T.C.
NECMETTİN ERBAKAN NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GRİ KURT OPTİMİZASYON ALGORİTMASININ VERİ MADENCİLİĞİ PROBLEMLERİNE UYGULANMASI

İhtisam AKTO

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ağustos-2021
KONYA

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.

İmza

İhtisam AKTO

Tarih: …../……/2021

ÖZET

YÜKSEK LİSANS TEZİ

GRİ KURT OPTİMİZASYON ALGORİTMASININ VERİ MADENCİLİĞİ PROBLEMLERİNE UYGULANMASI

İhtisam AKTO

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi ONUR İNAN

2021, 74 Sayfa

Jüri
Dr. Öğr. Üyesi ONUR İNAN
Dr. Öğr. Üyesi Şaban GÜLCÜ
Dr. Öğr. Üyesi Vahit TONGUR

Gelişen teknolojiyle beraber artan veri yoğunluğu, yüksek işleme gücü ve farklı hesaplama olanakları ile birlikte, karmaşık modellemelerin geliştirilmesi ve gelişmiş veri madenciliği uygulamaları kullanılarak temel kriterlere dayalı ileriye dönük daha başarılı tahmin modellemelerinin oluşturulması daha fazla mümkün hale gelmiştir. Veri kümeleri üzerinde uygulanan modelin tahmin başarısı, kuşkusuz hazırlanan modelin başarısına bağlı olarak değişmektedir. Bu yüzden veri madenciliği problemlerinin çözümünde kullanılan algoritma ve algoritmada oluşturulan modelin belirlenmesi uygulamanın başarısı açısından son derece önemlidir. Veri modellerinin karmaşık olması, veriye ait niteliklerin fazlalığı ve veri kümesindeki elemanların fazla olması, doğru analizlerin yapılmasını zorlaştırmaktadır. Bu nedenle özellik seçimi, ayrıklaştırma, kümeleme ve sınıflandırma gibi konular veri madenciliğinde büyük önem arz etmektedir. Gri Kurt Optimizasyonu Algoritması, veri madenciliği problemlerinde kullanılmaya elverişli olması ve yakın zamanda geliştirilen bir optimizasyon algoritması olması nedeniyle veri madenciliği uygulamalarında kümeleme ve sınıflandırma problemlerinin etkin çözümü için kullanılmıştır.

Bu çalışmada, gri kurtların toplumsal davranışlarına dayanan gri kurt optimizasyonu (GWO) algoritması ile en iyi küme merkezlerini aramak suretiyle veri nesnelerini bölümlemek ve kümelemenin akabinde yeni bir yaklaşımla sınıflandırmayı sağlamak amacıyla kullanılmıştır. GWO algoritmasının kümeleme performansı, K-means, K-medoids ve Fuzzy C-means algoritmalarının performanslarıyla karşılaştırılmıştır. Deneyler, GWO algoritmasının diğer kümeleme algoritmalarından daha iyi sonuçlar verdiğini ve alternatif olarak kümeleme problemlerinde kullanılabileceğini göstermektedir. Ayrıca seçilen 6 adet veri seti üzerinde GWO ile kümeleme tabanlı sınıflandırma modelini geliştirerek veri madenciliği uygulamalarının performansını arttırmaya yönelik deneysel çalışma yapılmıştır. GWO algoritması tabanlı geliştirdiğimiz yeni bir sınıflandırma modelinden elde edilen sonuçlar ile literatür taramasında elde edilen sonuçlar karşılaştırılmıştır. Sonuç olarak GWO algoritması tabanlı geliştirilen sınıflandırma modelinin daha başarılı sonuçlar verdiği tespit edilmiştir.

Anahtar Kelimeler: Gri kurt optimizasyon algoritması, kümeleme, sınıflandırma, veri madenciliği.

ABSTRACT

MS/Ph.D THESIS

APPLICATION OF GRAY WOLF OPTIMIZATION ALGORITHM TO DATA MINING PROBLEMS

İhtisam AKTO

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

Advisor: Assistant Professor Onur INAN

2021, 74 Pages

Jury
Advisor Assistant Professor Onur INAN
Assistant Professor Saban GULCU
Assistant Professor Vahit TONGUR

With the developing technology, increasing data density, high processing power and different computational possibilities, it has become more possible to develop complex models and to create more successful predictive models based on basic criteria by using advanced data mining applications. The estimation success of the model applied on the datasets undoubtedly varies depending on the success of the prepared model. Therefore, the algorithm used in solving data mining problems and the determination of the model created in the algorithm are extremely important for the success of the application. The complexity of data models, the excess of data attributes, and the excess of elements in the data set make it difficult to conduct accurate analysis. For this reason, issues such as feature selection, discretization, clustering and classification are of great importance in data mining. The Gray Wolf Optimization Algorithm was chosen because it is suitable for use in data mining problems and is a recently developed optimization algorithm.

In this study, the gray wolf optimization (GWO) algorithm based on the social behavior of gray wolves was used to search for the best cluster centers, to partition data objects and to provide classification with a new approach after clustering. The clustering performance of the GWO algorithm was compared with the performances of the K-means, K-medoids and Fuzzy C-means algorithms. Experiments show that the GWO algorithm gives better results than other clustering algorithms and can be used in clustering problems as an alternative. In addition, an experimental study was conducted to improve the performance of data mining applications by developing the clustering-based classification model with GWO on 6 selected data sets. The results obtained from a new classification model that we developed based on the GWO algorithm were compared with the results obtained in the literature review. As a result, it has been determined that the classification model developed based on the GWO algorithm gives more successful results.

Keywords: Classification, Clustering, data minig, gray wolf optimization algorithm

ÖNSÖZ

Yüksek lisans çalışmalarımın her aşamasında bana destek olan, yol gösteren ve çözüm sunan değerli hocam ve danışmanım Sayın Dr. Öğr. Üyesi Onur İNAN'a teşekkürlerimi sunarım. Bilgilerini ve önerilerini benimle paylaşan tüm hocalarıma, yüksek lisans çalışmam boyunca manevi destekleri ile her daim yanımda olan aileme sonsuz teşekkür ve şükranlarımı sunarım.

İhtisam AKTO
KONYA-2021

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Kısaltmalar

GWO : Gray Wolf Optimization
Min : Minimum
Max : Maximum
SSE : Sum of Squares for Error
YAK : Yapay Arı Kolonisi
PSO : Parçacık Sürü Optimizasyonu
BCWO : Breast Cancer Wisconsin (Original) Data Set
BCWD : Breast Cancer Wisconsin (Diagnostic) Data Set
K-NN : K-Nearest Neighbors (K En Yakın Komşuluk)
YSA : Yapay Sinir Ağları
SVM : Support Vector Machine

1. GİRİŞ

Bilişim alanındaki gelişmeler sayesinde hayatımızın her alanındaki faaliyetlerimiz bu gelişmelerden etkilenmektedir. Bilişimdeki bu gelişmeler sayesinde hayatı kolaylaştıran daha etkin teknolojiler üretilebilmektedir. Bu gelişmelerle beraber devasa boyutlarda veriler oluşmaktadır. Geleneksel yaklaşımlarla, klasik analiz ve raporlama teknikleriyle bu verilerden maksimum düzeyde bilgi almamız gittikçe zor olmaktadır. Aynı zamanda farklı anlama sahip bilgilerin keşfini zorlaştırmakta ve analiz için önemli verilerin gözden kaçmasına sebebiyet verebilmektedir. Bu aşamada, saklı tüm verilerin birbirleriyle olan tüm ilişkilerinin ortaya çıkarılarak değerlendirilmesi ve farklı simülasyonların kullanılması durumunda önemsiz gibi gözüken bir veri, yaşam için hayati bir bilgiye dönüşebilmektedir. Gelişen teknolojiyle beraber artan veri yoğunluğu, yüksek işleme gücü ve farklı hesaplama olanakları ile birlikte, karmaşık modellemelerin geliştirilmesi olanağı artmış ve gelişmiş veri madenciliği uygulamaları kullanılarak temel kriterlere dayalı ileriye dönük daha başarılı tahmin modellemelerinin oluşturulması daha fazla mümkün hale gelmiştir. Veri madenciliği yöntemleri ile eldeki veriler sınıflandırılarak, gruplandırılarak ya da veriler arasında ilişkiler, bağıntılar, istatistiksel sonuçlar oluşturularak farklı modeller oluşturulabilmektedir. Oluşturulan model ile veri kümesinde olmayan yeni bir veri geldiğinde, yeni gelen veri hakkında tahminleme yapma imkânını sağlanmış olur. Yapılan tahminlerin doğruluk derecesi oluşturulmuş olan modelin veri üzerindeki başarımını ortaya koyar. Dolayısı ile bir veri madenciliği uygulamasında hangi algoritma ile daha iyi sonuçlar üretildiği uygulamanın başarımı açısından son derece önemlidir (Coşkun ve Baykal 2011).

Kümeleme, veri madenciliği, istatistiksel veri analizi ve veri kıyaslama gibi konularda verileri gruplandırmak için kullanılan önemli bir yöntemdir. Aynı kümede toplanan verilerin benzerlik derecelerinin yüksek, ayrı kümelerin ise birbiri ile farklılık derecelerinin yüksek olması beklenir. Yani küme içi benzerlik maksimum ve kümeler arası benzerlik minimum olması istenir (Karaboga ve Ozturk 2011).

Literatürde birçok kümeleme algoritması bulunmaktadır. Ancak kümeleme algoritmaları temel olarak iki ana başlık altında toplanmaktadır. Hiyerarşik ve hiyerarşik olmayan kümeleme. Hiyerarşik kümeleme yönteminde gruplandırma işlemi yapılırken küme sayısı belli değildir. Diğer yandan hiyerarşik olmayan kümeleme yöntemi *n* adet

veriyi *K* tane kümeye ayırır. Yani küme sayısına daha önceden karar verilmiştir ve buna göre kümeleme işlemi gerçekleştirilir (Frigui ve Krishnapuram 1999).

Hiyerarşik olmayan kümeleme algoritmalarının başında K-means kümeleme algoritması gelmektedir. K-means algoritması hızlı, basit ve genel olarak başarılı bir algoritma olmasına karşın, başarım oranı, seçilen başlangıç konumuna bağlıdır ve bu sebeple de çoğu zaman başlangıç konumuna en yakın yerel minimuma takılır (MacQueen 1967). Bu sorunu aşmak için araştırmacılar çalışmalarında farklı kümeleme teknikleri geliştirmeye çalışmaktadırlar. İstatistik, graf teorisi, maksimizasyon algoritmaları, yapay sinir ağları, evrimsel algoritmalar ve sürü zekası algoritmaları gibi farklı teknikler, kümeleme problemlerinin çözümlerinde kullanılmaktadır (Karaboga ve Ozturk 2011).

Veri madenciliği algoritmalarıyla yapılan deneysel çalışmaların birbirleriyle karşılaştırılmalarına yönelik birçok çalışma mevcuttur. Bunlardan bir kısmı yakın zamanda geliştirilen algoritmaların, diğer kabul gören algoritmalarla karşılaştırması yapılarak aynı veri seti üzerinde yeni algoritmanın kabul edilebilirliğine dönük gerçekleştirilen çalışmalardan oluşmaktadır. Diğer bir kısmı da farklı veriler üzerinde hibrit çalışma ile veya farklı algoritmalar ile karşılaştırılarak değerlendirildiği çalışmalardan oluşmaktadır.

Günümüzün bilgisayar teknolojisi kadar güncel bir kavram olan optimizasyon kavramı çok çeşitli endüstri kesimlerinde uygulama olanağı bulmuştur. Optimizasyonda amaç maksimum kâr veya minimum maliyeti sağlayacak üretim miktarını, kısıtlara bağlı olarak tespit etmektir (Kara 1986).

Tez çalışmasında, hiyerarşik olmayan kümeleme amacıyla Gri Kurt Optimizasyon Algoritması (GWO) tercih edilmiş olup kullanılan veri setleri üzerinde kümeleme teknikleri kullanılarak sınıflandırma çalışması gerçekleştirilmiştir. Çalışmada kullanılan veri setleri UCI Machine Learning Repository veri tabanından alınmıştır. İlk etapta GWO algoritmasının farklı parametreler ile performansı incelenmiştir. Aynı veri setleri üzerinde K-means algoritması, Fuzzy C-means algortiması ve K-medoids algoritmasının kümeleme problemlerindeki performansı incelenmiştir. Daha sonra sınıflandırma amacıyla GWO algoritması ile öncelikle kümele yapılmış ardından K-nn algoritmasına benzer bir *K* değeri ile yine GWO algoritması kullanılarak yeni bir

yaklaşımla sınıflandırma yapılmış olup çalışmanın başarısı ölçülmüştür. Sınıflandırma sonucunda elde edilen doğruluk başarı oranlarına göre literatürdeki aynı veri setleri üzerinde yapılan çalışmalar incelenerek karşılaştırma yapılmıştır.

2. KAYNAK ARAŞTIRMASI

Literatürde, gerek farklı alanlardaki verilerin kümelenmesi gerekse de farklı algoritmaların kümeleme problemleri üzerinde kullanılması sonucu birçok çalışma mevcuttur. Bu bölümde kümeleme problemlerinin çözümü için yapılan çalışmalar, GWO Algoritması kullanılarak yapılan çalışmalar ile sınıflandırma problemlerinin çözümü üzerine yapılan çalışmalara yönelik kaynak araştırması yapılmıştır.

2.1. Kümeleme Problemi Üzerine Yapılan Çalışmalar

Ewees ve arkadaşları (2021) yaptıkları çalışmada, Diferansiyel Evrim (DE) Algoritmasını ve Muhalefete Dayalı Öğrenmeyi (OBL) birleştirerek iyileştirilmiş Balina Optimizasyon Algoritmasına (WOA - Whale Optimization Algorithm) dayalı yeni bir çok amaçlı optimizasyon yöntemi (MWDEO - Multi-Objective Optimization Method) geliştirmişlerdir. Yaptıkları çalışma sonucunda MWDEO'nun rekabetçi ve farklı türdeki problemleri çözmede etkili olduğunu göstermişlerdir (Ewees, Abd Elaziz et al. 2021).

Swiniarski ve Skowron (2003), yaptıkları çalışmada örüntü tanımada özellik seçimi için kaba kümeleme yöntemlerini uygulamışlardır. Özellik seçimi için kullandıkları algoritma, özellik azaltma için kullanılan temel bileşenler analizinin sonucuna kaba bir küme yönteminin uygulanmasına dayandırmaktadırlar. Sonuç olarak çalışmalarında, önerdikleri PICA ve kaba set yöntemlerine dayalı özellik seçimi ile birlikte sunmuşlardır (Swiniarski ve Skowron 2003).

Karami ve Zapata (2015) yaptıkları çalışmada, Parçacık Sürü Optimizasyonu (PSO) ve K-means algoritmasını kullanarak, içerik merkezli ağlara yapılan saldırıları bertaraf etmek için, gelen saldırıları kümeleyip elde edilen sonuçlara göre önlem alacak bir yaklaşımda bulunmuşlardır. K-means algoritması tek başına kullanıldığında elde edilen küme merkezlerinin yeterli derecede optimize edilmediğini belirterek, iki aşamalı bir çalışma gerçekleştirmişlerdir. Çalışmalarında önerdikleri algoritmanın en iyi küme sayısına, iyi ayrılmış küme sayısına ulaşabildiğini, aynı zamanda yüksek tespit oranını artırabileceğini ve diğer bazı iyi bilinen kümeleme algoritmalarına kıyasla yanlış pozitif oranı azaltabileceğini göstermişlerdir (Karami ve Guerrero-Zapata 2015).

Tsapanos ve arkadaşları (2015) yaptıkları çalışmada, Kernel K-means'e yeni bir yaklaşım getirerek yeni bir model sunmuşlardır. Kernel matrix hesaplama, yeni bir yaklaşımla kernel matris kırpma metodu ve Kernel K-means kümeleme algoritması bölümlerinden oluşan bu yeni yaklaşımlarında da büyük ölçekteki veri setleri üzerinde çalışma gerçekleştirerek kümeleme performansının arttırılmasını hedeflemişlerdir (Tsapanos, Tefas et al. 2015).

Rahman ve Islam (2014) yaptıkları çalışmada, Genetik Algoritma (GA) tabanlı olmak üzere K-means ile bütünleşmiş bir kümeleme yaklaşımı önererek kümeleme problemindeki doğru küme sayısını tespit etme konusunda çalışmışlardır. Doğru sayıda kümeyi otomatik olarak bulabilen ve yeni bir ilk popülasyon seçimi yaklaşımıyla doğru genleri belirleyebilen yeni bir Genetik Algoritma tabanlı kümeleme tekniği önermişlerdir. Bu yaklaşımları ile daha performanslı küme merkezlerinin tespitini yapmışlardır. Çalışmalarında kullandıkları 20 veri seti üzerinde önerdikleri yöntemin, kıyaslama yaptıkları diğer beş metottan genel olarak daha başarılı olduklarını tespit etmişlerdir (Rahman ve Islam 2014).

Banharnsakun ve ark.(2013) yaptıkları çalışmada, saf Yapay Arı Kolonisi algoritmasına yeni bir yaklaşım getirerek veri kümeleme üzerinde çalışmışlardır. Önerdikleri yeni yaklaşıma Best-so-far Artificial Bee Colony ismini vermişlerdir. Birçok veri kümesi üzerinde yaptıkları deneylerde, önerdikleri yaklaşımın literatürde kayda geçilmiş klasik kümeleme algoritmalarından daha iyi veya eş değer sonuçlar elde etmişlerdir (Banharnsakun, Sirinaovakul et al. 2013).

Korurek ve Nizam (2008) yaptıkları çalışmda, QRS komplekslerini gruplandırmak amacıyla Karınca Kolonisi Algoritması (KKA) tabanlı yeni bir kümeleme yaklaşımı üzerinde çalışmışlardır. Önerdikleri yeni yaklaşımlarında, sonuçların doğruluk oranının arttığını ve çalışma sürelerinde düşme görüldüğünü gözlemlemişlerdir (Korurek ve Nizam 2008).

Karaboga ve Ozturk (2011) yaptıkları çalışmada, metasezgisel algoritmalardan Yapay Arı Kolonisi (YAK) algoritması ve PSO ile beraber literatürde sıklıkla kullanılan bazı sınıflandırma algoritmalarının kümeleme analizindeki başarımlarını karşılaştırmışlardır. Çalışmalarında YAK Algoritması, kıyaslama problemlerinde veri kümeleme için kullanılmış ve ABC algoritmasının performansı, Parçacık Sürü

Optimizasyonu (PSO) algoritması ve literatürdeki diğer sınıflandırma tekniği ile on üç adet veri seti üzerinde karşılaştırılmışlardır. Yaptıkları deneysel çalışmaların sonuçlarına göre Yapay Arı Kolonisi algoritmasının kümeleme probleminde verimli sonuçlar elde ettiğini gözlemlemişlerdir (Karaboga ve Ozturk 2011).

Ebrahimi ve Khamehchi (2016) yaptıkları çalışmada, Üretim optimizasyon problemlerini çözmek için sperm balina algoritması (SWA) adı verilen yeni bir optimizasyon algoritmasını önermişlerdir. Bu algoritma, ispermeçet balinasının yaşam tarzına dayanmaktadır. SWA algoritması, GA (Genetik Algoritma) ve PSO (Particle Swarm Optimization) ile karşılaştırmışlardır. Çalışmalarında aldıkları sonuçlara göre SWA'nın problemlerin çoğunu çözmek için GA ve PSO'ya kıyasla daha az NFE'ye ihtiyaç duyduğunu ve aynı zamanda optimum cevabı bulmak için gereken süreyi yarı yarıya azaltabileceği sonucuyla SWA'nın optimizasyon problemlerinde güvenle kullanılabileceğini göstermişlerdir (Ebrahimi ve Khamehchi 2016).

