



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**ÇEKİRDEK FONKSİYONU KULLANAN
META SEZGİSEL TABANLI YENİ BİR
KÜMELEME ALGORİTMASI**

Hilal ELMAS

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

**Nisan-2024
KONYA
Her Hakkı Saklıdır.**

TEZ KABUL VE ONAYI

Hilal ELMAS tarafından hazırlanan “Çekirdek Fonksiyonu Kullanan Meta Sezgisel Tabanlı Yeni Bir Kümeleme Algoritması” adlı tez çalışması 26/04/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr.Öğr.Üyesi Vahit TONGUR

.....

Danışman

Dr.Öğr.Üyesi Ayşe Merve ACILAR

.....

Üye

Dr.Öğr.Üyesi Murat KARAKOYUN

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun/.../20.. gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Şerife Yurdağül KUMCU
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.

Hilal ELMAS

26.04.2024

ÖZET

YÜKSEK LİSANS TEZİ

ÇEKİRDEK FONKSİYONU KULLANAN META SEZGİSEL TABANLI YENİ BİR KÜMELEME ALGORİTMASI

Hilal ELMAS

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ayşe Merve ACILAR

2024, 44 Sayfa

Jüri

Dr. Öğr. Üyesi Vahit TONGUR
Dr. Öğr. Üyesi Ayşe Merve ACILAR
Dr. Öğr. Üyesi Murat KARAKOYUN

Kümeleme analizi, çok boyutlu uzayda yer alan etiketsiz nesnelerin benzerlik veya yakınlık temelli bir yaklaşımla belirli gruplar içerisinde gruplanmasını sağlayan algoritmalar. Günümüzde üretilen ham verilerin çoğunun etiketsiz olması kümeleme algoritmalarının önemini artırmıştır. Kümeleme algoritmaları, veri noktalarını benzerliklerine göre gruplara ayırma sürecinde, farklı kümeler arasındaki mesafenin mümkün olduğunca büyük olması ve aynı küme içindeki veri noktalarının birbirine olabildiğince yakın olması hedeflenmektedir. Kümeleme, bu iki kriteri dengeleyerek optimum küme merkezlerini bulmayı amaçlayan bir optimizasyon problemi olarak ele alınabilir. Böylece meta sezgisel optimizasyon algoritmaları kümeleme için kullanılabilir hale gelir. Meta-sezgisel algoritmalar, karmaşık optimizasyon problemlerini çözmek için geliştirilen ve geleneksel yöntemlerin yetersiz kaldığı durumlarda etkili sonuçlar üreten güçlü arama stratejileridir. Bu algoritmalar, kümelemede yüksek boyutlu veri kümeleri ve çoklu yerel minimumların bulunduğu karmaşık problem uzaylarında etkin bir şekilde kullanılarak, geleneksel kümeleme yöntemlerinin sınırlamalarını aşmada ve daha doğru ve esnek kümeleme çözümleri sunabilmektedirler. Kümeleme başarısını artırmak için kullanılan diğer bir araç ise Çekirdek (Kernel) fonksiyonlarıdır. Çekirdek fonksiyonları, veri noktalarını daha yüksek boyutlu bir uzaya dönüştürerek, lineer olmayan yapıların lineer yöntemlerle analiz edilmesini mümkün kılan matematiksel araçlardır. Kümeleme analizinde, veri noktalarının daha karmaşık ve gizli ilişkilerinin ortaya çıkarılmasını sağlarlar. Özellikle veri setinin orijinal uzayında lineer olarak ayrılmadığı durumlarda faydalıdır. Meta-sezgisel kümeleme algoritmalarında çekirdek fonksiyonları kullanımı, çözüm uzayının daha etkili bir şekilde taranmasını ve kümeleme sonuçlarının iyileştirilmesini sağlayabilmektedir. Bu motivasyondan yola çıkılarak gerçekleştirilen bu çalışmada, özellikle doğrusal olarak ayrılmayan veri kümelerinin kümeleme süreçlerinde etkinliği artırmak için, Radyal Tabanlı Çekirdek fonksiyonu (RBF) ile Balina Optimizasyon Algoritması'nı (WOA) birleştiren bir yaklaşımı sunulmuş ve önerilen yöntem KWOA ismi verilmiştir. WOA, balinaların avlanma davranışlarından esinlenerek tasarlan ve özellikle kambur balinaların balık sürülerini sarmal hareketlerle yakalama stratejilerini model alan meta sezgisel bir optimizasyon algoritmasıdır. Önerilen KWOA yönteminin etkinliği, UCI Makine öğrenmesi veri deposundan alınan sekiz veri seti üzerinde test edilmiş ve Rand İndeks (RI) ve Siluet İndeks (SI) kullanılarak kümeleme doğrulukları değerlendirilmiştir. Deneysel çalışmalar 30 kez tekrar edilmiştir. Literatürde başarısı ispatlanmış dokuz farklı kümeleme algoritması da aynı veri setleri üzerinde 30 kez çalıştırılmış ve elde edilen tüm sonuçlar tartışılmıştır. Sonuç olarak önerilen KWOA algoritması ile en yüksek RI ve SI değerlerinin elde ettiği görülmüştür. Özetle, gerçekleştirilen tez çalışmasında KWOA algoritmasında, çekirdek fonksiyonlarının veri setlerinin doğrusal olmayan yapısını etkili bir şekilde temsil edebilme kapasitesi ile ve WOA'nın optimizasyon yeteneği birleştirilmiş ve etkili sonuçlar elde edilmiştir.

Anahtar Kelimeler: Balina Optimizasyon Algoritması, Kernel Kümeleme, Kümeleme, Radyal Tabanlı Çekirdek Fonksiyonu

ABSTRACT

MS THESIS

A NEW META-HEURISTIC BASED CLUSTERING ALGORITHM USING KERNEL FUNCTION

Hilal ELMAS

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Asst. Prof. Dr. Ayşe Merve ACILAR

2024, 44 Pages

Jury

**Asst. Prof. Dr. Vahit TONGUR
Asst. Prof. Dr. Ayşe Merve ACILAR
Asst. Prof. Dr. Murat KARAKOYUN**

Clustering analysis is an algorithm that allows unlabeled objects in a multidimensional space to be grouped into certain groups with a similarity or distance-based approach. The fact that most of the raw data produced today is unlabeled has increased the importance of clustering algorithms. In the process of dividing data points into groups according to their similarities, clustering algorithms aim to ensure that the distance between different clusters is as large as possible and that the data points within the same cluster are as close to each other as possible. Clustering can be considered as an optimization problem that aims to find optimum cluster centers by balancing these two criteria. Thus, metaheuristic optimization algorithms become available for clustering. Meta-heuristic algorithms are powerful search strategies developed to solve complex optimization problems and produce effective results when traditional methods are insufficient. These algorithms can be used effectively in complex problem spaces with high-dimensional data sets and multiple local minima in clustering, overcoming the limitations of traditional clustering methods and providing more accurate and flexible clustering solutions. Another tool used to increase clustering success is Kernel functions. Kernel functions are mathematical tools that make it possible to analyze nonlinear structures with linear methods by transforming data points into a higher dimensional space. The use of kernel functions in meta-heuristic clustering algorithms can enable more effective scanning of the solution space and improvement of clustering results. In this study, an approach combining the Radial Basis Kernel function (RBF) and the Whale Optimization Algorithm (WOA) was presented to increase the efficiency in the clustering processes of data sets that are not linearly separable, and the proposed method was named KWOA. WOA is a meta-heuristic optimization algorithm inspired by the hunting behavior of whales and specifically modeled on the strategies of humpback whales to catch schools of fish in spiral movements. The effectiveness of the proposed KWOA method was evaluated on eight datasets retrieved from the UCI Machine learning data repository and their clustering accuracies were evaluated using Rand Index (RI) and Silhouette Index (SI). Experimental studies were repeated 30 times. Nine different clustering algorithms that have proven their success in the literature were run 30 times on the same data sets and all the results obtained were discussed. As a result, it was seen that the highest RI and SI values were obtained with the proposed KWOA algorithm. In summary, in the thesis study, the KWOA algorithm combined the capacity of kernel functions to effectively represent the non-linear structure of data sets and the optimization ability of WOA and effective results were obtained.

Key Words: Whale Optimization Algorithm, Kernel Clustering, Clustering, Radial Basis Kernel Function

ÖNSÖZ

Danışmanım Dr. Öğr. Üyesi Ayşe Merve ACILAR'a gösterdiği anlayış ve tüm desteklerinden dolayı teşekkür ederim.

Bütün eğitim ve öğrenim hayatım boyunca bana destek olan aileme teşekkür ederim.

Hilal ELMAS
KONYA-2024



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
KISALTMALAR	ix
1. GİRİŞ.....	1
1.1.Tezin Amacı ve Literatüre Katkısı.....	2
1.2.Tezin Yapısı.....	2
2. KAYNAK ARAŞTIRMASI	3
3.MATERYAL VE YÖNTEM.....	8
3.1.Çalışmada Kullanılan Veri Setleri	8
3.2. Kümeleme Yöntemleri.....	10
3.3. K-Ortalamlar Algoritması	11
3.4. Kernel Fonksiyonları	12
3.4.1. Doğrusal Kernel Fonksiyonu	14
3.4.2. Polinomal Kernel Fonksiyonu	14
3.4.3. Radyal Tabanlı Kernel Fonksiyonu	15
3.5. Kernel K Ortalamalar.....	16
3.6. Kümeleme Analizinde Kullanılan Küme Doğrulama Yöntemleri	16
3.6.1. Rand İndeks Küme Doğrulama Değeri.....	16
3.6.2. Siluet Küme Doğrulama Değeri	17
3.7. Balina Optimizasyon Algoritması	17
3.7.1. Balina Optimizasyon Algoritması Matematiksel Modeli	19
3.7.1.1. Avın Etrafını Kuşatma	19
3.7.1.2. Kabarcık Ağı Hareketi ile Saldırma.....	20
3.7.1.2.1. Daralan Dairesel Hareketler ile Kuşatma	20
3.7.1.2.2. Spiral Hareket Oluşturma	20
3.7.1.3. Av Arama.....	22
4.ÇEKİRDEK FONKSİYONU KULLANAN BALINA OPTİMİZASYON ALGORİTMASI TABANLI YENİ BİR KÜMELEME ALGORİTMASI- KWOA	
4.1. Veri Kümelerinin Hazırlanması.....	24
4.2. Arama Ajanlarının Kodlanması ve Başlangıç Popülasyonun Oluşturulması	25
4.3. Uygunluk Fonksiyonu.....	27
4.4. Önerilen KWOA Algoritması ve Örnek Bir Çözüm	28

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	33
5.1. Rand Index Karşılaştırma Sonuçları	33
5.2. Siluet Küme Doğrulama İndeksi Sonuçları	36
5.3. KWOA ile Kümelenmiş Verileri Görsel Sonuçları	38
6. SONUÇLAR VE ÖNERİLER.....	41
6.1 Sonuçlar	41
6.2 Öneriler	41
7. KAYNAKLAR	42
ÖZGEÇMİŞ	45



KISALTMALAR

- WOA :Balina Optimizasyon Algoritması (**W**hale **O**ptimization **A**lgorithm)
RBF :Radyal Tabanlı Kernel Fonksiyonu (**R**adial **B**asis **K**ernel **F**unction)
KWOA :Kernel fonksiyonlu Balina Optimizasyon Algoritması Tabanlı Kümeleme
(A New Whale Optimizastion Based Clustering Algorithm Using Kernel
Function)
RI :Rand İndeks (**R**and **I**ndex)
SI :Siluet Küme Doğrulama İndeksidir (**S**ilhouette **C**luster **V**alidation **I**ndex)
K :Veri setinin ayrılacağı küme sayısı
M :Veri setindeki özellik sayısı
X :Popülasyondaki bir arama ajanı (WOA'daki bir balinanın konumu)



1. GİRİŞ

Kümeleme, veri noktalarını benzerliklerine göre gruplara ayırma sürecidir ve bu süreçte, farklı kümeler arasındaki mesafenin mümkün olduğunca büyük olması ve aynı küme içindeki veri noktalarının birbirine olabildiğince yakın olması hedeflenmektedir. Kümeleme, bu iki kriteri dengeleyerek optimum kümeleri bulmayı amaçlayan bir optimizasyon problemi olarak ele alınabilir. Örneğin, veri madenciliği ve makine öğrenmesinde sıklıkla tercih edilen k-ortalamlar (k-means) algoritmasında amaç fonksiyonu, her bir küme içindeki veri noktalarının küme merkezlerine olan uzaklıklarının toplamını minimize etme problemidir.

Meta-sezgisel algoritmalar, karmaşık optimizasyon problemlerini çözmek için geliştirilen ve geleneksel yöntemlerin yetersiz kaldığı durumlarda etkili sonuçlar üreten güçlü arama stratejileridir. Bu algoritmalar, genellikle rastgelelik ve belirli bir çözüm uzayını geniş çapta tarayarak global optimuma ulaşmayı hedeflemektedir. Bu algoritmalar, kümelemede yüksek boyutlu veri kümeleri ve çoklu yerel minimumların bulunduğu karmaşık problem uzaylarında etkin bir şekilde kullanılarak, geleneksel kümeleme yöntemlerinin sınırlamalarını aşarak, daha doğru ve esnek kümeleme çözümleri sunabilirler.

Kernel fonksiyonları, veri noktalarını daha yüksek boyutlu bir uzaya dönüştürerek, lineer olmayan yapıların lineer yöntemlerle analiz edilmesini mümkün kılan matematiksel araçlardır. Kümeleme analizinde, veri noktalarının daha karmaşık ve gizli ilişkilerinin ortaya çıkarılmasını sağlar. Kernel fonksiyonları, kümeleme algoritmalarında, özellikle veri setinin orijinal uzayında lineer olarak ayrılmadığı durumlarda, daha anlamlı ve homojen kümelerin oluşturulmasına yardımcı olur. Meta-sezgisel kümeleme algoritmalarında kernel fonksiyonları, çözüm uzayının daha etkili bir şekilde taranmasını ve kümeleme sonuçlarının iyileştirilmesini sağlayabilir. Bu kombinasyon, algoritmaların hem doğruluğunu arttırmada hem de veri yapısının karmaşıklığıyla başa çıkmada kolaylık sağlayabilir. Bu motivasyondan yola çıkılarak gerçekleştirilen bu çalışmada, özellikle doğrusal olarak ayrılamayan veri kümelerinin kümeleme süreçlerinde etkinliği artırmak için, Radial Basis Function (RBF) kernel fonksiyonu ile Balina Optimizasyon Algoritması'nı (WOA) birleştiren bir yaklaşımı sunulmuş ve önerilen yönteme KWOA ismi verilmiştir.

WOA, balinaların avlanma davranışlarından esinlenerek tasarlan ve özellikle kambur balinaların balık sürülerini sarmal hareketlerle yakalama stratejilerini model

alan meta sezgisel bir optimizasyon algoritmasıdır. Bu algoritmanın temel amacı, arama alanındaki en iyi çözümü minimum iterasyonla bulmaktır. Kümeleme analizinde, çözüm uzayındaki her bir balina ajanı, veri noktalarının küme merkezlerine ilişkin potansiyel çözümleri temsil eder. WOA'nın her iterasyonunda, ajanlar arasındaki en iyi çözümü belirlemek için RBF kernel fonksiyonu kullanılır. Bu fonksiyon, veri noktalarının özelliklerini yüksek boyutlu bir uzaya dönüştürerek, ajanların küme merkezlerine olan yakınlığını ölçer ve böylece ajanların veri uzayındaki konumlarını güncelleyerek global optimuma doğru ilerlemelerine yardımcı olur. İterasyonlar boyunca ajanlar, RBF kernel yardımıyla hesaplanan bu yakınlık ölçütüne göre hareket eder ve nihai olarak en düşük maliyetli optimum çözüme yani veri noktalarını en iyi temsil eden küme merkezlerine ulaşmaya çalışılır. Önerilen KWOA yönteminin etkinliği, UCI Machine Learning Repository'den alınan veri setleri üzerinde test edilmiş ve Rand İndeks (RI) ve Siluet İndeks (SI) kullanılarak değerlendirilmiş, elde edilen sonuçların analizi yapılmıştır.

1.1. Tezin Amacı ve Literatüre Katkısı

Bu çalışmada, kümeleme algoritmalarının başarılarını arttırmak için, kullanılan verilerin doğrusal uzaydan, yüksek boyutlu bir uzaya dönüştürülmesini sağlayan kernel fonksiyonlarının kullanılması amaçlanmıştır. Bu amaç için özellikle doğrusal olarak ayıramayan veri kümelerinin kümeleme süreçlerinde etkinliği arttırmak için, Radial Basis Function (RBF) kernel fonksiyonu ile Balina Optimizasyon Algoritmasını (WOA) birleştiren bir yaklaşımı sunulmuştur. Kernel fonksiyonlarının veri dönüşüm kapasitesi ile WOA'nın optimizasyon yeteneği entegre edilerek, veri kümeleme problemlerinin çözümünde daha etkili sonuçlar elde edebilen, KWOA kümeleme algoritması ile literatüre katkı sağlanmıştır.

