlar ise araştırmacılar tarafından önerilen optimizasyon tabanlı kümeleme yöntemleri (K-Ortalaması, FCM, IFCM, KFCM, KIFCM, PSO-IFCM, GA-IFCM, ABC- IFCM, POS- KFCM, GA- KFCM ve ABC- KFCM) ile karşılaştırılmıştır (Kuo ve ark., 2018). Her veri seti için elde edilen RI sonuçları en iyi, en kötü, ortalama ve standart sapma kriterlerine göre değerlendirilmiş ve 100 ile çarpılmıştır (%RI). Yarasa tabanlı kümeleme algoritması ile bu çalışmada elde edilen sonuçlar ve Kuoa ve ark.'nın verdiği sonuçlar Çizelge 4.2'de verilmiştir (Kuo ve ark., 2018).

Çizelge 4.2. Kümeleme yöntemlerinin %RI kriterine göre kümeleme başarıları (Kuo ve ark., 2018).

Yöntemler		İris	Wine	Tae	Glass	WBC
K-Ortalaması	Ortalama	79.246	91.195	46.004	39.034	96.066
	Maksimum	88.670	94.940	56.291	45.327	96.193
	Minimum	57.330	67.416	38.411	33.178	96.047
	Std. Sapma	0.146	0.095	0.051	0.038	0.001
FCM	Ortalama	89.330	94.940	49.669	42.056	95.608
	Maksimum	89.330	94.940	49.669	42.056	95.608
	Minimum	89.330	94.940	49.669	42.056	95.608
	Std. Sapma	0.000	0.000	0.000	0.000	0.000
IFCM	Ortalama	91.333	71.178	46.203	44.860	96.340
	Maksimum	91.333	75.280	47.682	44.860	96.340
	Minimum	91.333	65.170	42.384	44.860	96.340
	Std. Sapma	0.000	0.035	0.013	0.000	0.000
KFCM	Ortalama	89.467	94.233	50.177	46.012	96.003
	Maksimum	96.000	94.944	51.656	48.598	96.047
	Minimum	84.667	92.700	42.384	43.925	95.754
	Std. Sapma	0.019	0.005	0.015	0.016	0.001
KIFCM	Ortalama	91.556	93.205	51.832	46.573	96.389
	Maksimum	96.000	94.944	56.954	49.065	97.365
	Minimum	89.333	92.700	50.331	43.925	96.193
	Std. Sapma	0.019	0.009	0.023	0.016	0.003
PSO-IFCM	Ortalama	80.667	65.506	44.768	41.869	95.666
	Maksimum	90.000	84.831	51.656	48.598	96.633
	Minimum	60.000	52.809	34.437	37.383	94.290
	Std. Sapma	12.009	14.605	7.646	4.341	0.980
GA-IFCM	Ortalama	84.933	65.169	44.238	39.346	95.959
	Maksimum	97.333	80.899	45.695	50.000	97.218
	Minimum	74.000	41.011	42.384	28.972	94.583
	Std. Sapma	9.691	17.465	1.510	7.912	1.094
ABC- IFCM	Ortalama	68.933	64.157	44.768	46.729	97.482
	Maksimum	78.667	71.348	47.682	51.869	97.804
	Minimum	61.333	60.112	42.384	40.187	97.218
	Std. Sapma	6.229	4.304	2.369	4.494	0.217
POS- KFCM	Ortalama	91.822	95.206	52.958	47.118	96.960
	Maksimum	94.667	96.629	56.954	49.065	97.365
	Minimum	90.000	94.382	50.331	45.327	96.340
	Std. Sapma	0.014	0.008	0.025	0.014	0.005
GA- KFCM	Ortalama	**91.844**	**95.243**	52.936	47.103	**97.057**
	Maksimum	96.000	97.191	55.629	49.065	97.365
	Minimum	90.000	94.382	51.656	46.729	96.633
	Std. Sapma	0.013	0.007	0.009	0.005	0.002
ABC- KFCM	Ortalama	91.733	95.187	52.848	47.134	96.911
	Maksimum	96.000	96.067	55.629	48.131	97.365
	Minimum	90.000	94.382	52.318	46.262	96.633
	Std. Sapma	0.015	0.005	0.008	0.004	0.002
Önerilen Yöntem	Ortalama	78.190	72.741	**56.130**	**53.832**	87.672
	Maksimum	80.786	73.636	56.501	60.17	91.347
	Minimum	75.594	71.847	55.76	47.494	83.997
	Std. Sapma	0.038	0.008	0.014	0.071	0.121