2.2. Sınıflandırma Problemi Üzerine Yapılan Çalışmalar

Moradi ve Rostami (2015), yaptıkları çalışmada sınıflandırma problemlerinin çözümü için graf kümeleme yaklaşımına ve karınca kolonisi optimizasyonuna dayanan yeni bir özellik seçme yöntemi önermişlerdir. Önerdikleri yöntem üç aşamadan oluşmaktadır. İlk adımda, özellik setinin tamamı bir graf ile temsil edilmiştir. İkinci adımda, özellikler bir topluluk algılama algoritması kullanılarak birkaç kümeye bölünmüş ve son olarak üçüncü aşamada ise özelliklerin son alt kümesini seçmek için karınca kolonisi optimizasyonuna dayalı yeni bir arama stratejisini kullanılmışlardır. Yaptıkları deneysel çalışmada, önerdikleri yöntemin sürekli olarak daha iyi sınıflandırma doğrulukları ürettiğini göstermişlerdir (Moradi ve Rostami, 2015).

Khozeimeh ve arkadaşları (2017) yaptıkları çalışmada, siğillerin tedavisinde kullanılan immünoterapi ve kriyoterapi tedavi yöntemlerinin tahmin doğruluğunu tahmin etmek ve uygun tedavi yöntemini belirlemek amacıyla Bulanık Mantık Kural tabanlı bir sistem geliştirmişlerdir. Geliştirdikleri sistem ile yaptıkları deneysel çalışmada doğru tedavi yöntemini % 83.33 ile % 80.7 arasında tahmin etme başarısını göstermişlerdir (Khozeimeh, Alizadehsani et al. 2017).

Liu ve arkadaşları (2011) yaptıkları çalışmada, kaba kümeleme algoritması ile Karınca Koloni Optimizasyonunu (KKO) birlikte kullanarak ayırt edici özellikleri seçmek için özellik seçimi yöntemini geliştirmişlerdir. Geliştirilen bu metotla önerdikleri yöntemin, sınıflandırma performansında diğer popüler özellik seçim algoritmalarından daha iyi performans gösterdiği sonucuna varmışlardır (Liu, Wu et al. 2011).

Choudhari ve Biday (2014) yaptıkları çalışmada, dermaskopik görüntüleri kullanarak cilt hastalığının kanserli olup olmadığının tahmini üzerinde bir sınıflandırma çalışmasını yapmışlardır. Dermaskopik görüntü özelliklerini kullanarak hastalığın kanserli hücreye sahip olup olmadığını tahmin etmişlerdir. Yaptıkları deneysel çalışmada Yapay Sinir Ağlarının cilt kanserinin teşhisinde % 86.66 doğruluk derecesini elde etmişlerdir (Choudhari ve Biday 2014).

Mettleq ve arkadaşları (2020) yaptıkları çalışmada, Dünyada 100'den fazla çeşide sahip mango meyvesine ait 1200'den fazla görüntü içeren bir veri seti üzerinde Mango meyvesinin sınıflandırılması üzerinde bir çalışma yapışlardır. Yaptıkları

çalışmada derin öğrenme tabanlı Convolutional Neural Network (CNN) algoritmasını kullanmışlardır. Deneysel çalışmalarında CNN algoritması ile Mango meyvesini, % 100 ile doğruluk derecesi ile sınıflandırma başarısını göstermişlerdir (Mettleq, Dheir et al. 2020).

Al-Kahlout ve arkadaşları (2021) yaptıkları çalışmada, Eritmato-Skuamöz Hastalıkları (ESD)'nı yüksek doğrulukla sınıflandırmak için geri yayılım ileri besleme metodolojisi dayalı JustNN modeline sahip Yapay Sinir Ağını kullanmışlardır. Bu amaçla UCI makine öğrenme veri deposundan Dermatology veri seti üzerinde yaptıkları deneysel çalışmada önerdikleri modelin %98,36 doğruluk derecesini elde etmişlerdir. Aldıkları sonuçlarla önerdikleri sınıflandırıcının faydalı olduğunu göstermişlerdir (Al-Kahlout, Naeem et al. 2021).

Chandel ve arkadaşları (2016) yaptıkları çalışmada, Tiroid hastalığının teşhisine yönelik KNN ve Naive Bayes sınıflandırıcıları kullanılmıştır. Elde ettikleri sonuçlar, K-NN için %93.44 doğrulukla en doğru sınıflandırıcı olduğunu, Naive Bayes sınıflandırıcının ise %22.56 doğrulukla olduğunu göstermişlerdir. Önerdikleri KNN tekniği ile elde edilen sınıflandırma doğruluğunun daha iyi sonuçlar verdiğini gözlemlemişleridir (Chandel, Kunwar et al. 2016).

Joshi ve ark (2020) yaptıkları çalışmada, tıp alanında karar destek sistemlerinde kullanım amacıyla en uygun algoritmanın belirlenmesine yönelik bir çalışma gerçekleştirmişlerdir. UCI veri deposundan aldıkları 5 adet veri seti üzerinde 6 adet algoritma (J48 karar ağacı, Naive Bayes, IBK, Support Vector Machine, ZeroR ve VFI) ile sınıflandırma çalışmasını yapmışlardır. Genel performansa dayalı gözlemlerine göre, en uygun algoritmaların sırasıyla %80.52, %80.27 ve %79.79 doğruluk değerleriyle VFI, J48 Karar Ağacı ve Naive Bayes olduğu sonucuna varmışlardır (Joshi ve Jetawat 2020).

2.3. Gri Kurt Optimizasyon Algoritması İle Yapılan Çalışmalar

Mirjalili ve arkadaşları (2014) yaptıkları çalışmada, Canis lupus isimli gri kurtların avlanma davranışlarını ve liderlik stratejilerinden ilham alarak meta sezgisel bir yaklaşım ile Gri Kurt Optimizasyonu (GWO - Grey Wolf Optimizer) algoritmasını geliştirmişlerdir. Geliştirdikleri GWO algoritmasının, Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO), Yerçekimi Arama Algoritması (Gravity Search

Algorithm - GSA), Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm - DEA) gibi birçok algoritmaya yakın sonuç verdiğini gözlemlemişlerdir. Önerdikleri algoritmanın mühendislik tasarımı problemlerinde kullanıp geçerli sonuçlar elde ettiğini, bilinmeyen arama uzayları ile zorlu problemlere uygulanabilir olduğunu göstermişlerdir (Mirjalili, Mirjalili et al. 2014).

Kamboj ve arkadaşları (2016) yaptıkları çalışmada, GWO algoritmasını kullanarak dışbükey olmayan ekonomik yük dağıtım sorununun çözümü ile ilgili bir çalışma yapmışlardır. Bir meta-sezgisel arama algoritması olarak geliştirilen av arama, avı çevreleme ve ava saldırma yaklaşımını benimseyen GWO algoritmasını ekonomik yük dağıtım (EYD) probleminin çözümünde kullanmışlardır. Yaptıkları deneysel çalışmada GWO algoritmasını diğer iyi bilinen geleneksel, sezgisel ve meta-sezgisel arama algoritmaları ile karşılaştırmışlar. Yaptıkları karşılaştırmada GWO algoritmasının çok rekabetçi sonuçlar elde ettiğini göstermişlerdir (Kamboj, Bath et al. 2016).

Yusof ve Mustaffa (2015) yaptıkları çalışmada, GWO algoritmasını kullanarak hammadde enerji ve zaman serisinin tahminine yönelik bir çalışma gerçekleştirmişlerdir. Yeni bir 'Swarm Intelligence' dayalı (SI) davranış, yani GWO ile kısa vadeli zaman serisi tahmin için geliştirilmiştir. West Texas Intermediate ham petrol ve benzin fiyatları üzerine model olarak diğer algoritmalarla karşılaştırmışlardır. Çalışma sonucunda GWO'nun diğer sezgisel algoritmalara bir rakip olabileceğini ortaya koymaya çalışmışlardır (Yusof ve Mustaffa 2015).

Mirjalili (2015) yaptığı çalışmada, Çok Katmanlı Algılayıcıların (MLP – Multi Layer Perceptron) eğitiminde GWO algoritmasını önermiştir. Önerilen yöntemin performansını karşılaştırmak için sekiz adet farklı veri setini kullanarak Parçacık Sürüsü Optimizasyonu (PSO), Genetik Algoritma (GA), Karınca Kolonisi Optimizasyonu (ACO), Evrim Stratejisi (ES) ve Popülasyon Tabanlı Artımlı Öğrenme (PBIL) ile test etmiştir. Sonuçlar, GWO algoritmasının rekabetçi sonuçlar elde ettiğini ve yakınlaştırmada yüksek düzeyde doğruluk gösterdiğini gözlemlemiştir (Mirjalili 2015).

Sharma ve Saikia (2015) yaptıkları çalışmada, klasik yöntemlerle kontrol edilen termik güç santrallerinin otomatik üretim sisteminin sağlanması amacıyla GWO algoritması kullanarak otomatik üretim kontrollerini yapmaya çalışmışlardır. Çalışmalarında GWO ile optimize edilen PID denetleyicisinin performansı ile STPP'li

ve STTP'siz sistemlerdeki salınımların büyüklüğü açısından diğer sistemlerden daha büyük olduğunu, PID denetleyici kazanımlarının sağlıklı olduğunu ve sistem koşullarında ve parametrelerindeki büyük değişiklikler için sıfırlamaya gerek olmadığını ortaya koymuşlardır. (Sharma ve Saikia 2015).

Salgotra ve arkadaşları (2020) yaptıkları çalışmada, GWO algoritmasının keşif yeteneklerinin geliştirmek amacıyla Gri Kurt Optimizasyonu Algoritmasının (GWO-E) genişletilmiş bir versiyonu üzerinde çalışma yapmışlardır. Bu amaçla liderlere atanan farklı pozisyonlar nedeniyle arama alanında yeni alanlar keşfetmeyi ve arama ajanları arasında çeşitliliği sağlamak için yinelemelerin ilk yarısında muhalefet tabanlı bir öğrenme yöntemini kullanmışlar. Çalışmalarında aldıkları sonuçlar, Genişletilmiş GWO (GWO-E) algoritmasının GWO, yarasa algoritması, yarasa çiçeği tozlayıcı, tavuk sürüsü optimizasyonu, diferansiyel evrim, ateş böceği algoritması, çiçek tozlaşma algoritması (FPA) ve çekirge optimizasyon algoritmasından daha iyi performans sergilediğini göstermişlerdir (Salgotra, Singh et al. 2020).

Elhariri ve arkadaşları (2015) yaptıkları çalışmada, Destek Vektör Makinaları (SVM) parametrelerinin optimum ayarlarını belirlenmesi amacıyla GWO algoritmasını kullanarak sınıflandırma doğruluğunu iyileştirecek karma model sunmaktadırlar. Önerdikleri yaklaşım üç aşamadan oluşmaktadır. Birinci aşamada ön işleme, ikinci aşamada özellik çıkarma ve son aşamada ise GWO ile SVM'yi birlikte kullanarak sınıflandırma işlemi yapmaktır. Deneysel sonuçlarında, önerdikleri GWO-SVMs yaklaşımının, tipik SVM sınıflandırma algoritmasına kıyasla daha iyi sınıflandırma doğruluğu sağladığını göstermişlerdir (Elhariri, El-Bendary et al. 2015).

Tunç (2016) yaptığı çalışmada, Finans Sektöründe kredi kullanmak için başvuruda bulunan ve müşterilerin kredi onayı için değerlendirilme sürecinde müşteriye ait farklı niteliklerden oluşan veriler, makine öğrenmesi tekniklerini kullanarak, müşteriye ait güvenilebilir bir kredi skorunu belirlemeye çalışmışlardır. Bu sınıflandırma probleminin çözümünde GWO algoritmasını tercih etmiştir. Yaptıkları deneysel çalışmadan elde ettiği sonuçlara göre müşteriye talep ettiği kredinin verilebilir veya verilemez durumunda olduğunu ortaya çıkarmıştır. Çalışmasındaki sonuçlara göre GWO algoritmasından elde edilen değerleri BAYES algoritması ile karşılaştırdığında çok yakın değerde sonuçlar verdiğini göstermiştir (Tunç 2016).

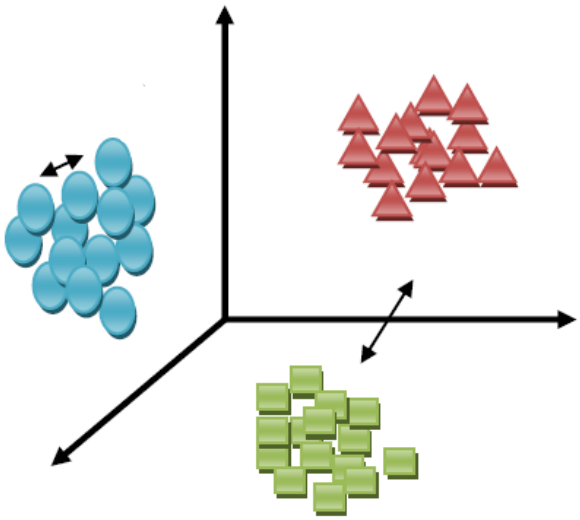
Karakoyun ve arkadaşları (2019) yaptıkları çalışmada, gri kurt optimizasyonu (GWO) algoritmasını kullanarak kümeleme problemlerinde kümelerin optimum merkezlerini araştırarak veri örneklerini bölümlere ayırmaya çalışmışlardır. GWO'nun kümeleme performansı, K-means, Fuzzy C-means ve K-medoids algoritmalarının performansları ile karşılaştırmışlardır. Deneyler sonuçlar neticesinde, GWO algoritmasının genellikle diğer kümeleme algoritmalarından daha iyi sonuçlar verdiğini ve alternatif olarak kümeleme problemine uygulanabileceğini göstermişlerdir (Karakoyun ve Inan 2019).

3. KÜMELEME PROBLEMİ

3.1. Problemin Tanımı ve Temel Kavramlar

Kümeleme, bir veri setinde bulunan öğelerin sahip oldukları özelliklerin benzerliklerine göre gruplara ayırılmasıdır. Bu sayede gruplandırılan veriler, işe yarar uygun anlam kazanmış bilgiler biçimine dönüşür. Daha açık bir ifadeyle, grup bilgisine sahip olmayıp doğal grupları kesin olarak bilinmeyen, bölümleri, değişkenleri ya da bölüm ve değişkenleri birbirleri ile benzer alt kümelere bölmeye yardımcı olan teknikler topluluğudur (Tatlıdil 1996, Özdamar 1999). Kümeleme, bölüm ve bu bölümlere ait verilerin gruplandırılabilmeleri hakkında kesin bir bilginin bulunmadığı bir popülasyondan alınan n tane birimin, p tane değişkene ilişkin gözlem sonuçları ile ilgilenir. Kümelemede benzer öğeleri bir araya getirerek farklı gruplar oluşturulur ve gruplar hiyerarşik bir biçime sokulur. Öğelerin kümelenmesi, gözlemlenen sonuçların minimum kayıpla bir arada toplanmasını sağlar (Lorr 1983).

Veri kümeleme, sınıflar arasındaki ilişkilerin türü hakkında herhangi bir ön bilgi bilmeden bir veri kümesini farklı sınıflara bölmeyi amaçlar (Taherdangkoo, Shirzadi et al. 2013). Aynı grupta olan nesnelerin benzer biçimde ve farklı gruptaki nesnelerin birbirinden farklı olacak şekilde kümelenmesine dikkat edilir. Şekil 3.1’ de görüldüğü gibi nesnelerin uzaklığı aynı küme içerisinde birbirine çok yakın iken, kümelerin birbirine uzaklığı daha fazladır (Hair, Black et al. 1998).



Şekil 3.1. Kümeleme örneği

Kümeleme analizi genel olarak belli aşamalardan meydana gelmektedir. Birinci aşamada, nesneler arasında bir benzerlik veya farklılık anlamına gelen uzaklığın hesaplanmasında kullanılacak uzaklık ölçütünün seçilmesidir. İkinci aşamada, hiyerarşik veya hiyerarşik olmayan bir kümeleme tekniği amaca uygun olarak karar verilir ve ardından seçilen teknik için kullanılacak olan kümeleme yöntemi türü seçilir. Üçüncü aşamada ise seçilen kümeleme yöntemine uygun bir kümeleme algoritması seçilir. Son aşamada ise küme sayısı belirlenerek algoritma ile elde edilen sonuçlar üzerinde değerlendirme yapılır (Sharma 1995).

3.2. Kümeleme Probleminde Benzerlik ve Uzaklık Kavramları

Kümeleme analizinin temeli benzerlik ve uzaklık kavramlarına dayanmaktadır. Kümeleme işleminde aynı kümeye sahip verilerin benzerlikleri dikkate alınırken, farklı kümelerdeki verilerin de birbirlerinden uzaklık durumları ön plana çıkar. İki nesnenin birbirine benzerlik durumu ne kadar fazla bu iki nesne arasındaki ilişki o kadar güçlüdür. Benzerliğin az olması durumunda ise iki nesne arasındaki farklılığın fazla olduğu söylenebilir. Nesneler arasındaki benzerlik ve farlılığın ölçülebilmesi amacıyla farklı metotlar kullanılabilir. Kullanılacak ölçüm metodunun seçilmesinde nesnelerin karakteristik özellikleri yani veri tipleri etkilidir. Bir nesneye ait özelliğin aldığı değerler sonlu sayıda veya sayılabilir sonsuzlukta ise bu veri tipi, kesikli (discrete) veri kategorisine dâhil olur. Ancak nesnenin özelliği birden çok aralıkta, o aralıklardaki her

değeri alabiliyorsa sürekli (continuous) veri kategorisine girer (Dogan 2002, Servi 2009).

Uzaklık, farklılıkların ölçülmesi ile elde edilir. Farklılıklar, çeşitli özelliklere sahip iki nesnenin birbirlerine karşı uyumsuzluk veya zıtlıkların ölçümüdür. Farklılık, aynı zamanda iki obje arasındaki bozukluğun veya düzensizliğin bir ölçüsü olarak da düşünülebilir. Benzerlik ölçümleri ile bir gözlemin diğerlerinden ayırt edilebilmesini sağlar. Bu sayede yapılan gözlemler farklılıklara veya benzerliklere dayalı olarak gruplandırma yapılabilmektedir. Gözlemler alt kümelere veya gruplara bir kere atandıktan sonra her grubun karakteristik özellikleri anlaşılabilir ve kümelerin özellikleri açıklanabilir olmaktadır (Servi 2009).

Küme: Birbirine benzeyen nesnelerden oluşan gruptur.