1.2. Tezin Yapısı

Tez çalışmasının ikinci bölümde kaynak araştırmasına yer verilmiştir. Üçüncü bölümünde kümeleme algoritma grupları, k ortalamalar, kernel k ortalamalar ve kernel fonksiyonları ve balina optimizasyon algoritması hakkında bilgiler sunulmuştur. Dördüncü bölümde, kernel yöntemini kullanarak balina optimizasyon algoritması tabanlı kümeleme yöntemi açıklanmıştır. Beşinci bölümde araştırma sonuçları verilmiş ve tartışılmıştır. Son bölümde ise uygulanan yöntemlerin sonuçları ve geleceğe yönelik önerilerden bahsedilmiştir.

2. KAYNAK ARAŞTIRMASI

Edward W. Forgy, 1965'te k ortalamalar algoritmasını detaylı bir şekilde tanımlamıştır. Bu çalışma, algoritmanın her iterasyonunda veri noktalarını onlara en yakın küme merkezlerine atama ve bu merkezleri yeniden hesaplama süreçlerini açıklar. Bu işlem, küme içi kare uzaklıkların toplamını azaltmayı hedefler. Forgy'nin bu çalışması, algoritmanın etkinliğini ve uygulanabilirliğini artıran önemli iyileştirmeler sunmuştur (Forgy,1965).

J.B. MacQueen, 1967'de "Some Methods for classification and Analysis of Multivariate Observations" makalesinde, k ortalamalar algoritmasını ilk kez tanıtmış ve çok değişkenli gözlemlerin sınıflandırılması ve analizi için bir yöntem olarak önermiştir. MacQueen, veri noktalarını kümelere atayarak ve küme merkezlerini iteratif olarak güncelleyerek optimum kümeleme çözümlerini nasıl elde edilebileceğini açıklamıştır (. MacQueen, 1967).

David Arthur ve Sergei Vassilvitskii (2006), çalışmalarında, k-ortalamalar algoritmasının başlangıç noktalarının dikkatli bir şekilde seçilmesiyle elde edilen avantajları ele almışlardır. Bu yöntem ile algoritmanın hızını ve kararlılığını önemli ölçüde artırarak daha iyi kümeleme sonuçları elde etmişlerdir (Arthur ve Vassilvitskii, 2006).

D. Li ve arkadaşları (2009) k-ortalamalar algoritmasını büyük ölçekli veri tabanına uygulamışlar ve aynı veri tabanını parçalara bölerek uygulaması arasında kalite açısından bir karşılaştırma gerçekleştirmişlerdir (Li ve ark., 2009).

Wang (2011) gürültülü veri tabanının standart halindeyken ve k-ortalamalar algoritmasını gürültülü nesnelere çıkardıktan sonra uygulayarak karşılaştırma yapmıştır (Wang, 2011).

Schölkopf, B., Smola, A. ve Müller, K.-R. (1998) çalışmalarında, veri setlerinin doğrusal olmayan yapılarını analiz etmek için kernel yöntemlerini kullanmışlardır. Yazarlar, K-Ortalamlar algoritmasının bir varyasyonu olan Kernel K-Ortalamlar algoritmasını tanıtmışlardır. Bu yöntem ile, verileri önce bir yüksek boyutlu özellik uzayına dönüştürüp, ardından bu uzayda kümeleme yaparak, orijinal veri uzayında doğrusal olarak ayrılabilir olmayan yapıları etkili bir şekilde ayırt etmeyi sağlamışlardır (Schölkopf, Smola ve Müller,1998).

Mark Girolami (2001) çalışmasında, veri örneklerinin denetimsiz bir şekilde bölümlendirilmesi ve verileri oluşturan potansiyel küme sayısının tahmin edilmesi için

bir yöntem sunmuştur. Çalışma, yüksek boyutlu bir özellik uzayına doğrusal olmayan bir veri dönüşümü gerçekleştirilmesinin, dönüştürülen uzaydaki örüntülerin doğrusal ayrılabilirliğinin olasılığını artırarak ilişkili veri yapısını basitleştirdiğini vurgulamıştır. Çekirdek matrisinin özvektörlerinin, veri içindeki doğal küme sayısını tahmin etme aracı olarak kullanılabilmesi ve ardından verilerin özellik uzayında bölümlendirilmesi için hesaplama açısından basit bir yinelemeli yöntem gösterilmiştir (Girolami, 2001).

Liu ve arkadaşları (2008), Agresif tümörlerin biyolojik ve klinik açıdan önemli olduğu düşünülerek tümör ilerlemesini özetlemek için kernel k ortalamalar metodunu kullanmıştır. Önerilen yöntem farklı tümör ilerleme yollarını keşfedip, daha fazlasına yol açan genleri (biyobelirteçler) de bulmuştur (Liu ve ark., 2008).

Filippone ve arkadaşları (2008), kümelemede örüntü tanıma sorunun çözümüne yönelik yeni fikirler ve yorumlar sunmuşlardır. Kümelemede spektral ve çekirdek yöntemleri için bulanık çekirdek yöntemlerine dikkat edilerek gözden geçirilmiştir. Kernel algoritmasının, klasik bölümlene kümeleme algoritmalarından farklı olarak doğrusal olmayan ayırıcı hiper yüzeyler üretebildiğini ön görmüşlerdir (Filippone ve ark., 2008).

Graves ve Pedrycz (2010), çalışmalarında İyonosfer, Halka ve Wisconsin meme kanseri veri setleri üzerinde yaptıkları çalışmalarda genel olarak, çekirdek tabanlı algoritmalar ile elde edilen sonuçlarda marjinal artışlar gözlemlemiş, çekirdek tabanlı bulanık kümelemede gelecekteki bir iyileştirme, kısmen denetimli bir bulanık kümeleme geliştirilebilir sonucuna ulaşmışlardır (Graves ve Pedrycz, 2010).

Wang ve arkadaşları (2020), doğrusal olmayan kümeleme problemini hedefleyen dağıtılmış P2P ağlarını, çekirdek tabanlı K-ortalama kümeleme algoritması kullanarak yüksek boyutlu uzayda doğrusal olmayan verilerin kümeleme probleminde çözüm getirmeye çalışmışlardır (Wang ve ark., 2020).

Zhang ve Cao (2011), kernel yöntemi ile entegre edilmiş yeni bir karınca tabanlı kümeleme algoritması (ACK) önermiştir. Önerilen algoritma, kernel tabanlı kümelemenin avantajlarını karınca tabanlı kümelemeye dahil etmiştir. 5 sentetik ve 5 gerçek veri seti kullanılarak diğer klasik algoritmalarla yapılan karşılaştırmalarında, ACK yönteminin verimlilik ve kümeleme kalitesi açısından yüksek performans sergilediğini göstermişlerdir (Zhang ve Cao, 2011).

Kuo ve arkadaşları (2014), birçok kümeleme algoritmasında kümelerin sayısını önceden tanımlamayı gerektirdiği için bu zayıflığı gidermek adına yeni bir otomatik kümeleme algoritması önermişlerdir: Arı koloni optimizasyonu ile otomatik kernel

kümeleme (AKC-BCO). AKC-BCO algoritmasının, uygun küme sayısını belirlediğini ve veri noktalarını doğru kümelerle yerleştirdiğini deneysel sonuçlar ile elde etmişlerdir (Kuo ve ark., 2014).

Fan ve arkadaşları (2016), karışık kurbağa sıçrama algoritması (SFLA) temel alan kernel kümeleme SFLA-SAKKM algoritmasını geliştirmişlerdir. SFLA-SAKKM'nin etkinliğini doğrulamak için yapay ve gerçek veri setleri üzerinde deneyler yapmış, deney sonuçları, SFLA-SAKKM'nin farklı veri setleri için optimal küme sayısı ve kernel parametresini bulduğu ve aynı zamanda kümeleme performansını artırdığı sonucunu elde etmişlerdir (Fan ve ark., 2016).

Hancer (2018), çalışmasında evrimsel hesaplama tekniği ile dizayn edilmiş bir çoklu kernel kümeleme algoritması geliştirmeyi amaçlamıştır. Çoklu kernel tabanlı bir amaç fonksiyonu tasarlamış ve bu fonksiyonu, diferansiyel gelişim adı verilen bir evrimsel hesaplama tekniği ile bütünleştirilerek DE-MKFC kümeleme algoritması oluşturmuştur. Gerçekleştirdiği sayısal ve görsel deneyler ile DE-MKFC kümeleme algoritmasının hem mevcut bilinen kümeleme algoritmalarından hem de çoklu kernel kümelemenin ilk örneği olarak görülen MKFC algoritmasından daha üstün sonuçlar sağladığı sonucuna ulaşmıştır (Hancer, 2018).

Nayak ve arkadaşları (2019), Çok Amaçlı Çekirdek Tabanlı Diferansiyel Evrim kümeleme algoritmasını (MOC-KDE) önermişlerdir. Gauss kerneli ile hibritleştirilmiş üç yenilikçi hedef sunmuşlardır. Bu çok amaçlı algoritma, bir kümedeki nesnelerin yakınlığına ve kümeler arası ayrımın yanı sıra, her bir kümenin sıklığını artıran üçüncü bir hedef olarak küme sayısının en aza indirilmesine önem vermiştir (Nayak ve ark., 2019).

Patel ve Vishwamitra (2023), dinamik kernel kümeleme ile bir örümcek maymun optimizasyon algoritması olan DKCSMO önermiştir. İyi kümeleme sonuçları için, örümcek maymun optimizasyon algoritmasının yerel lider aşaması, komşuluk arama stratejisi ile iyileştirilmiştir. Sonuçların kalitesini daha da artırmak için, CS-Index Gauss kernel dağılımı ile modifiye edilmiştir. Yaptıkları deneysel araştırmalarda, önerilen algoritmanın tahmin edilen kümeler, DB, SIL ve ARI ölçümleri açısından daha iyi sonuçlar ürettiği sonucunu elde etmişlerdir (Patel ve Vishwamitra, 2023).

Mirjalili ve Lewis (2016), kambur balinalarının sosyal davranışlarını taklit eden yeni bir doğa ilhamlı meta-sezgisel optimizasyon algoritması olan Balina Optimizasyon Algoritmasını (WOA) önermişlerdir. Algoritmanın, balinaların kabarcık ağı avlama stratejisinden esinlendiğinden bahsetmişlerdir. Önerilen algoritmanın keşif, sömürü,

yerel optimumlardan kaçınma ve yakınsama davranışlarını analiz etmek için 29 matematiksel referans fonksiyon üzerinde kapsamlı bir çalışma yapmışlardır. WOA'nın diğer en son meta-sezgisel yöntemlerle rekabetçi olduğunu bulmuşlardır (Mirjalili ve Lewis, 2016).

Nasiri ve Khiyabani (2018), kümele için Balina Optimizasyon Algoritması çalışmasında bulunmuşlardır. Bu çalışmada algoritmanın, başarılı bir şekilde kümeleme problemlerini çözebileceği gösterilmiştir. Bu algoritmanın, intra-küme mesafe fonksiyonu ve standart sapma açısından yapılan ön hesaplamalı deneyimlerde de başarılı sonuçlar gösterdiğini ifade etmişlerdir (Nasiri ve Khiyabani, 2018).

Singh (2020), veri kümeleme için, Balina Optimizasyon Algoritması tabanlı bir yaklaşım önermiştir. Önerilen yaklaşımın, veri kümeleme probleminin çözümünde bu çalışmada ele alınan diğer tüm yaklaşımlardan daha iyi performans sergilediğini elde etmiştir (Singh,2020).

Literatür genel olarak incelendiğinde, kernel fonksiyonlarının uzaklık hesabından önce kullanılmasının başarımı arttırdığı görülmektedir. Kernel fonksiyonları, verileri daha yüksek boyutlu bir uzaya taşıyarak makine öğrenmesi algoritmalarının verileri daha iyi anlamasını sağlar. Bu yüksek boyutlu dönüşüm, veriler arasındaki karmaşık ilişkilerin daha iyi yakalanmasına olanak tanır, böylece algoritmaların doğrusal olmayan yapıları tanıma kapasitesi artar. Bu sayede, sınıflandırma ve kümeleme gibi işlemlerde daha doğru sonuçlar elde edilebilmektedir. Algoritmalar veri setlerinin iç yapısını daha detaylı olarak işleyebilir.

Literatürdeki araştırmalar, kernel fonksiyonlarının ve Balina Optimizasyon Algoritması (WOA)'nın veri kümeleme ve makine öğrenmesi alanında önemli bir yer tuttuğunu göstermektedir. Kernel fonksiyonları, doğrusal olmayan veri setlerinin daha yüksek boyutlu bir özellik uzayına dönüştürülmesi yoluyla sınıflandırma ve regresyon problemlerinde başarı sağlar. Bu dönüşüm sayesinde, karmaşık yapıdaki veriler doğrusal olarak ayrılabilir hale gelir, böylece algoritmalar daha etkin bir şekilde çalışır. Özellikle, Gauss (RBF), polinom ve sigmoid gibi çeşitli kernel fonksiyonları farklı veri yapılarına uygun esnek çözümler sunar ve bu sayede daha doğru tahminler yapılmasını sağlar. Bu metodoloji, makine öğrenmesinde geniş kapsamlı uygulama alanları bulmuştur. Öte yandan, WOA'nın doğa ilhamlı yaklaşımı, veri kümeleme problemlerini çözmede etkili bir keşif ve sömürü mekanizması sunar. Bu algoritma, verileri gruplandırmada ve en uygun küme merkezlerini belirlemede literatürde yüksek performans gösterdiği raporlanmıştır. Kernel fonksiyonlarının veri dönüşüm kapasitesi

ile WOA'nın optimizasyon yeteneđini entegre ederek, veri kümeleme problemlerinin çözümünde daha etkili sonuçlar elde edilip, edilmeyeceđinin araştırılması bu tez çalışmasının ana konusudur.



3. MATERYAL VE YÖNTEM

3.1.Çalışmada Kullanılan Veri Setleri

Bu çalışmada kullanılan veri kümeleri materyal başlığı altında açıklanmıştır. Önerilen algoritmanın açıklanmasında kullanılmak üzere öncelikle bir sentetik veri seti oluşturulmuştur. Bu veri setinde iki nitelik (Özellik 1 ve Özellik 2) ve iki küme bulunmaktadır. Veri seti, her bir küme merkezinin etrafında normal dağılımlı verilerden oluşmakta olup, toplamda 8 örnek içermektedir. Küme merkezleri (0,0) ve (5,5) olarak belirlenmiştir ve her kümede 4'er örnek bulunmaktadır. Küme merkezlerinin etrafındaki örnekler, normal dağılımlı rastgele veriler olarak üretilmiştir.

Test çalışmasında önerilen algoritmanın başarısını ölçmek ve diğer algoritmalar ile başarısını tartışmak için UCI Makine Öğrenimi Deposundan (<https://archive.ics.uci.edu/>) alınan Balance, Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Original), Dermatology, Ecoli, Glass, Heart, Iris isimli sekiz farklı veri seti kullanılmıştır. Kullanılan veri setlerinin özellikleri Çizelge 3.1 sunulmuştur.

Çizelge 3.1. Çalışmada Kullanılan Veri Setleri ve Özellikleri

Veri Seti	Örnek Sayısı	Nitelik	Küme Sayısı
Balance	625	4	3
Breast Cancer W. Diagnostic	569	30	2
Breast Cancer W. Original	699	9	2
Dermatology	366	34	6
E. Coli	327	7	5
Glass	214	9	6
Heart Disease	303	13	2
Iris	150	4	3

Çizelge 3.1'de örnek sayısı, nitelik ve küme sayısı özellikleri verilen kümeler ayrıntılı özellikleri şu şekildedir.

Balance veri seti, psikolojik deneysel sonuçları modellemek için kullanılmaktadır. Bu veri setinde dört nitelik (sağ ve sol ağırlık ile sağ ve sol uzaklık) ve üç sınıf (sağa yatkın, sola yatkın ve dengeli) bulunmaktadır. Veri seti, dengeli sınıfta 49, sola yatkın sınıfta 288 ve sağa yatkın sınıfta 288 örnek olmak üzere toplamda 625 örnek içermektedir.

Breast Cancer Wisconsin (Diagnostic) veri seti, hastalık bölgesindeki tümörlü alanın özelliklerini sınıflandırmak için kullanılır. Nitelikler arasında tümörlü bölgenin kalınlığı, tümör hücrelerinin şekil ve büyüklük olarak benzerliği, tümörün vücuda yapışma oranı gibi özellikler bulunmaktadır. Veri setinde, tümörün iyi huylu (Benign) veya kötü huylu (Malignant) olup olmadığını belirleyen 30 nitelik ve iki sınıf bulunmaktadır. Benign sınıfında 357, Malignant sınıfında 212 örnek olmak üzere toplam 569 örnek vardır.