Çizelge 4.2'de 30 çalıştırma sonucunda ortalamaya göre en iyi sonuçlar koyu renkle gösterilmiştir. Çizelge 4.2'de bulunan kümeleme yöntemlerinin en iyi kümeleme başarısı (maksimum), en kötü kümeleme başarısı (minimum), ortalama ve standart sapma (Std Sapma) değerleri incelendiğinde iris veri setinde %78.190 ortalama kümeleme başarısı ile ABC- IFCM iyi sonucu vermiştir. İris veri seti için YA ile elde edilen standart sapma değeri 0.038 ile K-Ortalaması, PSO-IFCM, GA-IFCM ve ABC-IFCM kümeleme yöntemlerinden daha tutarlı sonuç elde ettiği gözlenmiştir. Wine veri setinde önerilen YA 72.741 ortalama kümeleme başarısı değeri ile IFCM, PSO-IFCM, GA-IFCM ve ABC- IFCM kümeleme yöntemlerinin önüne geçmiştir. Ayrıca bu veri seti için YA, 0.008 standart sapma değeri ile K-Ortalaması, IFCM, KIFCM, PSO-IFCM, GA-IFCM ve ABC- IFCM kümeleme yöntemlerinden daha iyi standart sapma değeri elde etmiştir. Tae ve Glass veri setlerinde önerilen YA tabanlı kümeleme algoritması sırasıyla %56.130 ve %53.832 değerleri ile Çizelge 4.2 de verilen tüm kümeleme yöntemlerinden daha iyi ortalama kümeleme başarısı elde etmişlerdir. Üstelik Tae veri seti için standart sapma değeri 0.014 elde ederek K-Ortalaması, FCM, KIFCM, PSO-IFCM, A-IFCM, ABC- IFCM ve POS- KFCM kümeleme yöntemlerinin önüne geçmiştir. Önerilen YA tabanlı kümeleme algoritması Glass veri setinde 0.071 standart sapma değeri ile PSO-IFCM, GA-IFCM ve ABC- IFCM kümeleme yöntemlerinden daha iyi standart sapma değeri elde etmiştir. WBC veri seti için önerilen yarasa algoritması tabanlı kümeleme algoritması en kötü kümeleme başarısını göstermiştir. Bunun yanında 0.121 standart sapma değeri ile PSO-IFCM, GA-IFCM ve ABC- IFCM kümeleme yöntemlerinden daha iyi standart sapma değeri elde edilmiştir. 30 çalıştırmada elde edilen en kötü değerler açısından bir karşılaştırma yapılırsa WBC veri seti hariç diğer veri setleri için YA en kötü sonucu vermemiştir. Diğer bir deyişle 30 çalıştırmasa YA'dan daha kötü kümeleme başarısına ulaşan algoritmalar olduğu Çizelge 4.2'de görülmektedir.

Önerilen YA'nın performansı İris, Wine, Tae, Glass ve WBC veri setlerinin kümelenmesi yanında tıbbı veri seti olan BCWD, Dermatoloji, Kalp Hastalıkları, SPECT_kalp, SPECTF_kalp ve Statlog_Heart veri setlerinin kümelenmesi problemine de uygulanmıştır. Tıbbı veri setlerinin özelikleri Çizelge 3.1'de gösterildiği gibidir.

Çizelge 4.3'de görüldüğü gibi 569 örnekli BCWD veri kümesi en küçük ortalama SSE değerine 40 yarasa ile ulaşmış fakat en iyi ortalama RI değerini 10 yarasa ile sağlamıştır. Bunun yanında BCWD veri seti için 0,78570 en yüksek RI değerine 20 yarasa ile ulaşılmıştır. Bu veri seti için ortalama RI değeri açısından 10 yarasa ve 40

yarasa ile yakın sonuçlar elde edildiğinden ve en iyi SSE değerine 40 yarasa ile ulaşıldığından en başarılı yarasa sayısı 40 olarak belirlenebilir.