Centroid: Kümenin merkezini temsil eden varlıktır. Kümenin merkezi, Eşitlik 3.1 ile hesaplanır.

$$X_o = \frac{\sum_{i=1}^{N} x_{mi}}{N} \tag{3.1}$$

Radius: Yarıçap, küme elemanlarının merkeze olan ortalama uzaklığını, ortalamadan sapmayı ifade eder. Radius, Eşitlik 3.2 ile hesaplanır.

$$R = \sqrt{\frac{\sum_{i=1}^{N} \left(x_{mi} - X_0\right)^2}{N}} \tag{3.2}$$

Diameter: Küme içinde bulunan iki nokta arasındaki ortalama mesafedir. Diameter, Eşitlik 3.3 ile hesaplanır.

$$D = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(x_{mi} - x_{mj}\right)^2}{N(N-1)}} \tag{3.3}$$

3.2.1. Verilerin kümelenmesinde kullanılan uzaklık yöntemleri

Her biri *p* adet özellikten (nitelik) oluşan, X_i ve X_j gibi iki nesne arasındaki uzaklık *d(i, j)* olsun. Bu durumda sürekli veriler için uzaklık bulma yöntemlerinden bazıları aşağıda sıralanmıştır.

Öklid (Euclidean) Uzaklığı: En yaygın olarak kullanılan Öklid uzaklığı metodu ile iki nesne arasındaki uzaklık, Eşitlik 3.4'e göre hesaplanır. Öklid uzaklık metodu belirlenen iki nokta (nesne) arasındaki doğrusal uzaklığı hesaplar.

$$d(i,j)=\left[\left|x_{i1}-x_{j1}\right|^2+\left|x_{i2}-x_{j2}\right|^2+\cdots+\left|x_{ip}-x_{jp}\right|^2\right]^{1/2} \tag{3.4}$$

Manhattan Uzaklığı: Manhattan uzaklık metodunda nesneler arasındaki uzaklık, Eşitlik 3.5'ye göre hesaplanır. Manhattan uzaklık bulma yönteminde nesneler arasındaki mutlak uzaklık değeri ele alınır.

$$d(i,j)=\left(\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|+\cdots+\left|x_{ip}-x_{jp}\right|\right) \tag{3.5}$$

Minkowski Uzaklığı: Minkowski metodu kullanılarak iki nesne arasındaki uzaklık, Eşitlik 3.6'e göre hesaplanır.

$$d(i,j)=\left[\left|x_{i1}-x_{j1}\right|^m+\left|x_{i2}-x_{j2}\right|^m+\cdots+\left|x_{ip}-x_{jp}\right|^m\right]^{1/m} \tag{3.6}$$

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

Pearson Uzaklığı: Pearson uzaklık metodu kullanılarak iki nesne arasındaki uzaklığı, Eşitlik 3.7 ile bulunur. Bu eşitlikte kullanılan S_p değeri, uzaklığın hesaplandığı nesneye ait varyanstır.

$$d(i,j)=\sqrt{\left(x_{i1}-x_{j1}\right)^2/S_1^2+\left(x_{i2}-x_{j2}\right)/S_2^2+\cdots+\left(x_{ip}-x_{jp}\right)^2/S_p^2} \tag{3.7}$$

Yukarıda verilen ve mesafe ölçmek için yaygın olarak kullanılan uzaklık bulma yöntemleri değişken değerlerinin sürekli olması durumunda uygulanmaktadır (Tatlıdil 1996, Dogan 2002).

3.2.2. Verilerin kümelenmesinde kullanılan benzerlik yöntemleri

Benzerlik, iki veri arasındaki yakınlığı ifade etmektedir. Aynı zamanda uzaklığın tersi olarak nitelendirilebilir. Benzerlik Dice Eşitlik 3.8, Cosine Eşitlik 3.9, Jaccard Eşitlik 3.10 ve Overlap Eşitlik 3.11 kullanılarak ölçülen benzerlik yöntemlerindendir.

$$sim(x_m, x_j)_{DICE} = \frac{2\sum_{i=1}^{n} x_{mi} x_{ji}}{\sum_{i=1}^{n} x_{mi}^{2} + \sum_{i=1}^{n} x_{ji}^{2}} \tag{3.8}$$

$$sim(x_m, x_j)_{COSINE} = \frac{\sum_{i=1}^{n} x_{mi} x_{ji}}{\sqrt{\sum_{i=1}^{n} x_{mi}^{2} * \sum_{i=1}^{n} x_{ji}^{2}}} \tag{3.9}$$

$$sim(x_m, x_j)_{JACCARD} = \frac{\sum_{i=1}^{n} x_{mi} x_{ji}}{\sum_{i=1}^{n} x_{mi}^{2} + \sum_{i=1}^{n} x_{ji}^{2} - \sum_{i=1}^{n} x_{mi} x_{ji}} \tag{3.10}$$

$$sim(x_m, x_j)_{OVERLAP} = \frac{\sum_{i=1}^{n} x_{mi} x_{ji}}{\left(\sum_{i=1}^{n} x_{mi}^{2}, \sum_{i=1}^{n} x_{ji}^{2}\right)} \tag{3.11}$$

İki vektör arasındaki benzerliği ölçerken değerlerin pozitif ve sürekli olması gerekir. Vektörler birbirinin aynı ise benzerlik 1 olacaktır. Vektörlerden birisi 0, diğer vektör 0'dan büyükse benzerlik 0 olacaktır. Benzerlikte elde edilen sonuç 0 ile 1aralığındadır.

3.3. Kümeleme Yöntemleri

Nesnelerin kümelenmesi sürecinde kullanılmak üzere birden fazla teknik geliştirilmiştir. Kümeleme yöntemlerini, Hiyerarşik yöntemler ve Hiyerarşik olmayan yöntemler şeklinde iki ayrı başlık halinde inceleyebiliriz.

3.3.1. Hiyerarşik kümeleme yöntemleri

Hiyerarşik kümeleme yöntemi, kümeleri art arda birleştirmeye verilen isimdir. Hiyerarşik tekniğin, ağaç yapısı ile gösterilmiş diyagram formuna Dendogram denilir (Lorr 1983). Bir grubu diğer bir gruba bağlayarak bir ağaç şeklinde bir hiyerarşi oluşturur. Hiyerarşik yapının inşa edilme yöntemine göre yöntemler, gruplayıcı yöntem ve bölücü yöntem olmak üzere iki bölümde incelenir.

Hiyerarşik kümeleme yönteminde küme sayısı önceden belirlenmez, dolayısıyla *K* (Küme Sayısı) parametresine ihtiyacı yoktur. Gruplayıcı yöntemde her bir nesne başlangıçta bir küme olarak kabul edilir. Daha sonra eklenen her yeni bir küme mevcut kümelere yakınlık durumlarına göre bir diğer iki küme veya nesne ile yeni bir kümede birleştirilir. Böylece her seferinde küme sayısı bir azaltılır. Bölücü yöntemde ise süreç gruplayıcı hiyerarşik yöntemin tam tersidir. Bu yöntemde tüm nesnelerden oluşan büyük bir küme ile işe başlanır. Birbirine benzemeyen nesneler mevcut kümeden çıkarılarak yeni kümeler oluşturulur. Her nesne tek başına bir küme oluncaya kadar işleme devam edilir (Everitt, Landau et al. 2001).

Ward Yöntemi: Hiyerarşik kümeleme yöntemleri arasında en iyi sonuç veren yöntemlerin başında Ward yöntemi gelmektedir (Hands ve Everitt 1987, Ferreira ve Hitchcock 2009). Ward yöntemi, klasik kareler toplamını ölçü olarak kabul etmekte ve grup içi dağılımı minimize ederek kümelerin oluşmasını sağlayan bir yöntemdir (Murtagh ve Legendre 2014).

Tek Bağlantı (single-link) Yöntemi: En yakın komşuluk tekniğini kullanan bu yöntem, en kısa mesafe esasına dayanır.

Tam Bağlantı (complete-link) Yöntemi: En uzak komşuluk tekniğini kullanan bu yöntem en uzun mesafe esasına dayanır.

Ortalama Bağlantı (average-link) Yöntemi: Karşılıklı iki küme arasındaki tüm mesafelerin ortalamasını esas alan bir yöntemdir.

Merkez (centroid) Bağlantı Yöntemi: Karşılıklı iki kümenin merkezlerinin birbirlerine olan uzaklıklarını esas alan bir yöntemdir.

3.3.2. Hiyerarşik olmayan kümeleme yöntemleri

Kümelenecek nesnelerin kaç kümeye bölüneceğini önceden belirleyen bir kümeleme yöntemidir. Bu yöntem, nesnelerin *K* (Küme Sayısı) adet kümede toplanması esasına dayanmaktadır. *K*'nın değeri, belirlenmiş kriterlere göre bir değer olarak belirlenebileceği gibi kümeleme tekniğinin bir parçası olarak da belirlenebilir. Çünkü benzerlik (uzaklık) matrisinin önceden belirlenmiş olması zorunlu değildir ve temel verinin bilgisayarın çalışması boyunca depolanması zorunlu değildir. Hiyerarşik olmayan yöntemler ile yapılan çalışmalar, hiyerarşik yöntemlere oranla daha hızlı sonuç verirler. Bu sebeple hiyerarşik olmayan yöntemler, büyük veri kümelerine sahip çalışmalarda uygulanabilir (Johnson ve Wichern 1988).

Hiyerarşik olmayan yöntemleri kullanarak geliştirilen birçok algoritma mevcuttur. K-means ve K-medoids algoritmaları örnek olarak verilebilir.

3.4. Optimizasyon

Optimizasyon, bir problemde belirli koşullar altında mümkün olan alternatif çözüm seçenekleri içerisinden en iyi çözümü seçme işlemidir. Optimizasyon problemlerini çözmek için geliştirilen algoritmalara, optimizasyon algoritmaları denmektedir. Optimizasyon problemlerini çözmek için klasik çözüm yöntemlerini kullanan algoritmalar ve sezgisel optimizasyon algoritmaları mevcuttur. Özellikle son yıllarda doğal süreçlerden esinlenilmiş birçok optimizasyon algoritması geliştirilmiştir.

Matematiksel olarak optimizasyon, bir fonksiyonunun amacına uygun olarak maksimize veya minimize edilmesi yani amaç fonksiyonunun en iyi değerini veren noktalardaki değişkenlerin değerinin bulunmasıdır (Kahaner, Cleve et al. 1989).

3.5. Veri Madenciliği

Günümüzde teknolojinin, hayatın her alanında kullanılmasıyla birlikte devasa boyutlarda veriler oluşmaktadır. Bu büyük veri yığınları içerisinde anlamlı bilgiler

ortaya çıkarmak zahmetli bir süreçtir. Veri tabanı sistemlerinin artan kullanımı ve hacimlerindeki olağanüstü artış, organizasyonlar için elde toplanan bu verilerden nasıl faydalanılabileceği problemi ile karşı karşıya kalınmıştır. Geleneksel sorgu (Query) dilleri veya raporlama araçlarının büyük boyutlu veri yığınlarına karşı yeterli olmaması, Veri Tabanlarında Bilgi Keşfi-VTBK (Knowledge Discovery in Databases) adı altında, sürekli ve yeni çözüm arayışlarına yöneltmektedir. Veri Tabanlarında Bilgi Keşfi sürecinde, modelin oluşturulması ve oluşturulan bu modelin değerlendirilmesi aşamaları, veri madenciliğinin (data mining) en önemli bölümünü oluşturmaktadır. Verilerin bilgiye dönüştürülmesi sürecini kolaylaştırmak amacıyla verileri ve veriler arasındaki ilişkileri inceleyerek anlamlı ifadeler veya yararlı bilgilerin keşfini sağlayan veri madenciliği ortaya çıkmıştır. Kısacası verinin bilgiye dönüştürülmesi sürecinde yapılan çalışmalara veri madenciliği denmektedir (Akpınar 2000).

Veri madenciliği alanında yapılan çalışmalar neticesinde elde edilen kazanımlar büyük önem arz etmektedir. Veri madenciliğinde kullanılmak suretiyle birçok yöntem geliştirilmiştir. Veri madenciliği teknikleri ile eldeki veriler gruplandırılarak veya sınıflandırılarak ya da veriler arasında bağlantılar, ilişkiler, istatistiksel sonuçlar oluşturularak modeller geliştirilir. Hazırlanan bu modeller, mevcut problemin çözümünde gösterdiği başarı açısından çok önemlidir.

Veri madenciliğinde kümeleme analizini yapabilmek için birçok kümeleme algoritması geliştirilmiştir. Kümeleme analizini gerçekleştirecek algoritmaların aşağıdaki gereksinimlere sahip olması gerekir (Xu, Wang et al. 2005).

- **Ölçeklenebilirlik:** Verimli ve etkili bir kümeleme yöntemi küçük veri setlerine sahip kümeler üzerinde etkin olmakla beraber, aynı zamanda milyonlarca veri seti içeren büyük bir veri tabanı üzerinde de etkin olmalıdır.

- **Farklı türlere sahip veri setleri üzerinde uygulanabilme:** Çeşitli veri türlerini, yalnızca sayısal değil, aynı zamanda ikili, kategorik ve sıralı verileri veya bu veri türlerinin karışımlarını da kümelemek için verimli ve etkili bir kümelemeye sahip olmalıdır.

- **Rastgele veya biçimsiz düzene sahip kümelerin keşfi:** Bir küme herhangi bir şekilde olabilir. Keyfi şekle sahip kümeleri tespit edebilen algoritmalar geliştirmek önemlidir.

- **Giriş parametrelerini belirlemek için minimum gereksinimler:** Bazı algoritmalar, kullanıcıların küme analizinde belirli parametreleri girmesini gerektirir. Ancak parametreleri belirlemek zordur.

- **Gürültülü veriler ile çalışma yeteneğine sahip olması:** Veri setindeki özniteliklerin bir kısmı eksik veya yanlış verilere sahip olması durumunda da etkinliğini korumalıdır. Gürültü verinin etkisinden bağımsız olmak için iyi bir kümeleme algoritması olması gerekir.

- **Verilerin giriş sırasından bağımsız olma:** Algoritmanın etkinliği veri setindeki verilerin giriş düzeninden bağımsız olabilmelidir.

- **Yüksek boyutluluk:** Yüksek boyutlu verileri işleme yeteneği, iyi bir algoritma için çok önemlidir.

3.6. Kümeleme Algoritmaları

Bu bölümde K-means kümeleme algoritması, Fuzzy C-means kümeleme algoritması ve K-medoids kümeleme algoritması hakkında bilgi verilmiştir.

3.6.1. K-means kümeleme algoritması

En eski veri kümeleme algoritmalarından biri olan K-means kümeleme algoritması 1967 yılında J.B. MacQueen tarafından geliştirilmiştir (MacQueen 1967). K-means kümeleme algoritmasının, küme merkezi hesaplama işleminde genellikle karesel hata kriterini (SSE) kullanılır. En düşük SSE değeri, kümeleme işleminin sonucu için en iyi sonuç olarak kabul edilir. Algoritma temel olarak nesnelerin bağlı oldukları küme merkezine olan karesel uzaklıklarının minimum olması esasına dayanmaktadır (Tan, Steinbach et al. 2013).

K-means algoritması eldeki *n* adet nesneden oluşan veri kümesini *K* adet kümeye bölme prensibine dayanır. Burada *K* tane kümenin, küme merkezlerinin birbirinden uzak olması ve küme içi nesnelerin birbirine yakın olması hedeflenir. Küme benzerliği kümedeki nesnelerin küme merkezine olan ortalama uzaklık değeri ile ölçülür. Ölçüm sonucunda elde edilen değer, kümenin ağırlık merkezini temsil eder. Bir küme içindeki nesnelerin merkeze olan karesel uzaklıklarının küçük olması o küme merkezinin iyi seçildiği anlamına gelir (Xu ve Wunsch 2005).

Elimizde *N* adet nesneden oluşan ve *K* adet kümeye ayrılacak bir veri seti olduğunu düşünelim. m_i, C_i kümesinin merkez noktası ve *x*, C_i kümesine ait bir nesne olmak üzere her bir küme için karesel hata Eşitlik 3.12 ile hesaplanır.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2 \left(m_i, x \right) \tag{3.12}$$

Her küme için bu değerin minimize edilmesi K-means algoritmasının temel hedefidir. K-means algoritmasının dezavantajlarından bir tanesi verinin bölüneceği küme sayısının belirlenmesi işleminin bir standardının olmamasıdır (Jain 2010).

Velmurugan (2014) yaptığı çalışmasında K-means algoritmasını temel olarak aşağıdaki adımlarla tanımlamıştır (Velmurugan 2014):

1. Her bir küme için küme merkezini temsil edecek bir nokta seçilir.
2. Veri kümesine ait her nesne için en yakın olduğu kümeye eklenir.
3. Her küme için küme merkezleri tekrar hesaplanarak güncellenir.
4. Küme merkezleri belirlenen hata payı kadar değişmeyene veya belirlenmişse iterasyon sayısı sonlanıncaya kadar 2. ve 3.adımlar tekrarlanır.

3.6.2. K-medoids kümeleme algoritması

K-medoids algoritması ilk olarak Kaufman ve Rousseeuw tarafından 1987 yılında geliştirilmiştir. Küme merkezlerini belirlerken K-means algoritmasındaki gibi kümedeki nesnelerin ortalama değerlerini kullanmak yerine, küme merkezine en yakın elemanı (medoids) bularak belirlemektedir. Veri topluluğunda aykırı veya gürültülü değerlerin bulunması durumunda K-means algoritmasına oranla daha az hassasiyete sahiptir (Han, Kamber et al. 2001).

K-medoids algoritması, verinin çeşitli yapısal özelliklerini niteleyen *K* adet temsilci nesneyi bulma temeli üzerine kurulmuştur. Temsilci olarak belirlenen nesne medoid olarak isimlendirilir ve kümenin merkezine en yakın temsilcidir. Bir grup veriyi *K* adet kümeye bölerken asıl hedef, benzerlikleri birbirine en yakın olan nesnelerin bir araya getirildiği ve farklı kümelerdeki nesnelerin birbirinden uzak olduğu kümeleri bulmaktır.

Temsilci nesne, diğer nesnelere olan ortalama uzaklığı minimum yapan kümenin en merkezi nesnesidir. Bu nedenle, bu bölünme metodu her bir nesne ve onun referans noktası arasındaki benzersizliklerin toplamını küçültme mantığı esas alınarak uygulanır. Kümeleme literatüründe temsilci nesnelere çoğunlukla merkez tipler (centrotypes) denilmektedir. PAM (Partitioning Around Medoids) algoritmasında temsilci nesneler medoid olarak adlandırılmaktadır (Kaufman ve Rousseeuw 1990). *K* adet temsilci nesneyi bulmayı amaç edinmesinden dolayı, K-medoids metodu olarak isimlendirilmiştir. *K* adet temsilci nesne tespit edildikten sonra gruptaki her bir nesne en yakın olduğu temsilciye atanarak *K* tane küme oluşturulur. Sonraki adımlarda her bir temsilci nesne temsilci olmayan nesne ile değiştirilerek kümelemenin kalitesi artıncaya kadar devam eder. Bu kalite nesne ile ait olduğu kümenin temsilci nesnesi arasındaki ortalama uzaklık maliyet fonksiyonu kullanılarak değerlendirme işlemi yapılır (Işık 2006).

3.6.3. Fuzzy C-means kümeleme algoritması

Fuzzy (Bulanık) C-Means (FCM) algoritması, bulanık bölünmeli kümeleme tekniklerinden en iyi bilinen ve yaygın kullanılan yöntemdir. Fuzzy C-means algoritması 1973 yılında Dunn tarafından ortaya atılmış ve 1981' de Bezdek tarafından geliştirilmiştir (Hoppner ve Klawonn 2000). Fuzzy C-means algoritması, amaç fonksiyonu temelli bir metottur. Fuzzy C-means metodu, nesnelerin iki veya daha fazla kümeye ait olabilmesine izin verir. Bulanık mantık prensibi gereği her veri, kümelerin her birine [0,1] arasında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı "1" olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır. Amaç fonksiyonun belirlenen minimum ilerleme değerine yakınsamasıyla kümeleme işlemi tamamlanır (Hoppner ve Klawonn 2000).