Breast Cancer Wisconsin (Original), Breast Cancer Wisconsin Diagnostic veri seti ile benzer özelliklere sahip olan bu veri setinde, dokuz nitelik ve iki sınıf (Benign, Malignant) bulunmaktadır. Veri seti, Benign sınıfında 458 ve Malignant sınıfında 241 örnek olmak üzere toplam 699 örnek içerir. Eksik olan 16 örnek veri setinden çıkarılmıştır.

Dermatology veri seti, cilt hastalıklarının özelliklerini modeller. Ciltte oluşan kızarıklık, sertleşme, kaşıntı, yumuşama, kepeklenme ve çatlama gibi belirtiler nitelik olarak kullanılır. Veri setinde 34 nitelik ve altı sınıf bulunmaktadır. Veri seti, Psoriasis (112 örnek), Seboreic Dermatitis (61 örnek), Lichen Planus (72 örnek), Pityriasis Rosea (49 örnek), Cronic Dermatitis (52 örnek) ve Pityriasis Rubra Pilaris (20 örnek) sınıflarından toplam 366 örnek içermektedir.

E.Coli veri setinde, normalde yedi nitelik, sekiz sınıf ve 336 örnek bulunmaktadır. Ancak, sınıflardan üç tanesi çok az örnek içerdiği için (sırasıyla 2, 2 ve 5 örnek), bu sınıflar çalışma dışında bırakılmıştır. Yapılan değişikliklerden sonra veri seti beş sınıf ve 327 örnekten oluşmaktadır.

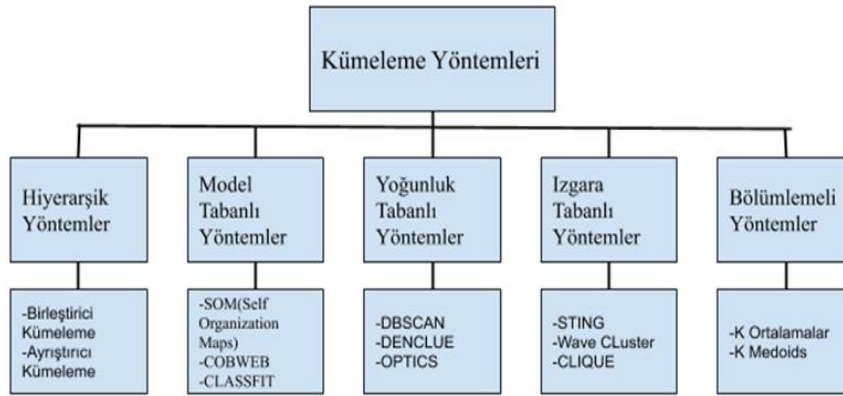
Glass veri seti, farklı cam tiplerini belirlemek amacıyla kullanılan altı sınıfa sahiptir ve dokuz nitelik ile 214 örnek içermektedir. Camın yapısındaki sodyum, magnezyum, alüminyum, silikon, potasyum, baryum, kalsiyum ve demir gibi elementlerin miktarları nitelik olarak kullanılmaktadır. Sınıflar arasında bina camları (float ve non-float), araç camları (float), kaplar, masa eşyaları ve far camları bulunmaktadır.

Heart veri seti, kalp hastalığı teşhisi için kullanılan 75 niteliğe sahiptir. Ancak, genellikle sınıflandırma çalışmalarında bunlardan 14 tanesi (1 tanesi sınıf özelliği olmak üzere) kullanılmaktadır. Bu çalışmada 13 nitelik değerlendirilmiştir. İki sınıf bulunan veri setinde toplamda 303 örnek yer almaktadır. Eksik veri içeren altı kayıt işleme alınmamıştır.

İris veri setinde, iris bitkisinin çanak ve taç yapraklarının uzunluk ve genişlik bilgilerini kullanan üç sınıf bulunmaktadır: Iris Setosa, Iris Versicolour ve Iris Virginica. Her sınıfın 50'şer örneği olmak üzere toplam 150 örnek içermektedir.

3.2. Kümeleme Yöntemleri

Danışmansız, gözetimsiz diye de anılan denetimsiz öğrenmede, modelin denetlenmesini gerektirmez. Bu tür algoritmalar, genellikle denetimli öğrenme yöntemlerine kıyasla daha karmaşık işlemleri başarabilirler. Gözetimsiz teknikler, verilerdeki kategorilere ayırmada faydalı özelliklerin keşfedilmesine olanak sağlar. Veriler etiketsizdir ve en sık örnek verilen türü kümeleme algoritmalarıdır.



Şekil 3.1. Kümeleme Yöntem Grupları ve Örnek Algoritmalar

Kümeleme, veri setindeki bilgilerin benzer özelliklere göre gruplara ayrılması işlemidir. Her bir gruba küme ismi verilir. Bu süreçte benzer özelliklere sahip veriler aynı kümeye yerleştirilirken, farklı kümeler arasındaki benzerlikler mümkün olduğunca düşük tutulmaya çalışılır. Kümeleme yöntemleri, Ar-Ge sorunlarının yanı sıra tıpta hastalıkların teşhisi ve tedavisi, durumların sınıflandırılması, veri bilimi, makine öğrenmesi, örüntü tanıma gibi birçok alanda kullanılır. Kümeleme Yöntem Grupları ve her kümeleme yöntemine ait örnek algoritmalar Şekil 3.1' de gösterilmiştir. Şekilden de görüldüğü üzere kümeleme yöntemleri; hiyerarşik, model tabanlı, yoğunluk tabanlı, ızgara tabanlı ve bölümlemeli olmak üzere beş başlık altında incelenmektedir.

Tez çalışmasında önerilen KWOA algoritması bölümlemeli kümeleme algoritmasına örnektir. Bu yöntem, önceden atanan k sayısı kadar veri kümesini alt gruplara ayırır. Bölümleme temelli kümeleme yöntemlerinin avantajları arasında, algoritmanın kolay uygulanabilirliği ve hesaplama verimliliği yer alır. Büyük veri setleri

üzerinde dahi, bu yöntemler nispeten hızlı çalışır ve karmaşık olmayan modeller ortaya koyar. Ayrıca, sonuçlarının görselleştirilmesi ve yorumlanması da oldukça basittir. Bununla birlikte, bölümlene temelli kümeleme yöntemlerinin dezavantajları da vardır. Örneğin, optimum kümelerin sayısının önceden bilinmesini gerektirir ve bu sayının seçimi sonuçları büyük ölçüde etkileyebilir. Ayrıca, bu yöntemler genellikle küresel şekilli kümeleri tespit etmede etkiliyken, farklı şekillerdeki veya yoğunluğu değişen kümeleri doğru bir şekilde ayırt etmekte zorlanabilirler. Ayrıca, aykırı değerler veya gürültülü veriler bu algoritmaların performansını olumsuz etkileyebilir. Bu sebeple, kullanılan veri setinin doğasına ve yapısal özelliklerine göre, bu yöntemlerin avantaj ve dezavantajlarının dikkatlice değerlendirilmesi gerekir. Küme sayısı konusunda ön bilgi var ise ya da araştırmacı anlamlı olacak küme sayısına karar vermişse bu durumda çok uzun zaman alan hiyerarşik teknikler yerine hiyerarşik olmayan bu teknikler tercih edilir (Yaraş, Akın, ve Şakacı,2011).

3.3. K-Ortalamlar Algoritması

K-Ortalamlar algoritması, gözetimsiz öğrenme kapsamında yer alan ve kümeleme metodolojileri arasında sıklıkla tercih edilen bir yaklaşımdır (Arthur, 2007).

Bu algoritma, denetimsiz bir öğrenme türüdür ve temel amacı, verilen bir N sayıda veri nesnesini, algoritmaya önceden belirlenen K sayıda kümeye ayırmaktır. Bu bölümlene sürecinin esas hedefi, elde edilen kümeler arasındaki benzerlikleri minimize ederken, aynı zamanda her bir kümenin içerisindeki benzerlikleri maksimize etmektir.

K-Ortalamlar algoritması, kullanımı kolay ve etkin bir algoritmadır. 'K', algoritmayı başlatmadan önce belirlenmesi gereken sabit küme sayısını temsil eder. Bu algoritma, iteratif bir yapıya sahip olup, her veri noktasının kümelerin merkezlerine olan uzaklıklarının toplamını minimize etmeye çalışır, böylece karesel hatayı en aza indirgeyerek K kümeye, elemanların optimum şekilde atanmasını sağlar.

Algoritmanın çalışma mekanizması, başlangıçta K adet noktanın rastgele seçilmesi ve bu noktaların her bir kümenin merkezini temsil etmesi şeklinde işler. Daha sonra, kalan noktalar kendilerine en yakın küme merkezlerine atanır. Her atama sonrasında, o kümeye ait veri noktalarının ortalaması alınarak küme merkezleri yeniden hesaplanır ve bu süreç, artık bir değişiklik olmayana kadar devam eder.

K-Ortalamlar algoritması, veri kümeleme sürecini dört temel aşamaya ayırarak sistematik bir yaklaşım sunar:

1. Bu aşamada, kümeleme işlemi için gerekli olan K adet küme merkezi belirlenir. Bu merkezler, başlangıçta rastgele seçilerek algoritmanın başlangıç noktaları olarak kullanılır.

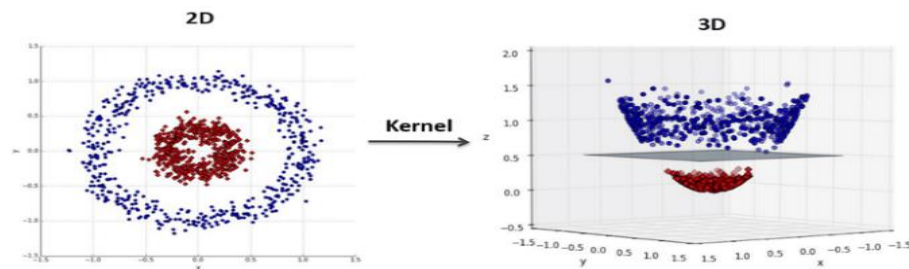
2. Algoritma, veri setindeki her bir veri noktasını, bu merkezlere olan uzaklıklarını hesaplayarak en yakın küme merkezine atar. Bu süreç, tüm veri noktalarının bir kümeye atanmasını içerir ve böylece bir ilk kümeleme gerçekleşir.

3. İlk atama işlemi tamamlandıktan sonra, her bir kümenin merkezi, ilgili kümeye ait veri noktalarının ortalaması alınarak yeniden belirlenir. Bu işlem, küme merkezlerinin veri noktalarının güncel dağılımını daha etkin bir şekilde yansıtmasını sağlamak amacıyla yapılır.

4. Algoritma, kararlı hale ulaşana kadar 2. ve 3. aşamaları sürekli olarak tekrarlar. Bu, herhangi bir veri noktasının küme atamasında veya küme merkezlerinde bir değişiklik olmadığı anlamına gelir. Bu noktada, algoritma optimum kümeleme çözümüne ulaştığı kabul edilir ve işlem sonlandırılır.

3.4. Kernel Fonksiyonları

Kernel fonksiyonu, genellikle iki vektör arasında yer alan iç çarpımını benzerlik ölçütünü belirlemek amacıyla kullanan bir matematiksel işlemdir. Bu tür fonksiyonlar, veriyi düşük boyutlu bir uzaydan yüksek boyutlu bir uzaya taşımadan, sanki yüksek boyutlu uzaydaymış gibi iç çarpım hesabı yapılmasını mümkün kılar. Bu sayede orijinal uzayda lineer olarak ayırt edilemeyen veri noktalarını, daha yüksek boyutlu bir uzayda lineer olarak ayırt edilebilir hale gelir. Makine öğrenimi algoritmalarına çeşitlilik ve uyarlanabilirlik sağlayan kernel fonksiyonları, farklı veri yapılarını etkin bir şekilde işleyebilme kapasitesiyle üstün performans sergileyebilir. Doğrusal olarak ayırtlamayan iki boyutlu bir veri örneği, Kernel fonksiyonundan geçtikten sonra doğrusal olarak işlem görebilmesi mümkün olmaktadır.



Şekil 3.2. Doğrusal ayrılabilir hale gelmesine bir örnek

Kernel fonksiyonları, aynı zamanda çekirdek fonksiyonları olarak da adlandırılır. Bu fonksiyonların genel bir tanımı Denklem (3.1) ile ifade edilmiştir.

$$\begin{aligned} x_i, x_j &\in X \\ K : X \times X &\rightarrow \mathbb{R} \quad K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_H \\ \phi : X &\rightarrow H \end{aligned} \quad (3.1)$$

Burada X veri uzayını, x_i ve x_j bu uzayda yer alan iki veri noktasını temsil ettiği kabul edilsin. K Kernel fonksiyonu, $X \times X$ kartezyen çarpımında \mathbb{R} 'ye (reel sayılara) bir dönüşümü tanımlar. Diğer bir deyişle, x_i ve x_j veri noktaları arasındaki benzerlik veya ilişki ölçü değeri reel sayı olarak Kernel fonksiyonu tarafından döndürülmektedir. ϕ fonksiyonu, veri uzayı X 'ten özellik uzayı H 'ye bir dönüşüm tanımlar. Burada H genellikle daha yüksek boyutlu (veya sonsuz boyutlu) bir Hilbert uzayıdır. ϕ fonksiyonu, veri noktalarını bu yüksek boyutlu uzaya projekte eder. Bu sayede, lineer olmayan ilişkiler lineer hale gelir ve analiz edilmesi kolaylaşır. $\phi(x_i)$ ve $\phi(x_j)$, x_i ve x_j 'nin özellik uzayı H 'daki görüntüleridir. $\langle \cdot, \cdot \rangle_H$ sembolü, bu özellik uzayında (Hilbert uzayında) tanımlanan iç çarpımı (dot product) göstermektedir. Diğer bir deyişle $\langle \phi(x_i), \phi(x_j) \rangle$, x_i ve x_j 'nin özellik uzayındaki görüntülerinin iç çarpımıdır. Bu sayede, veri noktaları arasındaki benzerlik, özellik uzayında hesaplanan iç çarpım ile belirlenir.

Birden fazla kernel fonksiyonu çeşidi bulunmaktadır. Bu fonksiyonlar arasında doğrusal kernel, doğrudan örnek vektörler arasındaki çarpımı kullanarak doğrusal ilişkileri yakalar. Polinomal kernel, özelliklerin yüksek dereceli polinomlarını dikkate alır. Radyal Tabanlı Fonksiyon (RBF) ya da Gaussian kernel, iki örnek arasındaki mesafeye bağlı olarak benzerlik ölçer. Kernel fonksiyonunun matematiksel olarak anlaşılmasını sağlamak için, sayısal bir örnek verilmiştir. x ve y olmak üzere iki veri noktası verildiği varsayalım:

$$x_i = (5,6,7) \text{ ve } x_j = (8,9,10)$$

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

$$\phi(x) = (x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_3x_1, x_3x_2, x_3^2)$$

$$\phi(y) = (y_1^2, y_1y_2, y_1y_3, y_2^2, y_2y_3, y_3^2, y_3y_1, y_3y_2, y_3^2)$$

işlemleri gerçekleştirilsin. Devamında iki vektörün iç çarpımı, her bir vektörün karşılık gelen elemanlarının çarpımlarının toplamı ile hesaplınsın:

$$\phi(5,6,7)=(25,30,35,30,36,42,35,42,49)$$

$$\phi(8,9,10)=(64,72,80,72,81,90,80,90,100)$$

$$\phi(x) * \phi(y) = \phi(5,6,7) * \phi(8,9,10) = (1600 + 2160 + 2800 + 2160 + 2916 + 3780 + 2800 + 3780 + 4900) = 26.896 \text{ şeklinde hesaplanır.}$$

Aynı işlem, Kernel Fonksiyonu kullanılarak gerçekleştirilirse, tek bir fonksiyon ve çok daha az bir maliyet ile aynı sonuç elde edilmektedir.

$$K(x,y) = (5*8 + 6*9 + 7*10)^2 = (40 + 54 + 70)^2 = 164 * 164 = 26.896$$

3.4.1. Doğrusal Kernel Fonksiyonu

Doğrusal kernel fonksiyonu, iki vektörün iç çarpımını hesaplayarak bu vektörleri yüksek boyutlu bir uzayda temsil etmeyi ve böylece veri noktalarını doğrusal olarak ayrılabilir bir hale getirmeyi amaçlar. Bu fonksiyon, Denklem (3.2) gösterilmiştir.

$$k(x,y) = x^T \cdot y + c \quad (3.2)$$

Bu denklemde, x ve y, veri noktalarını temsil eden vektörlerdir. x^T vektörlerin transpozunu ifade eder. c bir sabittir ve genellikle 0 ile 1 arasında bir değer alır. Doğrusal kernel, en çok kullanılan kernel fonksiyonlarından biridir ve özellikle yüksek sayıda özelliğe sahip veri kümelerinde tercih edilir. Metin sınıflandırma gibi, her bir harfin veya kelimenin ayrı bir özellik olarak ele alındığı durumlarda, doğrusal kernelin etkinliği öne çıkar. Çünkü bu tür veri setlerinde, verilerin yüksek boyutlu uzaydaki lineer ayrılabilirliği, modelin başarımını önemli ölçüde artırabilir.