Çizelge 4.3. BCWD veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	1148.4149	577.6448	1304.8986	122.8563	**0.52595**	0.72056	0.21793	0.14026
20	1108.8383	600.7155	1258.467	115.4681	0,47412	**0,78570**	0,21089	0,143149
30	1102.0841	548.1075	1193.2174	115.6908	0.47194	0.6696	0.1529	0.14626
40	**1093.5005**	599.2758	**1185.5387**	101.1202	0.52156	0.76274	0.26889	0.13836

33 özellikli Dermatology veri kümesi için 10 yarasa ile en yüksek RI değeri 0.40984 olarak elde edilmiştir. 40 yarasa sayısı ile 876.241 ortalama SSE değeri elde edilmiştir Fakat ortalama değerlere bakıldığında Dermatoloji veri seti için RI açısından başarı oldukça düşüktür. (Çizelge 4.4.).

Çizelge 4.4. Dermatology veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	895.7292	703.0395	946.8458	45.4364	**0.17668**	**0.40984**	0.0081967	0.12464
20	894.7616	706.7984	939.6493	40.1903	0.14262	0.27049	0.019126	0.05773
30	879.6228	700.3306	933.6046	45.2728	0.14536	0.34699	0.035519	0.074465
40	**876.241**	675.1906	**910.3305**	40.9466	0.15811	0.3306	0.032787	0.081158

2 sınıflı Kalp Hastalıkları veri kümesi kullanılarak YA ile elde edilen sonuçlar Çizelge 4.5'da verilmiştir. Ortalama değerler açısından bakıldığında SSE ve RI değerleri açısından sırasıyla 10 ve 20 yarasa sayılarında en iyi sonuçlar elde edilmiştir. 30 çalıştırma ile ulaşılan en iyi sonuçlar 40 yarasa ile olmuştur.

Çizelge 4.5. Kalp Hastalıkları veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	**360.117**	332.6216	436.6516	24.0236	0.51232	0.63366	0.33993	0.080579
20	410.5503	359.2319	440.8114	19.184	**0.51573**	0.71947	0.31353	0.11708
30	360.6103	362.8778	417.9046	15.136	0.48218	0.75248	0.22772	0.12593
40	385.4985	347.876	**412.5824**	16.7486	0.50308	**0.77228**	0.25083	0.12832

Çizelge 4.6 22 özelliğe sahip 2 sınıflı SPECT_kalp veri kümesi için elde edilen sonuçları göstermektedir. RI açısından 20 yarasa sayısı ile en yüksek ve ortalama değerlere ulaşılıyorken, en düşük SSE ve ortalama açısından 40 yarasa ile en iyi sonuçlar elde edilmiştir. Önerilen YA kümeleme algoritması ile en yüksek 0.71536 RI ve yarasa sayısı 40 da en düşük SSE değeri 642.0202 olarak elde edilmiştir (Çizelge 4.6).

Çizelge 4.6.SPECT_kalp veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	632.3881	537.8335	663.2361	23.9206	0.4985	0.6779	0.34831	0.095057
20	630.6045	530.310	659.804	22.7999	**0.51536**	**0.71536**	0.31835	0.11277
30	623.4791	521.4428	650.9503	23.7639	0.51273	0.70037	0.30712	0.11838
40	**622.1786**	549.1382	**642.0202**	16.9357	0.47478	0.63296	0.33333	0.091745

Çizelge 4.7. YA ile 44 özelliğe sahip SPECTF_kalp veri kümesinin 30 çalıştırma sonundaki sonuçlarını göstermektedir. Çizelge 4.7'e bakıldığında, hem en düşük ortalama hem de en düşük SSE değerine 40 yarasa sayısı ile ulaşılmıştır. En yüksek ortalama RI değerine 10 yarasa ile ulaşılırken en yüksek RI değeri (0.78652) 10, 30 ve 40 yarasa ile elde edilmiştir.

Çizelge 4.7. SPECTF_kalp veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	446.0533	242.9209	501.0543	47.1596	**0.55506**	**0.78652**	0.22472	0.21961
20	432.3198	232.9321	484.6728	44.2081	0.39401	0.77903	0.20974	0.21094
30	420.8562	255.4658	480.2127	41.8739	0.50874	**0.78652**	0.20974	0.23771
40	**413.8984**	200.4122	**465.4961**	46.4476	0.44669	**0.78652**	0.20974	0.23586

Çizelge 4.8'da 13 özelliğe sahip 2 sınıflı Statlog_Heart veri kümesinin sonuçları gösterilmiştir. Çizelge 4.9'da ifade edildiği gibi, en düşük ortalama SSE değerine 40 yarasa sayısında, 30 çalıştırma sırasında en düşük SSE değerine ise 30 yarasa ile varılmıştır. Bunun yanında RI açısından en iyi değerler de 30 yarasa ile elde edilmiştir.

Çizelge 4.8. Statlog_Heart veri seti kullanılarak elde edilen test sonuçları

Yarasa Sayısı	**SSE**				**RI**			
	Ortalama	Maksimum	Minimum	Std.Sapma	Ortalama	Maksimum	Minimum	Std.Sapma
10	387.3118	357.2624	418.466	15.292	0.45716	0.75926	0.23333	0.13394
20	378.5474	344.1558	398.4578	12.4756	0.49741	0.71111	0.25926	0.12191
30	373.6099	345.9507	**393.3526**	11.6144	**0.52815**	**0.77037**	0.25556	0.13013
40	**370.9521**	341.2462	400.3911	10.44	0.48358	0.76296	0.22222	0.14043

Sonuç olarak bu çalışmada kullanılan bütün tıbbi veri setleri için, önerilen YA tabanlı kümeleme yönteminden elde edilen kümeleme başarıları farklı yarasa sayıları ve 30 çalıştırma için Çizelge 4.9'da doğru kümelenen örnek sayısı açısından özetlenmiştir.

Çizelge 4.9. Tıbbı veri setlerinden elde edilen kümeleme başarıları

Veri seti adı	Yarasa sayısı(n)	Ortalama	Maksimum	Minimum	Std. Sapma
BCWD	10	**301.0667**	410	124	80.06351
	20	269.7667	**447**	120	81.4517
	30	268.5333	381	87	77.40443
	40	296.7667	434	153	**81.82409**
Dermatology	10	**64.6667**	150	3	**44.8526**
	20	61.73333	**171**	8	30.903
	30	26.79602	127	13	26.79602
	40	57.86667	121	12	29.20472
Kalp Hastalıkları	10	155.2333	192	103	24.00512
	20	**156.2667**	218	95	34.87782
	30	146.1	228	69	37.51475
	40	152.4333	**234**	76	**38.22711**
SPECT_kalp	10	133.1	181	93	24.95376
	20	**137.6**	**191**	85	29.6036
	30	136.9	187	82	**31.07663**
	40	126.7667	169	89	24.08413
SPECTF_kalp	10	**148.2**	**210**	60	57.65148
	20	105.2	208	56	55.37352
	30	135.8333	**210**	56	62.40304
	40	119.8667	**210**	56	**62.7472**
Statlog_Heart	10	132.3	215	61	**40.49786**
	20	134.3	192	70	32.36165
	30	**136.5333**	**221**	75	34.25564
	40	130.5667	206	60	37.27884

Çizelge 4.9'da görüldüğü gibi 30 çalıştırma sırasında tamamının doğru kümelendiği bir veri seti yoktur. BWCD veri seti için 569 örnekten en fazla 447 örnek 20 yarasa ile yapılan denemede doğru kümelenmiştir. Bunun yanında 10 yarasa için 30 denemede en yüksek ortalama doğru kümeleme sayısına ulaşılmıştır. YA, Dermatology veri setinde en düşük başarıyı göstermiştir. 366 örnekten en fazla 171 örnek 20 yarasa ile doğru kümelenebilmiş ve 0.46 RI değerine ulaşılabilmiştir. 303 örnekli Kalp

Hastalıkları veri seti için en yüksek kümeleme başarısı 40 yarasa sayısı ile 234'tür. SPECT_kalp ve SPECTF_kalp veri setlerinin örnek sayısı 267 ve maksimum doğru kümeleme sayısı sırasıyla 191 ve 210 olarak elde edilmiştir. Son olarak 270 örnekli Statlog_Heart veri seti kullanılarak yapılan denemelerde 30 yarasa sayısı ile 221 örnek doğru kümelenerek 0.796 RI değeri elde edilmiştir. 30 çalıştırmada doğru olarak kümelenen veri sayısının en düşük olduğu veri seti Dermatology veri setidir.

5. SONUÇLAR VE ÖNERİLER

Optimizasyon, arama alanını keşfetmek ve kullanmak için farklı kavramları birleştirerek yinelemeli bir üretim süreci olarak tanımlanır. Öğrenme stratejileri, verimli olmayan çözümler bulmak amacıyla bilgileri yapılandırmak için kullanılır. Optimizasyon yöntemleri, basit yerel arama prosedürlerinden karmaşık öğrenme süreçlerine kadar farklı optimizasyon algoritmalarından oluşan yaklaşık ve genellikle deterministtik olmayan tekniklerdir.

Optimizasyon algoritmaları genel olarak mühendislik, tıp ve havacılık gibi alanlarda sıkça kullanılmaktadır. Ayrıca, optimizasyon algoritmaları zor olan veri madenciliği ve makine öğrenmesi gibi problemlerin çözümünde de aktif olarak tercih edilmektedir. Kümeleme, veri madenciliği alanında çok önemli bir denetimsiz sınıflandırma şeklidir. Kümeleme, veriyi benzer nesnelerin gruplarına bölmesidir. Küme adı verilen her grup, küme içindeki benzer nesneler ve diğer kümelerin nesnelerine benzemeyen nesnelerden oluşmaktadır.

Verileri daha az küme ile temsil etmek, kesin olarak verilere ait bazı ince ayrıntıları kaybettirmekle birlikte basitleştirmeyi sağlar ve bu nedenle kümeleme bir veri sıkıştırma biçimi olarak kabul edilebilir. Dolayısıyla, birkaç küme tarafından birçok veri nesnesi temsil edilebilir ve kümelerine göre veri modelleme yapılabilir. Veri modelleme, kümelemeyi matematik, istatistik ve sayısal analizlere dayanan bir perspektife yerleştirir.

Kümeleme, istatistik, örüntü tanıma, veri madenciliği ve makine öğrenimi gibi çeşitli alanlarda aktif bir araştırma konusudur. Pratik açıdan bakıldığında, kümeleme bilimsel veri araştırma, bilgi alma ve metin madenciliği, web analizi, pazarlama, tıbbi teşhis, hesaplamalı biyoloji ve diğerleri gibi veri madenciliği uygulamalarında olağanüstü bir rol oynamaktadır. Kümelenme sorunu birçok bağlamda ve birçok disiplindeki araştırmacılar tarafından ele alınmıştır. Bu, geniş kapsamlı kullanışlılığını veri analizindeki adımlardan biri olarak yansıtmaktadır.

Günümüze kadar araştırmacılar tarafından birçok kümeleme algoritmaları geliştirilmiştir, geliştirilen bu kümeleme algoritmaları çeşitli veri setlerinde kullanılmıştır. Kümeleme algoritmalarının performansını etkileyen en önemli faktör küme merkezlerinin en iyi şekilde bulunmasıdır, kümeler için ne kadar iyi küme merkezleri elde edilirse o kadar iyi bir kümeye sahip olunmaktadır. Dolayısıyla verilerin küme merkezleri ciddi bir şekilde kümeleme algoritmalarının performansını

etkilemektedir. Her kümeleme algoritmasının kendine göre bir küme merkezi güncelleme formülü bulunmaktadır.

Kümeleme problemlerinin çözümünde yeni yaklaşımlardan biri optimizasyon algoritmalarının kullanılmasıdır. Bu tez çalışmasında YA'nın kümeleme performansı farklı veri setleri kullanılarak incelenmiştir. Önerilen YA tabanlı kümeleme yönteminde her iterasyonda YA'ya göre veri setlerinin küme merkezleri güncellenerek en iyi küme merkezleri araştırılmıştır. Önerilen yarasa algoritması tabanlı kümeleme yönteminin performansı araştırmacılar tarafından önerilen optimizasyon tabanlı kümeleme yöntemlerinden K-Ortalama, FCM, IFCM, KFCM, KIFCM, PSO-IFCM, GA-IFCM, ABC- IFCM, POS- KFCM, GA- KFCM ve ABC- KFCM ile İris, Wine, Tae, Glass ve Wbc veri setleri kullanılarak yapılan denemeler sonucunda elde edilen verilere dayanarak karşılaştırılmıştır. Karşılaştırma sonucunda önerilen YA tabanlı kümeleme yöntemi Tae ve Glass veri setlerinde tüm optimizasyon tabanlı ve standart kümeleme algoritmalarından daha iyi kümeleme başarısı göstermiştir. İris veri setinde önerilen yarasa algoritması tabanlı kümeleme yöntemi ABC- IFCM kümeleme yönteminden daha iyi sonuç vermiştir. Wine veri setinde önerilen yarasa algoritması tabanlı kümeleme algoritması IFCM, PSO-IFCM, GA-IFCM ve ABC- IFCM kümeleme yöntemlerinden iyi sonuç vermiştir. Önerilen yarasa algoritması tabanlı kümeleme yöntemi tıbbı veri setlerine uygulandığında genele olarak tıbbı veri setlerini iyi bir şekilde kümelediği görünmüştür. Fakat 30 çalıştırma için tamamının doğru kümelendiği bir veri seti yoktur. Kalp ile ilgili veri setlerinde (Statlog_Heart, Kalp Hastalıkları, SPECT_kalp ve SPECTF_kalp) yaklaşık %78 verinin doğru olarak kümelendiği çalıştırmalar mevcuttur. Fakat 30 çalıştırmanın ortalamasına bakıldığında bu değer %50 civarında kalmaktadır.

Sonuç olarak önerilen YA tabanlı kümeleme yöntemi farkı sayıda özelliğe ve sınıfa sahip verilere uygulanabilmektedir. Performansın iyileştirilebilmesi için farklı optimizasyon algoritmaları ile hibrit yapılar oluşturulabilir. Algoritmanın çalışması sırasında tek bir en iyisinin olması lokale takılma ihtimalini arttırmaktadır. Bu özellik iyileştirilebilirse daha iyi performans elde edilebilir.

KAYNAKLAR

Bezdek, J. C., 1974, Numerical taxonomy with fuzzy sets, *Journal of Mathematical Biology*, 1 (1), 57-71.

Bezdek, J. C., 1981, Objective function clustering, In: Pattern recognition with fuzzy objective function algorithms, Eds: Springer, p. 43-93.

Bezdek, J. C., Ehrlich, R. ve Full, W., 1984, FCM: The fuzzy c-means clustering algorithm, *Computers & Geosciences*, 10 (2-3), 191-203.

Basu T, Murthy C, 2015. A similarity assessment technique for effective grouping of documents. Information Sciences, 311, 149-62.

Bilekdemir, G., 2010, Veri madenciliği tekniklerini kullanarak üretim süresi tahmini ve bir uygulama, *DEÜ Sosyal Bilimleri Enstitüsü*.

Canayaz, M. ve Özdağ, R., 2017, Data clustering based on the whale optimization, *Middle East Journal of Technic*, 2 (1), 178-187.

Erilli, N., 2009, Kümeleme Analizine Bulanık Yaklaşım Algoritmaları ve Uygulamaları, 19 Mayıs Üniv, *Fen Bilimleri Enst., Yayınlanmamış Yüksek Lisans Tezi, Samsun. Bilimleri Enst., Samsun*.

Gandomi, A. H., Yang, X.-S., Alavi, A. H. ve Talatahari, S., 2013, Bat algorithm for constrained optimization tasks.

Işık, M. ve Çamurcu, A. Y., 2007, K-means, K-medoids ve bulanık C-means algoritmalarının uygulamalı olarak performanslarının tespiti.

James C. Bezdek, R. E., Wiliam Full, 1984, FCM: The fuzzy c-means clustering algorithm, *Computers & Geosciences*, 10 (2-3), 191-203.

Karaboga, D. ve Ozturk, C., 2011, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, *Applied Soft Computing*, 11 (1), 652-657.

Karakoyun, M., 2015, Kurbağa Siçrama Algorġtmasinin Kümeleme Problemlerġne Uygulanmasi, Selçuk ÜnİversİtesİFen BİlİmlerİEnstİtüsü, 84.

Karri, C. ve Jena, U., 2016, Fast vector quantization using a Bat algorithm for image compression, *Engineering Science and Technology, an International Journal*, 19 (2), 769-781.

Kuo, R., Lin, T., Zulvia, F. E. ve Tsai, C., 2018, A hybrid metaheuristic and kernel intuitionistic fuzzy c-means algorithm for cluster analysis, *Applied Soft Computing*, 67, 299-308.

Lamar, M., Maron, Y., Johnson, M. ve Bienenstock, E., 2010, SVD and clustering for unsupervised POS tagging, *Proceedings of the ACL 2010 Conference Short Papers*, 215-219.

Nguyen, Q. ve Kuo, R., 2019, Partition-and-merge based fuzzy genetic clustering algorithm for categorical data, *Applied Soft Computing*, 75, 254-264.

Osaba, E., Yang, X.-S., Diaz, F., Lopez Garci, P. ve Carballed, R., 2015, An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems, 48 (2016) 2059–2071.

Özturk, C., Hancer, E. ve Karaboga, D., 2014, Küresel En İyİ Yapay Ari KolonİAlgorİtmasi İle OtomatİK Kümeleme, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 29 (4).

Rahman, A. ve Islam, Z., 2014, A hybrid clustering technique combining a novel genetic algorithm with K-Means, *Knowledge-Based Systems*, 71, 345-365.

Ray, S. ve Turi, R. H., 1999, Determination of number of clusters in k-means clustering and application in colour image segmentation, *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, 137-143.

Senthilnath, J., Kulkarni, S., Benediktsson, J. A. ve Yang, X.-S., 2016, A novel approach for multispectral satellite image classification based on the bat algorithm, *IEEE Geoscience and Remote Sensing Letters*, 13 (4), 599-603.

Serkan, K. ve Fiğlali, N., 2018, Çok Amaçlı Esnek Atölye Tipi Çizelgeleme Problemlerinin Çözümünde Meta Sezgisel Yöntemlerin Kullanımı, *Harran Üniversitesi Mühendislik Dergisi*, 3 (3), 222-233.

Sharawi, M., Emary, E., Saroit, I. A. ve El-Mahdy, H., 2015, WSN's Energy-Aware Coverage Preserving Optimization Model based on Multi-Objective Bat Algorithm.

Soruri, M., Sadr, J. ve Zahiri, H., 2018, Gene clustering with hidden Markov model optimized by PSO algorithm, 21:1121–1126.

Şen, B., Atasoy, F. ve Aydın, N., 2010, Düşük maliyetli web tabanlı uzaktan eğitim sistemi uygulaması, *Akademik Bilişim*, 10-12.

Tripathi, A. K., Sharma, K. ve Bala, M., 2018, Dynamic frequency based parallel k-bat algorithm for massive data clustering (DFBPKBA), *International Journal of System Assurance Engineering and Management*, 9 (4), 866-874.

Tzortzis G, Likas A, 2014. The MinMax k-means clustering algorithm. Pattern Recognition, 47, 7, 2505-16.

Wanga, G., Ericchud, H. ve Seyedali, m., 2012, Three-dimensional path planning for UCAV using an improved bat algorithm.

UCI Repository Machine Learning, 2019, UCI Repository Machine Learning Database. https://archive.ics.uci.edu/ml/datasets.php: 25.02.2019

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, X.-S., 2013, Bat Algorithm: Literature Review and Applications, pp. 141–149.

Yetkin, M. ve Bilginer, Ö., 2018, Yarasa Algoritması ile Robust Kestirimin Nivelman Ağlarına Uygulanması, 3(3);196-202.

Yılmaza, S. ve Küçüksille, E. U., 2014, A new modification approach on bat algorithm for solving optimization problems, 28 (2015) 2259–2275.

Zweije, D., 2017, Solving the clustering problem using greedy partitioning and linkage tree approaches.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Mina Arjamand Fadhil FADHIL
Uyruğu	**:**	Irak
Doğum Yeri ve Tarihi	**:**	Bağdat/ 03-03-1995
Telefon	**:**	05379453151
Faks	**:**	
e-mail	**:**	Mena.fadhil95@gmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	**Aden-Bağdat-Irak**	**2012**
Üniversite	:	**Al-mansour-Irak**	**2016**
Yüksek Lisans	:	**S.Ü.Bilgisayar Mühendisliği A.B.D.. Selçuklu konya**	**2019**

UZMANLIK ALANI (Kümeleme, Optimizasyon)

YABANCI DİLLER (Arabca, ingilizca)

YAYINLAR

M. Fadhil, G.Tezel, Data Clustering With Bat Algorithm Using Benchmark Datasets *İnternational Conference On Engineering Technologies*,2019, ICENTE 19