Algoritma, en küçük kareler yönteminin genellemesi olan Eşitlik 3.13 amaç fonksiyonunu öteleyerek minimize etmek amaçlıdır (Hoppner ve Klawonn 2000).

$$Jm = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \qquad , \ 1 \leq \ \mathrm{m} < \infty \tag{3.13}$$

U üyelik matrisi rasgele atanarak algoritma başlatılır. İkinci adımda ise merkez vektörleri hesaplanır. Merkezler Eşitlik 3.14 ile hesaplanır (Hoppner ve Klawonn 2000).

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \, x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3.14}$$

Hesaplanan küme merkezlerine göre, *U* matrisi Eşitlik 3.15 kullanılarak yeniden hesaplanır. Eski *U* matrisi ile yeni *U* matrisi karşılaştırılır ve fark ε'dan küçük olana kadar işlemler devam eder (Moertini 2002).

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_i \right\|}{\left\| x_i - c_k \right\|} \right)^{2/(m-1)}} \tag{3.15}$$

Kümeleme işlemi sonucunda bulanık değerler içeren *U* üyelik matrisi kümelemenin sonucunu yansıtır. İstenirse, berraklaştırma yapılarak bu değerler yuvarlanıp 0 ve 1'lere dönüştürülebilir (Meltem ve Çamurcu 2011).

4. SINIFLANDIRMA

Sınıflandırma, bir veri kümesindeki nesnelerin sahip oldukları özelliklere veya belirli kriterlere göre sınıflara bölen veya kategorilere ayıran bir veri madenciliği işlevidir. Sınıflandırmada amaç, veri kümesindeki her nesne için hedeflenen sınıf değerini veya kategorisini doğru tahmin etmektir. Örneğin, bir bankanın gelen kredi başvuru taleplerini değerlendirirken kredi talebinde bulunan kişileri veya şirketleri düşük, orta veya yüksek riskli olarak tanımlamak için bir sınıflandırma modeli kullanılabilir. En basit sınıflandırma problemi türü sadece iki sınıfa veya kategoriye sahip sınıflandırmadır. Hedef tahmininde sadece iki değer vardır. Düşük kredi notu veya yüksek kredi notu, hasta veya hasta değil vb. şeklinde örneklendirilebilir (Oracle 2021).

Sınıflandırma, veri madenciliği alanındaki önemli araştırma alanlarından biridir. Sınıflandırma problemlerinin çözümünde geliştirilen algoritmalara Sınıflandırma Algoritmaları veya sınıflayıcı denmektedir. Bir sınıflandırma algoritması, eğitim sürecinde veri setindeki özniteliklerin değerleri ile hedefin değerleri arasındaki ilişkileri bulur. Bulunan bu ilişkiler bir model ile özetlenir. Daha sonra özetlenen bu model kullanılarak hedef değeri bilinmeyen yeni veriler üzerinden uygulanır. Tahmin edilen hedef değeri ile gerçek değer karşılaştırılarak sınıflayıcının başarısı değerlendirilir (Oracle 2021).

4.1. Sürü Tabanlı Algoritmalar

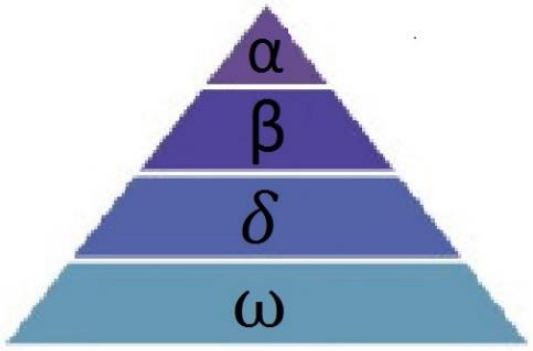
Veri madenciliği problemlerinin çözümünde doğadaki olaylardan veya canlıların davranışlarından esinlenerek birçok meta sezgisel teknik geliştirilmiştir. Özellikle canlıların sürü içerisindeki davranışlarını inceleyerek sürü optimizasyonu tabanlı birçok algoritma geliştirilmiştir. Parçacık Sürü Optimizasyonu (Kennedy ve Eberhart 1995), Karınca Kolonisi Optimizasyonu (Dorigo, Maniezzo et al. 1991), Balina Optimizasyon Algoritması (Mirjalili ve Lewis 2016) örnek olarak verilebilir. Bu algoritmalarla beraber doğadaki canlıların yaşam süreçlerinden esinlenerek birçok yeni optimizasyon algoritması önerilmekte ve önerilen algoritmalar üzerinde geliştirilmeler de yapılmaktadır.

4.2. Gri Kurt Optimizasyon Algoritması

Gri kurtların sosyal davranışına dayanan GWO (Gray Wolf Optimizer) algoritması (Mirjalili, Mirjalili et al. 2014) ilk olarak Mirjalili ve arkadaşları (2014) tarafından geliştirilmiştir. Gri kurtların avlanma davranışlarını, kurtlar arasındaki sosyal hiyerarşisini matematiksel olarak modellemişlerdir. Algoritmanın modellenmesi temel olarak dört adımdan oluşmaktadır:

1. Sosyal Hiyerarşi
2. Avı Çevreleme
3. Avlanma
4. Ava Saldırma

Gri kurt hiyerarşisinde Alfa, Beta, Delta ve Omega olmak üzere 4 tip simülasyon uygulanmakta olup Şekil 4.1. ile ifade edilmiştir (Mirjalili, Mirjalili et al. 2014).



Şekil 4.1. Gri Kurt Hiyerarşisi (Mirjalili, Mirjalili et al. 2014)

Şekil 4.1.'deki gri kurt hiyerarşisinde Alfa kurdu lider olarak bilinir. Alfa kurdu dişi veya erkek olabilir. Alfa kurdu, sürünün avlanmasında, uyku yerinin belirlenmesinde, ava saldırma vb. gibi durumlarda karar verme sorumluluğuna sahiptir (Mirjalili, Mirjalili et al. 2014).

Sosyal hiyerarşide Alfa kurttan sonra Beta kurt gelmektedir. Beta kurt, hiyerarşide Alfa kurdunun yardımcısı olarak kabul edilir. Alfa Kurt ile diğer kurtlar arasında koordinasyonu sağlar. Liderin kararlarının sürüde uygulanmasında önemli rol üstlenir. Alfa kurdunun liderlik özelliğini yitirmesi veya ölmesi durumunda Beta kurt, Alfa kurdun yerine geçer ve sürüye liderlik yapar (Mirjalili, Mirjalili et al. 2014).

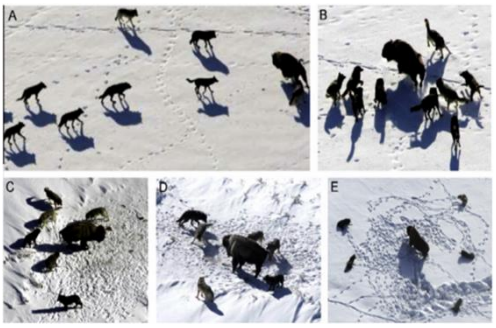
Hiyerarşide Beta kurdundan sonra Delta kurdu gelmektedir. Delta kurdu, yönetsel olarak Alfa ve Beta kurtlarından sonra gelir. Sürüde Alfa ve Beta kurdu bulunmadığı zaman sürüye liderlik etmektedir (Mirjalili, Mirjalili et al. 2014).

Hiyerarşide yönetsel faaliyetlerde bulunmayıp ve sürüdeki en alt aşamada bulunan kurt ise Omega'dır. Av organizasyonu Alfa, Beta ve Delta kurtları tarafından yönlendirilir ve Omega kurdu bu yönlendirmeye uyar. Hiyerarşi gereği en alt seviye olmasından dolayı yemek esnasında lider kurtların doymasını bekler ve daha sonra yemek yer. Omega kurtları, lider kurtlar tarafından seçilir. Omega kurtları sürü içerisinde her ne kadar değersiz gözükse bile kaybolmaları durumunda sürü içerisinde kargaşalar ve sorunlar gözlenmektedir (Mirjalili, Mirjalili et al. 2014).

Kurtların avlanma sürecinde grup olarak birlikte hareket ettikleri ve avlanma sürecini belli bir sıraya göre yönettikleri gözlemlenmiştir (Mirjalili, Mirjalili et al. 2014).

- Avı izleme ve takip
- Ava yaklaşma
- Avı çevreleme
- Avı rahatsız ederek avın durmasını sağlama
- Ava saldırma

Kurtların avlanma hiyerarşisini aşağıdaki Şekil 4.2 ile gösterilebilir (Mirjalili, Mirjalili et al. 2014).



Şekil 4.2. Gri Kurtların Avlanma Hiyerarşisi (Mirjalili, Mirjalili et al. 2014)

4.5.1. Sosyal hiyerarşi

Matematiksel algoritma modelinin sosyal hiyerarşisinde Alfa (α), Beta (β), Delta (δ) ve Omega (ω) kurtları (çözümleri) vardır. Modele göre en iyi çözüm Alfa ile temsil edilir. Daha sonra Beta ikinci en iyi çözüm ve Delta ise üçüncü en iyi çözüm olarak kabul edilir. Omega ise çözüm için en son aday olarak kullanılır. GWO algoritmasında optimizasyon (av) sırasıyla Alfa, Beta ve Delta tarafından yönlendirilir. Omega kurtları ise lider kurtları takip eder (Mirjalili, Mirjalili et al. 2014, Singh ve Singh 2017).

4.5.2. Avı çevreleme

Gri kurtlar, avlanma esnasında avın etrafını sarar. Avın pozisyonuna göre kurtların konumunun güncellenmesi, matematiksel modellemeye göre sırasıyla aşağıdaki Eşitlik 4.1 ve Eşitlik 4.2'de ki gibi olmaktadır. (Mirjalili, Mirjalili et al. 2014).

$$\vec{D} = |\vec{C} . \overrightarrow{X_p}(t) - \vec{X}(t)| \quad (4.1)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} . \vec{D} \quad (4.2)$$

t, mevcut iterasyonu,
$\vec{A}$ ve $\vec{C}$ katsayı vektörlerini,
$\overrightarrow{X_p}$, avın konum vektörünü,
$\vec{X}$, kurdun pozisyon vektörünü göstermektedir.

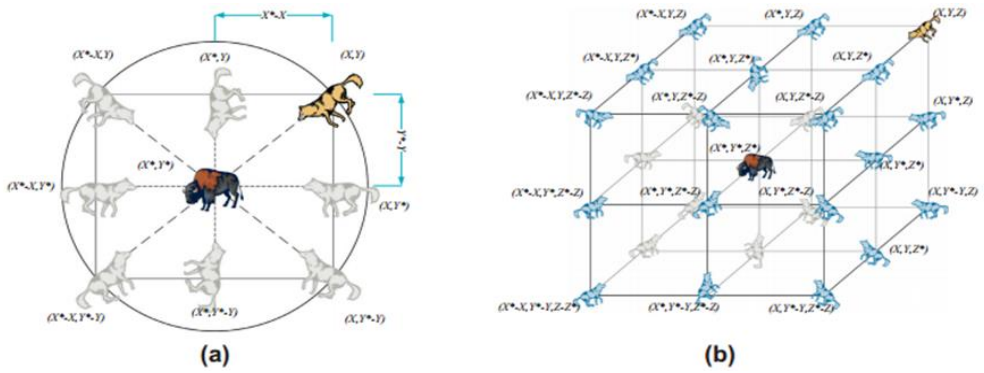
$\vec{A}$ ve $\vec{C}$ vektörleri aşağıdaki gibi hesaplanır:

$$\vec{A} = 2\vec{a} . \vec{r_1} - \vec{a} \quad (4.3)$$

$$\vec{C} = 2 . \vec{r_2} \quad (4.4)$$

Eşitlik 4.3'te $\vec{a}$, doğrusal olarak 2'den 0'a doğru azaltılır. Eşitlik 4.3 ve Eşitlik 4.4'teki $\vec{r_1}$ ve $\vec{r_2}$ ise 0 ile 1 arasında rastgele üretilen birer vektördür.

Şekil 4.3'te görüldüğü gibi, iki boyutlu bir pozisyon vektörü ve olası sonraki konumları gösterilmiştir. (X, Y) konumundaki gri kurt, avının konumuna (X^*, Y^*) göre konumunu güncelleyebilir. En iyi temsilcinin etrafındaki farklı yerlere, mevcut pozisyona göre değeri ayarlanarak ulaşılabilir. Örneğin, $\vec{A}$ ve $\vec{C}$ vektörleri için $\vec{A}$=(1,0) ve $\vec{C}$=(1,1) ayarına göre $(X^* - X, Y^*)$ değerine ulaşılabilir. En iyi ajan, A vektör değeri ile C vektör değerini güncel pozisyona göre değiştirerek daha farklı konumlara ulaşabilir. Şekil 4.3 (b)'de gösterilen 3B uzayda gri kurdun olası sonraki güncelleştirilmiş pozisyonu gösterilmiştir. Yani bir gri kurt (Şekil 4.3'de) herhangi bir rastgele bir yerde av etrafı alanı içinde konumunu güncelleyebilir (Mirjalili, Mirjalili et al. 2014).



Şekil 4.3. 2B ve 3B konum vektörleri ve olası sonraki konumları (Mirjalili, Mirjalili et al. 2014)

Aynı durum *n* boyutlu bir arama uzayında genişletilebilir ve gri kurtlar o ana kadar ki mevcut en iyi çözümün etrafında hiper küpler veya hiper küreler içinde hareket edecektir.

4.5.3. Avlanma

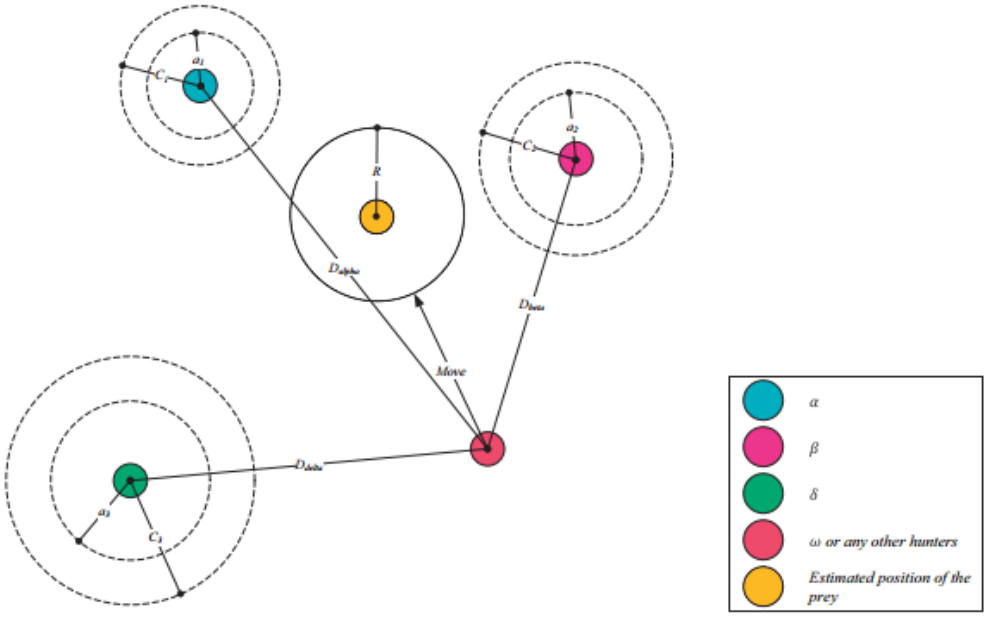
Gri Kurt popülasyonunda Alfa, Beta ve Delta kurtları en iyi konuma sahiptir. Bu sebepten potansiyel bir av hakkında daha iyi bilgiye sahiptirler. Popülasyondaki diğer üyeler avlanmaya uygun daha iyi bir konuma sahip olmak için mevcut konumlarını lider kurtların konumlarına göre güncellerler. Konum güncellemenin matematiksel modellenmesi Eşitlik 4.5, Eşitlik 4.6 ve Eşitlik 4.7 ile gösterilmektedir (Mirjalili, Mirjalili et al. 2014, Singh ve Singh 2017).

$$\overrightarrow{D_\alpha} = \left| \overrightarrow{C_1} . \overrightarrow{X_\alpha} - \vec{X} \right|$$
$$\overrightarrow{D_\beta} = \left| \overrightarrow{C_2} . \overrightarrow{X_\beta} - \vec{X} \right| \quad (4.5)$$
$$\overrightarrow{D_\delta} = \left| \overrightarrow{C_3} . \overrightarrow{X_\delta} - \vec{X} \right|$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} . \left(\overrightarrow{D_\alpha}\right)$$
$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} . \left(\overrightarrow{D_\beta}\right) \quad (4.6)$$
$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} . \left(\overrightarrow{D_\delta}\right)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad (4.7)$$

Şekil 4.4.'te bir 2B arama alanında arama yapan bir kurdun Alfa, Beta ve Delta kurtlarına göre konumunu nasıl güncellediğini göstermektedir. Arama uzayında Alfa, Beta ve Delta'nın konumlarıyla tanımlanan bir daire için son konumunun rastgele bir yerde olacağı gözlemlenebilir. Diğer bir ifadeyle Alfa, Beta, Delta avın o andaki konumunu tahmin ederek diğer kurtların avın etrafında konumlarını rastgele günceller (Mirjalili, Mirjalili et al. 2014).

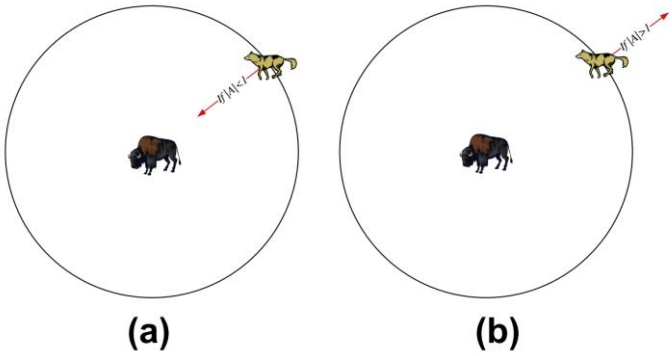


Şekil 4.4. GWO pozisyon güncelleme (Mirjalili, Mirjalili et al. 2014)

4.5.4. Ava saldırma (sömürü)

Gri kurtların av organizasyonundaki son adımları ava saldırmaktır. Saldırı aşaması avı sömürmektir. Algoritma çalışırken $\vec{a}$ vektörünün değerini 2'den 0'a doğru azaltıyoruz. Böylece $\vec{A}$ vektörünün değişimi, $\vec{a}$ vektörüne bağlı olduğundan gittikçe azalmaktadır. Diğer bir ifadeyle $\vec{A}$ vektörü [*-2$\vec{a}$, 2$\vec{a}$*] değerleri arasında rasgele üretilen bir sayıdır.

$\vec{A}$ vektörü [*-1,1*] aralığında rastgele bir değere sahip olduğunda, arama aracısının bir sonraki pozisyonu, o anki mevcut pozisyonu ve avın bulunduğu pozisyonu arasında herhangi bir pozisyonda olabilir. Şekil 4.5'te gösterildiği gibi $\vec{A} < 1$ olduğunda kurtlar ava saldırmaya başlar. Diğer yandan $\vec{A} > 1$ olduğunda kurtlar avdan uzaklaşmaya başlar (Mirjalili, Mirjalili et al. 2014, Rodriguez, Castillo et al. 2017, Singh ve Singh 2017).