3.4.2. Polinomal Kernel Fonksiyonu

Polinomal kernel fonksiyonu, iki vektör arasındaki iç çarpımı artı bir sabit kuvvetine yükseltir. Bu fonksiyonun formülü Denklem (3.3) ile aşağıdaki gibi ifade edilir:

$$k(x,y) = (x^T \cdot y + c)^d \quad (3.3)$$

Burada, x ve y , veri noktalarını temsil eden vektörlerdir. x^T vektörlerin transpozunu ifade eder. c bir sabittir ve genellikle 0 ile 1 arasında bir değer olarak polinomun asimptotik davranışını kontrol eder. d , polinomun derecesini belirtir ve bu, veri noktalarını daha yüksek boyutlu bir uzaya taşıyarak verilerin ayrılabilirliğini sağlayan faktördür. Polinomal kernel, özellikle veri setinin doğrusal olmayan özelliklerini yakalamak ve modelin karmaşık ilişkileri öğrenmesine olanak tanımak için kullanılır.

3.4.3. Radyal Tabanlı Kernel Fonksiyonu

Radyal Taban Fonksiyonu (RBF) Kernel fonksiyonu, iki vektör arasındaki benzerliği, bu vektörler arasındaki uzaklığın bir fonksiyonu olarak ölçer. RBF, özellikle iki örnek arasındaki uzaklık büyüdükçe benzerliğin azalmasını sağlayan bir yöntemdir. Bu özellik, RBF kernelini, veri noktalarını çok boyutlu bir uzayda temsil ederken, aynı zamanda bu noktalar arasındaki yerel yapıyı koruyabilen güçlü bir araç haline getirir. RBF kernel fonksiyonunun genel formülü Denklem (3.4) ile verilir.

$$k(x, y) = \exp\left(-\gamma \|x - y\|^2\right) \quad (3.4)$$

$$\gamma = \frac{1}{2\sigma^2}$$

Burada, x ve y , veri noktalarını temsil eden vektörlerdir. γ , genellikle 0 ile 1 arasında bir değer alan, RBF fonksiyonunun genişlik veya düzlüğünü kontrol eden serbest bir parametredir. $\|x - y\|^2$ ise x ve y vektörleri arasındaki Öklid mesafesinin karesidir.

Bu tez çalışmasında önerilen KWOA kümeleme algoritmasında da iki örnek arasındaki uzaklık büyüdükçe benzerliğin azalmasını sağladığı için RBF Kernel fonksiyonu kullanılmıştır. RBF'nin hesaplanmasını sayısal bir örnek üzerinde adım adım açıklanmıştır. $X_1(0,0)$, $X_2(4,4)$, $X_3(-4,4)$, $X_4(-4,-4)$, $X_5(4,-4)$ iki boyutlu bir uzayda, koordinatları verilen beş farklı nokta olsun ve $\sigma=4$ olarak seçilsin. Bu durumda $K(X_1, X_2)$ aşağıdaki gibi hesaplanır.

$$\|x_1 - x_2\|^2 = (0 - 4)^2 + (0 - 4)^2 = 32$$

$$K(X_1, X_2) = e^{-32/32} = e^{-1}$$

3.5. Kernel K Ortalamalar

Kernel K Ortalamalar algoritması bölümlenmeli bir kümeleme algoritmasıdır. Algoritmanın amacı, veri setinde bulunan noktaların kernel fonksiyonları ile çok boyutlu uzayda kümeleme işleminin yapılmasıdır (Schölkopf, Smola ve Müller,1998). Kernel K Ortalamalar algoritmasının sözde kodu çizelge 3.2’de verilmiştir.

Çizelge 3.2. Kernel K-Ortalamalar Sözde Kodu

<p>1.Adım : Rastgele k adet küme merkezi seçilir.</p> <p>2.Adım: $n=1, \dots, N$ e kadar her bir X_n noktası için ve her C_i kümesi için, $\ \Phi(X_n) - \mu_i \$ Kernel fonksiyonu kullanılarak hesaplanır. (μ_i, C_i kümesinin merkezini temsil etmektedir, $i=1 \dots k$)</p> $\text{Maliyet}(x_i, \mu_j) = K(x_i, x_i) - 2 \sum_{x_l \in C_j} \frac{K(x_i, x_l)}{ C_j } + \sum_{x_l, x_m \in C_j} \frac{K(x_l, x_m)}{ C_j ^2}$ <p>2.Adım : Her X_n en yakın olan C_i ‘ye atanır.</p> <p>3.Adım : Küme merkezleri, atanan veri noktalarına göre yeniden hesaplanır. Yeni küme merkezleri, atanan noktaların özellik uzayındaki ortalaması olarak güncellenir.</p> <p>4.Adım : Kümeler değişiyorsa Adım 1’e gidilir, değişmiyorsa sonuç küme merkezleri μ_i’ler çıktı olarak verilir.</p>
--

3.6. Kümeleme Analizinde Kullanılan Küme Doğrulama Yöntemleri

Kümeleme analizinin sonuçlarını değerlendirirken, veri türü ve kullanılan kümeleme yöntemine bağlı olarak çeşitli metrikler kullanılır. Bu çalışmada önerilen KWOA algoritmasının performansını değerlendirmek ve diğer kümeleme algoritmalarıyla karşılaştırmak amacıyla, Rand İndeks (RI-Rand Index) ve Siluet Küme Doğrulama İndeksi (SCVI - Silhouette Cluster Validation Index) ölçütleri kullanılmıştır. Rand İndeks, kümelemenin doğruluğunu ölçen bir istatistikken; ve Siluet Küme Doğrulama İndeksi, küme içi benzerlik ve kümeler arası farklılıkları temel olarak kümelemenin etkinliğini değerlendirir.

3.6.1. Rand İndeks Küme Doğrulama Değeri

Rand İndeks yöntemi, bir veri setindeki elemanların nasıl kümelenmiş olduğunu analiz ederken kullanılır. Veri setindeki eleman sayısı N olarak verilmiş olsun ve bu elemanlar k farklı kümeye ayrılmış olsunlar. Kümelemenin gerçek yapısını temsil eden bilgiler "Gerçek" olarak, algoritma tarafından oluşturulan küme yapıları ise "Tahmini" olarak adlandırılınsın. Veri setindeki her bir elemanın, kendisi hariç geriye kalan diğer elemanlar ile ikili incelemeye tabi tutulur. Her bir çift için, “Gerçek” ve “Tahmini”

küme yapılarında yer alma durumlarına göre dört farklı ilişki durumu tanımlanır. Bu ilişkiler sırası ile a,b,c ve d olarak isimlendirilir.

a: İki eleman hem "Tahmini" hem de "Gerçek" durumda aynı kümeye ait ise,

b: İki eleman "Tahmini" durumda aynı kümeye aitken, "Gerçek" durumda farklı kümelerde ise,

c: İki eleman "Tahmini" durumda farklı kümelerdeyken, "Gerçek" durumda aynı kümeye ait ise,

d: İki eleman hem "Tahmini" hem de "Gerçek" durumda farklı kümelerde ise.

Tüm gözlem sayılarının toplamı M ise, $M=a+b+c+d$ şeklinde hesaplanır. Rand İndeks, bu değerler kullanılarak $RI = \frac{a+d}{M}$ formülü ile hesaplanır.

RI değeri 0 ile 1 arasında bir değer alır. Rand İndeks değeri 1'e yaklaştıkça, "Tahmini" ve "Gerçek" durumlar arasındaki benzerlik artar, bu da kümelemenin başarılı olduğunu gösterir. Bu yöntem, kümeleme sonuçlarının objektif bir şekilde değerlendirilmesine olanak tanır.

3.6.2. Siluet Küme Doğrulama Değeri

Bu çalışmada, kümeleme performansını değerlendirmek için kullanılan diğer bir yöntem ise Siluet Küme Doğrulama İndeksidir (Silhouette Cluster Validation Index - SCVI). Matematiksel hesaplaması ve açıklamaları şekil 3.3'de verilmiştir.

$$SD(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

$$b(i) = \min_{C \neq A} d(i, C), C=1, \dots, k$$

Şekil 3.3. Siluet Küme İndeksi

Burada, k: Sistemdeki küme sayısını, A: i noktasının ait olduğu kümeyi temsil eden numarayı; a(i): i noktasının, kendi kümesi içerisindeki diğer noktalara olan uzaklıklarının ortalamasını; d(i,C) : i noktasının, C numaralı kümede bulunan noktalara olan uzaklıklarının ortalamasını göstermektedir. -1 ile +1 arasında değer alır. +1'e ne kadar yakınsa o kadar iyi kümelenecek demektir.

3.7. Balina Optimizasyon Algoritması

Balina Optimizasyon Algoritması (WOA - Whale Optimization Algorithm), kambur balinaların etkileyici ve sofistike avlanma stratejilerinden esinlenilerek geliştirilmiş bir meta-sezgisel optimizasyon yöntemidir.

Seyedali Mirjalili ve Andrew Lewis tarafından 2016 yılında önerilen Balina Optimizasyon Algoritması, kambur balinaların bu özgün avlanma tekniklerinden yararlanarak, optimizasyon problemlerine yönelik yeni ve etkili bir yaklaşım sunmaktadır (Mirjalili ve Lewis, 2016). Balinaların çevreleriyle etkileşimleri ve avlanma kapasiteleri, algoritmik süreçleri şekillendirmede bir metafor olarak kullanılmaktadır. Bu algoritma, özellikle karmaşık optimizasyon sorunlarının çözümünde, kril ve küçük balıkları avlamayı tercih eden kambur balinaların avlanma metotlarını matematiksel bir modelle yansıtmaktadır.

Kambur balinalar su altında nefes vererek hava kabarcıkları üretir ve bu kabarcıkları kullanarak avlarını merkezi bir noktada toplar. Yüzeye doğru yükselirken soluk alıp vermeye ve kabarcık oluşturmaya devam ederler. Bu süreç içerisinde, avlarına yaklaştıkça hava kabarcığı çemberini daraltırlar, böylece avın hareket alanını kısıtlar ve yakalama olasılığını maksimize ederler. Bu, balinaların avlarını etkili bir şekilde tuzağa düşürme ve hareketsiz hale getirme tekniklerinden biridir. Bu sürecin grafiksel gösterimi şekil 3.4’de verilmiştir.



Şekil 3.4. Kambur balinaların kabarcık ağı hareketi ile avlanma yöntemi (Mirjalili ve Lewis, 2016)

Kambur balinaların bu benzersiz beslenme davranışı, avı etkisiz hale getirme ve onları sürpriz bir saldırıyla yakalama kabiliyetlerini gösterir. Bu, algoritmik bağlamda, belirli bir hedefe odaklanma ve bu hedefe ulaşmak için stratejik hamleler yapma yeteneğiyle ilişkilendirilir. Balinaların beyin yapısı ve sosyal davranışlarının karmaşıklığı, algoritmada kullanılan stratejileri geliştirme açısından ek bir ilham kaynağı oluşturur. İnsan beyin yapısına benzer hücrelere sahip olmaları ve bu hücrelerin insanlarıkiyle benzer işlevlere sahip olması, balinaların öğrenme ve problem çözme yeteneklerini simgeler. Balina Optimizasyon Algoritması, bu biyolojik temellerden yola

çıkarak, verilen bir optimizasyon problemine en uygun çözümü arayan bir algoritma olarak karşımıza çıkmaktadır.

3.7.1. Balina Optimizasyon Algoritması Matematiksel Modeli

WOA'nın matematiksel modelinin temelini, balinaların içgüdüsel stratejilerinin üç aşamalı süreci oluşturur: avın çevresini sarma, avın bulunduğu yere doğru hareket etme ve av arama. Bu modelleme, gerçek hayattaki balina davranışlarından alınan soyutlamalarla optimizasyon problemlerine uygulanır ve bu problemlerin çözümünde kullanılır. Algoritmanın temel amacı, bir yandan global arama uzayını keşfederken diğer yandan yerel optimum çözümlere yönelik yoğun bir araştırma gerçekleştirmektir.

3.7.1.1. Avın Etrafını Kuşatma

Kambur balinaların avlanma yöntemleri, WOA için bir ilham kaynağıdır. Balinalar, avlarının yerini tahmin edebilir ve onları hava kabarcıkları ile sarmalayarak avlayabilirler. Balinaların avlarının yerini tahmin etme yeteneği, en uygun çözümün bulunması sürecine benzetilir. Optimizasyon problemlerinde, genellikle belirsiz olan en uygun çözüm, algoritma tarafından arama uzayında elde edilen en iyi sonuca veya ona yakın bir noktaya dayanarak belirlenir. WOA'da çözüm uzayını keşfederken, arama ajanları kullanır. Bu süreçte, en iyi arama ajanı keşfedildikten sonra, diğer arama ajanlarının pozisyonları en iyi arama ajanına göre güncellenir.

Avın etrafını kuşatma davranışının matematiksel modeli denklem (3.5) ve (3.6)'da gösterilmiştir.

$$\vec{D}' = |\vec{C} \cdot \vec{X}^*_{rand}(t) - \vec{X}(t)| \quad (3.5)$$

$$\vec{X}(t+1) = |\vec{X}^*_{rand}(t) - \vec{A} \cdot \vec{D}'| \quad (3.6)$$

Denklemlerde yer alan t iterasyon numarasını, \vec{A} ve \vec{C} yakınsama vektörlerini, \vec{X}^* en iyi konum vektörünü, \vec{X} arama ajanının konum vektörünü temsil eder. \vec{A} ve \vec{C} yakınsama vektörlerinin hesaplanması için kullanılan denklemler (3.7) ve (3.8)'de gösterilmiştir.

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \quad (3.7)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (3.8)$$

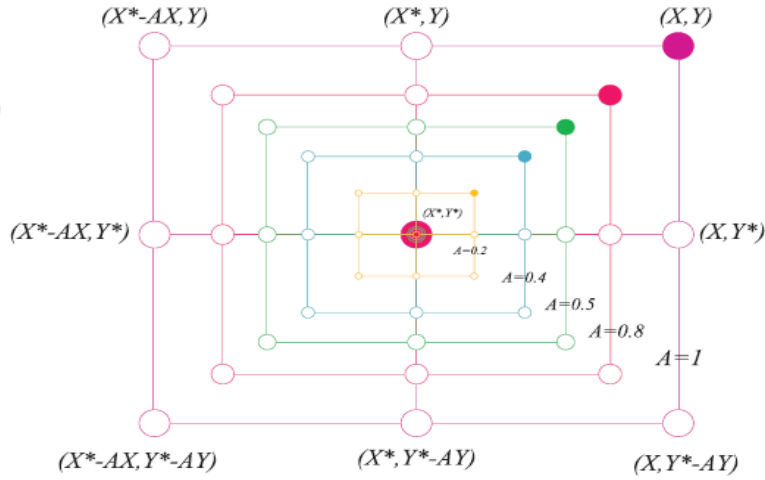
Denklemlerde; \vec{r} $[0,1]$ aralığında değer alan rastgele bir vektörü temsil eder. \vec{a} iterasyonlar boyunca 2'den 0'a doğru lineer olarak azalan bir vektörü gösterir. $a = 2 - \text{mevcut_iterasyon_numarası} * (2 / \text{maksimumIterasyonSayısı})$ formülü ile güncellenir.

3.7.1.2. Kabarcık Ağı Hareketi ile Saldırma

Kambur balinaların avlarına yönelik kabarcık ağı taktiği, algoritmanın benzetiminde iki temel mekanizmayla yansıtılır. Bunlar, avın etrafını daralan dairesel hareketler ile kuşatma ve avın konumuna bağlı olarak spiaral hareket oluşturma yöntemleridir.

3.7.1.2.1. Daralan Dairesel Hareketler ile Kuşatma

Bu azalma, \vec{A} 'nın $[-a, a]$ aralığında değişken değerler alabilmesini sağlar. \vec{a} değeri azaldıkça çember de daralır.

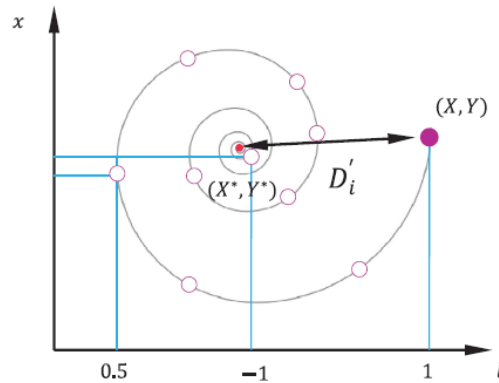


Şekil 3.5. Daralan Dairesel Hareketler ile Kuşatma (Mirjalili ve Lewis, 2016)

Bir arama ajanının yeni konumu, ajanın orijinal konumu ile mevcut en iyi ajanın konumu arasında herhangi bir yerde tanımlanabilir. Şekil 3.5, 2D uzayda $0 \leq A \leq 1$ aralığında, (X, Y) noktasından (X^*, Y^*) noktasına doğru ulaşılacak olası pozisyonları gösterir (Mirjalili ve Lewis, 2016). Bu, ajanın hareket kabiliyetini ve arama uzayındaki çeşitliliğini görselleştirir ve algoritmanın etkinliğini artıran bir faktördür.

3.7.1.2.2. Spiral Hareket Oluşturma

Bir arama ajanının spiral hareketi ve en iyi ajanın konumu Şekil 3.6'da, gösterilmektedir.



Şekil 3.6. Spiral Hareket Oluşturma (Mirjalili ve Lewis, 2016)

Bu mekanizmada, balina ile avı arasındaki mesafe önce hesaplanır, ardından bu bilgi, balinanın konumunu hedef avın konumuna doğru sarmal bir şekilde günceller.

Burada, hedef konum (en iyi ajan) ile arama ajanı arasındaki mesafe hesaplanarak, kambur balinaların sarmal şekilli hareketlerini simüle eden bir sarmal konum güncelleme denklemi oluşturulmuş ve matematiksel formülü denklem (3.9)'da verilmiştir.

$$\vec{X}(t + 1) = \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (3.9)$$

Denklem (3.9) 'da yer alan b logaritmik spiral sabiti, l $[-1,1]$ aralığında rastgele bir sayıyı temsil eder.

\vec{D}^l değeri, $\vec{D} = \vec{X}^*(t) - \vec{X}(t)$ matematiksel formülü ile ifade edilir. Bu ifade arama ajanı ile bilinen en iyi çözüm (av) arasındaki mesafeyi vermektedir.

Kambur balinaların avlarına yaklaşırken gösterdikleri hem spiral hem de dairesel kuşatma hareketleri, Balina Optimizasyon Algoritmasında modellenmiştir. Bu algoritma, balinaların su yüzeyine doğru spiral şekilde yükselirken aynı zamanda avlarını kuşatan hareketlerini taklit eder. Algoritmanın bu iki hareket türünden hangisini seçeceği, Denklem 3.10'da belirtildiği üzere belirlenir. Denklem 3.10'da yer alan p değeri $[0,1]$ aralığında rastgele bir değerdir.

$$\vec{X}(t + 1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & (\text{Denklem 3.6}) & p < 0,5 \\ \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & (\text{Denklem 3.9}) & p \geq 0,5 \end{cases} \quad (3.10)$$

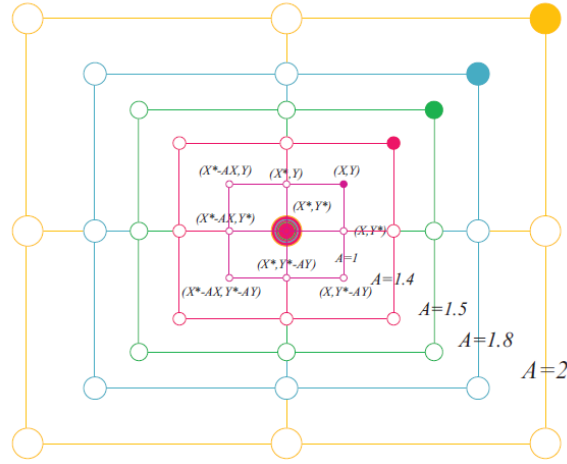
3.7.1.3. Av Arama

Kambur balinaların avlanma stratejileri içinde Kabarcık-Ağ metodu ve rastgele av arama yöntemleri bulunur. Balina Optimizasyon Algoritması'nda, global arama yapısını simüle etmek için rastgele av arama mekanizması kullanılır. Bu durumda, arama ajanının konumu güncellenirken, şimdiye kadar bulunan en iyi ajan yerine rastgele bir ajan seçilir. \vec{A} vektörünün 1'den büyük ya da -1'den küçük değerler alması, ajanların arama uzayında geniş bir alanı keşfetmesini sağlar. Bu durumda kullanılan matematiksel denklemler ise Denklem (3.11) ve (3.12)' de ifade edilmiştir.

$$\vec{D}' = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (3.11)$$

$$\vec{X}(t + 1) = |\vec{X}_{rand} - \vec{A} \cdot \vec{D}'| \quad (3.12)$$

\vec{X}_{rand} , algoritmanın içerisinde rastgele belirlenen bir arama ajanını temsil etmektedir.



Şekil 3.7. WOA'da uygulanan global arama (Mirjalili ve Lewis, 2016)

Şekil 3.7'de bir arama uzayı ve bu uzayda yer alan çeşitli arama ajanları gösterilmiştir.

Arama ajanları, algoritmanın potansiyel çözümleri keşfetmek için kullandığı temsilcilerdir ve farklı renklerde noktalar şeklinde gösterilmiştir. En iyi bulunan çözümü temsil eden arama ajanı X^* olarak belirtilmiş ve bu özel durumda merkezde kırmızı bir daire ile işaretlenmiştir.

X^* etrafında yer alan diğer ajanların pozisyonları, X^* arama ajanına göre değişik mesafe ve yönlerde gösterilmiştir. Bu mesafeler, A değerleri ile etiketlenmiş; A değeri,

ajanların X^* 'e göre olan konumlarını güncellemek için kullanılan bir katsayıdır. Renkli çerçeveler (pembe, yeşil ve mavi) arama ajanlarının X^* etrafındaki farklı yarıçaplar içinde nasıl yer aldıklarını göstermektedir ve bu, WOA'nın arama mekanizmasının çeşitli yarıçaplar ve yönlerde nasıl çalıştığını temsil etmektedir.

Algoritma, bu ajanların yerlerini, belirli bir iterasyon için en iyi çözüm olarak değerlendirilen X^* 'e göre güncelleyerek çözüm uzayını keşfeder. Ajanların bu şekilde hareket ettirilmesi, global (genel) arama yapısının bir parçasıdır ve algoritmanın potansiyel olarak daha iyi çözümlere yönelmesini sağlar. Amaç, en iyi arama ajanının pozisyonunu sürekli güncelleyerek en uygun çözüme yaklaşımdır. Yukarıda açıklanan tüm süreçlerin yer aldığı Balina Optimizasyon algoritmasının sözde kodu Çizelge 3.3'de verilmiştir.

Çizelge 3.3. Balina Optimizasyon Algoritmasının Sözde Kodu

```

Balina popülasyonunu ayarla  $X_i (i = 1, 2, \dots, n)$ 
Her bir arama ajanı için uygunluk değerini hesapla
 $X^*$  = Bilinen en iyi arama ajanı
while ( t < maksimum iterasyon sayısı)
  for (her bir arama ajanı için)
    Güncelle a, A, C, l, p
    if (p < 0.5)
      if (|A| < 1)
        Mevcut arama ajanının konumunu Denklem (3.5)'e göre güncelle
      else if (|A| >= 1)
        Rastgele bir arama ajanı seç ( $X_{rand}$ )
        Mevcut arama ajanının konumunu Denklem (3.12)'ye göre güncelle
      end if
    else if ( p >= 0.5)
      Mevcut arama ajanının konumunu Denklem (3.9)'a göre güncelle
    end if
  end for
  Arama ajanı kısıt dışına çıkıyorsa sınır değerini ver
  Her bir arama ajanının uygunluk değerini hesapla
  Daha iyi bir çözüm varsa  $X^*$  değerini güncelle.
  t=t+1
end while
Sonuç  $X^*$  (En iyi ajan) değerini döndür

```

4. ÇEKİRDEK FONKSİYONU KULLANAN BALİNA OPTİMİZASYON ALGORİTMASI TABANLI YENİ BİR KÜMELEME ALGORİTMASI-KWOA

Bu çalışmada, Balina Optimizasyon Algoritması (WOA) ve RBF Kernel Fonksiyonu, veri kümeleme problem çözümü için birleştirilmiştir. WOA, doğal davranışlardan esinlenen ve çözüm adaylarının (balinaların) global optimuma iteratif bir yaklaşımla hareket etmelerini modelleyen bir algoritmadır. Kernel Dönüşümü ise, veri setlerini yüksek boyutlu bir uzayda çalışma imkânı vererek karmaşık veri yapılarının analizini kolaylaştırır. Bu çalışmada, "Kernel Çekirdek fonksiyonu kullanan Balina Optimizasyon Algoritması Tabanlı Yeni bir Kümeleme" yöntemi, kısaca KWOA olarak adlandırılmış, WOA'nın pozisyon güncelleme mekanizmaları, RBF kernel dönüşümleri ile entegre edilmiştir. Bu entegrasyon ile algoritmanın veri kümeleme performansını ve analiz kapasitesini önemli ölçüde artırması hedeflenmektedir. KWOA algoritması sonucunda elde edilecek çözüm küme merkezleridir. Balina optimizasyon algoritmasındaki popülasyonu oluşturan her bir ajan (balina) olası bir çözümü temsil eder, diğer bir deyişle veri setine en uygun küme merkezlerini temsil etmektedir. Popülasyondaki her bir ajan için çözüm uzayında $[0,1]$ arasında rasgele üretilen küme merkezleri ile algoritma başlar. $[0,1]$ arasında üretilmesinin sebebi, verilerin kümeleme işleminde önce normalize edilmiş olmasıdır. Rasgele üretilen küme merkezleri, kümeleri temsil etmektedir. Önerilen KWOA kümeleme algoritmasına ait işlem adımları bölümün devamında açıklanmıştır.

4.1. Veri Kümelerinin Hazırlanması

KWOA, veri noktalarını yüksek boyutlu özellik uzayında etkin bir şekilde modellemekte ve WOA'yı daha karmaşık veri yapılarında etkili kümeleme yapabilen bir araç haline getirmektedir.

Bu çalışmada, algoritmaların başarılarını ölçmek ve test etmek amacıyla gerçek veri setlerine başvurulmuştur. Seçilen veri setleri, UCI Makine Öğrenimi Deposu'ndan elde edilmiştir ve çalışma kapsamında Balance, Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Original), Dermatology, Ecoli, Glass, Heart ve Iris olmak üzere çeşitli alanlardan sekiz veri seti kullanılmıştır. Veri setlerindeki tüm özellikler kullanılarak kapsamlı bir analiz gerçekleştirilmiştir. Bu süreçte, veri setlerindeki özellikler Min-Max normalizasyon tekniği ile ölçeklendirilmiştir.

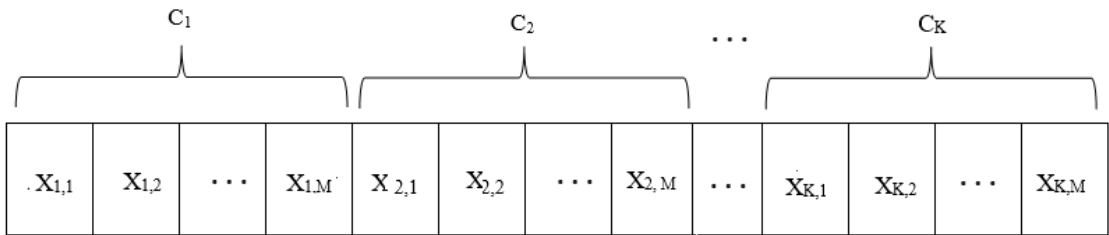
Min-Max normalizasyonu, veri setindeki sayısal özelliklerin değerlerini belirli bir aralığa ölçeklendirmek için kullanılan bir yöntemdir. Bu çalışmada, Min-Max normalizasyon işlemi, her bir özelliğin değerlerini 0 ile 1 arasında bir aralığa dönüştürmek için uygulanmıştır. İşlem, her özellik için aşağıdaki formülle gerçekleştirilir:

$$Yeni\ Değer = \frac{(Orijinal\ Değer - Min\ Değer)}{(Max\ Değer - Min\ Değer)} \quad (4.1)$$

Bu formülde, "Orijinal Değer" özelliğin orijinal sayısal değerini, "Min Değer" ve "Max Değer" ise sırasıyla bu özelliğin veri setindeki minimum ve maksimum değerlerini temsil eder. Sonuç olarak, tüm özellikler aynı ölçek üzerinde değerlendirilir, böylece algoritmaların performansı özelliklerin farklı değer aralıklarından etkilenmez. Min-Max normalizasyonu, özellikle farklı birimlerde veya çok farklı değer aralıklarına sahip özellikleri içeren veri setlerinde algoritmaların daha adil ve tutarlı bir şekilde karşılaştırılmasını sağlar. Bu yöntem, makine öğrenimi modellerinin eğitim sürecini de iyileştirir, çünkü model parametreleri daha hızlı ve etkili bir şekilde ayarlanabilir.

4.2. Arama Ajanlarının Kodlanması ve Başlangıç Popülasyonunun Oluşturulması

Balina Optimizasyon algoritmasında, metaforik bir yaklaşımla ifade edilen balina (whale) terimi, algoritmik bir popülasyon içindeki çözüm adaylarını tanımlamak için kullanılır. Balina Optimizasyon Algoritmasında, her bir balina aslında bir arama ajanıdır ve olası bir çözümü temsil eder. Kümeleme problemleri için olası çözüm, en uygun küme merkezlerinin bulunmasıdır. Bu sebeple küme merkezi adayları arama ajanlarında tutulmuştur. Her bir kaydı M tane nitelik içeren ve K adet kümeye ayrılacak olan DS isimli bir veri seti için bir arama ajanının genel kodlama yapısı Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Popülasyondaki bir arama ajanının kodlanması

Burada;

$C_i = i.$ Kümeyi ,

$X_{i,j} =$ Arama ajanının $i.$ küme merkezinin $j.$ niteliği için konum bilgisini,

($i = 1,2, \dots, K$ ve $j = 1,2, \dots, M$) ,

$K =$ Küme Sayısını ve

$M =$ Nitelik Sayısını gösterir.

Popülasyonda yer alan her bir arama ajanının boyutu denklem 4.2 kullanılarak hesaplanır.

$$\text{Arama_Ajani}_{\text{boyut}} = K * M \quad (4.2)$$

Kodlama yapısı şekil 4.1’de verilen bir arama ajanının her bir konum bilgisi için çözüm uzayında [0,1] arasında rasgele üretilen değerler ile algoritma başlar. [0,1] arasında üretilmesinin sebebi, verilerin kümeleme işleminden önce normalize edilmiş olmasıdır. Her bir küme merkezi M adet konum bilgisi ile temsil edilir. İlk M konum, birinci küme merkezini, ikinci M konum ikinci küme merkezini ve son M konum $K.$ Küme merkezini gösterir. Bu çalışmada esas amaç, veri setine uygun K adet küme merkezinin bulunmasıdır.

Başlangıç popülasyonu, çözüm uzayındaki çeşitli noktaları temsil eden ve muhtemel küme merkezlerine karşılık gelen arama ajanlarından oluşur. Başlangıç Popülasyonu oluşturmak için ilk olarak, popülasyon büyüklüğü belirlenmiştir. Daha sonra, popülasyon büyüklüğü kadar arama ajanı Şekil 4.1’de verilen kodlama yapısına uygun olacak şekilde üretilmiştir. Bu işlem sonucunda balina sayısı kadar satır ve $\text{Arama_Ajani}_{\text{boyut}}$ kadar sütundan oluşan bir matris elde edilir ve başlangıç popülasyonu olarak kullanılır. Bu şekilde, her bir balina için $K \times M$ boyutunda bir vektör elde edilir. Bu vektörler, balinaların konumlarını temsil ederler.

Sonuç olarak, bu yöntemle, algoritmanın optimizasyon sürecine başlamadan önce çözüm alanının temel yapısını yansıtan bir başlangıç popülasyonu elde edilmiştir. Bu popülasyon, algoritmanın arama sürecini daha verimli hale getirmekte ve potansiyel çözüm uzayında daha geniş bir kapsam sağlamaktadır.

Her bir balina, optimizasyon probleminin potansiyel bir çözüm adayı olarak düşünülebilir ve algoritmanın ilerleyen aşamalarında bu adaylar arasından en uygun olanı seçilecektir. Bu süreç, doğadan esinlenmiş optimizasyon yaklaşımları olan genetik algoritmalar ve parçacık sürü optimizasyonu gibi yöntemlerde sıklıkla uygulanan bir

başlatma stratejisini temsil eder. Bu algoritmalar, genellikle geniş bir arama uzayında global optimuma yakınsama hedefiyle iteratif bir şekilde çalışır.