Şekil 4.5. Av arama ve saldırma (Mirjalili, Mirjalili et al. 2014)

Şu ana kadar önerilen operatörlerle, GWO algoritmasının arama ajanlarının *Alfa*, *Beta* ve *Delta*'nın konumlarına göre kendi konumlarını güncellemektedirler ve ava doğru saldırmaktadırlar. Bununla beraber, GWO algoritması bu operatörlerle yerel çözümler üzerinde durgunluğa eğilim göstermektedir. Önerilen avı çevreleme mekanizmasının keşfi bir dereceye kadar doğruyu göstermektedir. Ancak GWO, av aramayı vurgulamak için daha fazla destek elemanına ihtiyaç duymaktadır (Mirjalili, Mirjalili et al. 2014).

4.5.5. Av arama (keşif)

Gri kurtların av arama süreci, genellikle *Alfa*, *Beta*, *Delta* kurtlarının konumuna göre gerçekleştirir. Gri kurtlar av aramak için birbirlerinden ayrılırlar ve avı

bulduklarında saldırmak için bir araya gelirler. Şekil 4.5'te olduğu gibi avın durumuna (mesafe, avın tehlike oluşturması vb.) göre gri kurtlar avdan vazgeçecektir (Mirjalili, Mirjalili et al. 2014).

Sapmayı matematiksel olarak modellemek gerekirse, $\vec{A}$ vektörü için -1'den küçük veya 1'den büyük rastgele değerler kullanılır. $\vec{A}$ vektörünün > 1 olması durumunda gri kurtlar avdan uzaklaşarak daha iyi bir av bulmaya yönelirler. GWO'nun av keşfini destekleyen bir başka bileşende $\vec{C}$ vektörüdür. $\vec{C}$ vektörü $[0,2]$ aralığında rasgele değerler içerir. Bu bileşen eşitlikteki mesafeyi tanımlamada avın etkisini skotastik olarak vurgulamak (C > 1) veya önemini azaltmak (C < 1) için rastgele ağırlıklar sağlar (Eşitlik 4.5). Bu durum GWO'nun yerel en yüksek değerden kaçınma durumuna uygun bir şekilde optimizasyon boyunca daha fazla rastgele davranışı göstermeyi sağlar. Buradaki $\vec{C}$ vektörü, $\vec{A}$ vektörünün aksine doğrusal olarak azalma eğilimini göstermez. Keşif sürecinde rastgele değerleri sağlayabilmek için *C* her zaman gereklidir. Bu bileşen, yerel optimal durgunluk durumu veya yineleme durumlarında rastgele değerle sağlar (Mirjalili, Mirjalili et al. 2014).

$\vec{C}$ vektörü, aynı zamanda doğal olarak ava yaklaşmayı engelleyen etmenlerin etkisi olarak kabul edilebilir. Genel olarak, doğadaki engeller kurtların avlanma yollarında ortaya çıkar ve bu durum aslında avlarına hızlı ve rahat bir şekilde yaklaşmalarını engeller. C vektörünün yaptığı tam olarak budur. Bir kurdun konumuna bağlı olarak, avına rastgele bir ağırlık değeri verebilir ve kurtların ava ulaşmasını zorlaştırabilir (Mirjalili, Mirjalili et al. 2014).

Özetle, GWO algoritmasında av arama süreci, rastgele bir popülasyon (aday çözümler) oluşturmakla başlatılır. Bu süreçte her iterasyonda avın muhtemel konumu, *Alfa*, *Beta*, *Delta* kurtları tarafından tahmin edilir. Her aday çözüm, avın konumuna göre uzaklığını günceller. Sırasıyla keşif ve ava saldırı için a parametresi 2'den 0'a doğru düşürülür. Aday çözümler, $|\vec{A}| > 1$ olması durumunda avdan uzaklaşma eğilimini, $|\vec{A}| < 1$ olması durumunda da saldırı eğilimini gösterir. GWO algoritması son olarak, bir bitiş kriteri ile sonlandırılır. GWO algoritmasının sahte kodu Çizelge 4.1 de gösterilmiştir (Mirjalili, Mirjalili et al. 2014).

Çizelge 4.1. GWO Algoritmasının sahte kodu

- Gri Kurt Popülasyonunu başlat

- Başlangıç değerlerini ata
- Herbir temsilci için uygunluk fonksiyon metodu hesapla
- X_α en iyi arama temsilcisini ata
- X_β en iyi 2. arama temsilcisini ata
- X_δ en iyi 3. arama temsilcisini ata
- **Tekrarla**
    - **Her Bir Temsilci İçi Tekrarla**
        - Temsilcinin konumunu güncelle (Eşitlik 4.7)
    - Başlangıç değerlerini güncelle
    - Herbir temsilci için uygunluk fonksiyonu metodu hesapla
    - $X_\alpha - X_\beta - X_\delta$ değerlerini güncelle
    - t = t+1
- **Çevrim sayısı maksimum çevrim sayısından küçük olduğu sürece**
- X_α 'yı geri döndür.

GWO'nun teorik olarak optimizasyon problemlerini nasıl çözebildiğini görmek için aşağıdaki bazı noktalara dikkat edilebilir (Mirjalili, Mirjalili et al. 2014):

- Önerilen sosyal hiyerarşi, GWO'nun yineleme süresince şimdiye kadar elde edilen en iyi çözümleri kaydetmesine yardımcı olur.
- Önerilen çevreleyici mekanizma, çözümlerin etrafında bir hiper-küre olarak daha yüksek boyutlara genişletilebilen daire şeklinde bir mahalleyi tanımlar.
- Rastgele parametreler *A* ve *C*, aday çözümlerin farklı rastgele yarıçaplara sahip hiper-kürelere sahip olmasına yardımcı olur.
- Önerilen avlanma yöntemi, aday çözümlerin avın olası konumunu tespit etmesine izin verir.
- Keşif ve sömürü, a ve A'nın uyarlanabilir değerleriyle garanti edilir.
- Parametrelerinin uyarlamalı değerleri bir ve bir arama ve çıkarma arasında yumuşak bir geçiş sağlar.
- A'nın azalmasıyla, yinelemelerin yarısı keşfe ($|\vec{A}|$ >= 1) ve diğer yarısı sömürüye ($|\vec{A}|$<1) ayrılmıştır.
- GWO'nun ayarlanacak yalnızca iki ana parametresi vardır (*a* ve *C*).

Gri kurtların tüm yaşam döngüsünü taklit etmek için mutasyonu ve diğer evrimsel operatörleri ilave etme ihtimali bulunmaktadır.

5. MATERYAL VE METOT

Bu bölümde tez çalışmasında kullanılan veri setlerinin hazırlanması ve veri setleri üzerinde kullanılan veri ön işleme teknikleri hakkında bilgi verilmiştir. Öncelikle kullanılacak veri setlerinin belirlenmesi ve ardından seçilen veri setleri üzerinde uygulanacak veri ön işleme tekniklerinin uygulanmasında takip edilecek adımlar hakkında genel bilgiler verilmiştir. Yapılan bu uygulamalar neticesinde başarılı sonuçlar elde etmek için algoritmanın eğitiminde kullanılacak optimum eğitim setinin oluşturulması hedeflenmiştir. Bu amaçla veri setindeki nitelikler üzerinde kullanılan normalizasyon işlemi hakkında bilgi paylaşılmıştır. Ayrıca algoritmanın performansını ölçmek amacıyla kullanılan metrikler ile ilgili bilgi verilmiştir.

5.1. Materyal

Bu çalışmada UCI veri ambarından alınan Balance, BCWD, BCWO, Credit, Dermatology, E. Coli, Glass, Iris, Thyroid, Wine ve Hepatit olmak üzere toplam 11 adet veri seti kullanılmıştır.

5.1.1. Veri setleri

UCI veri ambarından alınan ve literatürde sıklıkla kullanılan veri kümeleri ve özellikleri Çizelge 5.1'de gösterilmiştir.

Veri Seti	Özellik Sayısı	Sınıf Sayısı	Örnek Sayısı
Balance	4	3	625
BCWD	30	2	569
BCWO	10	2	683
Credit	14	2	690
Dermatology	34	6	366
E. Coli	7	5	327
Glass	9	6	214
Iris	4	3	150
Thyroid	5	3	215
Wine	13	3	178
Hepatit	19	2	155

Çizelge 5.1. Veri setleri ve özellikleri (Repository)

5.1.2. Eğitim ve test veri setlerinin belirlenmesi

Eğitim ve test verilerin seçimi modelin başarımı üzerinde etkili olan faktörlerdendir. Modelin başarımını test etmek için farklı veri seçme yöntemleri bulunmaktadır. Bu yöntemlerden bir tanesi de Çapraz Doğrulama (k-fold Cross Validation) yöntemidir. Bu yöntemde verilerin toplam sayısı k sayısına bölünür ve k sayısınca veri grubu meydana gelir. Bu veri gruplarından her biri önce eğitimde daha sonra test işleminde değerlendirilir. Böylece her bir veri grubu için eğitim ve test işlemi ayrı ayrı değerlendirilmiş olur. Bu sayede oluşturulan modelin başarısı daha iyi ortaya çıkarılmış olur.

5.1.3. Veri setleri üzerinde yapılan çalışmalar

Algoritmaların öğrenme süreçlerini maksimum seviyede sağlamak amacıyla veri setleri üzerinde normalizasyon işlemleri gerçekleştirilmiştir.

5.1.4. Normalizasyon

Normalizasyon, veri setindeki niteliklerin sahip olduğu ayrık verileri veya veriler arasındaki değerlerin birbirinden farklılıklarının büyük olduğu durumlarda veriyi sabit bir aralık içerisine alma yöntemine verilen isimdir. Sürekli veya ayrık değerlere sahip veriler, algoritmanın öğrenme sürecini ve buna bağlı olarak algoritmanın performansını büyük oranda etkilemektedir. Bu sebepten veriler birbirlerine yaklaştırılarak uzaklıklar minimize edilir ve bu sayede algoritmanın bu durumdan olumsuz etkilenmesi engellenmiş olur. Kümeleme veya sınıflandırma işlemine başlamadan önce veriler belli bir aralığa indirgenerek normalizasyon işlemi yapılır. Normalizasyon işlemini yapmak için farklı teknikler geliştirilmiştir.

Veri setindeki değerleri temsil eden ölçü birimi veri analizini etkileyebilmektedir. Örneğin, ölçü birimlerinin yükseklik için metre'den inç'e veya ağırlık için kilogramdan gram'a dönüştürülmesi çok farklı sonuçlara yol açabilir. Genel olarak, bir özniteliğin daha küçük birimlerle ifade edilmesi, o öznitelik için daha geniş bir aralığa yol açmaktadır. Bu nedenle, böyle bir özniteliğe sahip veriler daha büyük etki göstermekte veya daha fazla ağırlık verme eğiliminde olmaktadır. Ölçüm birimi seçimine bağımlılığı önlemeye yardımcı olmak için, veriler normalize edilmeli veya standartlaştırılmalıdır. Normalize işlemi, genellikle verilerin [−1,1] veya [0.0, 1.0] gibi

daha küçük veya ortak bir aralığa düşecek şekilde dönüştürülmesini içermektedir. (Han, Kamber et al. 2012). Bu çalışmada kullanılan veri setleri üzerinde normalizasyonu sağlamak amacıyla Min-Max yöntemi kullanılmıştır. Bu yöntemle veriler, belirli bir aralıkta aldıkları yeni değerler ile temsil edilmektedirler. Kullanılan Min-Max yöntemi, Eşitlik 5.1'de gösterilmiştir.

$$nX = \frac{X - \min X}{\max X - \min X}\left(nMax - nMin\right) + nMin \tag{5.1}$$

X: Normalize edilecek değeri,

nX: Normalizasyon işleminden sonra *X*'in alacağı değeri,

min*X:* Normalize edilecek değerlerin en küçük (minimum) değerini,

max*X:* Normalize edilecek değerlerin en büyük (maksimum) değerini,

nMax: Yeni maksimum değeri (verilerin normalize edileceği üst sınırını),

nMin: Yeni minimum değer (verilerin normalize edileceği alt sınırını) temsil etmektedir.

Tez çalışmasında veriler [0, 1] aralığına sahip yeni değerler ile normalize edilmiştir. Eşitlik 5.1'deki *nMin=0* ve *nMax=1* olarak belirlenmiş olup Eşitlik 5.2 şeklinde kullanılmıştır.

$$nX = \frac{X - \min X}{\max X - \min X} \tag{5.2}$$

5.1.5. Veri ön işleme teknikleri

Yüksek boyutlara sahip veri tabanları, gürültülü, eksik ve tutarsız verilere sahip karmaşık bir yapı karşısında oldukça hassastır. Bu karmaşıklığa sahip bir veri seti üzerinde yapılacak veri madenciliği uygulaması da düşük kaliteli sonuçlar üretmesine yol açacaktır (Han, Kamber et al. 2012).

Veri Ön İşleme, veri madenciliğinin önemli bir aşamasıdır. Veri madenciliği uygulamalarında arzu edilen sonuçlara ulaşmak için yapılacak uygulama öncesinde verilerin ön işlemden geçirilmesi gerekir. Veri ön işlemede kullanılan teknikler bütününe Veri Ön İşleme Teknikleri adı verilmektedir.

Veri ön işleme teknikleri aşağıdaki gibi sınıflandırılabilir (Han, Kamber et al. 2012).

1. Verinin Temizlenmesi
2. Verinin Birleştirilmesi
3. Verinin İndirgenmesi
4. Verinin Dönüştürülmesi

5.1.5.1. Verinin temizlenmesi

Verilerdeki gürültüyü gidermek, tutarsız ve hatalı verileri düzeltmek için veri temizleme tekniği kullanılabilir. Gürültülü, tutarsız, eksik veya hatalı verilerin yerine sabit bir değer veya aynı niteliğin sahip olduğu değerlerin ortalamasını atayarak düzeltme yapılabilir. Bir diğer yöntem de gürültülü verinin veri setinden çıkarılmasıdır. Ayrıca çeşitli algoritmalar veya istatistiksel analiz yardımıyla da tahmin edilecek uygun bir değer ile de düzeltme yapılabilir. Eksik değer için; bulunduğu öz niteliği devre dışı bırakma veya manuel olarak doldurma, sabit bir değer atama, öz niteliğin ortalama değerini atama yöntemleri kullanılabilir.

5.1.5.2. Verinin bütünleştirilmesi

Veri setlerinde aynı niteliği temsil eden verilerin farklı veri setlerinde farklı kategorik değerlerle gösterilebilmektedir. Örneğin cinsiyet bilgisini saklayan bir öz niteliğin değerleri Erkek/Kadın, E/K, 1/0 veya M/F şeklinde temsil edilebilmektedir. Buna benzer bir niteliğin aynı değeri temsil eden değerlerini tek bir alan ile gösterilmesi veri madenciliği sürecinin hızını ve doğruluğunu arttırmaya yardımcı olur. Aynı zamanda yapılacak uygulamanın kaliteli sonuçlar üretmesi açısından büyük önem taşımaktadır. Verinin tek tip haline getirilmesine, verinin birleştirilmesi adını almaktadır. Verinin birleştirilmesi ile veri setindeki veriler arasında uyumu sağlar ve farklı veri setleri ile ilişkilendirilmesi ihtiyacı karşısında bütünselliği ve doğru eşleştirmeyi sağlar.

5.1.5.3. Verinin indirgemesi

Veri madenciliği uygulamalarında veri setinin çok büyük boyutlarda olması veri analinizi ve uygulamanın kaliteli sonuçlar üretmesini zorlaştırmaktadır. Karmaşık yapıdaki büyük miktardaki veriler üzerinde kısa sürede analiz yapmak uzun zaman alabileceği gibi imkânsız hale de gelebilir. Veri setinde bulunan niteliklerden bazılarının uygulamaya hiçbir etkisinin olmadığı durumlarda etkisiz niteliklerin veri setinden çıkarılması veya hesaplama sürecinde devre dışı bırakılması işlemine verinin indirgenmesi denmektedir. Uygulamaya hiçbir etkisi olmayan verilerin veri setinden çıkarılmasıyla verinin boyutu küçültülebilir. Bu durum uygulamanın daha kısa sürede çalışmasını sağlayacaktır. Verinin indirgenebilmesi için uygulamanın verinin indirgenmesinden önce elde edilen sonuçlar ile verinin indirgenmesinden sonra elde edilen sonuçların aynı olması gerekir. Veri indirgeme işleminde rastgele nitelik azaltma, histogram, kümeleme, regresyon, örnekleme, genelleme, verinin sıkıştırılması, sayısal azaltma ve boyut küçültme gibi yöntemler kullanılabilmektedir (Han, Kamber et al. 2012).

5.1.5.4. Verinin dönüştürülmesi

Veri setindeki niteliklerin sahip olduğu değerlerin aynı anlamı taşıyacak veya aynı sonucu verecek başka değerler ile temsil edilmesi işlemine Verinin Dönüştürülmesi denmektedir. Veri setindeki nitelikler arasında ortalama değerleri ve varyans açısından büyük farklılıkların olması durumunda niteliklerin sonuca etkisi çok büyük veya çok az olabilmektedir. Bu nedenle dönüştürme işlemi, veri madenciliği uygulaması açısından büyük önem taşımaktadır. Verinin dönüştürülmesi yöntemlerine, sözel değerlere sahip niteliklerin sayısallaştırılması, değerler arasındaki büyük farklılıkların sabit bir aralığa çekilmesi gibi yöntemler örnek olarak verilebilir.

5.1.6. Kullanılan metrikler

Veri madenciliği uygulamalarında yapılan çalışmanın sonucunda elde edilen verilerin kabul edilebilir uygun bir karşılaştırma aracın tarafından değerlendirilmesine ihtiyaç vardır. Aynı veriler ile çalıştırılan algoritmalardan elde edilen sonuçların performanslarını karşılaştırıp yorumlanabilmesi için doğru bir performans karşılaştırma yönteminin belirlenmesi gerekir. Algoritmaların kümeleme performanslarını ölçmek amacıyla Toplam Karesel Uzaklık yöntemi tercih edilmiştir. Sınıflandırma çalışmasında performans kriteri olarak Doğruluk metriği kullanılmıştır.

5.1.6.1. Toplam karesel uzaklık yöntemi

Kümeleme analizinde nesneler kümelere ayrılırken nesnelerin birbirlerine olan benzerlikleri veya birbirlerinden farklılıkları dikkate alınarak yapılmaktadır. Aynı kümede yer alan nesnelerin küme merkezine olan uzaklıklarının minimum olması beklenmektedir. Nesnelerin küme merkezine olan uzaklıklarının hesaplanması için farklı yöntemler kullanılmaktadır. Bu modellerden bir tanesi de Toplam Karesel Uzaklık yöntemidir. İyi bir küme merkezinin belirlenmesi, Toplam karesel uzaklık değerinin en küçük değerine sahip olması ile mümkündür (Wan, Wang et al. 2012).