Örneğin; $K=3$ (küme sayısı), $M=4$ (veri boyutu), $BalinaSayisi = 10$ (popülasyon boyutu) için başlangıç popülasyonu, 10 satır ve 12 sütundan oluşur.

4.3. Uygunluk Fonksiyonu

Her bir bireyin çözüm kalitesinin değerlendirilmesi, balina optimizasyon algoritmasının temel bir unsuru olarak öne çıkar. Bölüm 4.2’te belirtildiği üzere, popülasyonda var olan her bir birey, farklı bir balina konumunu simgelemektedir. Dolayısıyla, eğer popülasyonda 20 birey bulunuyorsa, bu 20 farklı balina yapısının her biri, incelenen problem için bir çözüm niteliğindedir. Bu çözüm yapısının etkinliği ve kalitesi, tahmin edilen değerlerin ne derece doğru olduğu üzerinden ölçülebilir. Bu değerlendirme, algoritmanın genel başarımını ve uygulanabilirliğini belirlemede kritik bir rol oynar.

Önerilen KWOA algoritmasında uygunluk fonksiyonu olarak bölüm 3.6.1’de açıklanan Rand İndeks Küme Doğrulama İndeksinin 1’den çıkarılması ile elde edilen *Maliyet* değeri kullanılmıştır. KWOA algoritması bir minimizasyon algoritması olacak şekilde tasarlanmıştır ve amacı bu maliyet değerini minimize etmektir. Rand indeks değeri maksimum 1 değerini aldığı için, elde edilen değer 1’den çıkarılmış ve kümeleme işlemi bir minimizasyon problemine çevrilmiştir. Maliyet, çözümün kalitesinin bir ölçütü olarak işlev görmektedir ve düşük maliyet, çözüm adayının problemi ne kadar iyi çözdüğünü göstermektedir.

Maliyet değerinin hesaplanabilmesi için, popülasyondaki her bir arama ajanı küme merkezi olarak çözümlenir ve bu merkezler kullanılarak veri setindeki her bir kaydın kümelerle olan uzaklıkları hesaplanır ve en yakın olduğu kümeye eleman olarak atanır. Klasik kümeleme algoritmalarında uzaklık hesabı için Öklid vb. uzaklık fonksiyonları kullanılmaktadır. KWOA algoritmasında bu aşamada uzaklık fonksiyonu yerine, Bölüm 3.4.3’de açıklanan ve iki örnek arasındaki uzaklık büyüdükçe benzerliğin azalmasını sağlayan bir çekirdek fonksiyonu olan Radial Basis Function (RBF) kullanılmıştır. Hesaplanan bu çekirdek değeri, 1’den çıkarılarak bir uzaklık skoru olarak değerlendirilir. Bu işlemler sonucunda, veri setindeki her bir kaydın hangi kümeye atandığını içeren bir etiketler dizisi oluşur. Bu etiket dizisi, Rand indeks için küme değerlerini oluşturur. Rand İndeks, algoritma tarafından oluşturulan “Tahmini” kümeler ile “Gerçek” etiketler arasındaki benzerliği ölçer.

4.4. Önerilen KWOA Algoritması ve Örnek Bir Çözüm

Balina Optimizasyon Algoritması, doğal davranışlardan esinlenen bir yöntem olup, global arama kapasitesi ve yüksek çözüm çeşitliliği sağlama avantajlarına sahiptir. Algoritmanın esnekliği ve etkili arama stratejisi, karmaşık veri yapılarını etkin bir şekilde kümeler halinde organize etme potansiyeline sahiptir. Bu çalışmada, kernel dönüşümünün avantajları da göz önünde bulundurularak WOA ile birleştirilmiştir. Çizelge 4.1’de önerilen KWOA algoritmasının sözde kodu verilmiştir. Bu kod, veri kümeleme için WOA ve RBF kernel yöntemlerinin etkili bir entegrasyonunu sunar. Bu yaklaşım, verileri yüksek boyutlu bir uzayda modelleyerek ve balina optimizasyon algoritması ile bu uzayda etkin bir şekilde arama yaparak karmaşık kümeleme problemlerini çözmeyi hedeflemiştir.

Algoritma, iterasyonlar boyunca, maliyet fonksiyonunu minimize eden çözüm adaylarını seçer ve böylece en iyi çözüme yaklaşır. Nihai hedef, algoritmanın mümkün olan en düşük maliyet değerine ulaşarak, veriyi en iyi şekilde kümelemesidir.

KWOA Algoritmanın daha iyi anlaşılması için, iki kümeli ($K=2$) sentetik basit bir veri seti oluşturulmuş ve gerçekleştirilen işlemler adım adım gösterilmiştir.

Oluşturulan sentetik veri setindeki örnek sayısı 8, her örneğin sahip olduğu nitelik sayısı (M) 2 olarak belirlenmiştir. Her küme için merkezler rastgele olarak seçilmiş ve küme merkezleri etrafında normal dağılımlı rastgele veriler oluşturulmuştur. Bu işlem her küme için ayrı ayrı yapılır ve veri noktaları ile sınıf etiketleri de belirlenir.

Oluşturulan sentetik veri seti Çizelge 4.2’ te gösterilmiştir. Min-Max yöntemi ile normalize edilmiş veri seti Çizelge 4.3’ te gösterilmiştir.

Çizelge 4.2 Oluşturulan Sentetik Veri Seti

Veri Noktaları		Etiket Değerleri
-0,5382	-0,9960	1
0,8672	-0,5232	1
0,9759	-1,2974	1
0,3373	0,9173	1
5,1766	6,8169	2
5,7551	4,8761	2
4,4085	3,8893	2
6,8443	4,3190	2

Çizelge 4.1. KWOA (Kernel Yöntemi ile Balina Optimizasyon Algoritması Tabanlı Kümeleme) Algoritmasının Sözde Kodu

Fonksiyonlar:

maliyetHesapla (çözüm, X, K, M, Kernel, Baslangic_Etiketleri)

Merkezler = çözümü K x M boyutlarında diziye dönüştür

Mesafeler = **kernelMesafeleriniHesapla**(X, Merkezler, kernel)

Etiketler = Mesafeler içindeki her satır için en küçük mesafenin indeksini al

cost = 1 - rand_indeks(Baslangic_Etiketleri, Etiketler)

end

kernelMesafeleriniHesapla(X, Merkezler, kernel)

Mesafeler = numuneSayısı x merkezSayısı boyutunda sıfırlar matrisi oluştur

for(i=1: numuneSayısı)

for(j=1: merkezSayısı)

Mesafeler(i, j) = 1 - rbfKernel(X(i, :), Merkezler(j, :))

end

end

end

rbfKernel(x, y, sigma)

sonuç = Denklem 3.4'e göre hesapla

end

Ana Süreç:

Adım 1 : Küme sayısını(K), Max iterasyon değerini(Maxİtaration) ve özellik sayısını(M) oluştur.

Adım 2 : Başlangıç popülasyonunu oluştur.

Adım 3 : for index = 1 to Maxİtaration do

a = 2 - mevcut_iterasyon_numarası * (2 / maksimumIterasyonSayısı) ile güncelle

for(i=1 to whales)

r = rand()

A parametresini güncelle(Denklem 3.7)

C parametresini güncelle(Denklem 3.8)

l = rand()

b = 0.01

for(j=1 : (K * M))

Küme İndeks'ini hesapla : (j-1) mod M + 1

Küme Grubunu hesapla : ((j-1) / M) + 1

if (r < 0.5)

% Daralan dairesel hareketler ile kuşatma

MesafeD = |C * mevcutPopülasyon(i, j) - mevcutPopülasyon(i, j)| hesapla

mevcutPopülasyon(i, j) = mevcutPopülasyon(i, j) - A * MesafeD güncelle

else

% Spiral hareket oluşturma

MesafeD = |mevcutPopülasyon(i, j) - mevcutPopülasyon(i, j)| hesapla

mevcutPopülasyon(i, j) = MesafeD * e^(b * l) * cos(1 * 2 * pi) +

mevcutPopülasyon(i, j) güncelle

end

End for

maliyet = maliyetHesapla(popülasyon(i, :), X, K, M, Kernel, Baslangic_Etiketleri)

if maliyet < enİyiMaliyet ise

enİyiMaliyet = maliyet

enİyiÇözüm = popülasyon(i, :)

end

End for

En iyi küme merkezlerini, en iyi çözümü KxM boyutlarında bir matrise dönüştürerek oluştur.

Her veri noktası için, RBF kernel kullanarak en iyi küme merkezleriyle olan benzerlikleri hesapla ve en yüksek benzerlik gösteren merkezin etiketini ata.

Çizelge 4.3. Normalize Edilmiş Veri Noktaları

Normalize Edilmiş Veri Noktaları		Etiket Değerleri
0	0,0371	1
0,1903	0,0954	1
0,2051	0	1
0,1186	0,2729	1
0,7741	1	2
0,8525	0,7609	2
0,6701	0,6392	2
1	0,6922	2

Konunun daha net anlaşılması için Çizelge 4.1’de KWOA algoritmasına ait adım adım bir örnek çözümü gerçekleştirilmiştir.

1. Başlangıç (Adım1 ve Adım2, İterasyon=0):

KWOA algoritmasına ait başlangıç parametreleri değerleri çizelge 4.4’de verilmiştir. Çizelge de bulunan A değeri, Denklem (3.7)’ye göre, C değeri Denklem (3.8)’e göre hesaplanmıştır. r $[0,1]$ aralığında değer alan rastgele bir vektörü temsil eder. a iterasyonlar boyunca 2’den 0’a doğru lineer olarak azalan bir vektörü gösterir. $a = 2 - \text{mevcut_iterasyon_numarası} * (2 / \text{maksimumIterasyonSayısı})$ formülü ile güncellenir. b başlangıçta belirlenen sabit bir değerdir. l $[0,1]$ aralığında değer alan rastgele bir vektörü temsil eder.

Çizelge 4.4.İterasyon=0 için KWOA parametre değerleri

Parametre	Değer
a	1.3333
r	0.0464
A	-1.2096
C	0.0928
b	0.01
l	0,0651

Bu veri seti için çalıştırılacak KWOA algoritmasında *Balina (Arama Ajanı)* $\text{Sayısı}=1$, $\text{maksimum iterasyon sayısı}=1$ ve $\text{küme sayısı } K=2$ olarak belirlenmiştir. Her bir küme için başlangıç merkezleri rastgele olarak belirlenir. Başlangıç popülasyonu oluşturulur. Bu örnek için bir satır ve $K*M = 4$ sütun içeren bir matristir. Her bir konum değeri için, 0 ile 1 arasında rastgele değer üretilmiş ve çizelge 4.5’te verilmiştir.

Çizelge 4.5. Başlangıç Popülasyonu(X(0))

C ₁		C ₂		
X _{1,1}	X _{1,2}	X _{2,1}	X _{2,2}	Maliyet
0,5248	0,2912	0,1395	0,3664	0.5714

Çizelge 4.5’de ayrıntılı olarak konum değerleri verilen arama ajanı, Çizelge 3.3’de $\vec{X}(0)$ ’a karşılık gelmektedir ($\vec{X}(0) = [0,5248, 0,2912, 0,1395, 0,3664]$) Maliyet fonksiyonu diğer bir deyişle bölüm 4.3’de ayrıntıları verilen uygunluk fonksiyonu hesaplanacak $\vec{X}(0)$ çözüm vektörü, K adet küme merkezi içerecek şekilde K x M boyutunda bir matrise dönüştürülür. Ardından, RBF Kernel fonksiyonu kullanılarak, veri noktalarının her bir merkeze olan benzerlikleri hesaplanır ve benzerlik değerleri 1’den çıkarılarak *Mesafeler* (Çizelge 4.1-Maliyet Hesapla Fonksiyonu içinde yer almaktadır) matrisine atanır. Daha sonra, her bir veri noktası için en yakın merkezi bulmak amacıyla min fonksiyonu kullanılır ve bu merkezlerin indeksleri *Etiketler* vektörüne atanır. *Etiketler*, bölüm 3.6.1’de açıklanan Rand indeks küme doğrulama metriğindeki “Tahmini” yapısına karşılık gelmektedir. “Gerçek” yapısı ise, çizelge 4.3’de etiket değerleri sütununda verilmiştir. Bu değerleri kullanarak Rand İndeks hesaplanır. Rand indeks’in maksimum olması istenir ancak kümeleme işlemi minimizasyon problemi olarak kodlandığı için, hesaplan rand indeks değeri 1’de çıkarılarak Maliyet fonksiyonu hesaplanır. Bu şekilde, Rand İndeks 1’e yaklaştıkça, yani kümelerin doğruluğu arttıkça, maliyet değeri azalır. Bu formül, algoritmanın kümeleme doğruluğunu en üst düzeye çıkarmayı hedefler. Hesaplamalar sonucunda 0.5714 olarak elde edildi. Tek değer olduğu için en iyi birey $X^*(0)$ olarak atanmıştır. İterasyon numarası bir artırılarak işleme devam edilir.

İterasyon=1:

2. Avın Etrafını Kuşatma:

Arama ajanı $X_{1,1} = 0,5248$ konumu için:

$a = 2 - \text{mevcut_iterasyon_numarası} * (2 / \text{maksimumIterasyonSayısı})$ formülüne göre, 1. İterasyona geçildiği için $a=2$ olarak hesaplanır ve $r=0.7$ olarak üretildiği kabul edilsin.

- $A=2 \times 0.7 - 2 = -0.6$
- $C=2 \times 0.7 = 1.4$
- $D=|C \times X^*(0) - X(0)| = |1.4 \times 0.5714 - 0.5248| = 0.2751$

- $X(1)$ 'in yeni pozisyonu hesaplanıyor:

$$X(1) = X^*(0) - A \times D = 0.5714 - (-0.6) \times 0.2751 = 0.7364$$

Arama ajanının diğer konumları da benzer şekilde hesaplanır. Bunun sonucunda aşağıdaki gibi bir vektör elde edildiği kabul edilsin. Görüldüğü üzere 0,5248 değeri, 0.7364 olarak güncellendi.

$$\vec{X}(0) = [0,5248; 0,2912; 0,1395; 0,3664]$$

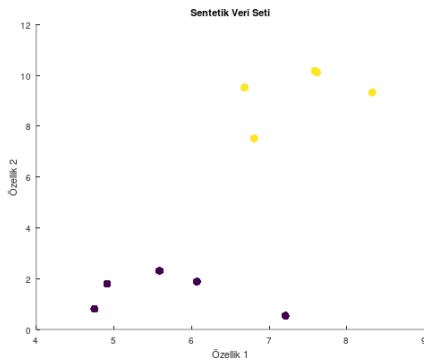
$$\vec{X}(1) = [0.7364; 0.2912; 0.1395; 0.1626]$$

$X(1)$ maliyet fonksiyonu hesaplanır.

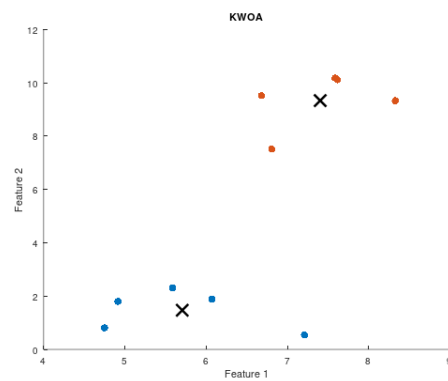
En iyi birey ve maliyet kontrolü yapıldıktan sonra, işlemler belirtilen iterasyon sayısı kadar tekrar edilir. Rastgele üretilen r 'nin durumuna göre av kuşatma veya av arama işlemine göre konumlar güncellenir. Algoritma, iterasyonlar boyunca bu maliyet fonksiyonunu minimize eden çözüm adaylarını seçer ve böylece en iyi çözüme yaklaşır. Elde edilen etiket ve yeni etiket değerlerine göre Rand İndeks ve Siluet İndeks değerleri hesaplanmıştır. Çizelge 4.6'da oluşturulan sentetik veri seti için RI ve SI değerleri sonuçlar verilmiştir.

Çizelge 4.6. KWOA Sentetik Veri Seti Sonuçları

Yöntem	Sonuç
RI	1
SI	0.9263



Şekil 4.2. Sentetik Veri Seti



Şekil 4.3. KWOA ile kümelmiş sentetik veri

Başlangıçta oluşturulan sentetik veri setinin KWOA ile kümelmiş hali Şekil 4.3'te gösterilmiştir. Küme merkezleri x ile işaretlenmiştir. Şekiller incelendiğinde küçük bir örnek için bile verimli bir sonuç alındığı gözlemlenmiştir.

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu çalışmada, Balina Optimizasyon Algoritması (WOA) kullanılarak, veri kümeleme probleminin çözümüne yönelik bir yaklaşım önerilmiştir. RBF kernel fonksiyonu ile dönüştürülen özellik uzayında, WOA'nın kabarcık ağı avlanma yöntemi temel alınarak geliştirilmiş bir optimizasyon stratejisi uygulanmıştır. Bu strateji, veri noktalarını kümeler halinde gruplamak için kullanılmış ve her kümeye uygun küme merkezi aranmıştır. Bu bölümde önerilen KWOA algoritmasının etkinliği, Bölüm 3.1'de açıklaması verilen veri setleri üzerinde test edilmiş ve elde edilen sonuçlar literatüre başarısını ispat etmiş ve kabul edilmiş dokuz farklı kümeleme algoritması ile kıyaslanmıştır.

Kullanılan algoritmalar Kernel K Ortalamalar, K Ortalamalar, Affinity Propagation, MeanShift, Spectral Clustering, DBSCAN, Optics, Birch ve Gaussian Mixture'dır. Kullanılan veri setleri Balance, Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Original), Dermatology, Ecoli, Glass, Heart ve Iris'tir. Her bir algoritma bu veri setleri üzerinde 30 kez çalıştırılarak elde edilen sonuçlar değerlendirilmiştir. Çalıştırma sonuçları Rand Index ve Siluet Küme Doğrulama Index değerleri kullanılarak tartışılmıştır.

5.1. Rand Index Karşılaştırma Sonuçları

Rand Index Karşılaştırma sonuçları, farklı algoritmaların veri setleri üzerindeki performanslarını göstermek üzere çizelge 5.1'de sunulmuştur. Bu karşılaştırma, her bir algoritmanın veri setleri üzerindeki 30 kez çalıştırmalarından elde edilen Rand Index değerlerinin minimum, maksimum, ortalama ve standart sapma gibi istatistiksel ölçümlerini içerir. Çizelge 5.1'deki algoritmaların her bir veri kümesi üzerindeki ortalama rand indeks değerleri sıralanmış ve numaralandırılmıştır. En yüksek rand indeksine sahip algoritma 1. Sıra değerini alır. Aynı başarıya sahip olan algoritmaların, buldukları sıra değerleri de aynı olur. En son sıra değeri, en düşük başarıya sahip algoritmaya verilir. Algoritmalara ait sıra değerleri çizelge 5.2'de sunulmuştur.

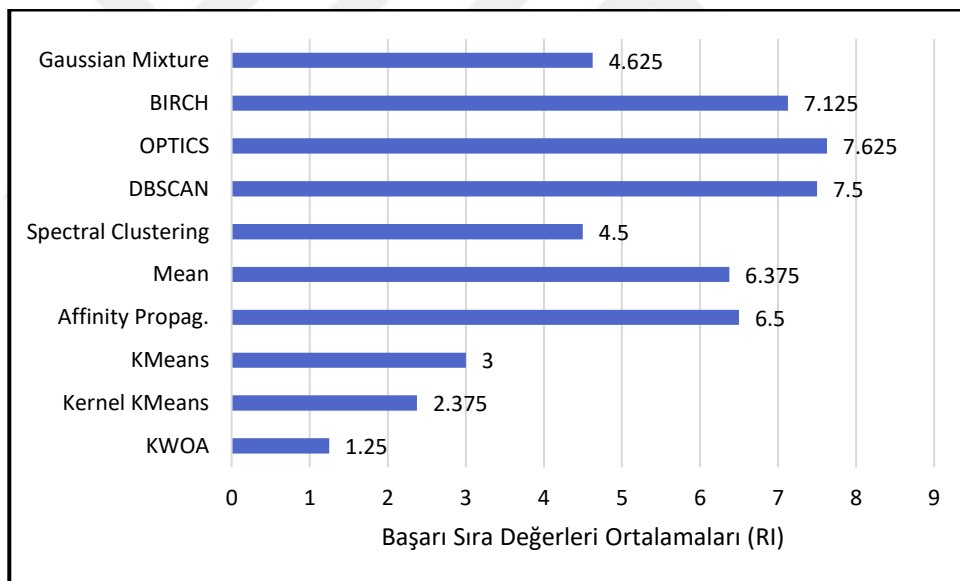
Çizelge 5.1. Her Bir Kümeleme Algoritmasına ait Rand Index Sonuçları

RI		KWOA	Kernel KMeans	KMeans	Affinity Propag.	Mean Shift	Spectral Clust.	DBSC AN	OPTI CS	BIRC H	Gaussian Mixture
Balance	Min	0,652	0,523	0,522	0,580	0,430	0,499	0,430	0,430	0,582	0,550
	Max	0,689	0,635	0,619	0,586	0,433	0,846	0,433	0,430	0,582	0,689
	Ort	0,672	0,585	0,583	0,582	0,430	0,575	0,430	0,430	0,582	0,588
	Std. S.	0,011	0,028	0,018	0,001	0,001	0,094	0,001	0,000	0,000	0,023
BCWO	Min	0,920	0,900	0,924	0,488	0,870	0,864	0,649	0,462	0,793	0,748
	Max	0,954	0,916	0,929	0,857	0,870	0,864	0,649	0,462	0,793	0,750
	Ort	0,932	0,904	0,925	0,727	0,870	0,864	0,649	0,462	0,793	0,748
	Std.S.	0,009	0,005	0,002	0,171	0,000	0,000	0,000	0,000	0,000	0,000
BCWD	Min	0,839	0,837	0,866	0,857	0,625	0,753	0,552	0,549	0,545	0,831
	Max	0,913	0,837	0,866	0,857	0,625	0,753	0,552	0,549	0,545	0,831
	Ort	0,866	0,837	0,866	0,857	0,625	0,753	0,552	0,549	0,545	0,831
	Std.S.	0,020	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Dermatology	Min	0,836	0,806	0,812	0,838	0,519	0,839	0,213	0,606	0,850	0,747
	Max	0,882	0,951	0,976	0,839	0,519	0,839	0,213	0,606	0,850	0,875
	Ort	0,851	0,900	0,891	0,838	0,519	0,839	0,213	0,606	0,850	0,850
	Std.S.	0,011	0,039	0,040	0,000	0,000	0,000	0,000	0,000	0,000	0,025
E. Coli	Min	0,796	0,782	0,787	0,286	0,708	0,829	0,298	0,445	0,286	0,642
	Max	0,884	0,877	0,887	0,290	0,708	0,829	0,298	0,445	0,290	0,878
	Ort	0,835	0,830	0,829	0,288	0,708	0,829	0,298	0,445	0,287	0,786
	Std.S.	0,022	0,035	0,029	0,002	0,000	0,000	0,000	0,000	0,002	0,083
Glass	Min	0,665	0,661	0,634	0,263	0,597	0,624	0,321	0,590	0,519	0,565
	Max	0,724	0,724	0,701	0,268	0,597	0,652	0,321	0,590	0,519	0,693
	Ort	0,690	0,688	0,664	0,264	0,597	0,627	0,321	0,590	0,519	0,675
	Std.S.	0,015	0,012	0,016	0,002	0,000	0,007	0,000	0,000	0,000	0,023
Heart	Min	0,500	0,513	0,501	0,511	0,500	0,604	0,505	0,503	0,501	0,530
	Max	0,696	0,612	0,589	0,512	0,500	0,604	0,505	0,503	0,501	0,589
	Ort	0,635	0,595	0,563	0,512	0,500	0,604	0,505	0,503	0,501	0,570
	Std.S.	0,036	0,043	0,034	0,000	0,000	0,000	0,000	0,000	0,000	0,027
İris	Min	0,780	0,716	0,721	0,756	0,776	0,831	0,776	0,655	0,717	0,333
	Max	0,974	0,982	0,874	0,756	0,776	0,831	0,776	0,655	0,717	0,333
	Ort	0,876	0,869	0,861	0,756	0,776	0,831	0,776	0,655	0,717	0,333
	Std.S.	0,056	0,134	0,038	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Çizelge 5.1’de raporlanan sonuçlar incelendiğinde sekiz veri setinin altısında daha başarılı sonuçlar ürettiği, BCWD veri setinde k-ortalama algoritması ile birinciliği paylaştığı görülmektedir. Dermatology veri setinde ise üçüncü en başarılı algoritma olmuştur. Algoritmaların ayrıntılı başarı sıra değerleri çizelge 5.2’de sunulmuştur.

Çizelge 5.2. Her Bir Kümeleme Algoritmasına ait Başarı Sıra Değerleri

	KWOA	Kernel KMeans	KMeans	Affinity Propag.	Mean Shift	Spectral Clustering	DBSCAN	OPTICS	BIRCH	Gaussian Mixture
Balance	1	3	4	5	7	6	7	7	5	2
BCWO	1	3	2	8	4	5	9	10	6	7
BCWD	1	3	1	2	6	5	7	8	9	4
Dermatology	3	1	2	7	8	6	9	7	4	4
E. Coli	1	2	3	8	5	3	7	6	9	4
Glass	1	2	4	10	6	5	9	7	8	3
Heart	1	3	5	6	10	2	7	8	9	4
İris	1	2	3	6	5	4	5	8	7	9
Ortalama	1.25	2.38	3	6.5	6.38	4.5	7.5	7.63	7.13	4.63



Şekil 5.1. Her Bir Kümeleme Algoritmasına ait RI'e göre Başarı Sıra Değerleri Ortalaması

Her veri seti ve algoritma için gözlemlenen minimum, maksimum ve ortalama RI değerleri, algoritmaların kümeleme başarısını göstermektedir. Standart sapma değerleri ise, algoritmaların performansının tutarlılığını ifade eder.

Tablolar incelendiğinde, KWOA algoritması veri setlerinde dikkate değer sonuçlar elde etmiştir. Örneğin, Balance veri seti üzerinde yapılan çalışmalarda, KWOA en yüksek ortalama Rand Index değerine ulaşarak diğer algoritmalar arasında ön plana

çıkıştır. Bu, algoritmanın bu özel veri yapısına ne kadar iyi uyum sağladığını ve kümeleme performansının ne derece etkili olduğunu göstermektedir.

Benzer şekilde, Heart, Breast Cancer Wisconsin (Diagnostic) ve Breast Cancer Wisconsin (Original) veri setlerinde KWOA algoritması, Rand Index ortalamalarında en yüksek değerlere sahip olmuştur. Bu sonuçlar, algoritmanın bu tıbbi veri setlerindeki karmaşık yapıları başarıyla yakalayıp gruplandırabildiğini ortaya koymaktadır.

Ayrıca, Iris veri setinde de KWOA'nın ortalama Rand Index değeri en yüksek çıkıştır. Bu sonuçlar, KWOA'nın çeşitli özellikler içeren ve farklı boyutlara sahip veri kümelerini etkili bir şekilde kümeleyebildiğini ve bu tür biyolojik veri kümeleri üzerindeki tutarlı ve güçlü performansını vurgulamaktadır.

Genel olarak, KWOA'nın tüm veri setlerinde istikrarlı ve yüksek RI değerleri sergilediği görülmektedir, bu da onun güvenilir ve etkili bir kümeleme yöntemi olduğuna işaret etmektedir. Özellikle, minimum ve maksimum RI değerleri arasındaki farkın genellikle küçük olması, algoritmanın tutarlılık açısından üstün olduğunu belirtir.

Bu sonuçlar, KWOA'nın geniş bir veri seti yelpazesinde güçlü ve stabil kümeleme performansı sunabileceğini, bu algoritmaların karmaşık veri setleri üzerinde de etkin olabileceğini ve algoritmanın daha geniş kapsamlı uygulamalar için de uygun olabileceğini düşündürmektedir.

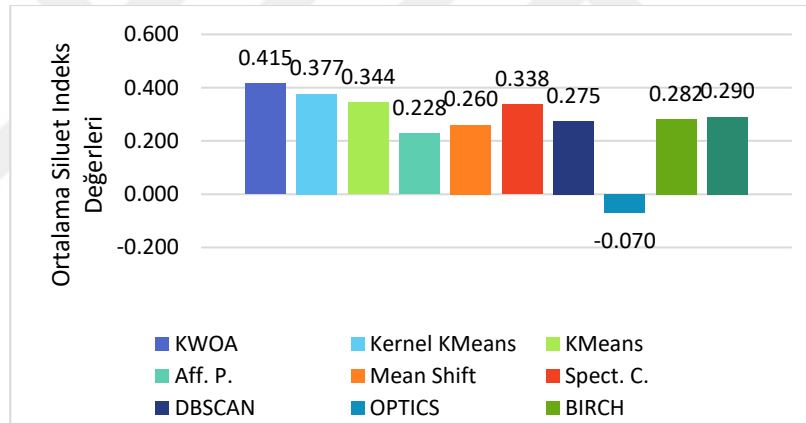
5.2. Siluet Küme Doğrulama İndeksi Sonuçları

Siluet Küme Doğrulama İndeksi kümeleme algoritmasının etkinliğini değerlendirmede kullanılan bir metriktir. Bu metrik, bir noktanın kendi kümesinde ne kadar iyi yerleştiğini ve diğer kümelere ne kadar ayrı olduğunu ölçer. İdeal olarak, yüksek bir Siluet İndeks değeri, iyi tanımlanmış ve ayrılmış kümeler anlamına gelir.

Siluet Doğrulama İndeksi sonuçları, veri setleri üzerindeki performanslarını göstermek üzere çizelge 5.3 sunulmuştur. Bu karşılaştırma, her bir algoritmanın veri setleri üzerindeki çalışmalarından elde edilen Siluet Küme Doğrulama İndeks değerlerinin ölçümlerini içerir.

Çizelge 5.3. Her Bir Kümeleme Algoritmasına ait Ortalama Siluet İndeks Sonuçları

SI	KWOA	Kernel KMeans	KMeans	Aff. P.	Mean Shift	Spect. C.	DBSCAN	OPTICS	BIRCH	Gauss. Mixture
Balance	0,269	0,313	0,172	0,122	-0,043	0,169	-0,008	-0,039	0,129	0,169
BCWO	0,723	0,585	0,606	0,376	0,511	0,580	0,334	-0,010	0,525	0,392
BCWD	0,537	0,399	0,385	0,388	0,286	0,401	0,356	-0,141	0,407	0,356
Dermatology	0,214	0,211	0,213	0,100	0,111	0,139	-0,182	-0,175	0,255	0,181
E. Coli	0,393	0,342	0,295	0,068	0,369	0,329	0,507	-0,348	-0,059	0,262
Glass	0,352	0,343	0,343	0,285	0,241	0,346	0,342	-0,143	0,336	0,262
Heart	0,256	0,248	0,244	0,168	-0,022	0,249	0,293	0,250	0,220	0,247
İris	0,577	0,573	0,497	0,319	0,630	0,487	0,556	0,044	0,439	0,453
Ortalama Siluet	0,415	0,377	0,344	0,228	0,260	0,338	0,275	-0,070	0,282	0,290



Şekil 5.2. Her Bir Kümeleme Algoritmasının Siluet İndeks Ortalaması

KWOA algoritması, 0,415 ortalama SI değeri ile diğer algoritmalarla kıyasla en yüksek değeri elde etmiştir. Özellikle Balance, Breast Cancer Wisconsin (Original), Breast Cancer Wisconsin (Diagnostic), Iris veri setlerinde KWOA, diğer kümeleme algoritmasına kıyasla üstün performans sergilemiştir. Balance veri setinde, KWOA algoritması, Kernel KMeans algoritmasının ardından ikinci sırada yer almış ve veri noktalarını kümeler halinde gruplamadaki etkinliğini göstermiştir. Breast Cancer Wisconsin (Original) veri setinde KWOA, en yüksek Siluet İndeks değerini elde ederek, tıbbi verilerdeki uyum ve doğruluğunu kanıtlamıştır. Benzer şekilde, Breast Cancer Wisconsin (Diagnostic) veri setinde de en yüksek performansı gösteren KWOA, tıbbi

verilerin doğru bir şekilde kümelenmesine olanak tanımıştır. Iris veri setinde, KWOA algoritması yüksek bir Siluet İndeks değeri elde ederek çiçek türlerini ayırt etme ve sınıflandırma konusundaki üstün performansını sergilemiştir.

Genel olarak, KWOA algoritması, farklı tipte ve boyutta veri setlerini analiz etme konusundaki güçlü yönlerini ortaya koyarak, en yüksek ortalama Siluet İndeks değerine ulaşmıştır. Elde edilen bu veriler, KWOA algoritmasının farklı tipte ve boyutta veri setlerini analiz etme konusundaki güçlü yönlerini ortaya koymaktadır ve bu algoritmanın çeşitli veri yapılarında kullanımının önerilmesine temel oluşturmaktadır.

5.3. KWOA ile Kümelenmiş Verileri Görsel Sonuçları

Bu bölümde, KWOA algoritması ile kümelenen sekiz veri setinin her biri için küme merkezleri, ilk iki nitelik kullanılarak görselleştirilmiş ve yorumlanmıştır. Görseller şekil 5.3'de sunulmuştur.

Iris veri seti: Üç kümesi olan veri setinde, her kümenin diğerlerinden net bir şekilde ayrıldığı gözlemlenmiştir. Özellikle, kümelerden bir tanesi diğer iki kümeye göre biraz daha izole durumdadır. Bu, o niteliğim diğer iki Iris türünden daha belirgin farklı özelliklere sahip olduğunu göstermektedir.

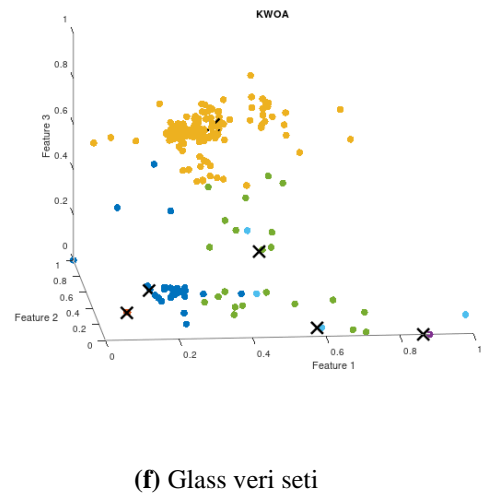
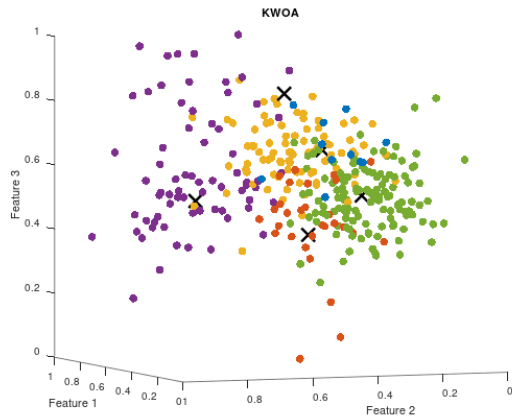
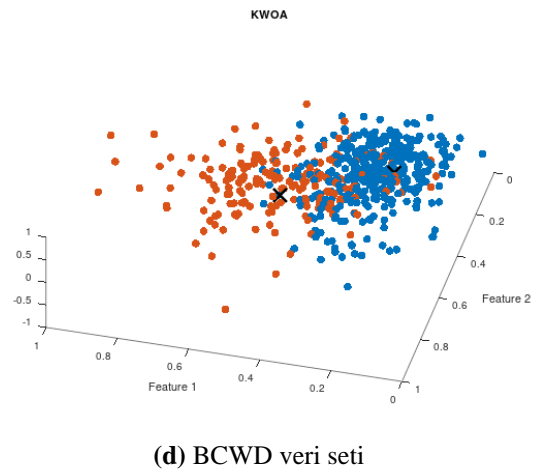
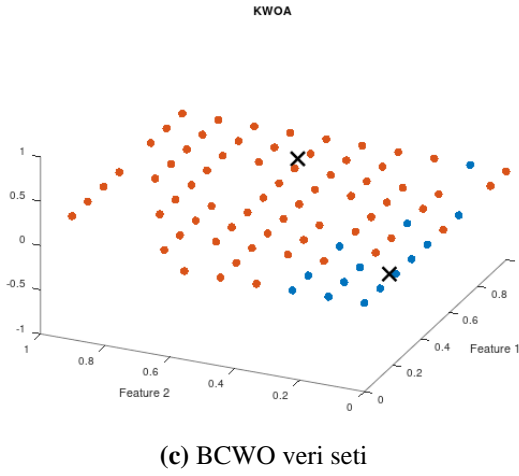
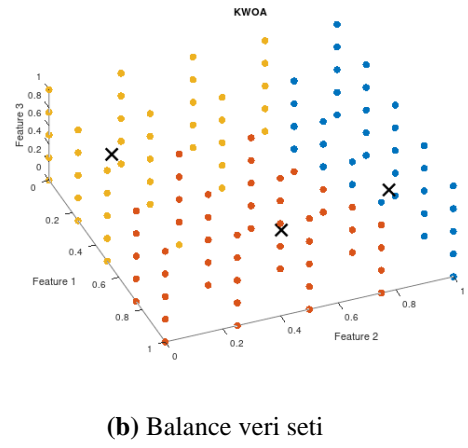
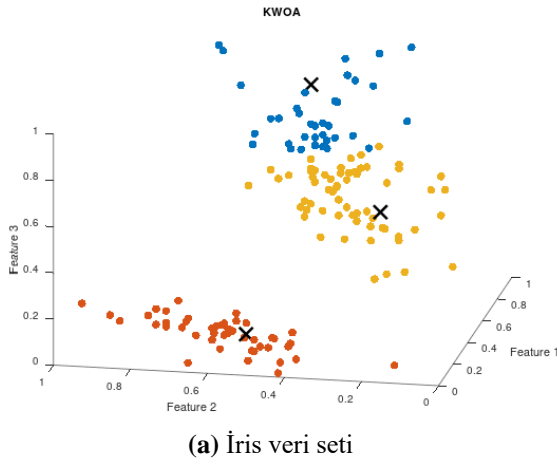
Balance veri seti: Dört farklı küme değeri olan veri seti, üç boyutlu uzayda birbirlerinden ayrılmış olduğu gözlemlenmiştir. Kümelerin birbirinden uzak oluşu, her bir kümeyi oluşturan veri noktalarının özellik uzayında birbirlerinden belirgin bir şekilde farklı olduğunu göstermektedir.

BCWO veri seti: İki ayrı küme, veri noktalarının iki boyutlu bir düzlemde birbirine yakın oluş biçiminden dolayı oluşmuştur. Her bir küme içindeki noktalar benzer yoğunluk ve yayılım gösterirken, iki küme arasında açık bir sınır belirlidir.

BCWD veri seti: Her bir kümesi içindeki noktalar benzer yoğunluk ve yayılım gösterirken, bu kümelemede veri noktalarının birbirine daha yakın olduğunu göstermektedir.

Ecoli veri seti: Noktalar, birbirleriyle örtüşen birden fazla renkte olduğu görülmektedir. Bu, kümelenmenin daha karmaşık olduğuna ve veri noktalarının birbirine daha yakın olduğuna işaret eder. KWOA algoritması bu zorluğun üstesinden geldiği görülmüştür.

Glass veri seti: Kümeleme iki boyutlu düzlemde gerçekleşmiş ve bazı kümeler diğerlerine göre daha yoğun görünmektedir. Bu durum, bazı cam türlerinin özelliklerinin diğerlerinden daha benzer olduğunu gösterebilir.



Şekil 5.3. KWOA algoritması ile kümelendikten sonra veri setlerinin ilk iki niteliğine göre küme merkezlerinin ve dağılımlarının görselleştirilmesi

Şekil 5.3’de görüldüğü üzere, KWOA algoritması özellikler arasındaki ilişkilere göre veri noktalarını gruplandırmış ve her kümenin merkezi bir 'X' ile işaretlenmiştir. Kümelene kalitesi, kümeler arası mesafe ve kümeler içindeki yoğunluk ile değerlendirilebilir. Her grafik, kümelene algoritmasının o veri setindeki özellik ilişkilerini ne kadar iyi yakaladığını görselleştirir.



6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Bu çalışmada, veri kümelemesi için yeni bir yöntem olarak Radial Basis Function (RBF) kernel fonksiyonu kullanılan Balina Optimizasyon Algoritması (WOA) geliştirilmiş ve değerlendirilmiştir. Araştırmanın temel bulguları, önerilen çalışmanın, veri noktalarını iç tutarlılığı yüksek kümeler şeklinde gruplandırırken aynı zamanda kümeler arası belirgin ayrımı sağlama kapasitesine sahip olduğunu göstermektedir. Rand Index ve Siluet doğrulama İndeks değerleri ile elde edilen metrikler, kümelenecek grupların ayırt ediciliğinin ve bütünlüğünün, algoritmanın etkinliğini doğrulayan yüksek seviyelerde olduğunu ortaya koymaktadır.

Elde edilen bu olumlu sonuçlar göz önüne alındığında, özellikle doğrusal olarak ayrıştırılamayan karmaşık yapıdaki veri kümelerinin analizinde kernel kümelemenin kullanımının desteklenmesi önerilir. RBF kernel ile yapılan dönüşümün, veri setlerinin daha uygun bir biçimde temsil edilmesine ve bu sayede algoritmanın kümeleme performansının artırılmasına olanak tanıdığı gözlemlenmiştir. Bu bulgular, veri madenciliği ve makine öğrenimi uygulamalarında, özellikle doğrusal olmayan ilişkilerin baskın olduğu durumlar için, WOA ve RBF kernel entegrasyonunun önemli avantajlar sağlayabileceğini işaret etmektedir.

6.2 Öneriler

Farklı uygulama alanları ve gerçek dünya veri setleri üzerinde WOA'nın daha kapsamlı bir şekilde test edilmesi önerilmektedir. Özellikle, algoritmanın parametrelerinin ve kernel fonksiyonlarının çeşitliliğinin, kümeleme performansı üzerindeki etkilerinin daha detaylı analizi yapılabilir. Bunun yanı sıra, algoritmanın ölçeklenebilirliği ve farklı boyutlardaki veri setlerine adaptasyonu gibi konular üzerine gelecek çalışmalar yapılması faydalı olacaktır.

Son olarak, bu çalışma, WOA ve kernel yöntemlerinin birlikte çalışma konusunda bir başlangıç noktası sunmakta olup, bu yaklaşımın farklı algoritmalar ve tekniklerle çalışma konusunda da araştırmalar yapılması önerilmektedir. Bu tür bir entegrasyon, algoritmanın veri keşfetme kabiliyetini artırabilir ve böylece daha karmaşık veri yapılarının çözümlenmesinde kullanılabilir.

7. KAYNAKLAR

- Arthur, D. (2007). kMeans++. *Soda '07*, 1027–1035. <http://www.stanford.edu/~darthur/kMeansPlusPlus.pdf> adresinden erişildi.
- Bayá, A. E. ve Larese, M. G. (2020). Pixel sampling by clustering. *Expert Systems with Applications*, 159. doi:10.1016/j.eswa.2020.113576
- Brest, J. ve Neri, F. (2019). Editorial of the special issue on differential evolution. *Swarm and Evolutionary Computation*, 50, 100583. doi:10.1016/j.swevo.2019.100583
- Camastra, F. ve Vinciarelli, A. (2015). Kernel methods. *Advanced Information and Knowledge Processing* içinde . doi:10.1007/978-1-4471-6735-8_9
- Ding, Y. ve Fu, X. (2016). Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing*. doi:10.1016/j.neucom.2015.01.106
- Fan, S., Ding, S. ve Xue, Y. (2018). Self-adaptive kernel K-means algorithm based on the shuffled frog leaping algorithm. *Soft Computing*, 22(3), 861–872. doi:10.1007/s00500-016-2389-2
- Faramarzi, A., Heidarinejad, M., Stephens, B. ve Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191. doi:10.1016/J.KNOSYS.2019.105190
- Ferreira, M. R. P., De Carvalho, F. D. A. T. ve Simões, E. C. (2016). Kernel-based hard clustering methods with kernelization of the metric and automatic weighting of the variables. *Pattern Recognition*. doi:10.1016/j.patcog.2015.09.025
- Filippone, M., Camastra, F., Masulli, F. ve Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1), 176–190. doi:10.1016/j.patcog.2007.05.018
- Filippone, M., Camastra, F., Masulli, F. ve Rovetta, S. (y.y.). A Survey of Kernel Clustering Methods.
- Forgy, E.W. (1965) Cluster Analysis of Multivariate Data: Efficiency vs Interpretability of Classifications. *Biometrics*, 21, 768-780.
- Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 13, 780–784. doi:10.1109/TNN.2002.1000150
- Graves, D. ve Pedrycz, W. (2010). Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy Sets and Systems*, 161(4), 522–543. doi:10.1016/j.fss.2009.10.021

- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. ve Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*.
- Hancer, E. (2018). Diferansiyel Gelişim Tabanlı Çoklu Bulanık Kernel Kümeleme. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 2018. doi:10.17341/gazimmfd.460525
- <https://acikbilim.yok.gov.tr/handle/20.500.12812/125033> adresinden erişildi.
- Kırmızıbiber, A. ve Gökgöz, T. (2019). Hiyerarşik ve K - Ortalamalar Yöntemleriyle Grid N oktarının K ümeleme nmesi, (0442), 25–27.
- Kuo, R. J., Huang, Y. D., Lin, C.-C., Wu, Y.-H. ve Zulvia, F. E. (2014). Automatic kernel clustering with bee colony optimization algorithm. *Information Sciences*, 283, 107–122. doi:<https://doi.org/10.1016/j.ins.2014.06.019>
- Li, C. Y., Li, J., Chen, H. L., Jin, M. ve Ren, H. (2021). Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. *Expert Systems with Applications*, 185, 115499. doi:10.1016/J.ESWA.2021.115499
- Liu, Z., Guo, Z. ve Tan, M. (2008). Constructing tumor progression pathways and biomarker discovery with fuzzy kernel kmeans and DNA methylation data. *Cancer Informatics*, 6, 1–7. doi:10.1177/117693510800600007
- MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).
- Manning, C. D., Raghavan, P. ve Hinrich, S. (2009). Information Retrieval - Online edition (c) 2009 Cambridge UP, (c), 569. <http://www-nlp.stanford.edu/IR-book/> adresinden erişildi.
- Matioli, L. C., Santos, S. R., Kleina, M. ve Leite, E. A. (2018). A new algorithm for clustering based on kernel density estimation. *Journal of Applied Statistics*. doi:10.1080/02664763.2016.1277191
- Mirjalili, S. ve Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. doi:10.1016/j.advengsoft.2016.01.008
- Nasiri, J. ve Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. *Cogent Mathematics & Statistics*, 5(1), 1483565. doi:10.1080/25742558.2018.1483565
- Nayak, S. K., Rout, P. K. ve Jagadev, A. K. (2019). Multi-objective clustering: a kernel based approach using Differential Evolution. *Connection Science*, 31(3), 294–321. doi:10.1080/09540091.2019.1603201
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C), 53–65. doi:10.1016/0377-0427(87)90125-7

- Sariman, G. (2011). Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma: K-Means ve K-Medoids Kümeleme Algoritmalarının Karşılaştırılması. *SDÜ Fen Bilimleri Enstitüsü Dergisi*, 15(3), 192–202. doi:10.19113/sdubed.41288
- Schölkopf, B., Smola, A. ve Müller, K.-R. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5), 1299–1319. doi:10.1162/089976698300017467
- Singh, T. (2020). A novel data clustering approach based on whale optimization algorithm. *Expert Systems*, 38. doi:10.1111/exsy.12657
- Tez, O. ve Taşdelen, B. (2017). Çeşitli çekirdek fonksiyonları ile oluşturulan destek vektör makinesi modellerinin performanslarının incelenmesi: Bir klinik uygulama.
- Vaishali P. Patel and L. K. Vishwamitra. 2023. Dynamic Kernel Clustering by Spider Monkey Optimization Algorithm. *J. Classif.* 40, 2 (Jul 2023), 382–406. <https://doi.org/10.1007/s00357-023-09439-x>
- Wang, J. ve Su, X. (2011). An improved K-Means clustering algorithm. *2011 IEEE 3rd International Conference on Communication Software and Networks* içinde (ss. 44–46). doi:10.1109/ICCSN.2011.6014384
- Wang, T., Wang, R., Zhou, J., Jiang, H., Han, S., Wang, L. ve Chen, Y. (2020). A Collaborative Kernel Clustering Algorithm for Non-Linear Data in Peer-to-Peer Networks. *2020 7th International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS 2020*, 849–853. doi:10.1109/ICCSS52145.2020.9336887
- Wilson, A. G., Hu, Z., Salakhutdinov, R. ve Xing, E. P. (2016). Deep kernel learning. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016* içinde .
- Yaraş, E., Akın, E. ve Şakacı, B. K. (2011). Tüketicilerin Çevre Bilinci Düzeylerini Belirlemeye Yönelik Bir Araştırma. *Öneri Dergisi*, 9(35), 117–126.
- YOUSEFİ, T., ODABAS, M. S. ve OKTAŞ, R. (2020). Overview of Different Methods Used in Clustering Algorithms. *Black Sea Journal of Engineering and Science*, 3(4), 13–14. doi:10.34248/bsengineering.698741
- Zhang, L. ve Cao, Q. (2011). A novel ant-based clustering algorithm using the kernel method. *Information Sciences*, 181(20), 4658–4672. doi:10.1016/j.ins.2010.11.005
- Zhao, B., Kwok, J. ve Zhang, C. (2009). *Multiple Kernel Clustering. Proc. 9th SIAM Int. Conf. Data Mining (C. 2)*. doi:10.1137/1.9781611972795.55