5.1.6.2. Doğruluk (Accuracy) yöntemi

Algoritmaların performanslarını ölçmek amacıyla bir çok kriter kullanılmaktadır. Sınıflandırıcıların ne kadar başarılı sonuçlar verdiğini bilmek için kullanılan sınıflandırma tekniklerinin performansının değerlendirilmesi gerekir. Doğrulama tekniği, sınıfları doğru tahmin etmek için kullanılan bir ölçüm yöntemidir. Doğru tahmin edilen sınıflar, gerçek sınıfa benzediği için doğru olarak değerlendirilir. Diğer yandan, yanlış sınıf ise hata olarak kabul edilir. Sınıflandırıcının öğrenme performansını ölçme yöntemlerinden bir tanesi de hata oranına ve ayrıca doğru tahmin edilen sınıfların toplam gerçek veriler üzerindeki yüzdesine bağlıdır. Doğruluk yüzdesi Eşitlik 5.3'te verilmiştir.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.3}$$

Ayrıca, algoritmanın sınıflandırma performanslarını ölçmek amacıyla duyarlılık (sensitivity) ve özgüllük (specificity) metrikleri kullanılmaktadır. Bu metrikler sırasıyla Eşitlik 5.4 ve Eşitlik 5.5'te verilmiştir.

$$Duyarlılık\ (Sensitivity) = \frac{TP}{TP + FN} \tag{5.4}$$

$$Özgüllük\ (Specificity) = \frac{TN}{TN + FP} \tag{5.5}$$

TP: Pozitif sınıfa ait verilerden kaç tanesinin doğru sınıflandırıldığını,

TN: Negatif sınıfa ait verilerden kaç tanesinin doğru sınıflandırıldığını

FP: Pozitif sınıfa ait olması gereken verilerden kaç tanesinin negatif sınıflandırıldığını,

FN: Negatif sınıfa ait olması gereken verilerden kaç tanesinin pozitif sınıflandırıldığını temsil etmektedir.

Sensitivity, sınıflandırıcının pozitifleri doğru belirlemedeki başarısı ortaya koymaktadır. Specificity, sınıflandırıcının negatifleri doğru belirlemedeki başarısını göstermektedir. Eğer sensitivity 1 ise tüm pozitiflerin doğru sınıflandırıldığı, specificity 1 ise tüm negatiflerin doğru bir şekilde sınıflandırıldığı anlamına gelir.

5.2. Metot

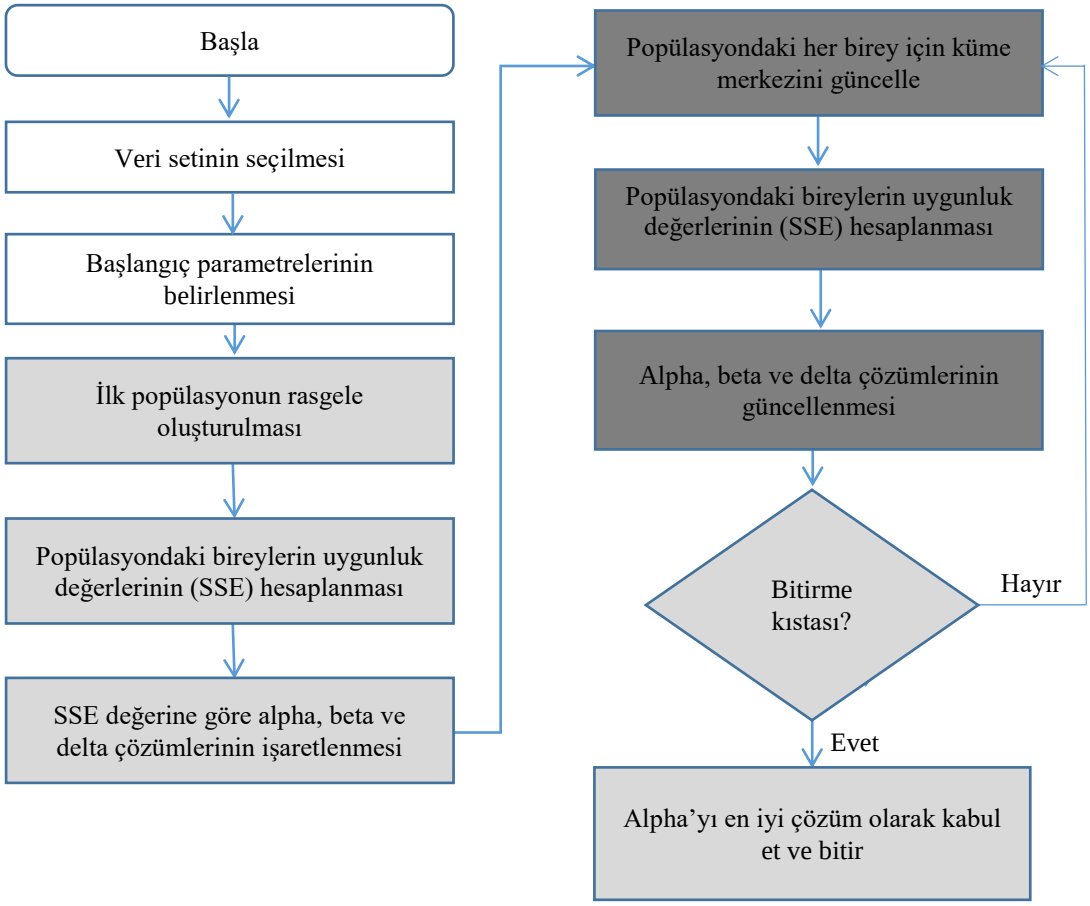
Bu çalışmada kümeleme ile ilgili deneysel çalışmada GWO algoritması kullanılmıştır. Sınıflandırma ile ilgili yapılan çalışmada ise GWO algoritması ile kümeleme yaptıktan sonra K-NN algoritmasına benzer şekilde yine GWO algoritması ile K sayısınca elaman içerisinde maksimum sayıya sahip sınıf etiketi ile sınıflandırma yapılmıştır.

5.2.1. GWO algoritmasının kümeleme problemlerinde kullanılması

Kümeleme çalışmasında kullanılan veri setleri üzerinde öncelikle veri ön işleme adımlarından biri olan normalizasyon (min-max) uygulanarak aykırı verilerin olumsuz etkileri azaltılmıştır. Kıyas yapılan diğer algoritmalar ile GWO algoritması 30 tekrar ile çalıştırılmıştır. Bu uygulama kısmında algoritmaların temel amacı SSE değerini

minimize etmek olmuştur. Kıyas metriği olarak algoritmaların 30 tekrarda elde ettikleri ortalama SSE değeri kullanılmıştır.

GWO algoritmasının kümeleme problemine uygulanması Şekil 5.1'de gösterilmiştir. Algoritma, parametrelerin belirlenip başlangıç değerlerinin ataması ile başlar. Daha sonra sınırlar içinde rastgele küme merkezleri ile ilk popülasyon oluşturulur. Popülasyondaki her bir birey küme merkezlerinin değerlerini tutacak şekilde oluşturulur. Rasgele üretilen popülasyonun her bir bireyi için küme merkezlerine göre uygunluk değeri (SSE) hesaplanır. En iyi uygunluk değerine sahip üç birey iyiden kötüye doğru sırasıyla alfa, beta ve delta olarak atanır. Bundan sonra popülasyondaki üyelerin (küme merkezleri) konumlarını güncellemek için bir döngü oluşturulur. Konum güncelleme aşamasında alfa, beta ve delta bireyleri popülasyona liderlik yapmaktadır. Ayrıca güncellenen konumların çözüm sınırları içerisinde olmasına dikkat edilir. Çözüm uzayının dışına taşan değerler sınırlara çekilir. Konum güncelleme döngüsünün her iterasyonunda, elde edilen yeni konumlara ve bu konumların sahip olduğu uygunluk değerlerine göre; alfa, beta ve delta bireyleri güncellenir. Döngüyü bitiren kıstas sağlandığında algoritmanın sonucu olarak alfa bireyinin sahip olduğu konum en iyi küme merkezleri olarak algoritmanın çıkış verisi olur.



Şekil 5.1. GWO algoritmasının kümeleme problemine uygulanması

Kümeleme için kullanılan parametreler Çizelge 5.2'te verilmiştir.

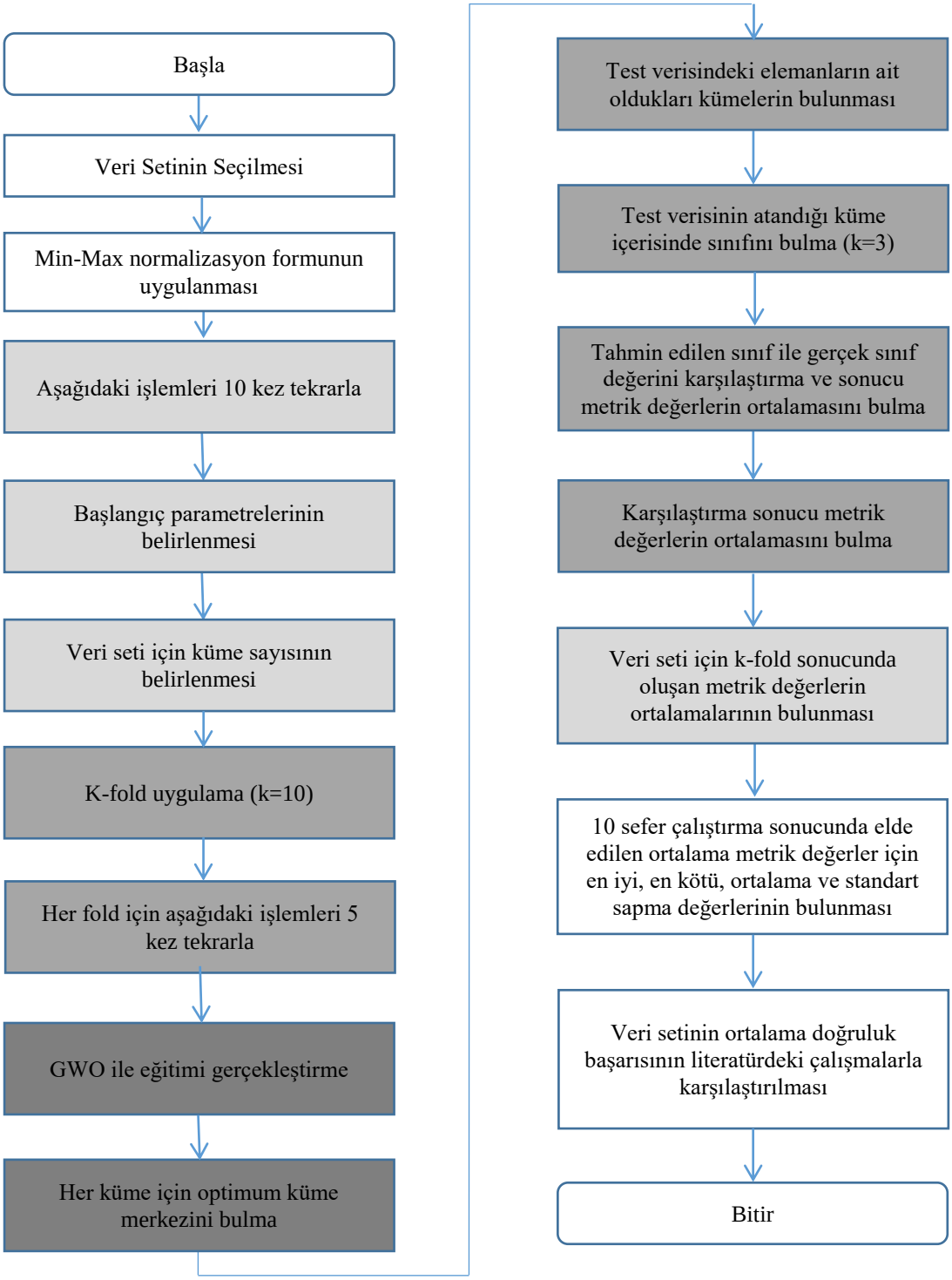
Parametre	**Değer**
Uygulamayı çalıştırma sayısı	30
İterasyon	1000
Popülasyon	100

Çizelge 5.2. Sınıflandırma işleminde kullanılan başlangıç parametreleri

5.2.2. GWO algoritmasının sınıflandırma problemlerinde kullanılması

Tez çalışmasında kullanılan veri setleri üzerinde öncelikle Min-Max tekniği ile normalizasyon işlemleri yapılmıştır. Ardından çapraz doğrulama ile her bir veri seti için foldlar belirlenmiştir. Rastgele başlangıç parametreleriyle Her fold (çapraz doğrulama) için optimum küme merkezleri belirlenerek algoritmanın eğitimi sağlanmıştır. Ardından test veri setindeki her eleman için küme merkezine olan uzaklıkları hesaplanarak ait

oldukları kümenin tespiti yapılmıştır. Kümesi belirlenen test verisinin atandığı küme içerisindeki k-en yakın komşuluk sayısı kadar (K=3) veriden en fazla sınıfa sahip veri ile aynı sınıfa atanmıştır. Tüm foldlar (çapraz doğrulamalar) için işlem bittikten sonra tüm metriklerin ortalama değerleri tespit edilmiştir. 10 sefer çalıştırılan ve her seferinde elde edilen ortalama değerlerin doğruluk, özgünlük ve öznellik için en iyi, en kötü, ortalama ve standart sapma değerleri tespit edilmiştir. Her veri seti için elde edilen ortalama doğruluk dereceleri ile yapılan literatür taraması sonucunda aynı veri setleri üzerinde yapılan çalışmalarda elde edilen doğruluk dereceleri karşılaştırılmıştır. Uygulama süreci aşağıdaki Şekil 5.2'deki diyagramda gösterilmiştir.



Şekil 5.2. Uygulama Akış Diyagramı

Şekil 5.2'de görüldüğü gibi sınıflandırma yapılırken ki uygulamanın akışı aşağıdaki gibi gerçekleştirilmiştir.

1. Deneysel çalışmada kullanılacak veri setleri belirlenir.
2. Veri setleri üzerinde Min-Max normalizasyonu uygulanır.
3. Uygulama 10 kez çalıştırılmıştır.
    - **a.** Başlangıç parametreleri belirlenir.
    - **b.** Her bir veri setindeki sınıf sayısı kadar küme sayısı belirlenir.
    - **c.** K-fold (kfold=10) ile veri seti parçaya ayrılır.
        - **i.** Her fold için 5 sefer aşağıdaki işlemler tekrarlanır.
            1. Eğitim verisi üzerinde algoritmanın eğitimi gerçekleştirilir.
            2. Her bir kümenin en iyi küme merkezi tespit edilir.
        - **ii.** Test verilerindeki her bir verinin küme merkezlerine ait olan uzaklıkları kullanılarak ait oldukları küme tespit edilir.
        - **iii.** Kümesi belirlenen test verisinin atandığı küme içerisindeki K sayısı kadar (K=3) veriden en fazla sınıfa sahip veri ile aynı sınıfa atanır.
        - **iv.** Fold'daki test için ayrılmış veri setindeki verilerin tahmin edilen sınıfları ile gerçek sınıfları karşılaştırılarak ortalama doğruluk, sensitivity ve specificity değerleri bulunur.
    - **d.** Tüm fold'lar için işlem bittikten sonra tüm değerlerin ortalaması tespit edilir.
4. Algoritmanın her çalıştırılmasında (sayaç=10) tüm foldlar sonucunda elde edilen ortalama değerler üzerinden doğruluk, sensitivity ve specificity için en iyi, en kötü, ortalama ve standart sapma değerleri tespit edilir.
5. Elde edilen ortalama doğruluk başarısı ile yapılan literatür taraması sonucunda aynı veri setleri üzerinde yapılan çalışmalarda elde edilen doğruluk dereceleri karşılaştırılmıştır.

Sınıflandırma için kullanılan parametreler Çizelge 5.3'te verilmiştir.

Parametre	**Değer**
Uygulamayı çalıştırma sayısı	10
k-fold	10
Her fold'da optimum küme merkezi için çalıştırma sayısı	5
İterasyon	1000
Popülasyon	100

Çizelge 5.3. Sınıflandırma işleminde kullanılan başlangıç parametreleri

6. DENEYSEL SONUÇLAR

Algoritmaların kümeleme problemindeki performanslarını karşılaştırmak için UCI Machine Learning Repository'den alınan veri setleri kullanılmıştır. Veri setleri hakkında ayrıntılı bilgi Çizelge 4.1'de verilmiştir.

Tez çalışmasında deneyler Intel® Core™ i7-8700 CPU 3.1 GHz İşlemci, 16 GB RAM, Windows 10 Pro (64-bit) İşletim Sistemine sahip makine üzerinde gerçekleştirilmiştir.

6.1. Kümeleme Sonuçları

K-means, K-medoids ve Fuzzy C-means algoritmalarının parametre olarak yalnızca maksimum yineleme sayısı mevcuttur. Bununla birlikte, GWO algoritmasının bir dizi parametresi vardır. Karşılıklı tek parametre, algoritmalar için maksimum yineleme sayısıdır ve iterasyon değeri 1000'dir. Popülasyon boyutu 100 olarak belirlendi, $\vec{a}$ 2 ile başlar ve doğrusal olarak 0'a doğru düşer ve $\vec{r}_1, \vec{r}_2$ aralığı [0, 1] arasında rastgele oluşturulur.

SSE değeri tüm algoritmalar için amaç fonksiyonu olarak kullanılır. Algoritmaların amacı, SSE değerini en aza indiren en iyi küme merkezlerini bulmaktır. Algoritmalar 30 kez çalışmış ve elde edilen sonuçlar Çizelge 6.2'de sunulmuştur; burada 30 kez çalışmanın sonucunda ***B*** en iyi, ***W*** en kötü, ***A*** ortalama ve ***S***, standart sapma sonucudur. Her veri kümesi için, herhangi bir algoritma tarafından üretilen en iyi ortalama sonuç **kalın (bold)** olarak işaretlenmiştir. Çizelge 6.1'de verilen sonuçlara göre, GWO algoritması altı veri seti (Balance, BCWD, Credit, Iris, Thyroid ve Wine) için diğer algoritmalardan daha iyi çözümler üretmiştir. BCWO veri seti için, GWO ve K-means aynı sonuca sahiptir ve diğer algoritmaların çözümünden daha iyidir. K-means algoritmasının, veri setlerinin geri kalanında (Dermotology, E.Coli ve Glass) daha iyi çözümler ürettiği görülmüştür. K-medoids ve Fuzzy C-means kümeleme algoritmaları diğer algoritmalara göre herhangi bir veri seti üzerinde daha iyi sonuç üretememiştir.

Data sets		K-means	K-medoids	Fuzzy C-means	GWO
Balance	B	1423.8514	1672.4587	1722.2446	1423.8205
	W	1433.0977	1822.7225	1722.2446	1423.8213
	A	1425.7619	1716.0266	1722.2446	**1423.8209**
	S	2.05E+00	3.44E+01	1.16E-12	1.95E-04
BCWO	B	1.34E+154	1.79E+308	7.63E+156	1.34E+154
	W	1.34E+154	1.79E+308	7.63E+156	1.34E+154
	A	**1.34E+154**	1.79E+308	7.63E+156	**1.34E+154**
	S	0.00E+00	0.00E+00	0.00E+00	0.00E+00
BCWD	B	2986.9613	3311.5612	3286.1132	2964.3876
	W	2988.4278	4755.7844	3286.1132	2964.3906
	A	2988.0856	3755.4299	3286.1132	**2964.3888**
	S	6.31E-01	4.51E+02	3.68E-13	7.96E-04
Credit	B	748491.65	562404.23	759180.47	556743.89
	W	808744.44	670524.54	759180.47	557162.85
	A	796620.05	596216.51	759180.47	**556797.08**
	S	2.33E+04	3.46E+04	6.49E-11	1.00E+02
Dermatology	B	2021.0276	2792.4665	5196.3797	2221.5083
	W	2298.0339	3275.9505	5196.3797	2439.2836
	A	**2088.3342**	2978.1435	5196.3797	2342.1929
	S	6.82E+01	1.19E+02	2.50E-11	4.49E+01
E. Coli	B	66.0246	127.784	108.4402	65.6273
	W	70.3212	182.2736	108.4402	74.9942
	A	**68.105**	152.0576	108.4402	71.5338
	S	1.10E+00	1.56E+01	9.05E-11	2.67E+00
Glass	B	213.4205	303.9722	400.9817	275.7853
	W	266.5812	424.9596	404.2808	436.4331
	A	**240.9204**	338.7686	402.0121	313.4421
	S	1.43E+01	2.54E+01	1.49E+00	2.74E+01
Iris	B	97.3259	182.0554	106.3591	96.6567
	W	122.2789	253.9361	106.3591	120.8957
	A	102.328	210.6296	106.3591	**98.4404**
	S	1.01E+01	2.09E+01	8.29E-14	6.11E+00
Thyroid	B	1988.0143	2076.709	2812.49	1868.262
	W	2019.3404	2390.4087	2812.49	1940.0841
	A	2009.7801	2194.24	2812.49	**1905.7626**
	S	9.09E+00	7.96E+01	3.90E-11	1.99E+01
Wine	B	16555.68	16901.08	17128.45	16305.47
	W	18436.95	26491.45	17128.45	16336.63
	A	17788.93	20476.13	17128.45	**16316.89**
	S	8.90E+02	2.68E+03	6.26E-12	8.49E+00

Çizelge 6.1. Algoritmaların 30 defa çalıştırma sonucu SSE değerleri

Çizelge 6.2'de algoritmalar 30 kez çalışma sonucunda amaç fonksiyonun ortalama değeri ve bu sonuçlara göre algoritmaların her bir veri seti üzerindeki sıralaması algoritmaların ortalama sıralaması ile verilmiştir. Ortalama sırasına göre, GWO algoritmasının diğer kümeleme algoritmalarına göre daha iyi sıralamaya sahip olduğunu göstermektedir. GWO algoritması 1.3 ile en iyi ortalama sıraya sahiptir ve K-means, Fuzzy C-means ve K-medoids algoritmaları sırasıyla 1.9, 3.3 ve 3.4 ortalamasına sahiptir.

	K-means	**K-medoids**	**Fuzzy C-means**	**GWO**
Balance	1425.7619	1716.0266	1722.2446	1423.8209
	2	3	4	1
BCWO	1.34E+154	1.79E+308	7.63E+156	1.34E+154
	1	4	3	1
BCWD	2988.0856	3755.4299	3286.1132	2964.3888
	2	4	3	1
Credit	796620.05	596216.51	759180.47	556797.08
	4	2	3	1
Dermatology	2088.3342	2978.1435	5196.3797	2342.1929
	1	3	4	2
E. Coli	68.105	152.0576	108.4402	71.5338
	1	4	3	2
Glass	240.9204	338.7686	402.0121	313.4421
	1	3	4	2
Iris	102.328	210.6296	106.3591	98.4404
	2	4	3	1
Thyroid	2009.7801	2194.24	2812.49	1905.7626
	2	3	4	1
Wine	17788.93	20476.13	17128.45	16316.89
	3	4	2	1
Avg. Rank	1.9	3.4	3.3	**1.3**

Çizelge 6.2. Algoritmaların SSE Ortalama Sıralama Sonuçları

Yapılan çalışmalar neticesinde GWO algoritmasının K-means, K-medoids ve Fuzzy C-means algoritmalarına göre daha iyi performans sergilediğini söyleyebiliriz. Bu sonuçlara bakarak kümeleme işlemlerinde GWO algoritmasının daha başarılı sonuçlar ürettiğini söylemek mümkündür. Farklı veri setleri ve farklı algoritmalarla GWO algoritmasının başarısı karşılaştırılabilir.

Kümeleme çalışmasında GWO, K-means, K-medoids ve Fuzzy C-means algoritmalarının kümeleme problemlerindeki performansları incelenmiştir. Genel olarak algoritmaların kümeleme başarılarının birbirine yakın olduğu gözlemekle beraber seçilen veri setlerinde GWO algoritmasının daha başarılı sonuçlar ürettiği görülmektedir.

6.2. Sınıflandırma Sonuçları

GWO algoritması ile aşağıdaki veri setleri üzerinde sınıflandırma çalışması yapılmıştır. Sınıflandırma sonucunda elde edilen doğruluk dereceleri ile literatürde aynı veri seti için yapılan çalışmaların karşılaştırması yapılmıştır.

6.2.1. Dermatology veri seti

Deneysel çalışma sonucunda Dermatology veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.3'te paylaşılmıştır.

	Accuracy	**Sensitivity**	**Specificity**
En iyi	% 96.56	% 96.17	% 99.34
En Kötü	% 95.13	% 95.78	% 99.06
Ortalama	% 96.11	% 95.98	% 99.25
Standart Sapma	0.00422	0.00275	0.00084

Çizelge 6.3. Dermatology veri setine ait deneysel sonuçlar

Çizelge 6.3'e bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 96.11 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve specitifity değerlerine bakıldığında ise sırasıyla % 95.98 ve % 99.25 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold (çapraz doğrulama) sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00422 gibi küçük bir değerin elde edildiği görülmektedir. Standart sapmanın düşük olması algoritmanın kararlılığını göstermektedir.

Algoritmanın Dermatology veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma

metriği olarak sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.4'te gösterilmiştir.

Al-Kahlout ve ark. (Al-Kahlout, Naeem et al. 2021)	YSA Algoritması	**% 98.36**
Hameed ve ark. (Hameed, Karlik et al. 2019)	Yenilenmiş SOM (Self-Organizing Map) algoritması	% 68.18
Wang ve ark. (Wang 2018)	Genetik Algoritma ve YSA	% 95.38
Liu ve ark. (Liu, Xu et al. 2016)	RELIEFF dayalı C4.5 Karar Ağacı	% 90.20
Huang ve ark. (Huang, Hung et al. 2014)	SVM ile Özellik Seçimi (SVM-RFE)	% 95.38
Luo ve ark. (Luo, Ding et al. 2009)	Nonnegative Laplacian Embedding	% 83.61
Önerilen çalışma		% 96.11

Çizelge 6.4. Dermatology veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.4'deki literatür çalışmalarından Hameed ve ark. yaptıkları çalışmada eğitim için % 70 ve test için % 30 ayırmışlardır ve ayrıca iterasyon değerini 10 olarak belirlemişlerdir. Luo ve ark. iterasyon değerini 300 olarak almışlardır. Al-Kahlout ve ark. yaptıkları çalışmada veri setinin % 67'lik kısmını eğitim ve kalan % 33'lük kısmını test amacıyla kullanmışlardır. Wang ve ark. yaptıkları çalışmada veri setinin % 80'lik kısmını eğitim ve kalan % 20'lik kısmını test amacıyla kullanmışlardır.

Çizelge 6.4'e bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 96.11 başarı oranı ile Al-Kahlout ve arkadaşlarının elde ettiği % 98.36'lık sonuca çok yakın olduğu görülmektedir. Ayrıca GWO algoritmasının kalan diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiği görülmektedir.

6.2.2. Hepatit veri seti

Deneysel çalışma sonucunda Hepatit veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.5'te paylaşılmıştır.

	Accuracy	**Sensitivity**	**Specificity**
En iyi	% 98.94	% 97.60	% 97.60
En Kötü	% 97.42	% 95.60	% 95.60
Ortalama	% 98.04	% 96.58	% 96.58
Standart Sapma	0.00439	0.00801	0.00801

Çizelge 6.5. Hepatit veri setine ait deneysel sonuçlar

Çizelge 6.5'e bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 98.04 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve specitifity değerlerine bakıldığında ise sırasıyla % 96.58 ve % 96.58 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00439 gibi küçük bir değerin elde edildiği görülmektedir.

Algoritmanın Hepatit veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metriği olarak sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.6'da gösterilmiştir.

Peng ve ark. (Peng, Zou et al. 2021)	Random Forest Algoritması	% 91.90
Novichasari ve ark. (Novichasari ve Wibisono 2020)	Parçacık Sürü Optimizasyonu	% 90.80
Joshi ve ark. (Joshi ve Jetawat 2020)	J48 karar ağacı, Naive Bayes, IBK, SVM, ZeroR ve VFI	% 85.16
Mitra ve ark. (Mitra ve Samanta 2017)	Levenberg Marquardt algortiması ile YSA	% 94.61
Tan ve ark. (Tan, Teoh et al. 2009)	Hibrit Çalışma(GA-SVM)	% 87.70
Önerilen çalışma		**% 98.04**

Çizelge 6.6. Hepatit veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.6'daki literatür çalışmalarından Peng ve ark. yaptıkları çalışmada çapraz doğrulama değeri için 5, 10 ve 20 değerlerine karışık sırasıyla % 88.2, %91.0 ve % 91.20 gibi doğruluk derecesini almışlardır. Diğer çalışmalardan Novichasari ve ark. ile Joshi ve ark. yaptıkları çalışmalarda çapraz doğrulama için 10 olarak kullanmışlardır. Tan ve ark. yaptıkları çalışmada popülasyon boyutu için 10, toplam çalıştırma sayısı için 100 ve çapraz doğrulama için 20 değerlerini kullanmışlardır.

Çizelge 6.6'ya bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 98.04 gibi bir başarı oranı ile diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiği görülmektedir. GWO algoritmasına en yakın sonucu % 94.61 ile Mitra ve arkadaşının elde ettiği görülmektedir.

6.2.3. Thyroid veri seti

Deneysel çalışma sonucunda Thyroid veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.7'de paylaşılmıştır.

	Accuracy	**Sensitivity**	**Specificity**
En iyi	% 95.04	% 90.99	% 95.36
En Kötü	% 92.86	% 88.39	% 93.85
Ortalama	% 94.28	% 89.77	% 94.73
Standart Sapma	0.00708	0.00838	0.00502

Çizelge 6.7. Thyroid veri setine ait deneysel sonuçlar

Çizelge 6.7'ye bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 94.28 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve specitifity değerlerine bakıldığında ise sırasıyla % 89.77 ve % 94.73 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00708 gibi küçük bir değerin elde edildiği görülmektedir.

Algoritmanın Thyroid veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metriği olarak sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.8'de gösterilmiştir.

Shivastuti ve ark. (Shivastuti, Manhas et al. 2021)	SVM ve RF (Random Forest) Algoritması	% 93.00
Sivasakthivel ve ark. (Sivasakthivel, Shrivakshan et al. 2017)	J8, CART ve Random Forest Algoritmaları	% 86.12
Chandel ve ark. (Chandel, Kunwar et al. 2016)	K-NN, SVM, Naive Bayes	% 93.44
Doğantekin ve ark. (Dogantekin, Dogantekin et al. 2011)	GDA_WSVM (Generalized Discriminant Analysis ve Wavelet Support Vector Machine System)	% 91.86
Polat ve ark. (Polat, Şahan et al. 2007)	AIRS (Artificial immune recognition system) Algoritması	% 85.00
Önerilen çalışma		**% 94.28**

Çizelge 6.8. Thyroid veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.8'deki literatür çalışmalarından Shivastuti ve ark. ile Sivasakthivel ve ark. yaptıkları çalışmada çapraz doğrulama için 10 olarak kullanmışlardır. Doğantekin ve ark. yaptıkları çalışmada verisetinin % 80'lik kısmını eğitimde kullanmış olup veri setinden kalan % 20'lik kısmını ise test amacıyla ayırmışlardır. Polat ve ark. yaptıkları çalışmada k en yakın komşu değerini 2 ve çapraz doğrulamayı 10 olarak kullanmışlardır.

Çizelge 6.8'e bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 94.28 başarı oranı ile diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiği görülmektedir. GWO algoritmasına en yakın sonucu % 93.44 ile Chandel ve arkadaşlarının elde ettiği görülmektedir.

6.2.4. BCWD veri seti

Deneysel çalışma sonucunda BCWD veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.9'da paylaşılmıştır.

	Accuracy	**Sensitivity**	**Specificity**
En iyi	% 96.66	% 95.61	% 96.09
En Kötü	% 95.75	% 95.11	% 94.99
Ortalama	% 96.13	% 95.30	% 95.42
Standart Sapma	0.00312	0.00232	0.00353

Çizelge 6.9. BCWD veri setine ait deneysel sonuçlar

Çizelge 6.9'a bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 96.13 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve specitifity değerlerine bakıldığında ise sırasıyla % 95.30 ve % 95.42 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00312 gibi küçük bir değerin elde edildiği görülmektedir.

Algoritmanın BCWD veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metriği olarak

sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.10'da gösterilmiştir.

Assegie (Assegie 2021)	K-NN	% 94.35
Yang ve ark. (Yang ve Yang 2021)	Relief ve ReliefF	% 93.15
Abdel-Baset ve ark. (Abdel-Basset, El-Shahat et al. 2020)	GWO ve KNN ile özellik seçimi	% 94.82
Lavanya ve ark. (Lavanya ve Rani 2011)	CART (Sınıflandırma ve Regresyon Ağaçları)	% 94.72
Rani (Rani 2010)	YSA	% 92.00
Önerilen çalışma		**% 96.13**

Çizelge 6.10. BCWD veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.10'daki literatür çalışmalarından Assegie, yaptığı çalışmada k-en yakın komşuluk değeri için farklı değerler kullanmış 40 ile düzenli bir başarı elde etmiştir. Abdel-Baset ve ark. yaptıkları çalışmada k-en yakın komşuluk değeri için 40, çapraz doğrulama için 10 ve iterasyon için 30 değerlerini kullanmışlardır.

Çizelge 6.10'a bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 96.13 gibi bir başarı oranı ile diğer çalışmalardan daha iyi Jabbar tarafından yapılan çalışmada % 97.42 gibi bir başarıya yakın bir değer elde etmiştir. Ayrıca GWO algoritmasının kalan diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiğini ve en yakın sonucun % 94.82 ile Abdel-Baset ve arkadaşlarının elde ettiği görülmektedir.

6.2.5. BCWO veri seti

Deneysel çalışma sonucunda BCWO veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.11'de paylaşılmıştır.

	Accuracy	**Sensitivity**	**Specificity**
En iyi	% 97.37	% 97.21	% 97.21
En Kötü	% 96.77	% 96.46	% 96.35
Ortalama	% 97.03	% 96.77	% 96.73
Standart Sapma	0.00233	0.00255	0.00276

Çizelge 6.11. BCWO veri setine ait deneysel sonuçlar

Çizelge 6.11'e bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 97.03 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve

specitifity değerlerine bakıldığında ise sırasıyla % 96.77 ve % 96.73 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00233 gibi küçük bir değerin elde edildiği görülmektedir.

Algoritmanın BCWO veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metriği olarak sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.12'de gösterilmiştir.

Jijitha ve ark. (Jijitha ve Amudha 2021)	Genetik Algoritma (GA), Logistic Regression (LR) ve K-NN	**% 99.60**
Bayrak ve ark. (Bayrak, Kırcı et al. 2019)	YSA ve SVM	% 96.99
Janghorbani ve ark. (Janghorbani ve Moradi 2017)	Fuzzy Evidential Network	% 96.85
Das ve ark. (Das, Panigrahi et al. 2012)	PSO ve SVM	% 93.55
Abonyi ve ark. (Abonyi ve Szeifert 2003)	Supervised Fuzzy Clustering	% 95.57
Önerilen çalışma		% 97.03

Çizelge 6.12. BCWO veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.12'deki literatür çalışmalarından Bayrak ve ark. ile Abonyi ve ark. yaptıkları çalışmada çapraz doğrulama için 10 değerlerini kullanmışlardır.

Çizelge 6.12'ye bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 97.03 gibi bir başarı oranı ile Jijitha ve arkadaşlarının elde ettiği % 99.60'lık sonuca yakın olduğu görülmektedir. Ayrıca GWO algoritmasının kalan diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiğini ve en yakın sonucun % 96.99 ile Bayrak ve arkadaşlarının elde ettiği görülmektedir.

6.2.6. Wine veri seti

Deneysel çalışma sonucunda Wine veri seti için elde edilen doğruluk (accuracy), öznellik (specificity) ve özgünlük (sensitivity) metriklerinin en iyi, en kötü, ortalama değerleri ile birlikte standart sapma bilgisi de Çizelge 6.13'te paylaşılmıştır.

	Accuracy	Sensitivity	Specificity
En iyi	% 98.94	% 97.60	% 97.60
En Kötü	% 97.42	% 95.60	% 95.60
Ortalama	% 98.04	% 96.58	% 96.58
Standart Sapma	0.00439	0.00801	0.00801

Çizelge 6.13. Wine veri setine ait deneysel sonuçlar

Çizelge 6.13'e bakıldığında algoritmanın sınıflandırma problemi üzerinde ortalama % 98.04 gibi bir başarı değeri elde ettiği görülmektedir. Sensitivity ve specitifity değerlerine bakıldığında ise sırasıyla % 96.58 ve % 96.58 olduğu görülmektedir. Algoritmanın 10 tekrar ve her tekrardaki 10 fold sonucunda elde ettiği değerlerin standart sapmasına bakıldığında 0.00439 gibi küçük bir değerin elde edildiği görülmektedir.

Algoritmanın Wine veri seti üzerinde elde ettiği deneysel sonuçlar literatürden alınan bazı çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metriği olarak sınıflandırma başarısı yüzdesi (accuracy) kullanılmıştır. Yapılan bu karşılaştırmanın sonuçları Çizelge 6.14'te gösterilmiştir.

Wang ve ark. (Wang, Jiang et al. 2021)	MRMI (Maximum-Relevance and Maximum-Interaction)	% 97.70
Mijwil ve ark. (Mijwil ve Abttan 2021)	Genetik Algoritma ve C4.5 Karar Ağacı	% 93.25
Prasetyo ve ark. (Prasetyo, Purbaningtyas et al. 2020)	CosineKNN (Kosinüs ağırlıklı K-NN)	% 96.79
Hameed ve ark. (Hameed, Karlik et al. 2019)	Yenilenmiş SOM (Self-Organizing Map) algoritması	% 81.13
Chen ve ark. (Chen, Yuan et al. 2016)	DCQGA-SVM	% 90.41
Önerilen çalışma		**% 98.04**

Çizelge 6.14. Wine veri setine ait karşılaştırmalı literatür sonuçları

Çizelge 6.14'deki literatür çalışmalarından Wang ve ark. ile Mijwil ve ark. yaptıkları çalışmada çapraz doğrulama için 10, Prasetyo ve ark. yaptıkları çalışmada çapraz doğrulama için 2 değerini kullanmışlardır. Hameed ve ark. yaptıkları çalışmada iterasyon için 50 değerini kullanmışlardır. Chen ve ark. yaptıkları çalışmada iterasyon için 100 ve popülasyon boyutu için 20 değerlerini kullanmışlardır.

Çizelge 6.14'e bakıldığında tez çalışmasında kullanılan GWO algoritmasının % 98.04 gibi bir başarı oranı ile diğer çalışmalardaki sonuçlardan daha iyi sonuç elde ettiği

görülmektedir. GWO algoritmasına en yakın sonucu % 97.70 ile Wang ve arkadaşlarının elde ettiği görülmektedir.

Yapılan deneysel çalışmada GWO algoritması ile literatürde yakından zamandan geçmişe doğru aynı veri setleri üzerinde yapılan deneysel çalışmalar karşılaştırılmıştır. Yapılan literatür karşılaştırması neticesinde Hepatit, Thyroid, BCWO ve Wine veri setleri üzerinde GWO algoritmasının elde ettiği doğruluk oranları diğer çalışmalarda elde edilen doğruluk oranlarından daha iyi olduğu gözlenmektedir. Dermatology ve BCWD veri setlerinde ise GWO algoritmasından daha iyi sonuçların da olduğunu fakat GWO'nun en başarılı çalışmanın elde ettiği sonuca çok yakın olduğunu söylemek mümkündür. Bu sonuçlara bakılarak sınıflandırma problemlerinde önerdiğimiz çalışma ile GWO algoritmasının başarılı sonuçlar ürettiğini söylemek mümkündür. Farklı veri setleri ve farklı çalışmalar ile algoritmalarla GWO algoritmasının başarısı karşılaştırılabilir.

Bu çalışmada GWO algoritmasının sınıflandırma başarısı literatürdeki aynı veri setleri üzerinde yapılan çalışmalar ile karşılaştırılmıştır. Genel olarak seçilen veri setlerinde GWO algoritmasının daha başarılı sonuçlar ürettiği görülmektedir.

7. TARTIŞMA VE ÖNERİLER

Bu çalışmada, Gri Kurt Optimizasyon Algoritması ile iki aşamalı çalışma gerçekleştirilmiştir.

Birinci aşamada, hiyerarşik olmayan (bölümsel) kümeleme için UCI Machine Learning Repository'den alınan 10 veri kümesinde GWO algoritması kullanılmıştır. GWO algoritmasının kümeleme performansı, K-means, K-medoids ve Fuzzy C-means kümeleme algoritmalarının performanslarıyla karşılaştırılmıştır. Sonuç olarak, GWO algoritması genellikle diğer kümeleme algoritmalarından daha iyi çözümler üretmiştir. Bu nedenle, GWO Algoritması, kümeleme problemlerinde kullanılacak alternatif bir algoritma olarak önerilebilir.

İkinci aşamada ise, UCI Machine Learning Repository'den alınan hem az sınıflı ve hem de çok sınıflı 6 veri seti üzerinde GWO algoritması kullanılmıştır. Seçilen veri setleri üzerinde önce GWO algoritması ile küme merkezleri belirlenmiş olup ardından test verilerindeki her elman için uzaklık hesabı ile ait oldukları kümeler belirlenmiştir. Daha sonra küme içerisindeki sınıfları bulunarak algoritmanın doğrulukları tespit edilmiştir. Veri setleri üzerinde yapılan deneysel çalışmada elde edilen ortalama doğruluk dereceleri ile için aynı veri setleri üzerinde literatürde yapılan deneysel çalışmalar karşılaştırılmıştır. Karşılaştırma neticesine GWO algoritmasının sınıflandırma problemlerinde başarılı sonuçlar ürettiği gözlemlenmiştir. Sonuç olarak, GWO algoritması kümeleme ve sınıflandırma problemlerinde daha iyi çözümler üretmiştir. Bu nedenle, GWO Algoritması, hem kümeleme ve hem de GWO ile geliştirdiğimiz yeni bir yaklaşım ile sınıflandırma problemlerinde kullanılacak alternatif bir algoritma olarak önerilebilir.

Yapılacak çalışmalarda daha farklı veri setleri kullanılarak geliştirilebilir. Gerek sentetik, gerekirse gerçek verilerin daha yüksek boyutlarda, daha fazla sınıflı problemlerde, eğitimde kullanılan örnek sayısının oranlarına göre çalışma performansları değerlendirilebilir. Ayrıca farklı metasezgisel algoritmaların başarı performansları değerlendirilebilir. Algoritmaların uzaklık hesaplama işlemlerinde Öklid metodundan farklı diğer metotlar da kullanılarak algoritmaların performansını arttıracak en uygun metot tespit edilebilir. Böylelikle algoritmaların başarımları arttırılabilir.

KAYNAKLAR

Abdel-Basset, M., D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque and S. Mirjalili (2020). "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection." Expert Systems with Applications **139**: 112824.

Abonyi, J. and F. Szeifert (2003). "Supervised fuzzy clustering for the identification of fuzzy classifiers." Pattern Recognition Letters **24**(14): 2195-2207.

Akpınar, H. (2000). Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İstanbul Üniversitesi İşletme Fakültesi Dergisi. Cilt.29, Sayı.1.

Al-Kahlout, B. I., M. M. Naeem and M. J. Shepherd (2021). "ANN for the Classification of Eryhemato-Squamous Disease."

Assegie, T. A. (2021). "An optimized K-Nearest Neighbor based breast cancer detection." Journal of Robotics and Control (JRC) **2**(3): 115-118.

Banharnsakun, A., B. Sirinaovakul and T. Achalakul (2013). "The best-so-far ABC with multiple patrilines for clustering problems." Neurocomputing **116**: 355-366.

Bayrak, E. A., P. Kırcı and T. Ensari (2019). Comparison of machine learning methods for breast cancer diagnosis. 2019 Scientific meeting on electrical-electronics & biomedical engineering and computer science (EBBT), Ieee.

Chandel, K., V. Kunwar, S. Sabitha, T. Choudhury and S. Mukherjee (2016). "A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques." CSI transactions on ICT **4**(2-4): 313-319.

Chen, P., L. Yuan, Y. He and S. Luo (2016). "An improved SVM classifier based on double chains quantum genetic algorithm and its application in analogue circuit diagnosis." Neurocomputing **211**: 202-211.

Choudhari, S. and S. Biday (2014). "Artificial neural network for skin cancer detection." International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) **3**(5): 147-153.

Coşkun, C. and A. Baykal (2011). "Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması." Akademik Bilişim **2011**: 1-8.

Das, S. R., P. K. Panigrahi, K. Das and D. Mishra (2012). "Improving rbf kernel function of support vector machine using particle swarm optimization." International Journal of Advanced Computer Research **2**(7): 130-135.

Dogan, I. (2002). "Selection by cluster analysis." Turkish Journal of Veterinary & Animal Sciences **26**(1): 47-53.

Dogantekin, E., A. Dogantekin and D. Avci (2011). "An expert system based on Generalized Discriminant Analysis and Wavelet Support Vector Machine for diagnosis of thyroid diseases." Expert Systems with Applications **38**(1): 146-150.

Dorigo, M., V. Maniezzo and A. Colorni (1991). "The Ant System: An Autocatalytic Optimizing Process."

Ebrahimi, A. and E. Khamehchi (2016). "Sperm whale algorithm: An effective metaheuristic algorithm for production optimization problems." Journal of Natural Gas Science and Engineering **29**: 211-222.

Elhariri, E., N. El-Bendary, A. E. Hassanien and A. Abraham (2015). "Grey Wolf Optimization for One-Against-One Multi-class Support Vector Machines." Proceedings of the 2015 Seventh International Conference of Soft Computing and Pattern Recognition (Socpar 2015): 7-12.

Everitt, S. B., S. Landau and M. Leese (2001). Cluster Analysis. Oxford University Press Inc, New York.

Ewees, A. A., M. Abd Elaziz and D. Oliva (2021). "A new multi-objective optimization algorithm combined with opposition-based learning." Expert Systems with Applications **165**.

Ferreira, L. and D. B. Hitchcock (2009). "A Comparison of Hierarchical Methods for Clustering Functional Data." Communications in Statistics-Simulation and Computation **38**(9): 1925-1949.

Frigui, H. and R. Krishnapuram (1999). "A robust competitive clustering algorithm with applications in computer vision." Ieee Transactions on Pattern Analysis and Machine Intelligence **21**(5): 450-465.

Hair, J., W. Black, R. Tatham and R. Anderson (1998). Multivariate Data Analysis. Prentice Hall College Div; 5th edition (March 1, 1998).

Hameed, A. A., B. Karlik, M. S. Salman and G. Eleyan (2019). "Robust adaptive learning approach to self-organizing maps." Knowledge-Based Systems **171**: 25-36.

Han, J., M. Kamber and J. Pei (2012). "Data Mining: Concepts and Techniques, 3rd Edition." Data Mining: Concepts and Techniques, 3rd Edition: 1-703.

Han, J., M. Kamber and A. Tung (2001). "Spatial clustering methods in data mining: a survey." Data Mining and Knowledge Discovery - DATAMINE.

Hands, S. and B. Everitt (1987). "A Monte-Carlo Study of the Recovery of Cluster Structure in Binary Data by Hierarchical-Clustering Techniques." Multivariate Behavioral Research **22**(2): 235-243.

Hoppner, F. and F. Klawonn (2000). "Fuzzy clustering of sampled functions." Peachfuzz 2000 : 19th International Conference of the North American Fuzzy Information Processing Society - Nafips: 251-255.

Huang, M.-L., Y.-H. Hung, W. Lee, R.-K. Li and B.-R. Jiang (2014). "SVM-RFE based feature selection and Taguchi parameters optimization for multiclass SVM classifier." The Scientific World Journal **2014**.

Işık, M. (2006). "Bölünmeli kümeleme yöntemleri ile veri madenciliği uygulamaları." Y. Lisans Tezi, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, İstanbul.

Jain, A. K. (2010). "Data clustering: 50 years beyond K-means." Pattern Recognition Letters **31**(8): 651-666.

Janghorbani, A. and M. H. Moradi (2017). "Fuzzy evidential network and its application as medical prognosis and diagnosis models." Journal of biomedical informatics **72**: 96-107.

Jijitha, S. and T. Amudha (2021). Breast cancer prognosis using machine learning techniques and genetic algorithm: experiment on six different datasets. Evolutionary Computing and Mobile Sustainable Networks, Springer**:** 703-711.

Johnson, R. A. and D. W. Wichern (1988). Applied multivariate statistical analysis (2nd edition). Prentice Hall, Englewood Cliffs, New Jersey.

Joshi, M. and A. Jetawat (2020). Evaluation of Classification Algorithms used in Medical Decision Support Systems. 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), IEEE.

Kahaner, D., M. Cleve and N. Stephen (1989). Numerical Methods and Software. Prentice-Hall, Englewood Cliffs, NJ.

Kamboj, V. K., S. K. Bath and J. S. Dhillon (2016). "Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer." Neural Computing & Applications **27**(5): 1301-1316.

Kara, İ. (1986). Yöneylem araştırması: doğrusal olmayan modeller, Anadolu Üniversitesi.

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

Karakoyun, M. and O. A. Inan, İhtisam (2019). "Grey Wolf Optimizer (GWO) Algorithm to Solve the Partitional Clustering Problem." International Journal of Intelligent Systems and Applications in Engineering: 201-206.

Karami, A. and M. Guerrero-Zapata (2015). "A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks." Neurocomputing **149**: 1253-1269.

Kaufman, L. and P. Rousseeuw (1990). "Finding Groups in Data: An Introduction to Cluster Analysis." John Wiley & Sons, New York.

Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks.

Khozeimeh, F., R. Alizadehsani, M. Roshanzamir, A. Khosravi, P. Layegh and S. Nahavandi (2017). "An expert system for selecting wart treatment method." Computers in biology and medicine **81**: 167-175.

Korurek, M. and A. Nizam (2008). "A new arrhythmia clustering technique based on Ant Colony Optimization." Journal of Biomedical Informatics **41**(6): 874-881.

Lavanya, D. and D. K. U. Rani (2011). "Analysis of feature selection with classification: Breast cancer datasets." Indian Journal of Computer Science and Engineering (IJCSE) **2**(5): 756-763.

Liu, H., X. Wu and S. Zhang (2011). "Feature selection using hierarchical feature clustering." International Conference on Information and Knowledge Management, Proceedings: 979-984.

Liu, Q., X. Xu, Y. Tao and X. Wang (2016). An Improved Decision Tree Method Base on RELIEFF for Medical Diagnosis. 2016 6th International Conference on Digital Home (ICDH), IEEE.

Lorr, M. (1983). Cluster Analysis for Social Scientists, Jossey-Bass Inc Pub; 1st edition (August 1, 1983).

Luo, D., C. Ding, H. Huang and T. Li (2009). Non-negative laplacian embedding. 2009 Ninth IEEE International Conference on Data Mining, IEEE.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley**:** 281-297.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations."

Meltem, I. and A. Y. Çamurcu (2011). "K-MEANS VE AŞIRI KÜRESEL C-MEANS ALGORITMALARI İLE BELGE MADENCİLİĞİ." Marmara Fen Bilimleri Dergisi **22**(1): 1-18.

Mettleq, A. S. A., I. M. Dheir, A. A. Elsharif and S. S. Abu-Naser (2020). "Mango Classification Using Deep Learning." International Journal of Academic Engineering Research (IJAER) **3**(12).

Mijwil, M. M. and R. A. Abttan (2021). "Utilizing the Genetic Algorithm to Pruning the C4. 5 Decision Tree Algorithm." Asian J. Appl. Sci. ISSN 2321 **893**.

Mirjalili, S. (2015). "How effective is the Grey Wolf optimizer in training multi-layer perceptrons." Applied Intelligence **43**(1): 150-161.

Mirjalili, S. and A. Lewis (2016). "The Whale Optimization Algorithm." Advances in Engineering Software **95**: 51-67.

Mirjalili, S., S. M. Mirjalili and A. Lewis (2014). "Grey Wolf Optimizer." Advances in Engineering Software **69**: 46-61.

Mitra, M. and R. Samanta (2017). "A Study on UCI Hepatitis Disease Dataset Using Soft Computing."

Moertini, V. S. (2002). "Introduction To Five Data Clustering Algorithms." Integral, vol. 7, no. 2, pp. 87–96,.

Murtagh, F. and P. Legendre (2014). "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" Journal of Classification **31**(3): 274-295.

Novichasari, S. I. and I. S. Wibisono (2020). "Particle Swarm Optimization For Improved Accuracy of Disease Diagnosis." Journal of Applied Intelligent System **5**(2): 57-68.

Oracle. (2021). "Data Mining Concepts " Retrieved 22.02.2021, 2021, from https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/classify.htm#DMCON004.

Özdamar, K. (1999). Paket Programlar ile İstatistiksel Veri Analizi, Kaan Kitabevi.

Peng, J., K. Zou, M. Zhou, Y. Teng, X. Zhu, F. Zhang and J. Xu (2021). "An Explainable Artificial Intelligence Framework for the Deterioration Risk Prediction of Hepatitis Patients." Journal of Medical Systems **45**(5): 1-9.

Polat, K., S. Şahan and S. Güneş (2007). "A novel hybrid method based on artificial immune recognition system (AIRS) with fuzzy weighted pre-processing for thyroid disease diagnosis." Expert Systems with Applications **32**(4): 1141-1147.

Prasetyo, E., R. Purbaningtyas and R. D. Adityo (2020). "Cosine K-nearest neighbor in milkfish eye classification." International Journal of Intelligent Engineering and Systems **13**(3): 11-25.

Rahman, M. A. and M. Z. Islam (2014). "A hybrid clustering technique combining a novel genetic algorithm with K-Means." Knowledge-Based Systems **71**: 345-365.

Rani, K. U. (2010). "Parallel approach for diagnosis of breast cancer using neural network technique." International Journal of Computer Applications **10**(3): 1-5.

Repository, U. M. L. "UCI Machine Learning Repository." 2020, from https://archive.ics.uci.edu/ml/index.php.

Rodriguez, L., O. Castillo, J. Soria, P. Melin, F. Valdez, C. I. Gonzalez, G. E. Martinez and J. Soto (2017). "A fuzzy hierarchical operator in the grey wolf optimizer algorithm." Applied Soft Computing **57**: 315-328.

Salgotra, R., U. Singh and S. Sharma (2020). "On the improvement in grey wolf optimization." Neural Computing & Applications **32**(8): 3709-3748.

Servi, T. (2009). Çok Değişkenli Karma Dağılım Modeline Dayalı Kümeleme Analizi. Doktora Tezi, Çukurova Üniversitesi.

Sharma, S. (1995). Applied Multivariate Techniques.

Sharma, Y. and L. C. Saikia (2015). "Automatic generation control of a multi-area ST - Thermal power system using Grey Wolf Optimizer algorithm based classical

controllers." International Journal of Electrical Power & Energy Systems **73**: 853-862.

Shivastuti, H. K., J. Manhas and V. Sharma (2021). "Performance Evaluation of SVM and Random Forest for the Diagnosis of Thyroid Disorder."

Singh, N. and S. B. Singh (2017). "A novel hybrid GWO-SCA approach for optimization problems." Engineering Science and Technology-an International Journal-Jestech **20**(6): 1586-1601.

Sivasakthivel, A., G. Shrivakshan and G. Shrivakshan (2017). "A comparative study of diagnosing thyroid diseases using classification algorithm." International Journals of Advanced Research in Computer Science and Software Engineering **7**.

Swiniarski, R. W. and A. Skowron (2003). "Rough set methods in feature selection and recognition." Pattern Recognition Letters **24**(6): 833-849.

Taherdangkoo, M., M. H. Shirzadi, M. Yazdi and M. H. Bagheri (2013). "A robust clustering method based on blind, naked mole-rats (BNMR) algorithm." Swarm and Evolutionary Computation **10**: 1-11.

Tan, K. C., E. J. Teoh, Q. Yu and K. C. Goh (2009). "A hybrid evolutionary algorithm for attribute selection in data mining." Expert Systems with Applications **36**(4): 8616-8630.

Tan, P.-N., M. Steinbach and V. Kumar (2013). Introduction to Data Mining, Pearson Education.

Tatlıdil, H. (1996). Uygulamalı Çok Değişkenli İstatistiksel Analiz, Cem Ofset Ltd.Şti. Eylül, Ankara.

Tsapanos, N., A. Tefas, N. Nikolaidis and I. Pitas (2015). "A distributed framework for trimmed Kernel k-Means clustering." Pattern Recognition **48**(8): 2685-2698.

Tunç, A. (2016). Finans sektörü için yapay öğrenme teknikleri kullanarak kredi kullanabilirliğin tespiti. Yüksek Lisans Tezi, Selçuk Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar Mühendisliği Anabilim Dalı.

Velmurugan, T. (2014). "Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data." Applied Soft Computing **19**: 134-146.

Wan, M., C. Wang, L. X. Li and Y. X. Yang (2012). "Chaotic ant swarm approach for data clustering." Applied Soft Computing **12**(8): 2387-2393.

Wang, L.-x., S.-y. Jiang and S.-y. Jiang (2021). "A feature selection method via analysis of relevance, redundancy, and interaction." Expert Systems with Applications: 115365.

Wang, Q. (2018). "Feed-forward Neural Networks with Genetic Algorithm and Network Reduction: A Case Study of Dermatologist-level Classification of Skin Cancer."

Xu, R. and D. Wunsch (2005). "Survey of clustering algorithms." Ieee Transactions on Neural Networks **16**(3): 645-678.

Xu, Z. J., L. S. Wang, J. C. Luo and J. Q. Zhang (2005). "A modified clustering algorithm for data mining." IGARSS 2005: IEEE International Geoscience and Remote Sensing Symposium, Vols 1-8, Proceedings: 741-744.

Yang, M. and J. Yang (2021). "Feature Selection Based on Distance Measurement." Journal of New Media **3**(1): 19.

Yusof, Y. and Z. Mustaffa (2015). Time Series Forecasting of Energy Commodity using Grey Wolf Optimizer. Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong.