T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

METASEZGİSEL ALGORİTMALAR İLE VERİ MADENCİLİĞİNDE AYKIRI DEĞERLERİN TESPİTİ UYGULAMALARI

Merve GİTMEZ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2018

Yrd. Doç. Dr. İbrahim Berkan AYDİLEK danışmanlığında, Merve GİTMEZ'in hazırladığı **"Metasezgisel algoritmalar ile veri madenciliğinde aykırı değerlerin tespiti uygulamaları "** konulu bu çalışma 23/02/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

İmza

Danışman: Yrd. Doç. Dr. İbrahim Berkan AYDİLEK ……..

Üye : Prof. Dr. Murat KAYRİ ……..

Üye : Yrd. Doç. Dr. Abdülkadir GÜMÜŞÇÜ ……..……..

Bu Tezin Elektrik Elektronik Mühendisliği Anabilim Dalında Yapıldığını ve Enstitümüz Kurallarına Göre Düzenlendiğini Onaylarım.

Prof. Dr. Halil Murat ALĞIN
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirimlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

Sayfa No

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
ŞEKİLLER DİZİNİ iv
ÇİZELGELER DİZİNİ v
SİMGELER DİZİNİ vi
1.GİRİŞ 1
1.1.Aykırı Değer Tespiti 4
1.1.1.Veri madenciliğinde aykırı değer tespiti 4
1.1.2.Aykırı değer analizi 5
1.1.3. Aykırı değerlerin türleri 6
1.1.3.1.Küresel aykırı değerler 6
1.1.3.2.Bağlamsal (Kontekste bağlı) aykırı değerler 6
1.1.3.3.Kolektif aykırı değerler 7
1.1.4. Aykırı değer tespitinin zorlukları 8
1.1.5. Aykırı değer tespit metotları 8
1.1.5.1.Aşırı değer analizi 10
1.1.5.2.Kümeleme tabanlı metotlar 14
1.1.5.3.İstatistiksel yaklaşımlar 15
1.1.5.4.Yakınlık temelli yaklaşımlar 18
1.1.5.5.Sınıflamaya dayalı yaklaşımlar 18
1.1.6. Bağlamsal ve kolektif aykırı değerlerin madenciliği 20
1.2. Metasezgisel Algoritmalar 21
1.2.1.Genetik algoritmalar 21
1.2.1.1.Genetik algoritma temel kavramları 22
1.2.1.2.Genetik algoritmada kullanılan operatörler 23
1.2.1.3.Genetik algoritması temel çalışma mantığı 25
1.2.2. Parçacık sürü algoritması 26
1.2.2.1.Parçacık sürü optimizasyonu temel kavramları 28
1.2.2.2.Parçacık sürü optimizasyonu parametre kontrolü 29
1.2.2.3.Parçacık sürü algoritması temel çalışma mantığı 30
1.2.3.Diferansiyel gelişim algoritması 32
1.2.3.1.Diferansiyel gelişim algoritması kontrol parametreleri 32
1.2.3.2.Diferansiyel gelişim algoritması kullanılan operatörler 33
1.2.3.3.Diferansiyel gelişim algoritması temel çalışma mantığı 34
1.3.Sınıflandırma Algoritmaları 35
1.3.1.Destek vektör makineleri 35

1.3.1.1.Doğrusal ayrılabilen verilerde sınıflandırma 36
1.3.1.2.Doğrusal ayrılamayan verilerde sınıflandırma 38
1.3.2.En yakın k komşu algoritması 39
1.3.2.1. En yakın k komşu algoritmasının parametreleri 40
1.3.3.Çok katmanlı algılayıcılar 41
1.3.3.1.İleri doğru hesaplama 42
1.3.3.2.Geriye doğru hesaplama 42
2. ÖNCEKİ ÇALIŞMALAR 44
3. MATERYAL ve YÖNTEM 46
3.1.Materyal 46
3.2.Yöntem 48
3.2.1.Uygulamada kullanılan parametreler 59
3.2.2.Uygulamada kullanılan algoritma 50
4. ARAŞTIRMA BULGULARI ve TARTIŞMA 53
4.1.Veri Setleri İçin Elde Edilen Aykırı Değerler 53
4.2.Veri Setlerinin Elde Edilen Sınıflandırma Başarıları 71
5. SONUÇLAR ve ÖNERİLER 110
5.1.Sonuçlar 110
5.2.Öneriler 114
KAYNAKÇA 116
ÖZGEÇMİŞ 121

ÖZET

Yüksek Lisans Tezi

METASEZGİSEL ALGORİTMALAR İLE VERİ MADENCİLİĞİNDE AYKIRI DEĞERLERİN TESPİTİ UYGULAMALARI

Merve GİTMEZ

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. İbrahim Berkan AYDİLEK
Yıl: 2018, Sayfa: 121

Metasezgisel yöntemler, arama uzayında yüksek kalitede çözümleri kapsayan bölgeleri etkin bir şekilde kullanarak optimizasyon problemlerine ait en uygun çözümü bulmayı amaçlayan yöntemlerdir. Bu yöntemler, oldukça az değişiklikle farklı optimizasyon problemlerine uygulanabilmektedirler. Ayrıca veri madenciliğinin de önemli problemlerine uygulanabilmektedirler. Çok değişkenli değişkenler bağlamında aykırı değerlerin tespit edilmesi, veri madenciliği için önemli bir konudur. Bu konuya çözüm getirmek için önerilen yöntemlerin çoğu verilerin dağılım merkezine olan Mahalanobis mesafesine, Öklid mesafesine veya bir iz düşüm takibi yaklaşımına (projection pursuit) dayanmaktadır. Metasezgisel yöntemler, aykırı değerleri tespit etmek amacıyla kullanıldığı uygulamalarda mesafe parametrelerini manuel olarak ayarlamak yerine otomatik olarak optimize etmektedir. Bu çalışmadaki amaç, metasezgisel algoritmaları kullanarak veri madenciliğinin önemli problemi ve konusu olan çok değişkenli değişkenler bağlamında aykırı değerleri tespit etmeye çalışmaktır ve aykırı değerlerin sınıflandırma başarısı üzerindeki etkisini incelemektedir.

ANAHTAR KELİMELER: Aykırı değer, diferansiyel gelişim, genetik, metasezgisel algoritmalar, parçacık sürü

ABSTRACT

MSc Thesis

OUTLİERS DETECTİON İN DATA MİNİNG BY METAHEURİSTİC ALGORİTHMS

Merve GİTMEZ

Harran University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronic Engineering

Supervisor: Assist. Prof. Dr. İbrahim Berkan AYDİLEK
Year: 2018, Page:121

The aim of metaheuristic methods is to find the most appropriate solution of optimization problems by using the regions covering high quality solutions in search space effectively. These methods can be applied to different optimization problems with very little modification. Also these methods are adapted to the important problems of data mining. Detection of outliers in the context of multivariate data is an important issue for data mining. Most of recommended methods for solving this problem are based on mahalanobis distance, euclidean distance, which is the most common distribution of observations or it is based on projection pursuit approach. Metaheuristic methods, when they are used in the applications in order to detect outliers, are able to optimize the distance parameters automatically rather than adjusting them manually. The purpose of this work is to detect the outlier at the multivariate variable context, which is an important problem and subject of data mining using metaheuristic algorithms and is to examine the impact of the detected outliers on the classification success.

KEY WORDS: Outlier, differential evolution, genetic, metaheuristic algorithm, particle swarm

TEŞEKKÜR

Tezin konusunun seçiminde ve tez çalışmaları süresince benden yardımlarını esirgemeyen danışmanım Sayın Yrd. Doç. Dr. İbrahim Berkan AYDİLEK' e ve attığım her adımda yanımda olan, beni destekleyen aileme teşekkür ederim.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.1. R bölgesindeki aykırı olan nesnelerin gösterimi 5
Şekil 1.2. Kolektif aykırı değerlerin gösterimi 7
Şekil 1.3. Aşırı değer ve aykırı değer farkının gösterimi 11
Şekil 1.4. Simetrik aşırı değer analizi 12
Şekil 1.5. Asimetrik aşırı değer analizi 12
Şekil 1.6. Çok değişkenli aşırı değer analizi 13
Şekil 1.7. Kutu grafiği gösterimi 16
Şekil 1.8. Histogram gösterimi 17
Şekil 1.9. Normal sınıflardan bir model öğrenme 19
Şekil 1.10.Yarı gözetimli metotlarla aykırı değer tespiti 20
Şekil 1.11. Genetik algoritma akış diyagramı 25
Şekil 1.12. Parçacık sürü algoritması akış diyagramı 30
Şekil 1.13. Diferansiyel gelişim algoritması akış diyagramı 34
Şekil 1.14. Destek vektörlerinin gösterimi 36
Şekil 1.15. Doğrusal olarak ayrılabilen verilerde hiperdüzlemin belirlenmesi 37
Şekil 1.16. Doğrusal olarak ayrılamayan verilerde hiperdüzlemin belirlenmesi 38
Şekil 1.17. Kernel fonksiyonu ile verilerin daha yüksek bir boyuta dönüştürülmesi 39
Şekil 1.18. k en yakın komşuluğa göre sınıflandırma 40
Şekil 1.19. Çok katmanlı algılayıcılar için ağ modeli 41
Şekil 4.1. İris veri seti için aykırı değerlerin grafiksel gösterimi 55
Şekil 4.2. İris veri seti için 100 iterasyondaki BestCost değerleri..... 55
Şekil 4.3. İris veri seti için 200 iterasyondaki BestCost değerleri..... 56
Şekil 4.4. Haberman veri seti için aykırı değerlerin grafiksel gösterimi 58
Şekil 4.5. Haberman veri seti için 100 iterasyondaki BestCost değerleri..... 58
Şekil 4.6. Haberman veri seti için 200 iterasyondaki BestCost değerleri 59
Şekil 4.7. Yeast veri seti için aykırı değerlerin grafiksel gösterimi 63
Şekil 4.8. Yeast veri seti için 100 iterasyondaki BestCost değerleri 63
Şekil 4.9. Yeast veri seti için 200 iterasyondaki BestCost değerleri 64
Şekil 4.10. Blood veri seti için aykırı değerlerin grafiksel gösterimi 67
Şekil 4.11.Blood veri seti için 100 iterasyondaki BestCost değerleri 67
Şekil 4.12. Blood veri seti için 200 iterasyondaki BestCost değerleri 68
Şekil 4.13. Bupa veri seti için aykırı değerlerin grafiksel gösterimi 70
Şekil 4.14. Bupa veri seti için 100 iterasyondaki BestCost değerleri 70
Şekil 4.15. Bupa veri seti için 200 iterasyondaki BestCost değerleri 71
Şekil 4.16. İris veri setinin DVM-C ile elde edilen accuracy değerleri 74
Şekil 4.17. İris veri setinin DVM-P ile elde edilen accuracy değerleri 75
Şekil 4.18. İris veri setinin ÇKA-C ile elde edilen accuracy değerleri 76

Şekil 4.19. İris veri setinin ÇKA-P ile elde edilen accuracy değerleri 76
Şekil 4.20. İris veri setinin KNN-C ile elde edilen accuracy değerleri 77
Şekil 4.21. İris veri setinin KNN-P ile elde edilen accuracy değerleri 78
Şekil 4.22. Bupa veri setinin DVM-C ile elde edilen accuracy değerleri 80
Şekil 4.23. Bupa veri setinin DVM-P ile elde edilen accuracy değerleri 81
Şekil 4.24. Bupa veri setinin ÇKA-C ile elde edilen accuracy değerleri 82
Şekil 4.25. Bupa veri setinin ÇKA-P ile elde edilen accuracy değerleri 83
Şekil 4.26. Bupa veri setinin KNN-C ile elde edilen accuracy değerleri 84
Şekil 4.27. Bupa veri setinin KNN-P ile elde edilen accuracy değerleri 85
Şekil 4.28. Blood veri setinin DVM-C ile elde edilen accuracy değerleri 88
Şekil 4.29. Blood veri setinin DVM-P ile elde edilen accuracy değerleri 89
Şekil 4.30. Blood veri setinin ÇKA-C ile elde edilen accuracy değerleri 90
Şekil 4.31. Blood veri setinin ÇKA-P ile elde edilen accuracy değerleri 91
Şekil 4.32. Blood veri setinin KNN-C ile elde edilen accuracy değerleri 92
Şekil 4.33. Blood veri setinin KNN-P ile elde edilen accuracy değerleri 93
Şekil 4.34. Haberman veri setinin DVM-C ile elde edilen accuracy değerleri 96
Şekil 4.35. Haberman veri setinin DVM-P ile elde edilen accuracy değerleri 97
Şekil 4.36. Haberman veri setinin ÇKA-C ile elde edilen accuracy değerleri 98
Şekil 4.37. Haberman veri setinin ÇKA-P ile elde edilen accuracy değerleri 99
Şekil 4.38. Haberman veri setinin KNN-C ile elde edilen accuracy değerleri 100
Şekil 4.39. Haberman veri setinin KNN-P ile elde edilen accuracy değerleri 101
Şekil 4.40. Yeast veri setinin DVM-C ile elde edilen accuracy değerleri 104
Şekil 4.41. Yeast veri setinin DVM-P ile elde edilen accuracy değerleri 105
Şekil 4.42. Yeast veri setinin ÇKA-C ile elde edilen accuracy değerleri 106
Şekil 4.43. Yeast veri setinin ÇKA-P ile elde edilen accuracy değerleri 107
Şekil 4.44. Yeast veri setinin KNN-C ile elde edilen accuracy değerleri 108
Şekil 4.45. Yeast veri setinin KNN-P ile elde edilen accuracy değerleri 109

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1. Uygulamada kullanılan veri setleri 46
Çizelge 3.2. Uygulamada kullanılan algoritmanın kaba kodu..... 52
Çizelge 4.1. İris veri seti için elde edilen aykırı değerler 54
Çizelge 4.2. Haberman veri seti için elde edilen aykırı değerler 56
Çizelge 4.3. Yeast veri seti için elde edilen aykırı değerler 60
Çizelge 4.4. Blood veri seti için elde edilen aykırı değerler 64
Çizelge 4.5. Bupa veri seti için elde edilen aykırı değerler 68
Çizelge 4.6. Veri setlerinin sınıflandırma başarısındaki değerleri..... 72
Çizelge 4.7. İris veri seti için elde edilen accuracy değerleri 73
Çizelge 4.8. Bupa veri seti için elde edilen accuracy değerleri 79
Çizelge 4.19. Blood veri seti için elde edilen accuracy değerleri 86
Çizelge 4.10. Haberman veri seti için elde edilen accuracy değerleri 94
Çizelge 4.11. Yeast veri seti için elde edilen accuracy değerleri 102

SİMGELER DİZİNİ

ÇKA-C	Çok Katmanlı Algılayıcılar - Çapraz Doğrulama (Multilayer Perceptron - Cross Validation)
ÇKA-P	Çok Katmanlı Algılayıcılar - Belli Bir Orana Göre Bölme (Multilayer Perceptron - Percentage Split)
DGA	Diferansiyel Gelişim Algoritması
DVM-C	Destek Vektör Makineleri - Çapraz Doğrulama (Support Vector Machine - Cross Validation)
DVM-P	Destek Vektör Makineleri - Belli Orana Göre Bölme (Support Vector Machine - Percentage Split)
GA	Genetik Algoritma
IQR	İnterquartile Range (Çeyrekler Arası Açıklık)
K	Küme Sayısı
KNN-C	K En Yakın Komşuluk - Çapraz Doğrulama (K - Nearest Neighbor - Cross Validation)
KNN-P	K En Yakın Komşuluk - Belli Bir Orana Göre Bölme (K - Nearest Neighbor - Percentage Split)
npop	Popülasyon Sayısı
PSO	Parçacık Sürü Optimizasyonu Algoritması
T	İterasyon Sayısı

1. GİRİŞ

Bilgi teknolojilerinin hızlı gelişmesi sonucu veri hacmi, çeşidi ve karmaşıklığı hızlı büyümekte ve heterojen yapılı bilginin otomatik olarak analiz edilmesine daha fazla ihtiyaç duyulmaktadır. Bu nedenle, veri madenciliği yöntemlerine olan ilgi artmakta ve gün geçtikçe daha fazla kullanılmaktadır. Bu yöntemlerin esas amacı, veri içindeki gizli bağıntıların ortaya çıkarılmasıdır (Han ve Kamber, 2000). Veri madenciliği analiz sonuçları daha sonra konu ile ilgili karar alma ve karar oluşturma süreçlerinde kullanılmaktadır. Alınan bu kararların niteliği tüm veri madenciliği sürecinin kalitesine doğrudan bağlı olmaktadır.

Veri madenciliği, büyük veri kümelerinin içinden işe yarayacak bilgi keşfinin analiz sürecini içeren bilgisayar biliminin disiplinler arası bir alt alanını oluşturmaktadır. Bu gelişmiş analiz sürecinin amacı, bir veri kümesinden bilgi çıkarmak ve çıkardığı bu bilgiyi daha ileri bir kullanım için anlaşılabilir bir yapıya dönüştürmektir (Zernik, 2010). Bu dönüştürme sürecinde veri, aşağıdaki ön işlemlerden geçmektedir (Han, Kamber, 2006):

- Veri temizleme; gürültülü ve tutarsız verileri çıkarmak.
- Veri bütünleştirme, birçok veri kaynağını birleştirmek.
- Veri seçme, yapılacak olan analiz ile ilgili olan verileri belirlemek.
- Veri dönüşümü, verinin veri madenciliği teknikleriyle kullanılabilecek hale dönüşümünü gerçekleştirmek.
- Veri madenciliği, veri örüntülerini yakalayabilmek için akıllı metotları uygulamak.
- Örüntü değerlendirme, bazı ölçümlere göre elde edilmiş bilgiyi temsil eden ilginç örüntüleri tanımlamak.
- Bilgi sunumu, madenciliği yapıldıktan sonra elde edilen bilginin kullanıcıya sunumunu gerçekleştirmek.

Veri dönüşümü sürecinden sonra kullanılacak veri madenciliği modeline karar verilmektedir. Bu modeller sınıflandırma, regresyon, kümeleme, aykırı değer analizi ve evrimsel analiz gibi tekniklerdir.

Veri madenciliği, veri modellerini çıkarmak için akıllı yöntemlerin uygulandığı temel işlemler olarak da ifade edilmektedir. Veri madenciliğinin temel işlemleri aşağıdaki gibi maddeler halinde özetlenebilmektedir (Jain, Srivastava, 2013):

- Verileri çok boyutlu bir veri tabanı sisteminde depolama ve yönetme.
- İşlem verilerini veri ambarı sistemine ayıklama, dönüştürme ve yükleme.
- İş analistlerine ve bilgi teknolojisi profesyonellerine veri erişimi sağlama.
- Verileri uygulama yazılımı ile analiz etme.
- Verileri grafik gibi yararlı bir biçimde sunma.

Veri madenciliği modelleri, veri madenciliğindeki kullanım amaçlarına göre gruplandırılabilir. Veri madenciliği modelleri, tanımlayıcı ve öngörücü olmak üzere iki kategoriye ayrılmaktadır. Tanımlayıcı bir model, veri tabanındaki verilerin genel özelliklerini karakterize etmektedir. Veri madenciliğinin tanımlayıcı görevleri arasında kümeleme, bağlantı kuralları, sıralı dizi analizi, özetleme, tanımsal istatistik, aykırı değer analizi gibi metotlar bulunmaktadır. Öngörücü model ise, belirli bir değişkenin (hedef değişken) bilinmeyen (genellikle gelecekteki) bir değerinin tahmin edilmesine izin vermektedir. Veri madenciliğinin öngörücü modelleri arasında sınıflandırma ve regresyon gibi metotlar bulunmaktadır (Alagöz ve ark., 2014).

Veri madenciliği pazarlama, finans, görüntü işleme ve tanıma, robot görüş sistemleri, yüzey analizi ve coğrafi bilgi sistemleri, uzay bilimleri, atmosfer bilimleri, fen bilimleri, sağlık bilimi, sosyal bilimler ve davranış bilimleri gibi alanlarda yaygın olarak kullanılmaktadır. Bu alanlarda çalışırken veri madenciliği yapay zekâ, yapay sinir ağları, bulanık mantık, neuro fuzzy sistemler, bayesian ağları, metasezgisel algoritmalar, karar ağaçları, çoklu regresyon analizi ve zaman serileri analizi gibi teknikler kullanmaktadır (Karacan, 2010).

Bu tez çalışmasında metasezgisel algoritmalar kullanılmıştır. Metasezgisel algoritmalar, verilen kısıtlar kümesi altında seçilen bir optimizasyon problemi için hedef fonksiyonuna uygun olarak karar değişkenlerinin mümkün olan en iyi değerlerinin elde edildiği bir prosedürdür. En yaygın optimizasyon prosedürleri herhangi bir özel hedefi maksimize ya da minimize edecek bir tasarım için geçerlidir.

Günümüzde deterministik ve stokastik bağlamlarda sezgisel tabanlı arama teknikleriyle çalışan birçok optimizasyon algoritması bulunmaktadır. Optimizasyon yaklaşımının çeşitli problemlere uygulanabilirliğini arttırmada güçlü optimizasyon algoritmaları geliştirmek için doğal ve fiziksel ilkelerden ilham alınmaktadır. Evrimsel algoritmalar, benzetimli tavlama, karınca koloni optimizasyonu, memetik algoritmalar, parçacık sürüsü optimizasyonu bu tür algoritmalara örnektir (Arunachalam, 2008).

Optimizasyon bir amaç için belirtilen şartları sağlaması koşuluyla en uygun çözümü bulmayı amaçlamaktadır. Optimizasyon üzerine çalışmalar yapıldıkça farklı teknikler de ortaya çıkmaktadır. Yaygın olarak kullanılan tekniklerden bazıları genetik algoritmalar, diferansiyel gelişim algoritması, karınca kolonisi algoritması ve parçacık sürü algoritmasıdır. Bu teknikler doğadaki olaylardan esinlenerek geliştirilmiş popülasyon tabanlı sezgisel yöntemlerdir. Genetik algoritmalar, canlıların genetik yapısından esinlenerek geliştirilmiştir. Diferansiyel gelişim algoritması yapısı itibariyle genetik algoritmaya benzemektedir. Diferansiyel gelişim algoritması da canlıların genetik yapısından esinlenerek geliştirilmiştir. Karınca kolonisi algoritması, karınca sürülerinin yiyecek ararken yaptıkları davranışlarından esinlenerek geliştirilmiştir. Parçacık sürü optimizasyonu algoritması ise balık, kuş, böcek gibi hayvan sürülerinin davranışlarından esinlenerek geliştirilmiş popülasyon tabanlı stokastik bir optimizasyon yöntemidir.

Bu tez çalışmasında, metasezgisel algoritmalardan genetik algoritma, parçacık sürü algoritması ve diferansiyel gelişim algoritması kullanılarak veri madenciliğinde aykırı değer tespiti yapılacaktır. Daha sonra tespit edilen aykırı değerlerin sınıflandırma başarısı üzerindeki etkisi değerlendirilecektir. Bu bağlamda 1. bölümde

aykırı değer tespiti ve metasezgisel algoritmalara yer verilecektir. 2. bölümde konuyla ilgili literatür araştırmasına yer verilecektir. 3. bölümde uygulamanın materyal ve yöntemine yer verilecektir. 4. bölümde araştırma bulgularına yer verilecektir ve 5. bölümde ise sonuç ve öneriler kısmına yer verilecektir.

1.1. Aykırı Değer Tespiti

1.1.1. Veri madenciliğinde aykırı değer tespiti

Veri madenciliğinde anormallik tespiti (sapma tespiti) bir veri setinde beklenen bir kalıba veya diğer maddelere uymayan öğelerin, olayların ve gözlemlerin tanımlanmasıdır (Chandola ve ark., 2009).

Genellikle anormal değerler banka dolandırıcılığı, yapısal bir kusur, tıbbi problemler veya bir metindeki hatalar gibi bir takım problemlere dönüşebilmektedir. Anormalliklere aykırı değerler, yenilik algılama, sapma ve istisnalar gibi tanımlamalar yapılmaktadır (Hodge ve Austin, 2004).

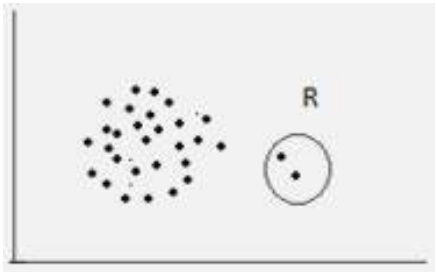
Aykırı değerlerin birçok veri madenciliği senaryosunda çok sayıda uygulaması vardır. Bu uygulamalardan bazılarının amaçları aşağıdaki gibidir (Hodge ve Austin, 2004):

- Kredi başvuru işlemi, örneğin sahte uygulamaları veya potansiyel olarak sorunlu müşterileri tespit etmek.
- İzinsiz giriş tespiti, örneğin bilgisayar ağlarında yetkisiz erişim tespiti yapmak.
- Faaliyet izleme, örneğin telefon faaliyetini izleyerek cep telefonu dolandırıcılığını tespit etmek.
- Ağ performansı, örneğin bilgisayar ağlarının performansını izleyerek ağ saldırılarını tespit etmek.
- Dolandırıcılık tespiti, örneğin kredi kartı veya cep telefonlarının hileli kullanımını tespit etmek.

- Arıza teşhisi, örneğin uzay mekiğindeki motorlar, jeneratörler, boru hatları veya uzay araçlarındaki arızaları tespit etmek.
- Yapısal hata tespiti, örneğin üretimdeki ürün hatalarını tespit etmek.
- Uydu görüntü analizi, örneğin yeni özellikler veya yanlış sınıflandırılmış özellikleri tanımlamak.
- Hareket bölütleme, örneğin arka plandan bağımsız olarak hareket eden görüntünün özelliklerini algılamak.
- Tıbbi durum izleme, örneğin kalp atış hızını denetlemek.
- Metindeki yeniliğin tespit edilmesi, örneğin haberlerin başlangıcını tespit etmek, konu algılama ve konuyu takip etmek.
- Veri tabanında beklenmedik girdileri tespit etmek, örneğin hataları, dolandırıcılıkları veya geçerli ancak beklenmedik girişleri saptamak.
- Bir eğitim verisinde yanlış etiketlenmiş verileri saptamak.

1.1.2. Aykırı değer analizi

Şekil 1.1.'de, nesnelerin çoğu kabaca Gauss dağılımını izlemektedir. Ancak R bölgesindeki nesneler diğerlerinden oldukça farklıdır. Bu nesneler veri kümesindeki diğer nesneler gibi aynı dağılımı takip etmemektedirler. Böylece R bölgesindeki nesneler veri setinde aykırı veri noktaları olarak tanımlanmaktadır (Han ve ark., 2012).



Şekil 1.1. R bölgesindeki aykırı olan nesnelerin gösterimi

Bir veri noktasının "aykırılık değeri" kendisine karşılık gelen sayısal bir değerle ölçülmektedir. Aykırılık değeri için aykırı değer tespit uygulamaları iki tür çıktı üretmektedir. Birincisi, gerçek değerli aykırılık skorudur. Böyle bir skor bir veri

noktasının bir aykırı değer olarak kabul edilme eğilimini nicelleştirmektedir. Skorun daha yüksek değerleri belirli bir veri noktasının bir aykırı değer olduğu olasılığını artırmaktadır. Bazı algoritmalar belirli bir veri noktasının bir aykırı değer olma ihtimalini nicelendiren bir olasılık değeri bile çıkartabilmektedir. İkincisi, bir veri noktasının bir aykırı değer olup olmadığını gösteren bir ikili değer çıktısı veren bir etikettir. Bu tür çıktılar gerçek değerli aykırılık skoruna göre daha az bilgi içerir çünkü ikili etiketler haline getirmek için aykırı skorlara bir eşik uygulanmaktadır. Bu nedenle aykırı skorlar ikili etiketlerden daha geneldir (Aggarwal, 2015).

1.1.3. Aykırı değerlerin türleri

Genelde aykırı değerler aşağıdaki gibi 3 kategori içerisinde sınıflandırılmaktadır:

- Küresel aykırı değerler
- Bağlamsal aykırı değerler
- Kolektif aykırı değerlerdir

1.1.3.1. Küresel aykırı değerler

Verilen bir veri setinde bir veri nesnesi bazen veri kümesinin geri kalanından önemli derecede ayrılabilmektedir. Bu durumda bu veri nesnesi küresel bir aykırı değer olarak algılanmaktadır. Küresel aykırı değerler, aykırı değer türlerinin en basitini oluştururlar. Bu nedenle çoğu aykırı değer tespit metodu küresel aykırı değerleri bulmayı amaçlamaktadır (Han ve ark., 2012).

1.1.3.2. Bağlamsal (kontekste bağlı) aykırı değerler

Verilen bir veri setinde, bir veri nesnesi eğer veri setinin belirli bir içeriğine göre önemli derecede bir sapma gösteriyorsa bu bir kontekste dayalı aykırı değerdir. Kontekste dayalı aykırı değerler aynı zamanda koşula bağlı aykırı değerler olarak da bilinmektedir. Kontekste dayalı aykırı değer tespitinde kontekst, problem tanımının

özel bir parçası olmak zorundadır. Genellikle kontekste dayalı aykırı değer tespitinde söz konusu veri nesnelerinin özellikleri iki gruba ayrılmaktadır. Birincisi, kontekste dayalı özelliklerdir. Bunlar nesnenin kontekstini belirlemektedir. İkincisi, davranışsal özelliklerdir. Davranışsal özellikler nesnenin özelliklerini belirlemektedir ve nesnenin ait olduğu bağlamda bir aykırı değer olup olmadığını değerlendirmek için kullanılmaktadır (Han ve ark., 2012).

1.1.3.3. Kolektif aykırı değerler

Bir veri kümesi verildiğinde veri kümesinin bir alt kümesini oluşturan bir takım nesneler bütün olarak tüm veri kümesinden büyük ölçüde ayrılabilmektir. Bu tür durumlarda bu nesneler kolektif aykırı değer oluşturmaktadır.

Şekil 1.2.'de bir bütün olarak siyah nesneler kolektif bir aykırı değer oluşturmaktadır. Çünkü bu nesnelerin kendi aralarındaki yoğunluğu, veri setinin diğer nesneleriyle olan yoğunluğundan daha fazladır. Ancak bu siyah nesneler bireysel olarak değerlendirildiğinde veri setinin bütünüyle ilgili bir aykırı değer oldukları söylenemez.



Şekil 1.2. Kolektif aykırı değerlerin gösterimi

Küresel ya da kontekste dayalı aykırı değer tespitinin aksine, kolektif aykırı değerde sadece bireysel nesnelerin davranışı değil aynı zamanda nesnenin grup olduğu da göz önünde bulundurulmaktadır. Bu yüzden kolektif aykırı değer tespiti için verinin nesneleri arasındaki mesafeleri ve benzerlikleri gibi bilgilerine gereksinim duyulmaktadır (Han ve ark., 2012).

1.1.4. Aykırı değer tespitinin zorlukları

Normal nesneleri ve aykırı değerleri etkin bir şekilde modellemede bir takım zorluklarla karşılaşılmaktadır. Çünkü veri normalliği ve anormalliği arasındaki sınır genellikle net değildir. Bunun yerine geniş bir gri alan aralığı olabilmektedir. Sonuç olarak bazı aykırı değer tespit yöntemleri girdi verisindeki her nesneye "normal" veya "aykırı" bir etiket verirken diğer yöntemler her nesneye nesnenin ”aykırılık değerini" ölçen bir puan atamaktadır.

Uygulamaya özel aykırı değerleri tespit etmede de zorluklarla karşılaşılır. Çünkü farklı uygulamalar farklı gereksinimlere ihtiyaç duymaktadır. Örneğin klinik veri analizinde küçük bir ayırım bir aykırı değer olduğunu kabul etmekte yeterliyken bir pazarlama analizinde çok büyük dalgalanmalara gereksinim duyulmaktadır.

Aykırı değer tespitinde gürültünün varlığından dolayı aykırı değerler yanlış değerlendirilmektedir. Düşük veri kalitesi ve gürültünün var olması aykırı değer tespitinde normal nesneler ve aykırı değerler arasındaki farkı bulanıklaştırmaktadır. Üstelik gürültü ve eksik veriler aykırı değerleri gizleyerek aykırı değer belirleme etkinliğini azaltabilmektedir. Bununla birlikte bir gürültü noktası da bir aykırı değer gibi görünebilmektedir. Bu yüzden aykırı değer tespit metodu yanlışlıkla bir gürültü noktasını bir aykırı değer gibi tanımlayabilmektedir.

Aykırı değer tespitinde aykırılığın anlaşılmasının gerektiği durumlar olabilmektedir. Bazı uygulama senaryolarında bir kullanıcı sadece aykırı değer tespiti yapmak istemez, aynı zamanda tespit ettiği nesnelerin neden aykırı değer olduğunu anlamak ister. Bu yüzden aykırı değer tespit metodu aykırılığı açıklayacak doğrulamaları da sağlamalıdır (Han ve ark., 2012).

1.1.5. Aykırı değer tespit metotları

Literatürde ve pratikte çok fazla sayıda aykırı değer tespit metodu bulunmaktadır. Aykırı değer tespit metotları genel olarak iki yöntemle kategorize

edilmektedir. İlk yöntemde aykırı değer tespit yöntemleri, aykırı değer tespit modeli oluşturmak için alan uzmanı tarafından sağlanan etiketlere göre sınıflandırılır. Bu yöntemler gözetimli, yarı gözetimli ve gözetimsiz metotlardır. İkinci yöntemde aykırı değer tespit yöntemleri, normal nesnelerle aykırı değerler arasındaki varsayımlarına göre gruplandırılır. Bu yöntemler istatistiksel metotlar, yakınlık tabanlı metotlar ve kümeleme tabanlı metotlardır.

Gözetimli metotlar normal ve anormal verileri modellemektedir. Gözetimli aykırı değer tespitinde aykırı değerlerin popülasyonu tipik olarak normal nesnelerden daha azdır. Bundan dolayı alan uzmanları tarafından incelenen ve eğitimde kullanılan örnek veriler, aykırı değer dağılımını temsil etmek için yeterli değildir. Bu problemlerle başa çıkmak için bazı yöntemler yapay aykırı değerler oluşturmaktadır. Ayrıca gözetimli aykırı değer tespiti uygulamalarında aykırı değerlerin çoğunu yakalamak mümkün olduğu gibi normal nesnelerin aykırı değer gibi yanlış etiketlenmesi de görülmektedir.

Gözetimsiz metotlar aykırı değerden ziyade daha çok normal nesneleri bir kalıp olarak takip etmeyi seçmektedir. Gözetimsiz metotlarda normal nesnelerin yüksek benzerliği paylaşan tek bir gruba girmesi gerekmemektedir. Bunun yerine her grubun belirgin özelliklere sahip olduğu birden fazla grup oluşturulmaktadır.

Çoğu kümeleme metotları gözetimsiz aykırı değer tespit metotları gibi hareket etmeye adapte olmaktadır. Temel fikir kümeleme işlemini yapmak ve daha sonra herhangi bir kümeye ait olmayan veri nesnelerini aykırı değer olarak algılamaktır. Ancak bu işlemleri yaparken herhangi bir kümeye ait olmayan bir veri nesnesinin bir aykırı değer değil de gürültü olduğu durumlar oluşmaktadır. Bu durumlarda gürültü aykırı değer olarak görülmektedir.

Çoğu aykırı değer tespit uygulamalarında, bazı etiketli örnekleri elde etmek kolay olmasına rağmen, etiketli örneklerin sayısı genelde düşük olabilmektedir. Bu durumda aykırı ya da normal nesnelerin sadece küçük bir bölümünün etiketlendiği ya da verilerin çoğunun yanlış etiketlendiği durumlarla karşılaşılabilmektedir. Yarı

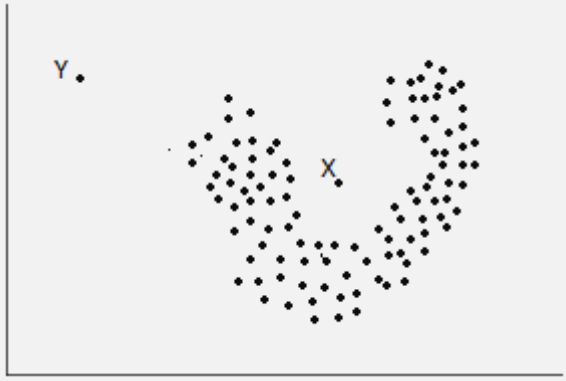
gözetimli aykırı değer tespit metotları bu türden senaryoları ele almak için geliştirilmiştir. Ayrıca yarı gözetimli aykırı değer tespit metotları, yarı gözetimli öğrenme metotlarının uygulamaları olarak görülmektedir (Han ve ark., 2012).

1.1.5.1. Aşırı değer analizi

Aşırı değer analizi, verilerin dışında kalan veri noktalarının aykırı değerler olarak rapor edildiği çok belirgin bir aykırı değer analizi türüdür. Bu türden aykırı değerler olasılık dağılımlarının istatistiksel kuyruklarına karşılık gelmektedirler.

Aşırı değerlerin çok belirgin şekil bozuklukları olduğunu anlamak önemlidir. Başka bir deyişle, tüm aşırı değerler aykırı değerlerdir, ancak tersi doğru olmayabilmektedir. Örneğin, 1 boyutlu değerlerin {1, 2, 2, 50, 98, 98, 99} veri setinde 1 ve 99 değerleri aşırı değerler olarak düşünülmektedir. Öte yandan 50 değeri, veri kümesinin ortalamasıdır ve kesinlikle aşırı bir değer değildir. Bununla birlikte 50 değeri, {1, 2, 2} ve {98, 98, 99} gibi küçük aralıklarla gruplandırılan diğer veri değerlerinin çoğundan izole edilmiştir. Bu nedenle, çoğu olasılık ve yoğunluk tabanlı modeller, 50 değerini veride güçlü bir aykırı değer olarak sınıflandırmaktadır. Aşırı değer analizi ve aykırı değer analizi arasındaki karışıklıklar özellikle çok değişkenli veriler bağlamında yaygındır (Aggarwal, 2015).

Şekil 1.3.'deki örneğe göre Y veri noktası aşırı bir değerdir ve aynı zamanda bir aykırı değer olarak düşünülebilir. Bununla birlikte, X veri noktası da ayrılmıştır ve bu nedenle bir aykırı değer olarak düşünülmelidir. Ancak, çok değişkenli bir aşırı değer olarak kabul edilemez.



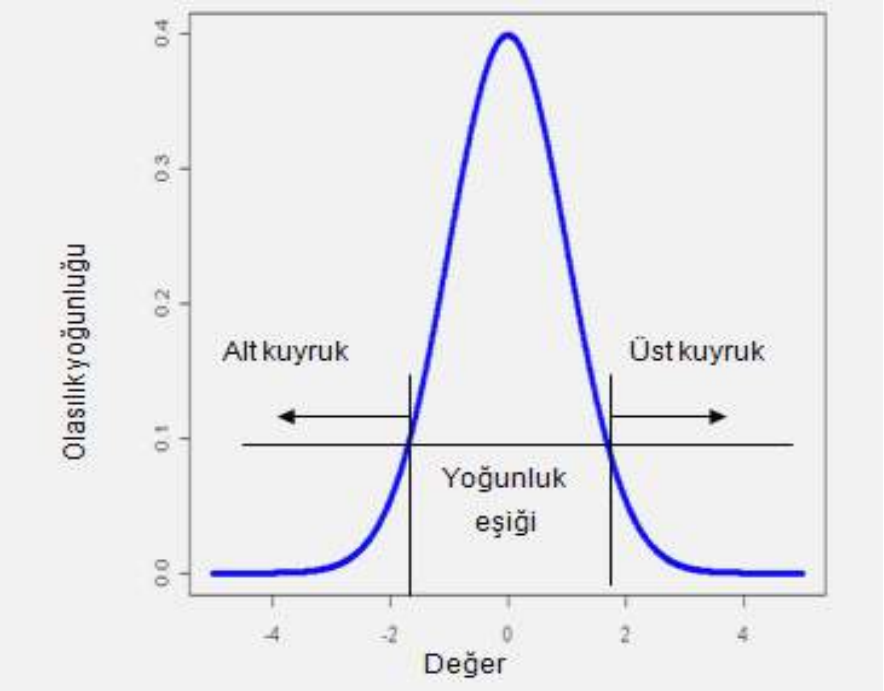
Şekil 1.3. Aşırı değer ve aykırı değer farkının gösterimi

Aşırı değer analizi, tek değişkenli ve çok değişkenli aşırı değer analizi şeklinde iki kategoride incelenmektedir.

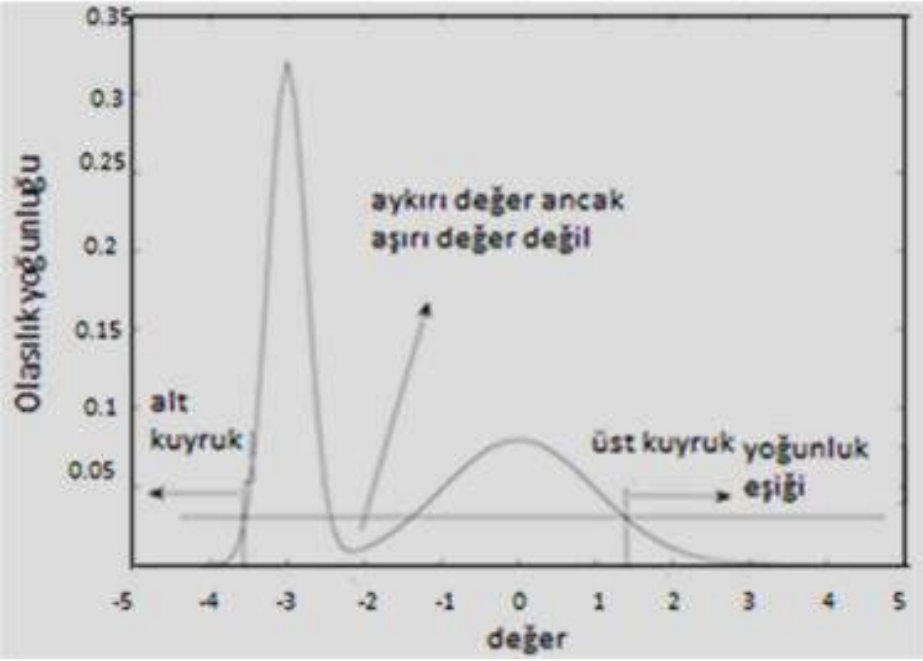
Tek değişkenli aşırı değer analizi, istatistiksel kuyruk güven testi kavramıyla yakından ilgilidir. Tipik olarak istatistiksel kuyruk güven testleri 1 boyutlu verilerin belirli bir dağılım ile tanımlandığını varsaymaktadır. Bu yöntemler, bu dağılım varsayımlarına dayanarak bir veri noktasından daha aşırı olan nesnelerin kesimini belirlemeye çalışmaktadır (Aggarwal, 2015).

Simetrik olmayan bir dağılımın kuyruğu tanımlanırken üst kuyruk ve alt kuyruk kavramları tanımlanmalıdır. Üst kuyruk, belirli bir eşiğin üzerinde olan tüm aşırı değerler olarak tanımlanmaktadır ve alt kuyruk, belirli bir eşiğin altındaki tüm aşırı değerler olarak tanımlanmaktadır. Yoğunluk dağılımı f(x) ve θ kullanıcı tanımlı yoğunluk eşiği olmak üzere, kuyruk dağılımın uç bölgesi $f(x) \leq \theta$ şeklinde tanımlanmaktadır.

Simetrik ve asimetrik dağılımlar için alt kuyruk ve üst kuyruk örnekleri sırasıyla Şekil 1.4.'de ve Şekil 1.5.'de gösterilmiştir. Şekil 1.5.'de görüldüğü gibi asimetrik dağılımın üst kuyruk ve alt kuyruk bölgesindeki alanlar aynı değildir. Ayrıca dağılımının iç kısmındaki bazı bölgeler yoğunluk eşiğinin (θ) altındaki yoğunluğa sahip olmalarına rağmen aşırı değer değildirler. Çünkü dağılımın sonuna gelmemektedirler. Bu bölgedeki veri noktaları aykırı olarak kabul edilebilir, ancak aşırı değer olarak kabul edilemezler.



Şekil 1.4. Simetrik aşırı değer analizi



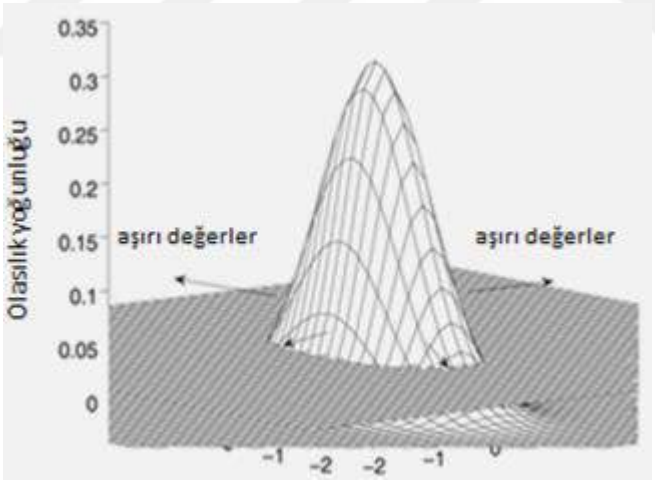
Şekil 1.5. Asimetrik aşırı değer analizi

Yoğunluk eşiği kavramı, özellikle asimetrik tek değişkenli veya çok değişkenli dağılımlar söz konusu olduğunda kuyruk için tanımlayıcı özelliğe sahiptir.

Kuyruk olasılığını belirlemek için bir model dağılımı seçilir. En yaygın kullanılan model normal dağılımdır. Normal dağılımın ortalama μ ve standart sapma σ olmak üzere, yoğunluk fonksiyonu f(x) aşağıdaki gibi tanımlanır (Aggarwal, 2015).

$$f(x) = \frac{1}{\sigma.\sqrt{2.\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}} \quad (1.1)$$

Tek değişkenli aşırı değer analizinde olasılık yoğunluğunun belirli bir eşikten daha az olduğu bölgeler aşırı değer olarak tanımlanırken çok değişkenli dağılımlar için de benzer bir kavram tanımlanmaktadır. Kavram tek değişkenli olgudan daha karmaşıktır ve Şekil 1.6.'da da görüldüğü gibi tek bir tepe noktası olan tekdüzelik olasılık dağılımları için tanımlanmıştır. Bir önceki durumda olduğu gibi çok değişkenli bir Gauss modeli kullanılır ve ilgili parametreler veri odaklı bir şekilde tahmin edilir.



Şekil 1.6. Çok değişkenli aşırı değer analizi

Çok değişkenli aşırı değer analizinin örtük modelleme varsayımı, tüm veri noktalarının tek bir zirveye sahip bir olasılık dağılımında bulunmasına dayanmakta ve tüm yönlerden kümenin merkezine mümkün olduğunca uzak olan veri noktalarını aşırı değerler olarak kabul etmektedir.

Çok değişkenli Gauss dağılımının parametreleri için $\bar{\mu}$, d-boyutlu bir veri kümesinin ortalama vektörü olsun ve σ, d × d kovaryans matrisi olsun. Bu durumda d-boyutlu $\overline{X}$ veri noktası için olasılık yoğunluk fonksiyonu $f(\overline{X})$ aşağıdaki gibi tanımlanabilmektedir.

$$f(\overline{X}) = \frac{e^{\frac{-1(\overline{X}-\bar{\mu}).\sigma^{-1}.(\overline{X}-\bar{\mu})^T}{2}}}{\sqrt{|\sigma|}.(2.\pi)^{\frac{d}{2}}} \quad (1.2)$$

Maha ($\overline{X}$, $\bar{\mu}$, σ), kovaryans matrisi σ'ye göre $\overline{X}$ ve $\bar{\mu}$ arasındaki Mahalanobis mesafesini temsil ediyorsa normal dağılımın olasılık yoğunluk fonksiyonu aşağıdaki gibi tanımlanmaktadır (Aggarwal, 2015).

$$f(\overline{X}) = \frac{e^{\frac{-1}{2}.Maha(\overline{X},\bar{\mu},\sigma)^2}}{\sqrt{|\sigma|}.(2.\pi)^{\frac{d}{2}}} \quad (1.3)$$

1.1.5.2. Kümeleme tabanlı modeller

Kümeleme tabanlı yaklaşımlar nesneler ve kümeler arasındaki ilişkiyi inceleyerek aykırı değerleri saptamaktadır. Kümeleme tabanlı aykırı değer tespiti için üç genel yaklaşım sunulur. Bu yaklaşımlar aşağıdaki gibidir:

- Nesne herhangi bir kümeye ait değilse aykırı değer olarak tanımlanır.
- Nesne ve ona en yakın küme arasında büyük bir mesafe varsa nesne aykırı değer olarak tanımlanır.
- Nesne küçük ya da seyrek bir kümenin parçasıysa, o kümedeki tüm nesneler aykırı değer olarak tanımlanır (Han ve ark., 2012).

Aykırı değerlerin çoğu zaman küçük kümeler halinde ortaya çıkma eğilimi görülmektedir. Bunun nedeni üretim sürecindeki aykırılıkların birkaç kez tekrarlanabilmesidir. Sonuç olarak aykırı değerlerden oluşan küçük gruplar oluşabilmektedir. Kümeleme yöntemleri genellikle bu tür senaryolara karşı

dayanıklıdır. Çünkü bu tür gruplar genellikle kendi kümelerini oluşturmak için gereken kritik kütleye sahip olmamaktadırlar (Aggarwal, 2015).

Bir veri noktasının aykırılık skorunu tanımlamanın basit bir yolu ilk olarak veri noktalarını kümelemek ve daha sonra her bir veri noktasının ham (raw) mesafesini en yakın küme merkezine doğru kullanmaktır. Ancak yerel veri dağılımı sıklıkla mesafeleri çarpıtabilmektedir. Bundan dolayı ham (raw) mesafeyi kullanmak çok doğru olmamaktadır. Bu tür durumlarda yerel Mahalanobis mesafesi en yakın kümenin merkezine göre kullanılmaktadır.

Mahalanobis uzaklığı, aykırı değerleri bulmaya yarayan bir uzaklık ölçütüdür. Mahalanobis uzaklığı kovaryans matrisini kullanarak iki nokta arasındaki mesafeyi hesaplamaya çalışmaktadır. Bu şekilde sadece aralarındaki mesafesi belirlenecek iki noktayı değil aynı zamanda veri setindeki diğer noktaları da göz önünde bulundurmuş olmaktadır.

D-boyutlu uzayda r kümesine karşılık gelen d-boyutlu ortalama vektörü μ_r ve d × d matrisinin kovaryans matrisi σ_r'ye sahip olduğu varsayılırsa, bir veri noktası $\bar{X}$ ve μ_r arasındaki Mahalanobis mesafesi aşağıdaki denklemde olduğu gibi tanımlanmaktadır.

$$\text{maha}(\bar{X}, \bar{\mu_r}, \sigma_r) = \sqrt{(\bar{X} - \bar{\mu_r}).\sigma_r{}^{-1}(\bar{X} - \bar{\mu_r})^T} \qquad (1.4)$$

Aykırı skor belirlendikten sonra skorları ikili etiketler haline dönüştürmek için tek değişkenli aykırı değer analizi kullanılmaktadır.

1.1.5.3. İstatistiksel yaklaşımlar

Kümeleme için istatistiksel metotlarda olduğu gibi aykırı değer tespitinde de istatistiksel metotlar veri normalliği ile ilgili varsayımlar yapmaktadır. İstatistiksel yaklaşımlar bir veri setinde normal nesnelerin istatistiksel bir süreç tarafından üretildiğini varsaymaktadır. Sonuç olarak nesnenin istatistiksel modelin bölgesinde

bulunma ihtimali yüksek ise normal bir nesnedir fakat bulunma ihtimali düşük ise aykırı bir nesnedir. Genellikle aykırı değer tespiti için istatistiksel metotlar iki temel kategoriye ayrılmaktadır. Bunlar parametrik ve nonparametrik metotlardır.

Parametrik metotlar, tek değişkenli ve çok değişkenli olmak üzere iki şekilde incelenir. Tek değişkenli aykırı değerlerin tespitinde sadece tek bir değişken ya da nitelik içeren veriler bulunmaktadır. Çok değişkenli aykırı değer tespitinde ise iki ya da daha fazla değişken ya da nitelik içeren veriler bulunmaktadır. Çok değişkenli veriyi işlemek için birçok tek değişkenli aykırı değer tespit yöntemleri kullanılmaktadır. Temel fikir, çok değişkenli aykırı değer tespit görevini tek değişkenli aykırı değer tespit problemine dönüştürmektir.

Kutu grafiği yöntemi, aykırı değerleri yakalamak için parametrik bir istatistiksel model olarak kullanılmaktadır.

Şekil 1.7.'de görüldüğü gibi Q2, dizi şeklinde sıralanmış bir verinin orta indeksine sahip verinin değeridir. Q1 ve Q3 ise sırasıyla veri boyutunun ¼ ve ¾ indeksine sahip veri değerleridir. IQR= Q3-Q1 olmak üzere,

Alt uç değer =Q1 - 1.5(IQR) ve
Üst uç değer= Q3 + 1.5(IQR) şeklinde hesaplanmaktadır.



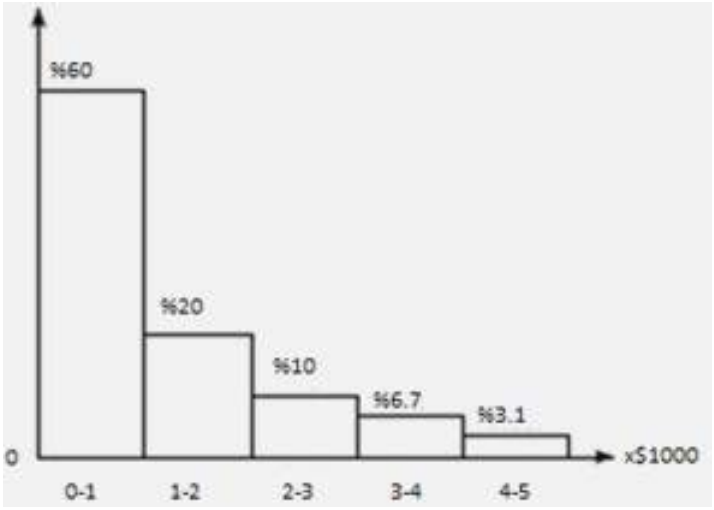
Şekil 1.7. Kutu grafiği gösterimi

Bu durumda alt uç değerden daha küçük uç değerler ve üst uç değerden daha büyük uç değerler aykırı değer olarak kabul edilmektedir (Taşdelen ve Ovla, 2012).

Parametrik olmayan yöntemler genellikle verilerle ilgili daha az varsayımda bulunurlar ve bu nedenle daha fazla senaryoda uygulanırlar.

Histogram, aykırı değerleri yakalamak için parametrik olmayan istatistiksel bir model olarak kullanılmaktadır. Ancak bunun bir dezavantajı, uygun bir kutu boyutunun seçilmesinin zor olmasıdır. Bir taraftan kutu boyutu çok küçük ayarlanmışsa birçok normal nesne boş kutulara denk gelmekte ve bu nedenle aykırı değerler olarak yanlış tanımlanmaktadır. Öte yandan kutu boyutu çok büyük olarak ayarlanmışsa aykırı değerler sık kullanılan kutulara denk gelmekte ve normal nesneler gibi algılanmaktadır.

Örneğin bir şirketin, her müşteri işlemi için satın alma tutarını kaydettiği kabul edilsin. İşlemlerin verilen miktarlarının yüzde olarak grafiğini oluşturmak için Şekil 1.8.'de görüldüğü gibi bir histogram kullanılsın ve işlem tutarlarının%60'ı 0 ile 1000 dolar arasında olduğu kabul edilsin. Bu durumda 7500 $ tutarındaki bir işlem, yalnızca 1-(%60 + %20 + %10 + %6.7 + %3.1)= %0.2 yüzdelik dilimine denk gelmektedir. Ayrıca 5000 $ 'dan yüksek bir miktara sahip olması nedeniyle aykırı bir değer olarak kabul edilmektedir. Öte yandan 385 $, işlemlerin %60'ını tutan kutuya düştüğü için normal bir değer olarak kabul edilmektedir (Han ve ark., 2012).



Şekil 1.8. Histogram gösterimi

1.1.5.4. Yakınlık temelli yaklaşımlar

Özellik uzayında bir dizi nesne göz önüne alındığında nesneler arasındaki benzerliği ölçmek için bir mesafe ölçüsü kullanılmaktadır. Yakınlık temelli yaklaşımlar, aykırı bir nesnenin en yakın komşularına yakınlığının veri kümesindeki diğer nesnelerin çoğuna yakınlığından büyük ölçüde farklı olduğunu varsaymaktadır. Yakınlık temelli yaklaşımlardan en çok kullanılan mesafe tabanlı aykırı değer tespitidir.

Mesafeye dayalı aykırı değer tespit yöntemi belirli bir yarıçapla tanımlanan bir nesnenin komşuluğunu ele almaktadır. Komşuluğunda yeterli sayıda başka nokta bulunmaması durumunda bu nesneyi bir aykırı değer olarak kabul etmektedir.

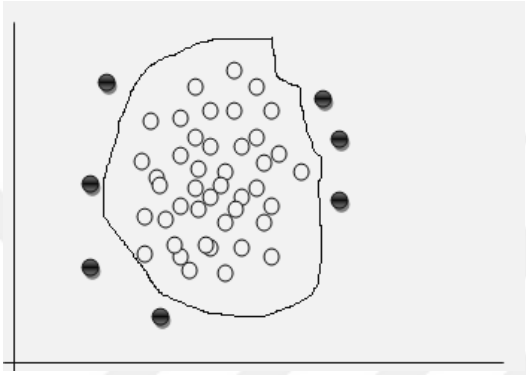
Analiz edilecek bir veri seti için kullanıcı bir nesnenin makul bir komşuluğunu tanımlamak için bir mesafe eşiği belirtmektedir. Her bir nesne için o nesnenin komşuluğundaki diğer nesnelerin sayısı incelenmektedir. Veri setinde nesnelerin çoğunun bir veri noktasından uzak olması, o veri noktasının aykırı bir değer olarak kabul edilmesine neden olmaktadır (Han ve ark., 2012).

1.1.5.5. Sınıflamaya dayalı yaklaşımlar

Sınıflamaya dayalı aykırı değer tespit yöntemlerinin genel fikri, normal verileri aykırı değerlerden ayıran bir sınıflandırma modeli oluşturmaktır.

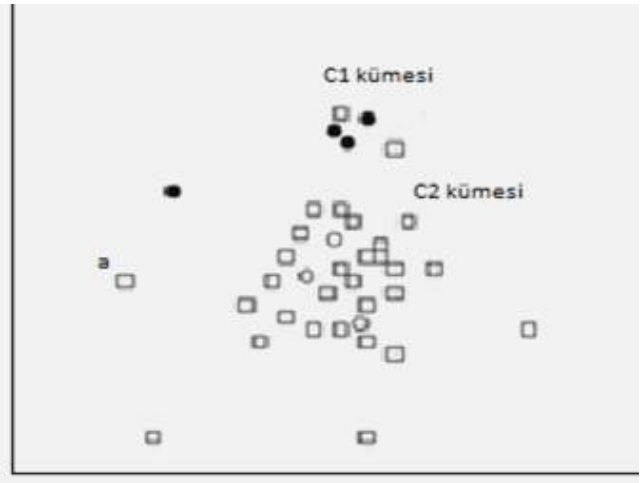
Normal örnekleri normal olarak etiketleyen ve normal olmayan örnekleri de aykırı değer olarak etiketleyen bir eğitim setinde normal örneklerin sayısı muhtemelen aykırı değer örneklerinin sayısını aşmaktadır. Aykırı örneklerin sayısının yetersiz kaldığı bu durumda doğru bir sınıflandırıcının oluşması zor olmaktadır. Bu sorunun üstesinden gelmek için sınıflandırma tabanlı aykırı değer tespit yöntemlerinde genellikle tek sınıf bir model kullanılmaktadır. Yani sınıflandırıcı yalnızca normal sınıfı tanımlamak için oluşturulmaktadır. Normal sınıfa ait olmayan örnekler aykırı değer olarak kabul edilmektedir.

Sınıflandırmaya dayalı yaklaşımlar için tek sınıf bir model kullanılarak aykırı değer tespiti yapılmaktadır. Şekil 1.9.'da bir veri seti için beyaz noktaların "normal" olarak etiketlenmiş örnekler olduğu ve siyah noktaların "aykırı" olarak etiketlenmiş örnekler olduğu kabul edilmektedir. Yeni bir nesne verildiğinde, nesne normal sınıfın sınırları içindeyse normal bir durum olarak kabul edilmektedir. Nesne normal sınıfın sınırları dışında ise aykırı değer olarak kabul edilmektedir.



Şekil 1.9. Normal sınıflardan bir model öğrenme

Sınıflandırmaya dayalı yaklaşımlar için yarı gözetimli aykırı değer tespiti yöntemleri de kullanılarak aykırı değer tespiti yapılmaktadır. Şekil 1.10.'da büyük bir küme C2 ve küçük bir küme C1 görülmektedir. C2'deki bazı nesneler "normal" etiketi taşıdığından, bu kümedeki tüm nesneler (etiketsiz olanlarla birlikte) normal nesneler olarak eğitilebilmektedir. Benzer şekilde küme C1'deki bazı nesneler " aykırı değer " etiketi taşıdığından, C1'deki tüm nesneler aykırı değer olarak kabul edilmektedir. C2 için modele girmeyen herhangi bir nesne (ör., a) de aykırı bir değer olarak kabul edilir (Han ve ark., 2012).



Şekil 1.10. Yarı gözetimli metotlarla aykırı değer tespiti

1.1.6. Bağlamsal ve kolektif aykırı değerlerin madenciliği

Belirli bir veri kümesindeki bir nesne, nesnenin belirli bir içeriğine göre belirgin bir sapma yapıyorsa içeriğe dayalı bir aykırı değerdir (veya koşullu aykırı değerdir). Bağlam, bağlamsal nitelikler kullanılarak tanımlanmaktadır. Bunlar ağırlıklı olarak uygulamaya bağlıdır ve çoğunlukla kullanıcıların içeriğe dayalı aykırı değer tespiti görevinin bir parçası olarak sağlanır. Bağlamsal nitelikler mekânsal nitelikler, zaman, ağ konumları ve uygun bir şekilde yapılandırılmış öznitelikleri içerebilir. Buna ek olarak davranışsal nitelikler nesnenin özelliklerini tanımlar ve nesnenin ait olduğu bağlamda bir aykırı değer olup olmadığını değerlendirmek için kullanılır.

Bağlamsal aykırı değerleri belirlemek, karşılık gelen içeriği analiz etmeyi gerektirir. Bağlamların açıkça tanımlandığı durumlarda bağlamsal aykırı değer tespiti geleneksel aykırı değer tespitine dönüştürülür. Buradaki amaç, bağlamsal aykırı değer tespit sorununu tipik bir sapma belirleme soruna dönüştürmektir. Belirli bir veri nesnesi için, nesnenin bir aykırı değer olup olmadığı iki aşamada değerlendirilebilir. İlk adımda, bağlamsal nitelikleri kullanarak nesnenin bağlamı tespit edilir. İkinci adımda, geleneksel bir aykırı değer tespit yöntemi kullanılarak bağlam içindeki nesnenin aykırı değer skoru hesaplanır (Han ve ark., 2012).

1.2. Metasezgisel Algoritmalar

1.2.1. Genetik algoritmalar

Genetik algoritmalar (GA) John Holland tarafından 1960'larda keşfedildi. 1960'larda ve 1970'lerde Michigan Üniversitesi'ndeki Holland, öğrencileri ve meslektaşları tarafından geliştirildi. Evrim stratejileri ve evrimsel programlamanın tersine, Holland'ın özgün amacı belirli sorunları çözmek için algoritmalar tasarlamak değil, doğada olduğu gibi adaptasyon olgusunu incelemek ve doğal adaptasyon mekanizmalarının bilgisayar sistemlerine aktarılacağı yollar geliştirmekti (Melanie,1999).

Genetik algoritmalar diğer optimizasyon algoritmalarında olduğu gibi doğada gözlemlenen süreci taklit eden eniyileme yöntemlerinden biridir. Genetik algoritmalar problemlere tek bir çözüm üretmek yerine birbirinden bağımsız olan farklı çözümler üretmektedir. Arama uzayında aynı anda farklı noktaların değerlendirilmesiyle elde edilen bu çözüm değerleri bütünsel olarak en iyi çözümü bulma olasılığını yükseltmektedir.

Genetik algoritma çoğu uygulamada diğer eniyileme problemlerine göre başarılı olduğunu kanıtlamıştır. Ancak hiçbir optimizasyon algoritması en iyi çözümü bulacağını garanti etmemektedir. Genetik algoritmaların asıl amacı, optimizasyon algoritmaları dışındaki diğer algoritmalarla çözülmesi zor olan problemlere makul süre içerisinde kabul edilebilir çözümler sunmaktır.

Genetik algoritmalar parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, planlama, yapay zeka uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel eniyileme problemleri ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer eniyileme yöntemlerinin yanında başarılı sonuçlar vermektedir (Beasley ve ark., 1993).

1.2.1.1. Genetik algoritma temel kavramları

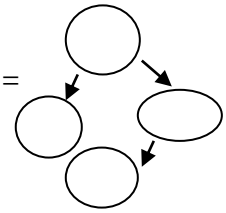
Genetik algoritmalar biyolojideki evrimsel süreci taklit ettiklerinden dolayı benzer temel kavramlara sahiptir. Bunlar gen, kromozom ve popülasyondur (Michalewicz, 1992).

Gen, canlıların en küçük kalıtım birimidir. Genetik algoritmada ise gen, popülasyonu oluşturan bireylerin kod yapısındaki en küçük birimdir. Bir genin yapısı genelde ikili tabandaki sayıları kullandığı gibi tamsayıları, karakterleri, ağaç biçimini ve benzeri sembolleri de kullanabilmektedir. Bu kodlama biçimlerinden hangisinin kullanılacağı problemin özelliğine göre programcıya bağlıdır. Ancak kodlama biçiminin algoritmanın performansı üzerinde büyük etkisi olduğundan kodlama biçimi seçilirken dikkat edilmelidir.

Kromozom, birden fazla genin bir araya gelmesiyle oluşur. Her bir kromozom popülasyonda bir bireyi temsil etmektedir.

Farklı kromozom yapıları aşağıdaki gibi gösterilmektedir:

Kromozom A= 010011001
Kromozom B=34223265
Kromozom C= 3.2 4.5 3.1 2.6
Kromozom D=ANHBCDALS
Kromozom E =



Kromozom A, bireyleri 0 ve 1 bitleri ile kodlamaktadır. Kromozom B, bireyleri rakamlarla kodlamaktadır. Kromozom C, bireyleri rasyonel değerlerle kodlamaktadır. Kromozom D, bireyleri karakterle kodlamaktadır ve kromozom E, bireyleri ağaç yapısıyla kodlamaktadır.

Popülasyon, muhtemel çözümleri içeren bireylerden meydana gelir. Popülasyondaki birey sayısı genin kodlama biçiminde olduğu gibi problemin özelliğine göre programcı tarafından belirlenir. Popülasyonun büyüklüğü problemin çözüm süresini etkilediğinden dolayı popülasyondaki birey sayısının fazla seçilmesi çözüm süresini uzatırken, az seçilmesi de popülasyonun istenen çözüm değerine ulaşılamamasına sebep olabilmektedir.

1.2.1.2. Genetik algoritmada kullanılan operatörler

Genetik algoritma doğal genetik çeşitlilikten esinlenen operatörleri kullanarak belirli bir problem için aday çözümler popülasyonunu geliştirmektedir. Bu operatörler seçme, mutasyon ve çaprazlamadır (Melanie, 1999).

Seçme yöntemleri, iterasyonlar boyunca mevcut topluluktan yeni bir topluluk elde etmede kullanılan işlemlerdir. En çok kullanılan seçim yöntemleri rulet seçimi, turnuva seçimi ve sıralı seçimdir.

Rulet seçiminde her birey uygunluk değerleriyle orantılı bir olasılık değerine sahiptir. Olasılık değeri büyük olan rulet tekerleği üzerinde daha fazla yer kaplar böylece tekerlek döndüğünde eşleşme havuzuna düşme ihtimali o kadar yüksek olur.

Turnuva seçiminde popülasyondan rastgele bazı bireyler seçilir. Seçilen bu bireylerden uygunluk değeri büyük olan eşleşme havuzuna atılır.

Sıralı seçimde her bir birey uygunluk değerleri oranında artan sıralı bir diziye atanır. En kötü uygunluk değerine sahip birey dizinin 1 numaralı indeksinde bulanacak şekilde sıralanır. Bu şekilde indeksi büyük olan eşleşme havuzuna atılır.

Mutasyon işlemi kromozomların genlerinde değişiklik yapma işlemidir. Bireylerin kromozom değerleri değişmedikçe uygunluk değerleri de değişmez ve program bir döngü gibi hep aynı sonuçları üretir. Bu da en iyi çözümü bulmada bir

engeldir. Mutasyon işleminde, bundan kaçınmak için kromozomu oluşturan genler üzerinde tersleme, yerleştirme, çıkarma, yer değiştirme gibi değişiklikler yapılır.

- Tersleme, kromozomdaki bir geni tersi değerle değiştirmektir.
  Örnek: 0110101 -> 0100101
- Yerleştirme, kromozoma yeni bir gen eklenmesiyle oluşur.
  Örnek: 1010101 -> 10110101
- Çıkarma, kromozomdan bir gen çıkartılarak yapılır.
  Örnek: 0110101 -> 010101 3. gen çıkarılmış.
- Yer değiştirme, kromozomdaki iki genin yeri değiştirilerek yapılır.
  Örnek: 110101 -> 100111 2. ve 5. gen yer değiştirmiş.

Çaprazlama, canlı genetiğindeki üremeye benzemektedir. Çaprazlamada birden fazla çözüm alınır ve bu çözümlerden başka bir çözüm üretilir. Amaç, uygunluk değeri yüksek olan bireylerden yeni uygunluk değeri yüksek bireyler elde etmektir. En çok kullanılan çaprazlama yöntemleri tek ve çift noktalı çaprazlama yöntemleridir.

Tek noktalı çaprazlamada popülasyondan iki tane kromozom seçilir. Bu kromozomların ilk ve son genleri hariç bir gen seçilir ve bu noktadan itibaren kromozomda kalan genler karşılıklı yer değiştirirler.

Kromozom-1: 11011|00100111011
Kromozom -2: 11001|11000010010
Çocuk-1: 1100111000010010
Çocuk 2: 1101100100111011

Çift noktalı çaprazlamada popülasyondan iki tane kromozom seçilir. Tek noktalı çaprazlamada olduğu gibi kromozomların ilk ve son genleri hariç iki nokta seçilir. Ve bu iki nokta arasındaki genler karşılıklı yer değiştirir.

Kromozom-1: 11000|00100|110110

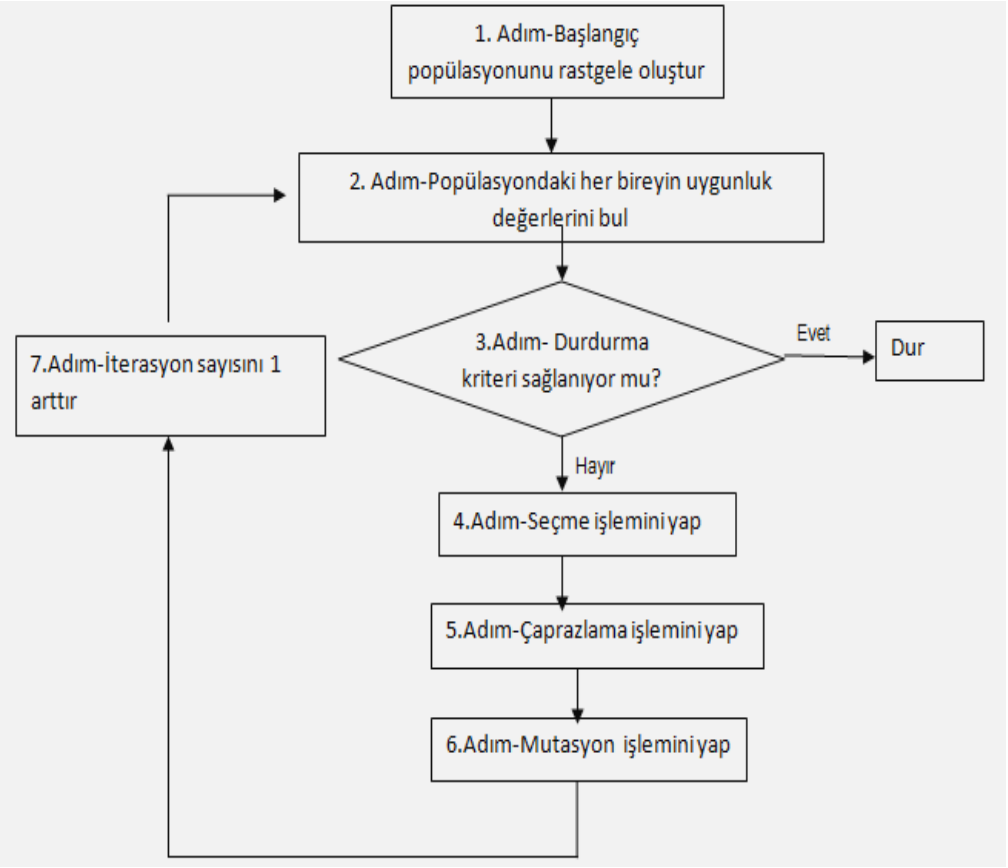
Kromozom-2: 11011|11000|011110

Çocuk-1: 1100011000110110

Çocuk-2: 1101100100011110

1.2.1.3. Genetik algoritması temel çalışma mantığı

Şekil 1.11.'de akış diyagramı verilmiş olan genetik algoritmanın temel çalışma mantığı aşağıda adımlar halinde değerlendirilmektedir.



Şekil 1.11. Genetik algoritma akış diyagramı

1. Adım: Başlangıç popülasyonu oluşturulur.

- Problemin kaç boyutlu olduğuna karar verilir.
- Probleme özel bir uygunluk fonksiyonu tanımlanır.
- Popülasyon sayısına, mutasyon türü ve olasılığına, çaprazlama türü ve olasılığına ve durdurma kriterine karar verilir.
- Popülasyon sayısı kadar birey yani kromozom kodlanır.

2. Adım: Uygunluk değerleri belirlenir.

- Popülasyonu oluşturan her bireyin uygunluk değeri hesaplanır.
- Tespit edilen uygunluk değerleri arasından en başarılı olanı iterasyonun sonuç değeri olarak kabul edilir.

3. Adım: Durdurma kriteri kontrol edilir.

- Durdurma kriteri sağlandıysa o anki sonuç problemin sonuç değeri olarak alınır.
- Durdurma kriteri sağlanmadıysa uygunluk değeri başarılı bulunan bireylere (çoğalma şansını artırmak için) başarılılıkları oranında bir olasılık değeri verilir.

4. Adım: Seçme işlemi yapılır.

- Seçme yöntemlerinden birine karar verilir ve bununla bazı bireyler seçilir.

5. Adım: Çaprazlama işlemi yapılır.

- Seçilen bu bireyler çaprazlama işlemine tabi tutulur. Amaç, bir önceki iterasyondan daha kaliteli bireyler elde etmektir.

6. Adım: Mutasyon işlemi yapılır.

- Yeni oluşan bireyin genlerinde değişiklik yapmak amacıyla mutasyon işlemine tabi tutulur. Amaç, birey çeşitliliğini sağlamaktır.

7. Adım: Yeni bir iterasyona geçilir.

- Mutasyon işlemi sonrasında yapısı değişen bireylerle birlikte yeni bir popülasyon elde edilir ve iterasyon sayısı bir artırılır ve 2. adımdan devam edilir.

1.2.2. Parçacık sürü algoritması

Parçacık sürü algoritması (PSO) 1995 yılında J.Kennedy ve R.C.Eberthart tarafından doğrusal olmayan problemlerin çözümü için geliştirilmiştir. Her bir parçacık, sürü içerisindeki her bir bireyi temsil etmektedir. Doğada sürünün hareket

halinde olduğu kabul edilmekte olduğundan, problemde popülasyondaki parçacıklar başlangıçta rastgele dağılmaktadır.

S konum vektörünü göstermek üzere her bir parçacığın ilk konumları, s=$\{x_{i,1},\ x_{i,2}, x_{i,3}\ ,.,\ x_{i,n}\}$ şeklinde gösterilir. i, i.parçacığı temsil etmektedir ve n problemin kaç boyutlu yani kaç değişkenli olduğunu ifade etmektedir. Parçacıkların kaydedilen bir önceki en iyi pozisyonları pbest ile ifade edilir. Parçacığın pbest değerleri p_i=$\{p_{i,1},\ p_{i,2}, p_{i,3}, \dots, p_{i,n}\}$ şeklinde gösterilir. Her iterasyondaki pbest değeri bir önceki iterasyondaki pbest değeriyle karşılaştırılır ve iyi olan pbest değeri yeni pbest değeri olarak güncellenir. Her bir iterasyondaki bütün parçacıkların pbest değerleri arasındaki en iyi pbest değeri, gbest değeri denilen popülasyonun en iyi değeri olarak alınmaktadır.

Parçacıkların bütün iterasyonlar boyunca hareket halinde olduğu varsayılmaktadır. Bu durumda hız vektörü v=$\{v_{i,1},\ v_{i,2}, v_{i,3}, \dots, v_{i,n}\}$ şeklinde tanımlanmaktadır. Hız vektörü her parçacığın sahip olduğu pbest değeri ve popülasyonun sahip olduğu gbest değeri dikkate alınarak her iterasyonda yeniden güncellenir. Hız vektörü güncellenen parçacığın önceki konumu da dikkate alınarak yeni bir konumu oluşturulur.

Hız formülü:

$$v_{i,d}^{t+1} = w.v_{i,d}^{t+1} + c_1.rand_1.\left(p_{i,d} - x_{i,d}\right) + c_2.rand_2.\left(p_{g,d} - x_{g,d}\right) \quad (1.5)$$

Bilişsel bileşen formül:

$$v_{i,d}^{t+1} = v_{i,d}^{t+1} + c_1.rand_1.\left(p_{i,d} - x_{i,d}\right) \quad (1.6)$$

Sosyal bileşen formül:

$$v_{i,d}^{t+1} = v_{i,d}^{t+1} + c_2.rand_2.\left(p_{g,d} - x_{g,d}\right) \quad (1.7)$$

Tam formül, tam parçacık sürü optimizasyon modelini; bilişsel formül, bilişsel sürü optimizasyonu modelini; sosyal bileşen formül, sosyal parçacık sürü optimizasyon modelini oluşturmaktadır.

PSO, bilişsel sosyal ve keşifsel davranış biçimleri arasındaki bir denge olduğu varsayımına dayanmaktadır. Bilişsel davranış biçimi her bireyin kendi hafızasında kalmış olan iyi konuma gitme eğilimi olarak tanımlanırken sosyal davranış biçimi her bireyin iyi konumlarda bulunan diğer bireyleri takip etme eğilimi olarak tanımlanmaktadır. Keşifsel davranış biçimi ise her bireyin rastgele olarak arama yapma eğilimi olarak tanımlanmaktadır (Shi ve Eberhart, 1998).

Parçacık sürü optimizasyonu için aşağıdaki değişkenler kullanılmaktadır.

c_1, c_2: pozitif sabitler
w: eylemsizlik ağırlığı
$v_{i,d}$: parçacığın o anki hızı
$rand_1, rand_2$: rastgele pozitif sayılar
$p_{i,d}$: parçacığın sahip olduğu en iyi pozisyonu
$p_{g,d}$: popülasyonun sahip olduğu en iyi pozisyonu

1.2.2.1. Parçacık sürü optimizasyonu temel kavramlar

PSO'da kullanılan temel kavramlar; hız engelleme, eylemsizlik ağırlığı, sınırlama katsayısı ve k sabitidir.

Hız engelleme, maksimum hızı dengede tutmaya çalışmaktadır. PSO'da iterasyonlar boyunca hızın çok arttığı durumlar bir sonraki pozisyonda da büyük değişimlere yol açmaktadır. Bunun önüne geçmek için hız sınırlandırmasına başvurulmuştur. Bunun için önceden bir maksimum hız kavramı belirtilir, parçacığın hızı bu maksimum hızın üstüne çıktığında yeniden maksimum hıza çekilir. Büyük seçilmesi durumunda küresel araştırma, küçük seçilmesi durumunda da yerel araştırma için iyi sonuçlar elde edilir.

$$v_{i,d}^{t+1} = \begin{cases} v_{i,d}^{t+1} & , v_{i,d}^{t+1} < V_{max,j} \\ V_{max,d} & , v_{i,d}^{t+1} \geq V_{max,j} \end{cases} \quad (1.8)$$

Eylemsizlik ağırlığı, parçacığın bir sonraki konumunu oluştururken bir önceki hızın konuma katkısını ayarlamaktır. Büyük eylemsizlik ağırlığı, araştırma bölgesinde tümel araştırma yapılmasını sağlarken, küçük eylemsizlik ağırlığı yerel araştırma yapılmasını sağlamaktadır (Shi ve Eberhart, 1998).

Sınırlama katsayısı, hız engelleme ve eylemsizlik ağırlığında olduğu gibi hızdaki ani artışları kontrol etmektir. Sınırlama katsayısı hesaplamasında kullanılan basit yöntemlerden biri aşağıdaki formüllerde gösterilmiştir (Clerc ve Kennedy, 2002).

$$v_{i,d}^{t+1} = w.v_{i,d}^{t+1} + c_1.U(0,\phi 1).\left(p_{i,d} - x_{i,d}\right) + c_2.U(0,\phi 2).(p_{g,d} - x_{i,d}) \quad (1.9)$$

$$x_{i,d} = x_{i,d} + v_{i,d} \quad (1.10)$$

$$\phi = \phi_1 + \phi_2 > 4 \quad (1.11)$$

$$x = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4.\phi}} \quad (1.12)$$

K sabiti, çok karmaşık optimizasyon problemlerinde eylemsizlik ağırlığı doğru sonucu vermede yeterli olmadığında hızı kontrol etmek için kullanılır. K sabiti, sapma sayısını azaltarak doğru sonuca ulaşılmasına olanak sağlamaktadır. Bu durumda K sabitinin de kullanıldığı denklem aşağıdaki gibi gösterilir (Shi ve Eberhart, 2000).

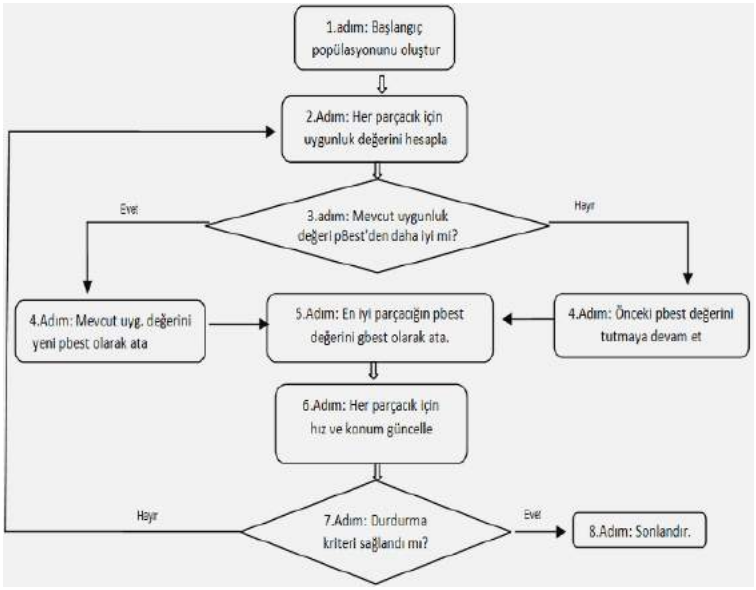
$$v_{i,d}^{t+1} = k.\left[w.v_{i,d}^{t+1} + c_1.rand_1.\left(p_{i,d} - x_{i,d}\right) + c_2.rand_2.(p_{g,d} - x_{i,d})\right] \quad (1.13)$$

1.2.2.2. Parçacık sürü algoritması parametre kontrolü

- Parçacık sayısı, problemin zorluğuna, boyutuna uygun olacak şekilde belirlenir.
- Parçacık boyutu, amaç fonksiyonundaki değişken sayısına göre belirlenir.
- Parçacık aralığı, amaç fonksiyonundaki değişkenlerin sınırlarını belirler.
- Vmax, ani pozisyon değişimlerin önüne geçmek için kullanılır.
- Öğrenme faktörleri, formülde c_1 ve c_2 olarak kullanılan değişkenleri temsil etmektedir. c_1 ve c_2 [0,4] aralığında değer alırlar.
- Durma koşulu, iterasyonlar sürdükçe programın bir yerde durması için kullanılır. Bunun için maksimum iterasyon sayısı belirlenebilir ya da kabul edilecek bir değer belirlenebilir ya da probleme uygun başka bir şart koşulabilir.
- Rand, her iterasyonda ve iterasyon içerisindeki her işlem için rastgele belirlenir. Genelde (0,1) arasında seçilir.

1.2.2.3. Parçacık sürü algoritması temel çalışma mantığı

Şekil 1.12.'de akış diyagramı verilmiş olan parçacık sürü algoritmasının temel çalışma mantığı aşağıda adımlar halinde değerlendirilmektedir.



Şekil 1.12. Parçacık sürü algoritması akış diyagramı

1. Adım: Başlangıç popülasyonu oluşturulur.

- Probleme uygun bir uygunluk fonksiyonu oluşturulur.
-Parçacık sayısı, sonlandırma kriteri, kısıtlama kriterleri kullanıcı tarafından belirlenir.
- Parçacık boyutu, probleme göre belirlenir.
- Her bir parçacığın rastgele hız ve konum vektörleri oluşturulur.

2. Adım: Her parçacık için uygunluk değeri hesaplanır.
- D boyut sayısı olmak üzere her bir parçacık, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ şeklinde gösterilir. Ve n, popülasyon sayısı olmak üzere popülasyon, $p= (x_1, x_2, \dots, x_n)$ şeklinde gösterilir.
- Bu durumda uygunluk fonksiyonu f(x) olmak üzere her parçacık için uygunluk değeri $f(x_n)$ şeklinde bulunabilir.

3. Adım: Mevcut uygunluk değeri pbest ile karşılaştırılır.
- Her parçacığın o ana kadarki iterasyonlar boyunca hep tutulan bir en iyi değeri bulunmaktadır, buna pbest değeri denilir. (Her bir parçacığın ilk pbest değerleri ilk iterasyondaki uygunluk değerleriyle aynı alınır.)
- İterasyonlar boyunca her parçacık için bulunan uygunluk değeri onun pbest değeri ile karşılaştırılır.

4. Adım: Karşılaştırmanın sonucunda uygunluk değeri;
- Eğer daha büyükse mevcut uygunluk değeri yeni pbest olarak atanır.
- Eğer daha büyük değilse önceki pbest yeni pbest değeri olarak tutulur.

5. Adım: En iyi parçacığın pbest değeri gbest olarak atanır.
- İterasyonda bütün parçacıkların pbest değerleri arasındaki en iyi pbest değeri iterasyonun gbest değeri olarak alınır.

6. Adım: Her parçacık için hız ve konum güncellenir.

$$V_i^{t+1} = V_i^t + c_1 \text{rand}_1^t (\text{pbest}_i^t - x_i^t) + c_2 \text{rand}_2^t (\text{gbest}^t - x_i^t) \quad (1.14)$$

$$X_i^{t+1} = x_i^t + V_i^{t+1} \quad (1.15)$$

Yukarıdaki formüller kullanılarak her bir parçacığın hız ve konumları güncellenir.

7. Adım: Sonlandırma kriteri sağlandı mı?
- İlk adımda kullanıcının oluşturduğu sonlandırma kriteri, çoğunlukla belirli bir iterasyon sayısına bağlı olmaktadır.
- Eğer kriter sağlandıysa uygulama sonlandırılır.
- Eğer sağlanmadıysa, her bir parçacığın güncellenmiş yeni hız ve konum değerleri kullanılarak 2. Adım'dan devam edilir.

1.2.3. Diferansiyel gelişim algoritması

Diferansiyel gelişim algoritması (DGA) algoritması, sürekli alanlar üzerinde optimizasyon problemleri için Rainer Storn ve Kenneth Price tarafından geliştirilen evrimsel programlamanın bir dalıdır. Price ve Storn'a göre diferansiyel gelişim algoritması ısıl işlem algoritması (simulated annealing) ve genetik algoritmaya göre daha etkilidir (Price ve Storn, 1997).

DGA'da her değişkenin değeri gerçek sayı ile temsil edilir. DGA, gerçek değerli değişkenlerle problemleri çözen en iyi genetik algoritmalardan biridir.

DGA'nın diğer algoritmalara göre avantajları basit yapısı, kullanım kolaylığı, hızlı olması ve sağlamlılığıdır. DGA, pratik uygulamalarda hızlı erişilebilen, faydalı bir tasarım aracıdır. DGA, uzman bilgisi veya karmaşık tasarım algoritmalarına başvurmadan çözülmesi zor olan sorunlara etkili çözümler bulmak için çeşitli bilim ve mühendislik uygulamalarında kullanılmaktadır (Arunachalam, 2008).

1.2.3.1. Diferansiyel gelişim algoritması kontrol parametreleri

F, mutasyon kontrol parametresi sabit ve gerçek bir değerdir.
CR, çaprazlama kontrol parametresidir.

NP, popülasyon büyüklüğüdür.
G, iterasyon sayısıdır.

Etkili bir çözüm elde etmede bu parametrelerin rolü büyüktür. Bu parametrelerin farklı değerlerinde farklı çözümler oluşur. En iyi çözümü elde edebilmek için parametrelerin en uygun değerleri bulunmalıdır.

1.2.3.2. Diferansiyel gelişim algoritması kullanılan operatörler

Diferansiyel gelişim algoritmasında genetik algoritmaya benzer operatörler kullanılmaktadır. Bu operatörlerin temel amacı aday çözümler popülasyonunu geliştirmektir. Bu operatörler kısıtlama fonksiyonu, mutasyon ve çaprazlama gibi işlemlerdir.

Kısıtlar fonksiyonu diğer optimizasyon algoritmalarından farklı olarak amaç fonksiyonundan ayrı değil, amaç fonksiyonun içerisinde yer alır.

$$x_{j,i,g=0} = x_j^l + rand_j[0,1].(x_j^u - x_j^l) \quad (1.16)$$

Denklemde $x_{j,i,g}$, g neslindeki i kromozomunun j parametresini göstermektedir. (x_j^u, x_j^l) değişkenlere ait sırasıyla üst ve alt değerleri göstermektedir.

Mutasyon işleminde hedef kromozom dışında üç farklı (r_1, r_2, r_3) kromozom seçilir. İlk İki kromozomun (r_1, r_2) farkı alınır ve (0,2) arasında değer alan F mutasyon kontrol parametresiyle çarpılır. Bu çarpım değeri seçilen üçüncü kromozoma olan, r_3 'e eklenir.

$$n_{j,i,g+1} = x_{j,r_3,g} + F.(x_{j,r_1,g} - x_{j,r_2,g}) \quad (1.17)$$

Çaprazlama yapılırken, mutasyon sonucu elde edilen fark kromozomu ve x_i,G kromozomu kullanılarak yeni iterasyona aday, deneme kromozomu (u_i,G+1) üretilir.

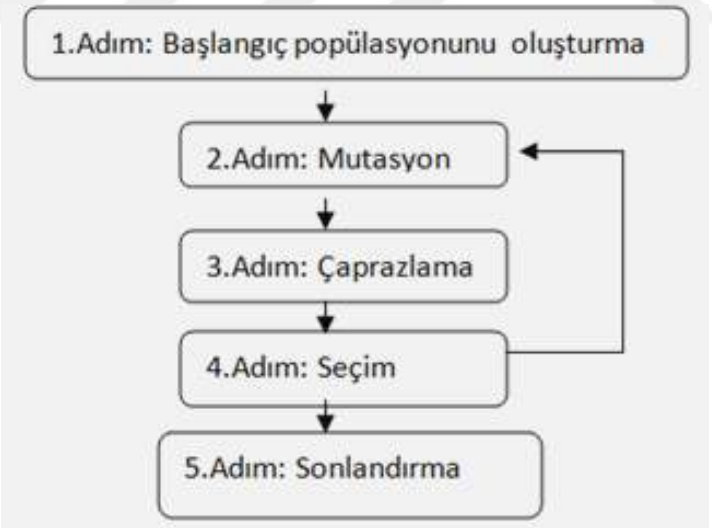
Deneme kromozomuna ait her bir gen CR olasılıkla fark kromozomundan 1-CR olasılıkla mevcut kromozomdan seçilir (Mohamed ve ark., 2012).

0 ile 1 arasında üretilen rastgele sayı CR'den küçükse gen, $n_{ji,G+1}$'den aksi takdirde mevcut kromozomdan seçilir. Amaç belirlenen oranda genin yeni fark kromozomundan alınmasıdır. Buradaki j =jrand koşulu, en az bir tane genin üretilen yeni kromozomdan alınmasını garanti etmek amacıyla konulmuştur. Rastgele seçilen jrand noktasındaki gen CR'ye bakılmaksızın $n_{ji,G+1}$'den seçilir (Keskintürk, 2006).

$$u_{j,i,g+1} = \begin{cases} n_{ji,G+1} & \text{if(randb(j)} \leq \text{CR) or j} = \text{jrand} \\ x_{ji,G} & \text{if(randb(j)} > CR\text{)and j} \neq \text{jrand} \end{cases} \quad (1.18)$$

1.2.3.3. Diferansiyel gelişim algoritması temel çalışma mantığı

Şekil 1.13.'de akış diyagramı verilmiş olan genetik algoritmanın temel çalışma mantığı aşağıda adımlar halinde değerlendirilmektedir.



Şekil 1.13. Diferansiyel gelişim algoritması akış diyagramı

1. Adım: Başlangıç popülasyonu oluşturulur.

- Problemle ilgili hiçbir bilgi mevcut değilse ilk popülasyon rastgele seçilir. Mevcut ön çözüm durumunda, başlangıç popülasyonu genellikle ön çözüme normal dağılmış rastgele sapmalar eklenerek üretilir (Arunachalam, 2008).

- Optimize edilecek problem için amaç fonksiyonu oluşturulur.
- Problemin kısıtlarını sağlayan kısıtlar fonksiyonu oluşturulur.
- F, CR, NP, G parametreleri kullanıcı tarafından belirlenir.

2. Adım: Mutasyon işlemleri yapılır.
- Bireylerin genlerinde değişiklik yapmak amacıyla mutasyon işlemi yapılır. Amaç birey çeşitliliğini artırmaktır.

3. Adım: Çaprazlama işlemleri yapılır.
- Çaprazlama işlemi mutasyona uğramış bireylerin bir sonraki iterasyona geçerken genlerinde değişiklik yapmaktadır.

4. Adım: Seçim işlemi yapılır.
- Çaprazlamadan elde edilen vektör ile en başta seçilen mevcut hedef vektör arasında bir değerlendirme yapılır. Uygunluğu daha yüksek olan vektör bir sonraki iterasyona aktarılır.

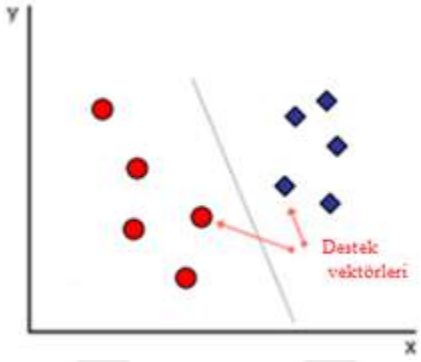
5. Adım: Sonlandırma işlemi yapılır.
- Diğer optimizasyon algoritmalarında olduğu gibi sonlandırma kriteri kullanıcı tarafından belirlenir. Belirlenecek bir iterasyonun sonunda durması istenilebileceği gibi, hedef bir sonuç değerine ulaşıncaya kadar da sürdürülebilir.

1.3. Sınıflandırma Algoritmaları

1.3.1. Destek vektör makineleri

1990'larda geliştirilen destek vektör makineleri (DVM) sınıflandırma ve regresyon amaçlı kullanılan önemli makine öğrenme algoritmalarından biridir. DVM algoritmasının sınıflandırma amaçlı kullanımında, verilen bir veri setini uygun bir şekilde iki sınıfa ayıracak hiperdüzlem tanımlanması gerekmektedir (Vapnik, 1995).

Şekil 1.14.'de gösterilen destek vektörleri, hiperdüzleme en yakın veri noktalarıdır, bu nedenle çıkarılmaları durumunda hiperdüzlemin konumunu değiştireceklerdir. Dolayısıyla destek vektörleri bir veri kümesinin kritik unsurları olarak görülebilirler.



Şekil 1.14. Destek vektörlerinin gösterimi

İki sınıflı doğrusal verilerin sınıflandırılması için tasarlanmış olan DVM algoritması daha sonra çok sınıflı ve doğrusal olmayan verilerin sınıflandırılması için geliştirilmiştir (Vapnik, 1995).

1.3.1.1. Doğrusal ayrılabilen verilerde sınıflandırma

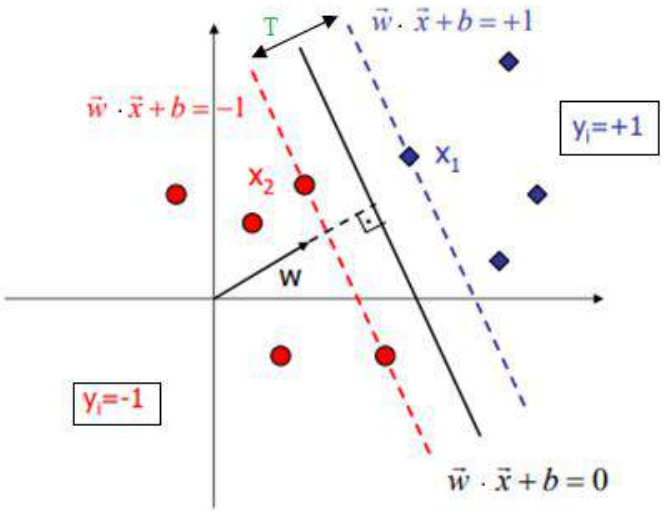
Doğrusal olarak ayrılabilen verilerde destek vektör makineleri ile sınıflandırma yapılacağı zaman iki sınıflı örneklerden oluşan bir veri setinin eğitim verisi ile elde edilen karar fonksiyonu kullanılarak verilerin birbirinden ayrılması amaçlanır. Karar fonksiyonu eğitim verisini en uygun şekilde ayırabilecek en büyük sınıra sahip hiperdüzlemi bulmaya çalışmaktadır (Osuna, 1997).

Eğitim verisinin k tane örnekten oluştuğu varsayılırsa i=1,2,3,..,k olmak üzere optimum hiperdüzleme ait eşitsizlik 1.19'daki denklemde olduğu gibi gösterilmektedir.

$$\begin{cases} w.x_i + b \geq +1 & \text{y} = +1 \\ w.x_i + b \leq +1 & \text{y} = -1 \end{cases} \tag{1.19}$$

x ∈ R^N olup N-boyutlu bir uzayı, y ∈ {-1, +1} ise sınıf etiketlerini, w ağırlık vektörünü ve b eğilim değerini göstermektedir (Kavzaoğlu, 2010).

Şekil 1.15.'de gösterildiği gibi optimum hiperdüzlemin T sınırının maksimuma çıkarılması için w ifadesinin minimum olması amaçlanmaktadır. Bu durumda en uygun hiperdüzlemin belirlenmesi için $\left[\frac{\|w\|^2}{2}\right]$ değerinin minimum olması gerekmektedir.



Şekil 1.15. Doğrusal olarak ayrılabilen verilerde hiperdüzlemin belirlenmesi

DVM'de optimal ayırma hiperdüzleminin bulunması için Lagrange çarpanlarından yararlanılır. 1.20'deki denklemde formülü verilen Lagrange çarpanları en küçükleme tipindeki problemi ikincil probleme dönüştürerek problemin çözülmesini daha kolay hale getirmektedir (Gunn, 1998).

$$L(w, b, \alpha) = \left[\frac{\|w\|^2}{2}\right] - \sum_{i=1}^{k} \alpha_i . y_i \, (w . x_i + b) + \sum_{i=1}^{k} \alpha_i \qquad (1.20)$$

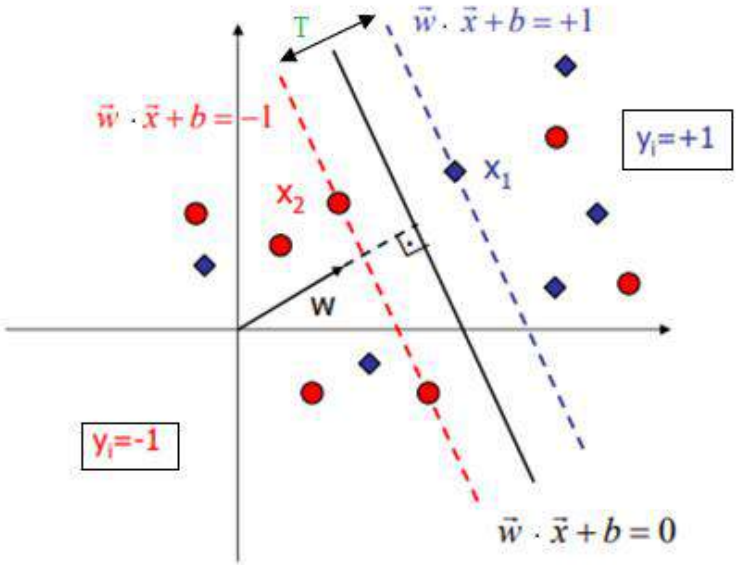
Lagrange çarpanı kullanılarak doğrusal olarak ayrılabilen iki sınıflı bir veri setinin uygun bir şekilde birbirinden ayrılmasını amaçlayan karar fonksiyonu 1.21'deki denklemde gösterilmektedir (Osuna, 1997).

$$f(x) = \text{sign}\left(\sum_{i=1}^{k} \lambda_i \, y_i (x . x_i) + b\right) \quad (1.21)$$

1.3.1.2. Doğrusal ayrılamayan verilerde sınıflandırma

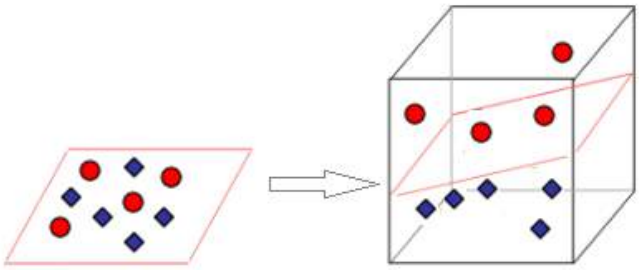
Veri setlerinin hiperdüzlemle doğrusal olarak ayrılamaması durumunda kullanılan algoritmalardır. Bu durumda verileri ayırmak için doğrusal olmayan bir hiperdüzlemin kullanılması gerekmektedir.

Şekil 1.16.'da görüldüğü gibi doğrusal olarak ayrılmayan verilerde T sınırını maksimum yapacak ve sınıflandırma hata oranını minimum yapacak bir düzlem belirlenmelidir.



Şekil 1.16. Doğrusal olarak ayrılamayan verilerde hiperdüzlemin belirlenmesi

Destek vektör makineleri Şekil 1.17.'de olduğu gibi doğrusal olmayan veriler için matematiksel olarak 1.22'deki denklemde ifade edilen bir kernel fonksiyonu kullanarak verilerin yüksek boyutta doğrusal olarak ayrılabilmesini sağlamaktadır.



Şekil 1.17. Kernel fonksiyonu ile verilerin daha yüksek bir boyuta dönüştürülmesi

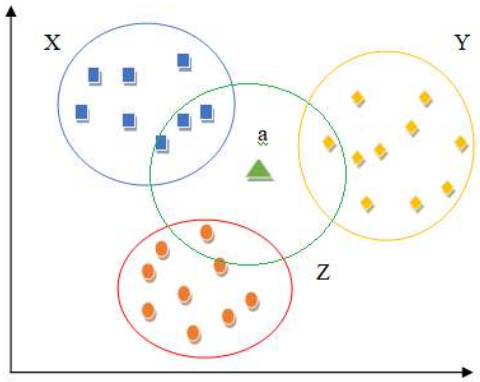
Doğrusal olarak ayrılamayan iki sınıflı bir problemin çözümü için Kernel fonksiyonu kullanılarak oluşturulan karar fonksiyonu 1.23'deki denklemde gösterilmektedir (Osuna, 1997).

$$K(x_i, x_j) = \varphi(x).\varphi(x_j) \tag{1.22}$$

$$f(x) = \text{sign}(\sum_i \alpha_i y_i \varphi(x).\varphi(x_i) + b) \tag{1.23}$$

1.3.2. En yakın k komşu algoritması

En yakın k komşu algoritması (KNN- K Nearest Neighbor's) sınıflandırma işlemlerinde kullanılan gözetimli öğrenme metotlarından biridir. KNN algoritması nesnelerin birbirine olan mesafelerine göre birbirine yakın olan nesneleri aynı sınıfa almaktadır. Daha sonra sınıfı belli olmayan bir test verisi geldiğinde test verisinin önceden sınıflandırılmış diğer nesnelere olan uzaklıkları hesaplanarak en yakınındaki k tane nesne çoğunlukla hangi sınıfa ait ise test verisi de o sınıfa dahil edilmektedir (Larose, 2005). Bu durumda Şekil 1.18.'de gösterildiği gibi a nesnesi X sınıfına dahil edilmektedir.



Şekil 1.18. k en yakın komşuluğa göre sınıflandırma

1.3.2.1.En yakın k komşu algoritmasının parametreleri

Uzaklık ölçütü olarak nesnelerin birbirlerine olan mesafelerini bulmada Öklid, Manhattan ve Minkowski uzaklık formüllerinden yararlanılmaktadır. Veri sayısı n olmak üzere p=$(x_1, x_2, x_3, \ldots, x_n)$ ve q=$(y_1, y_2, y_3, \ldots, y_n)$ noktaları arasındaki uzaklık değerleri aşağıdaki denklemlerde gösterilmektedir.

Öklid uzaklığı= $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ (1.24)

Manhattan uzaklığı = $\sqrt{\sum_{i=1}^{n}|x_i - y_i|}$ (1.25)

Minkowski uzaklığı = $(\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}}$ (1.26)

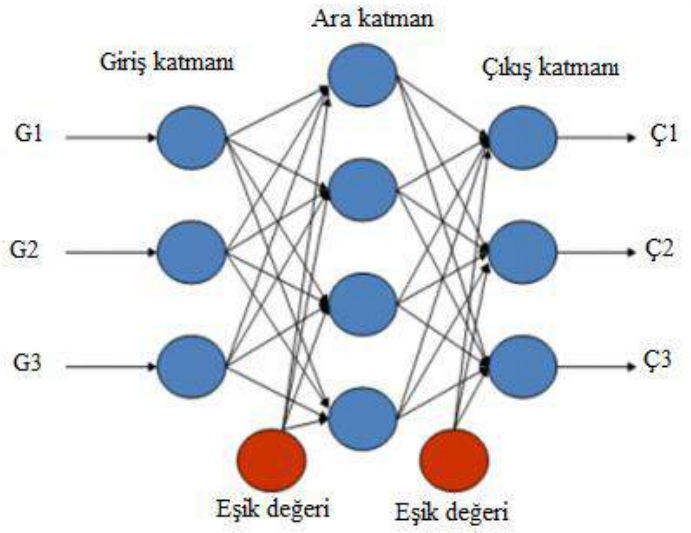
Komşu sayısı k en küçük değeri olan 1 değerini alırsa sınıfı belli olmayan bir a nesnesi geldiğinde sadece kendisine en yakın olan nesneye göre sınıflandırılmaktadır. En yakın olduğu nesne hangi sınıfa ait ise a nesnesi o sınıfa dahil edilmektedir. Veri sayısı n olmak üzere k en büyük değeri olan n değerini alırsa sınıfı belli olmayan bir a nesnesinin sınıflandırılması gerektiğinde a nesnesinin bütün nesnelere olan uzaklıkları alınarak a nesnesi oylamaya göre bir sınıfa dahil edilmektedir. Bu nedenle doğru k değerinin seçilmesi önemlidir (Mitchell, 1997).

Ağırlıklandırma, test verisine daha yakın olan verilerin seçilme şanslarının artırılması amacıyla kullanılmaktadır. Bu nedenle test verisine daha yakın olan komşu verilere, test verisine olan mesafelerine göre bir ağırlık değeri verilerek seçilme şansları artırılmaya çalışılmaktadır.

1.3.3. Çok katmanlı algılayıcılar

Rumelhart tarafından 1986'da geliştirilen çok katmanlı algılayıcılar (ÇKA) modeli çoğunlukla sınıflandırma amaçlı kullanılmaktadır. Modelin sınıflandırma amaçlı kullanımında öncelikle ağın eğitilmesi gerekmektedir.

Şekil 1.19.'da gösterildiği gibi ÇKA'da ağın yapısı giriş katmanı, ara katman ve çıktı katmanı olmak üzere üç katmandan oluşmaktadır. Veri setinin eğitim seti olarak kullanılacağı verilerinin her biri giriş katmanına gönderilip daha sonra ara katmanda bir takım işlemlerden geçerek çıkış katmanına gönderilir. Böylece her bir verinin girdi değeriyle birlikte bir çıktı değeri de oluşmaktadır (Haykin,1984).



Şekil 1.19. Çok katmanlı algılayıcılar için ağ modeli

Çok katmanlı algılayıcılar delta öğrenme kuralını kullanmaktadırlar. Delta modeline göre ağın eğitilmesi, ağın çıktı değerlerinin hesaplandığı (ileri doğru

hesaplama) ve hata değerini minimuma çekmek için ağırlıkların güncellendiği (geriye doğru hesaplama) iki aşamada gerçekleştirilmektedir.

1.3.3.1. İleri doğru hesaplama

İleri doğru hesaplama aşamasında eğitim setindeki her bir nesne giriş katmanından ağa gönderilir. Giriş katmanına gelen girdiler hiç bir değişiklik olmadan ara katmana gönderilir. Bu durumda giriş katmanındaki k. nesnenin çıktısı $Ç_k^i = G_k$ şeklinde hesaplanır (Mia ve arkd.,2015).

Ara katmandaki her bir nesne giriş katmanındaki bütün nesnelerden gelen bilgileri bağlantılı ağırlıklarının etkisi ile alır. Ara katmandaki nesnelere gelen net girdi (NET_j^a) değerlerinin hesaplandığı 1.27'deki denklemde A_{kj} , k. girdi nesnesinin j. ara katman nesnesine bağlandığı ağırlık değerini göstermektedir.

$$\mathrm{NET_j^a} = \sum_{k=1}^{n} \mathrm{A_{kj}} \,.\, Ç_k^i \tag{1.27}$$

Ara katmandaki nesnelerin NET girdi değerleri ile aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanıldığında j. ara katman nesnesinin çıktı değeri aşağıdaki denklemde olduğu gibi bulunmaktadır.

$$Ç_j^a = \frac{1}{1+e^{-\left(\mathrm{NET_j^a}+\beta_j^a\right)}} \tag{1.28}$$

Denklemdeki β_j^a olarak verilen değer, j. nesneye bağlanan eşik değer nesnesinin ağırlığını gösterir. Bu değer 1'e eşittir ve her zaman sabittir.

1.3.3.2. Geriye doğru hesaplama

Geriye doğru hesaplama aşamasında ağa gönderilen giriş değerleri (B_i) ile ağın ürettiği çıkış değerleri ($Ç_i$) karşılaştırılır. Giriş ve çıkış değerleri arasındaki fark (E_i)

hata değerini vermektedir. Bu durumda i.nesne için oluşan hata değeri $E_i = B_i - Ç_i$ şeklinde hesaplanmaktadır.

Bütün nesneler için ağda meydana toplam hata (TH) aşağıdaki denklemde olduğu gibi bütün nesnelerin hatalarının kullanılmasıyla hesaplanmaktadır.

$$\mathrm{TH} = \frac{1}{2}\left(\sum_{\mathrm{i}} \mathrm{i}^2\right) \tag{1.29}$$

ÇKA hatanın minimum olmasını amaçlanmaktadır. Bu nedenle tespit edilen hata değeri ağın ağırlık değerlerine dağıtılmaktadır. Hatanın geriye yayılması olarak bilinen hatayı dağıtma işlemi, hatanın bir sonraki iterasyon için azaltılmasını sağlamaktadır (Öztemel, 2012).

2. ÖNCEKİ ÇALIŞMALAR

Crawford ve Wainwright (1995), aykırı değerleri tespit etmek için alt küme noktalarının seçimini yönetmede genetik algoritmalar üzerinde çalışmışlardır. Elde ettikleri bulgular, çoklu durumlarda aykırı değerleri tespit etmede genetik algoritmaların başarılı olduğunu göstermiştir.

Rosseeuw ve Leroy (1987), dağılım tabanlı bir yaklaşım kullanmışlardır. Bu yaklaşımda verilen veriden istatistiksel yöntemler geliştirilmiş ve nesnenin belirli bir modele ait olup olmadığını bulmak için istatistiksel test uygulanmıştır. Olasılığı düşük nesne, istatistiksel modelde aykırı değer olarak tanımlanmıştır.

Breuni (2000), yoğunluğa dayalı bir yaklaşım kullanmıştır. Yerel komşuluklarına olan mesafeye bağlı olarak herhangi bir veri noktasına yönelik bir aykırılık değeri olarak LOF değeri (yerel aykırılık faktörü) atanmıştır.

Knorr (2000) ve Angiulli (2005), uzaklığa dayalı bir yaklaşım kullanmışlardır. Bu yaklaşımda sapmalar, özellik uzayında bir mesafe ölçüsü ile saptanmıştır.

Alma (2009), genetik algoritma tabanlı çoklu regresyon modellerinde, bilgi kriterlerini kullanarak aykırı değerlerin belirlenmesi üzerine çalışma yapmıştır.

Doğan (2009), aykırı durum tespiti, kümeleme ve sınıflandırma gibi veri madenciliği paradigmalarını kullanarak ağ güvenliği alanına yeni bir yaklaşım getirmiştir. Kümeleme ve sınıflandırma için en yakın k komşuluk (KNN) algoritmasını kullanmıştır. Ağdaki kullanıcılar, k en yakın komşu ve eşik değer parametresinin hesaplanmasıyla kümelendirilmiştir. Bu kümeler üzerinde yapılan işlemler sonucunda yeni isteklerin bir aykırı durum mu yoksa normal mi oldukları ayırt edilmiştir.

Mohemmed ve ark. (2010), aykırı değer tespitinde parçacık sürü optimizasyonu algoritmasını mesafe ölçümlerini optimize etmek için kullanmışlardır. PSO yönteminin yerel aykırılık faktöründen (LOF) daha etkili olduklarını tespit etmişlerdir.

Gazen ve ark. (2010), projeksiyon takibi (projection pursuit) bağlamında çok değişkenli aykırı değerlerin saptanması için tribes adlı yeni bir sezgisel optimizasyon algoritmasının kullanılmasını önermişlerdir.

Raja ve Bhaskaran (2012), aykırı değerleri tespit etmede genetik algoritmalar yaklaşımı kullanmışlardır. Bu yaklaşımda aykırı değerler üretilen uygunluk değerine göre belirlenmiş olup, uygunluk değeri daha düşük olan nesneler aykırı değer olarak belirlenmiştir. Deneysel olarak, önerilen algoritmanın belirli bir zaman diliminde aykırı değerlerin hesaplanmasında iyi olduğu gösterilmiştir.

Erdoğan (2012), etkin-aykırı, yerel aykırılık etkeni, tek sınıflı karar vektör makineleri ve parzen pencereleri yöntemlerini aykırı değer analizinde kullanmıştır. Bu yöntemlerde temel bileşenler çözümlemesi (TBÇ), laplasyen öz haritalar (LÖH) ve çok boyutlu ölçekleme teknikleriyle birleştirilerek farklı veri kümeleri üzerinde aykırı değer tespit çalışması yapılmıştır. Deneylerin sonucunda, LÖH yönteminin diğer yöntemlere göre daha başarılı performans sergilediği gösterilmiştir.

Wong ve ark. (2012), güvenlik yönetimi üzerine yaptıkları çalışmada, yapay bağışıklık sistemini (YBS) kredi kartı dolandırıcılık tespiti için modellemişlerdir. Önerdikleri bu sistem güvenlik yönetimi uygulamalarına yeni bir yaklaşım getirmiştir.

3. MATERYAL ve YÖNTEM

3.1. Materyal

Uygulamada, Çizelge 3.1.'de özellikleri açıklanan, "Uc Irvine Machine Learning Repository" sitesinin veri tabanından alınan 5 farklı veri seti kullanılmaktadır.

Çizelge 3.1. Uygulamada kullanılan veri setleri

Veri Seti	Veri setinin dağılım özelliği	Veri sayısı	Değişken sayısı	Değişkenlerin özelliği	Genel kullanım alanları
Blood Transfusion Service Center	Çok değişkenli	748	5	Gerçel Sayı	Sınıflandırma
Bupa (Liver Disorders)	Çok değişkenli	345	7	Gerçel Sayı, Tam sayı	N/A
Yeast	Çok değişkenli	1484	8	Gerçel Sayı	Sınıflandırma
Haberman	Çok değişkenli	306	3	Tam Sayı	Sınıflandırma
İris	Çok değişkenli	150	4	Gerçel Sayı	Sınıflandırma

Blood transfusion service center veri seti, bir kan bağışçısının başka bir tarihte kan verip vermediğini tahmin etmeye yönelik bir veri setidir. Veri setinin girdi parametreleri kan bağışçısının ilk ve son bağışından itibaren geçen zaman, toplam bağış sayısı ve toplam bağışçı değerlerini girdi olarak kabul etmektedir.

Liver disorder veri seti, aşırı alkol tüketiminden kaynaklanabilecek karaciğer rahatsızlıklarına duyarlı olduğu düşünülen bireylerin kan testi değerlerinden oluşmaktadır. Bu değerler ile birey hakkında karaciğer rahatsızlıklarıyla ilgili bir tespit yapılmaya çalışmaktadır.

Yeast (maya) veri seti, ökaryotik hücrelerde protein lokalizasyon alanlarını öngörmeye yönelik bir veri setidir. Birçok biyoteknolojik ürünün üretiminde kullanılan mayalar için proteinlerin lokalizasyon değerlerini tahmin etmek önemlidir.

Haberman veri seti, meme kanseri ameliyatı geçirmiş hastalara ait değerleri girdi parametresi olarak kullanarak hastaların ameliyattan sonra yaşama süresini tahmin etmeye çalışmaktadır.

İris veri seti, bitkinin taç yaprak ve çanak değerlerinin uzunluk ve genişliklerine göre hangi türe ait olduğunu öngören bir veri setidir.

Bu 5 farklı veri seti 3 farklı optimizasyon algoritması (diferansiyel gelişim algoritması, genetik algoritma ve parçacık sürü optimizasyonu algoritması) ile birlikte kullanılmaktadır.

Diferansiyel gelişim algoritması için; ölçekleme faktörünün alt sınırı için beta_min=0.2, ölçekleme faktörünün üst sınırı için beta_max=0.8, çaprazlama olasılığı için pcr=0.2 kullanılmaktadır.

Genetik algoritma için; çaprazlama yüzdesi için pc=0.8, mutasyon yüzdesi için pm=0.3, mutasyon oranı için mu=0.02 ve seçim yöntemi olarak rulet tekerleği kullanılmaktadır.

Parçacık sürü optimizasyonu algoritması için; eylemsizlik ağırlığı için w=1, eylemsizlik ağırlığı düşürme oranı için wdamp=0.99, kişisel öğrenme katsayısı için c1=2 ve global öğrenme katsayısı için c2=2 kullanılmaktadır.

Metasezgisel algoritmalarda yukarıda bahsedilen parametrelerin türünün seçimi ve değerinin seçimi, uygulamada istenilen çözüme ulaşmada çok önemli rollere sahip oldukları kadar, uygulamadaki iterasyon sayısı ve popülasyon sayısı da önemlidir. Bu yüzden iterasyon sayısı 100 popülasyon sayısı 20, iterasyon sayısı 100 popülasyon sayısı 50, iterasyon sayısı 200 popülasyon sayısı 20 ve iterasyon sayısı 200 popülasyon sayısı 50 olmak üzere her bir veri seti ve her bir algoritma için uygulama ayrı çalıştırıldı ve elde edilen değerler kaydedildi.

Uygulamanın kodu matlab programı 2016b sürümü kullanılarak geliştirildi. Geliştirilen bu kod windows 10 Professional 64 bit işletim sistemine sahip Intel Core i7-5500u,2.40 GHz işlemcili ve 8 GB RAM'li bir dizüstü bilgisayar üzerinde çalıştırılmıştır. Görsel grafikler için IBM SPSS programından yararlanıldı. Veri setlerinin sınıflandırma başarılarını değerlendirmek için de WEKA programından yararlanıldı.

3.2. Yöntem

Çoğu gerçek dünya veri setleri, genelde "aykırı değerli" olarak adlandırılan belirli bir miktar olağandışı değer içerir. Aykırı değerlerin izolasyonu (veriden temizlenmesi) hem orijinal verilerin kalitesini artırmak hem de veritabanlarındaki bilgi keşfi sürecinde belirsiz değerlerin etkisini azaltmak için önemlidir. Çoğu aykırı algılama yöntemleri, grafik olarak temsil edilen verilerin manuel olarak incelenmesine dayanmaktadır (Last ve Kandel, 2001). Bu tez çalışmasında, aykırı değerlerin saptanması sürecinin otomatikleştirilmesi için metasezgisel yöntemlere başvurulmaktadır.

Literatür araştırmalarında görüldüğü gibi çok sayıda aykırı değer tespiti uygulamasının temel mantığı veri setini önce kümelemek daha sonra kümelenmiş veri setinden farklı metotlar kullanılarak aykırı değerleri tespit etmektir. Burada yapılan çalışmada evrimsel kümeleme yöntemi kullanılmaktadır. Evrimsel kümeleme bir dizi kümeleme oluşturmak için iterasyonlar boyunca verilen verilerin işlenmesidir. Evrimsel kümeleme her bir iterasyon için bir kümeleme oluşturmaktadır. İterasyonlar boyunca oluşan her bir küme bir önceki iterasyona benzer olmalıdır ve bir önceki iterasyondan gelen verileri doğru bir şekilde yansıtmalıdır (Chakrabarti ve ark., 2006). Evrimsel kümeleme işlemleri için metasezgisel algoritmalar kullanılmaktadır. Daha sonra kümelenmiş verilerden aykırı değerleri tespit etmek için aykırı değer tespit yöntemlerinden istatistiksel yaklaşımlar kullanılmaktadır.

3.2.1. Uygulamada kullanılan parametreler

Uygulamada amaç metasezgisel algoritmalarla aykırı değerleri tespit etmek olduğu için yöntemde belirtildiği gibi öncelikle metasezgisel algoritmalarla veri en doğru şekilde kümelendirilmeye çalışılmaktadır. Bunun için küme sayısı ve benzerlik ölçütü iki önemli parametredir.

Kümeleme işleminde verilerin gruplandırılacağı optimum küme sayısı doğru belirlenmelidir. Literatürdeki çalışmalar da göz önünde bulundurularak küme sayısı için 3,5 ve 7 değerlerine göre sonuçlar ayrı elde edilmektedir.

Benzerlik ölçütü, verilerin kümelenmesi sırasında en doğru şekilde seçilmelidir. Aynı kümedeki örnekler birbirine daha çok benzemelidir, farklı kümedeki örnekler de birbirine daha az benzemelidirler. Bu yüzden kümeleme işlemi yapılırken verilerin benzerliğinin hangi ölçütle değerlendirileceği de kümeleme işlemleri açısından önemli bir konudur. Sıklıkla kullanılan başlıca benzerlik ölçüleri; korelasyon katsayıları, uzaklık ölçüleri, birliktelik kat sayıları ve olasılığa dayalı benzerlik katsayılarıdır (Aldenderfer ve Blashfield, 1984). Uygulamada nümerik değerler kullanıldığından benzerlik ölçütü için uzaklık ölçüleri baz alınmaktadır. Literatürde kullanılan uzaklık ölçülerinden Manhattan, Öklid, Minkowski ve Mahalanobis mesafeleri arasından en çok kullanılan öklid mesafesi seçilmektedir.

Uygulamada kullanılan metasezgisel algoritmalardan PSO, GA ve DGA için kullanılan parametreler de en doğru kümeleme işlemini gerçekleştirmede önemlidir. Kullanılan bu parametreler diferansiyel gelişim algoritması için ölçekleme faktörünün alt sınırı, ölçekleme faktörünün üst sınırı ve çaprazlama olasılığıdır. Genetik algoritma için; çaprazlama yüzdesi, mutasyon yüzdesi, mutasyon oranı ve seçim yöntemidir. Parçacık sürü optimizasyonu algoritması için de eylemsizlik ağırlığı, eylemsizlik ağırlığı düşürme oranı, kişisel öğrenme katsayısı ve global öğrenme katsayısı parametreleridir.

Uygulamada metasezgisel algoritmalarla en doğru şekilde evrimsel kümeleme işlemi yapıldıktan sonra aykırı değer tespit modelinin de doğru bir şekilde seçilmesi

önemlidir. Bu çalışmada, parametrik bir istatistiksel yöntem olan kutu grafiği (box-plot) yöntemi seçilmektedir.

3.2.2. Uygulamada kullanılan algoritma

1.Adım: Veri seti metasezgisel algoritmalarda işlenmek için hazır hale getirilir.
- Başlangıç için, veri seti alınır.
- Uygunluk fonksiyonu tanımlanır. Uygunluk fonksiyonu aşağıdaki adımları gerçekleştirir.

- K: küme sayısını, N: veri sayısını, P: boyut sayısını göstermektedir. Bu durumda NxP boyutlu Y veri matrisi ve KxP boyutlu kümelerin merkezini tutan X matrisi olmak üzere, verilerin küme merkezlerine olan uzaklıklarını bulmak için Öklid formülünden yararlanılmaktadır.

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{1P} \\ Y_{21} & Y_{22} & Y_{2P} \\ Y_{N1} & Y_{N2} & Y_{NP} \end{bmatrix} \qquad \begin{bmatrix} X_{11} & X_{12} & X_{1P} \\ X_{21} & X_{22} & X_{2P} \\ X_{K1} & X_{K2} & X_{KP} \end{bmatrix}$$

- $d(i,j)$, Y matrisindeki verilerin X matrisindeki verilere olan uzaklıklarını tutmaktadır. Bu uzaklıkları aşağıdaki Öklid uzaklık formülüyle hesaplamaktadır.

$$d(i,j) = \sqrt{\left|Y_{i1} - X_{j1}\right|^2 + \left|Y_{i2} - X_{j2}\right|^2 + \cdots + \left|Y_{ip} - X_{jp}\right|^2} \qquad (3.1)$$

- Hesaplanan Öklid mesafeleri d matrisinde tutulmaktadır.

$$\begin{bmatrix} d_{11} & d_{12} & d_{1K} \\ d_{21} & d_{22} & d_{2K} \\ d_{N1} & d_{N2} & d_{NK} \end{bmatrix}$$

- d matrisinde dNK, N. Verinin K. Kümeye olan uzaklığını göstermektedir.

- Öklid mesafesi birbirine en yakın olan veriler aynı kümeye alınır. Küme içerisindeki verilerin küme merkezine olan uzaklıkları dmin adında bir matriste tutulur.
- Küme içerisindeki verilerin küme merkezine olan toplam mesafeleri bulunur. Toplam mesafe, parçacığın maliyetini (BestCost değerini) belirler.

2. Adım: Kullanılacak metasezgisel algoritma seçilir.(PSO, GA, DGA)
- Algoritmanın parametreleri tanımlanır.
- Başlangıç popülasyonu oluşturulur.
- Popülasyondaki her bir parçacık küme merkezlerini temsil etmektedir.
- i. parçacık için K küme sayısı ve P boyut olmak üzere KxP boyutlu aşağıdaki gibi X matrisi oluşmaktadır.

$$\begin{bmatrix} X_{i,12} & X_{i,12} & X_{i,1P} \\ X_{i,21} & X_{i,22} & X_{i,2P} \\ X_{i,K1} & X_{i,K2} & X_{i,KP} \end{bmatrix}$$

3. Adım: Her bir parçacık için uygunluk fonksiyonu kullanılır ve uygunluk değeri bulunur.

4. Adım: Durdurma kriteri sağlanıncaya kadar;
- PSO için, pbest ve gbest değerleri güncellenir, hız ve konum güncellenir.
- DGA ve GA için, mutasyon çaprazlama ve seçim işlemleri yapılır.

5. Adım: İterasyon sonunda en iyi uygunluk değerine sahip parçacık, en iyi maliyeti yani en doğru kümelemeyi yaparak en düşük toplam mesafeyi bulmuş olur.

6. Adım: Metasezgisel algoritmalarla veri en doğru şekilde kümelendikten sonra veri noktalarının küme merkezlerine olan mesafeleri kullanılarak aykırı değerleri bulmaya yardımcı olacak olan kutu grafiği (boxplot) yöntemi kullanılır.

Algoritmanın kaba kodu için, p başlangıç popülasyonu ve n popülasyon sayısı olmak üzere, başlangıç popülasyonu p=$\{p_1, p_2, p_3, \dots, p_n\}$ şeklinde gösterilebilir. Uygunluk fonksiyonu da F olmak üzere parçacıkların uygunluk değerleri

F=$\{f_1, f_2, f_3, \dots, f_n\}$ şeklinde gösterilebilir. Popülasyonun elde ettiği en iyi değer gbest olmak üzere, gbest.dmin değerleri de veri noktalarının küme merkezine olan en yakın mesafelerini tutmaktadır. Boxplot(gbest.dmin), popülasyonun elde ettiği en iyi değerdeki verilerin küme merkezine olan en yakın mesafelerini kullanarak kutu grafiği yöntemiyle aykırı değerleri tespit etmektedir. Bu durumda uygulamanın kaba kodu Çizelge 3.2.'de olduğu gibi değerlendirilebilir:

Çizelge 3.2. Uygulamada kullanılan algoritmanın kaba kodu

```
başla
t=0;
başlangıç pop P(t);
uygunluk fonk F(t);
    koşul(durdurma koşulu sağlanıncaya kadar )
        başla
            t=t+1;
            güncelle p(t);
            değerlendir f(t);
        son
    son
son
gbest;
başla
boxplot(gbest.dmin);
son.
```

Gbest.dmin dizisinin tuttuğu değerler boxplot yönteminde kullanılarak uç değerler yani aykırı değerler tespit edilmektedir. Örneğin gbest.dmin değerleri {10,13,25,26,28,30,32,34,37,56} olsun. Eğer değerler küçükten büyüğe doğru sıralı değilse önce bu değerler küçükten büyüğe doğru sıralanmalıdır.

Median değeri =(28+30)= 29 olmak üzere Q1= 25, Q2(median değeri)=29, Q3=34 bulunmaktadır. Bu durumda alt uç değerler için Q1-1.5(Q3-Q1)= 11.5 olarak bulunur. Bu değerden küçük bütün değerler alt uç değer olarak kabul edildiği için 10 değeri bir alt uç değer olarak tespit edilir. Ve üst uç değerler için de Q3+ 1.5(Q3-Q1)= 47.5 olarak bulunur. Bu değerden büyük bütün değerler üst uç değer olarak kabul edildiği için 56 değeri bir uç değer olarak tespit edilir.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

4.1. Veri Setleri İçin Elde Edilen Aykırı Değerler

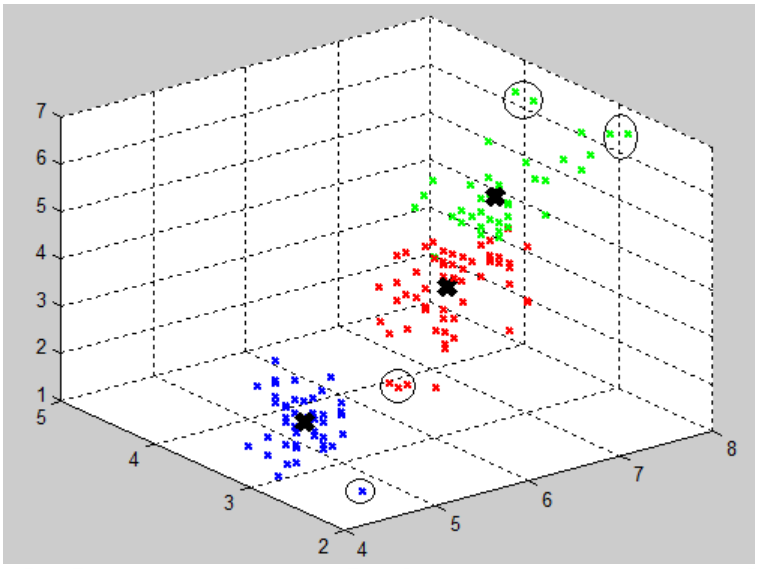
Her bir veri seti için küme sayısı 3,5 ve 7 ile iterasyon sayısı 100 popülasyon sayısı 20, iterasyon sayısı 100 popülasyon sayısı 50, iterasyon sayısı 200 popülasyon sayısı 20 ve iterasyon sayısı 200 popülasyon sayısı 50 olmak üzere GA, DGA ve PSO metasezgisel algoritmaları kullanılarak 12 farklı durumda aykırı değerler tespit edilmektedir. Aşağıda her bir veri seti için gösterilen tablolarda tespit edilen aykırı değerlerin indeksleri ve veri setine oranla yüzdelik değerleri gösterilmektedir. Daha sonra aykırı değerlerin normal değerlerden ayrıldığı, 3 boyutlu grafik üzerinde gösterilmektedir. Aykırı değerlerin grafiksel gösteriminde büyük koyu siyah 'X' işareti kümelerin merkezini göstermektedir. Kümelerin merkezinden diğer nesnelere göre daha uzak olan nesneler aykırı değer olarak kabul edilmektedir. Bu durumda aykırı değerler kabaca yuvarlak içerisine alınarak gösterilmeye çalışılmaktadır. Son olarak yine her bir algoritma için her bir veri setinden popülasyon sayısı 50 ile 100 iterasyonda ve 200 iterasyonda elde edilen BestCost değerleri gösterilmektedir. BestCost değeri, algoritmaların en doğru kümelemeyi yaptıkları andaki veri noktalarının küme merkezlerine olan toplam mesafeleridir.

150 veriden oluşan iris veri seti için elde edilen aykırı değerler Çizelge 4.1.'de gösterildiği gibidir. 12 farklı durumda elde edilen değerler göz önünde bulundurulduğunda GA'nın elde ettiği aykırı değerlerin ortalama sayısı %3.94, DGA'nın elde ettiği aykırı değerlerin ortalama sayısı %4.61 ve PSO'nun elde ettiği aykırı değerlerin ortalama sayısı %4.44 olarak tespit edilmektedir.

Çizelge 4.1. İris veri seti için elde edilen aykırı değerler

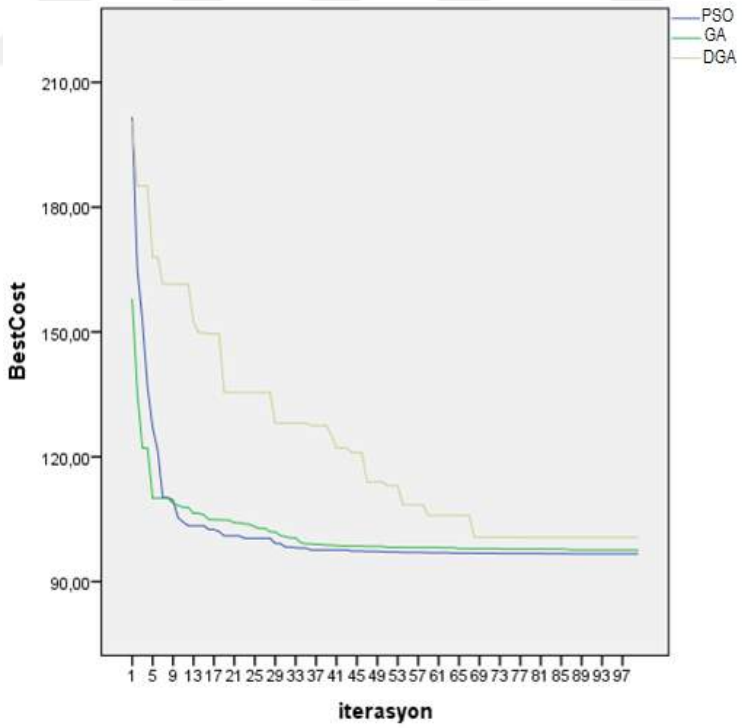
İris	GA	Adet	DGA	Adet	PSO	Adet
K=3 t=100 N=20	118, 119, 123, 132	4 %2.6	58, 61, 99, 106, 107, 118, 119, 123, 132	9 %6	99, 118, 119, 123, 132	5 %3.3
K=3 t=100 N=50	58, 99, 118, 119, 123, 132	6 %4	58, 61, 94, 99, 106, 107, 118, 119, 123, 132	10 %6.6	118, 119, 123, 132	4 %2.6
K=3 t=200 N=20	99, 118, 119, 123, 132	5 %3.3	58, 61, 94, 99, 118, 119, 123, 132	8 %5.3	118, 119, 123, 132	4 %2.6
K=3 t=200 N=50	107, 118, 119, 123, 132	5 %3.3	99, 106, 118, 119, 123, 132	6 %4	99, 118, 119, 123, 132	5 %3.3
K=5 t=100 N=20	106, 107, 118, 119, 123, 132	6 %4	58, 61, 94, 99, 118, 119, 123, 132	8 %5.3	16, 107, 118, 119, 123, 132	6 %4
K=5 t=100 N=50	16, 42, 106, 107, 118, 119, 123, 132	8 %5.3	58, 61, 94, 99, 107, 119	6 %4	42, 106, 107, 118, 119, 123, 132, 136	8 %5.3
K=5 t=200 N=20	106, 118, 119, 123, 132	5 %3.3	58, 61, 94, 99, 107, 123	6 %4	106, 107, 118, 119, 123, 132	6 %4
K=5 t=200 N=50	106, 107, 118, 119, 123, 132, 136	7 %4.6	58, 61, 94, 99, 107, 118, 119, 132	8 %5.3	16, 42, 106, 107, 118, 119, 120, 123, 132, 136	10 %6.6
K=7 t=100 N=20	107, 118, 119, 132	4 %2.6	106, 118, 119, 123, 132	5 %3.3	106, 107, 118, 119, 123, 132, 136	7 %4.6
K=7 t=100 N=50	106, 107, 118, 119, 123, 132	6 %4	58, 61, 94, 107, 118	5 %3.3	106, 107, 118, 119, 123, 132	6 %4
K=7 t=200 N=20	16, 106, 107, 118, 119, 123, 132	7 %4.6	118, 119, 123, 132	4 %2.6	16, 42, 106, 107, 118, 119, 120, 123, 132, 136	10 %6.6
K=7 t=200 N=50	16, 42, 99, 107, 118, 119, 123, 132	8 %5.3	107, 108, 110, 118, 119, 123, 132, 136	8 %5.3	16, 42, 106, 107, 118, 119, 120, 123, 132	9 %6

Bu aykırı değerlerin Şekil 4.1.'de 3 boyutlu grafiksel gösterimi bulunmaktadır. Grafiğe bakıldığında aykırı değerlerin normal verilerden ayrıldığı görülmektedir.

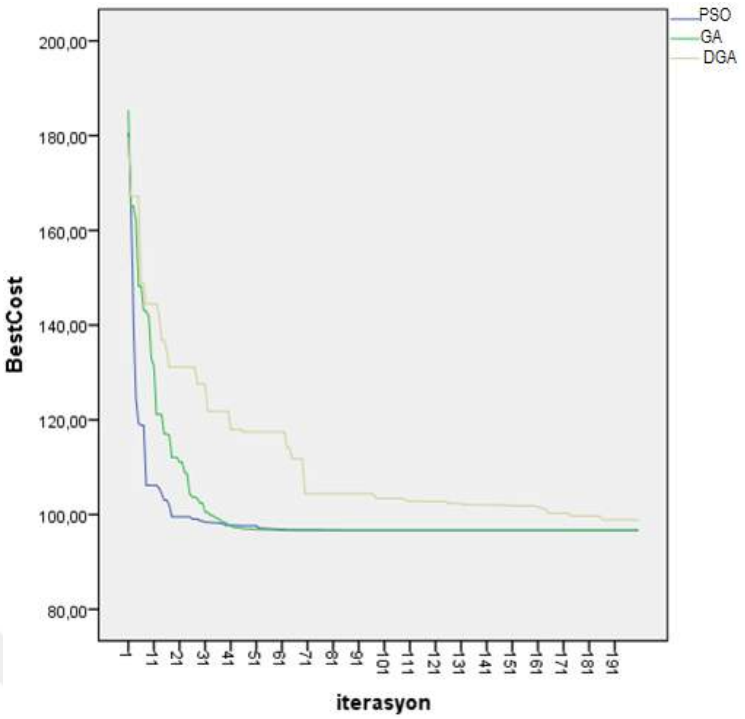


Şekil 4.1. İris veri seti için aykırı değerlerin grafiksel gösterimi

Şekil 4.2.'de ve Şekil 4.3.'de algoritmaların iterasyonlara bağlı elde ettikleri BestCost değerleri gösterilmektedir.



Şekil 4.2. İris veri seti için 100 iterasyondaki BestCost değerleri



Şekil 4.3. İris veri seti için 200 iterasyondaki BestCost değerleri

306 veriden oluşan haberman veri seti için elde edilen aykırı değerler Çizelge 4.2.'de gösterildiği gibidir. 12 farklı durumda elde edilen değerler göz önünde bulundurulduğunda GA'nın elde ettiği aykırı değerlerin ortalama sayısı %4.43, DGA'nın elde ettiği aykırı değerlerin ortalama sayısı %5.77 ve PSO'nun elde ettiği aykırı değerlerin ortalama sayısı %3.8 olarak tespit edilmektedir.

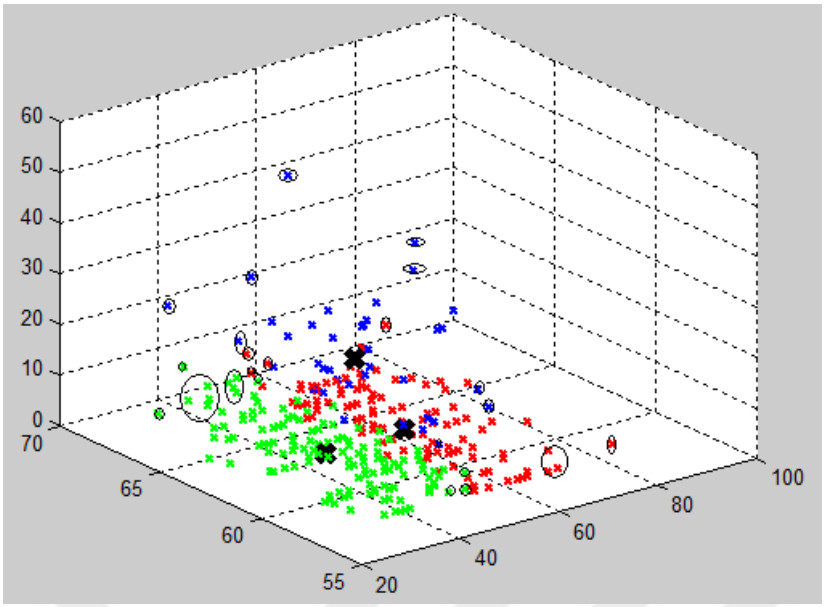
Çizelge 4.2. Haberman veri seti için elde edilen aykırı değerler

Haber-Man	GA	Adet	DGA	Adet	PSO	Adet
K=3 t=100 N=20	10, 23, 25, 63, 175, 216, 288, 303, 304, 305, 306	11 %3.1	10, 25, 44, 60, 63, 76, 80, 93, 97, 161, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 262, 288, 306	26 %7.7	6, 10, 15, 23, 25, 63, 175, 216, 253, 288, 304, 305, 306	13 %3.8
K=3 t=100 N=50	10, 23, 25, 44, 60, 63, 76, 80, 86, 93, 97, 161, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 288, 306	27 %8.0	10, 25, 44, 61, 63, 97, 161, 169, 175, 189, 217, 228, 253, 255, 261, 306	16 %4.7	3, 6, 10, 15, 23, 25, 63, 175, 216, 253, 288, 304, 305, 306	14 %4.1

Çizelge 4.2. (**devam**)

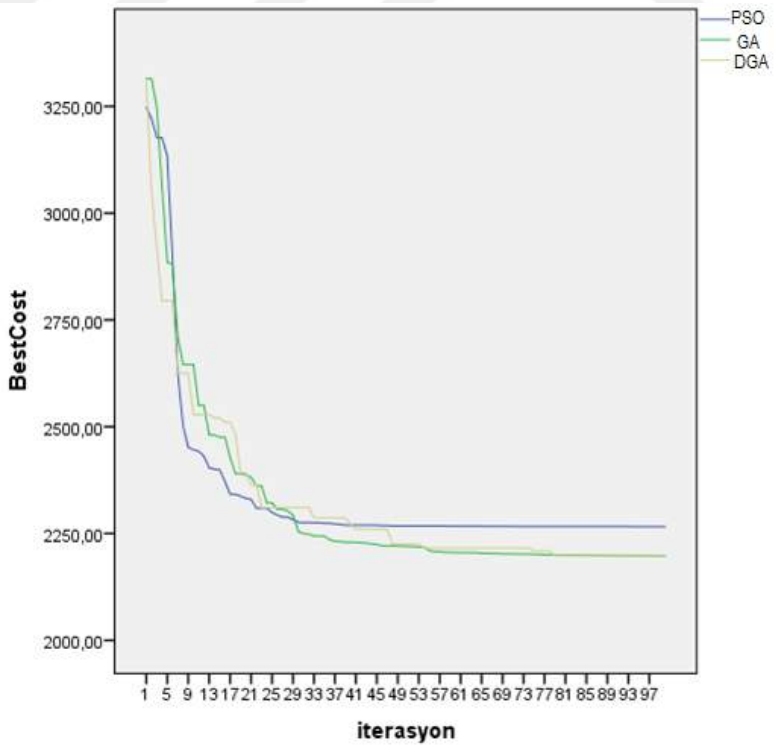
K=3 t=200 N=20	10, 23, 25, 44, 60, 63, 76, 80, 86, 93, 97, 161, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 288, 306	27 %8.0	10, 23, 25, 44, 60, 63, 76, 80, 86, 93, 97, 161, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 262, 288, 306	28 %8.3	10, 25, 44, 63, 97, 161, 169, 175, 189, 216, 228, 253, 255, 261	14 %4.1
K=3 t=200 N=50	6, 10, 23, 25, 63, 175, 216, 253, 288, 304, 305, 306	12 %3.5	10, 23, 25, 44, 60, 63, 76, 80, 93, 97, 161, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 262, 288, 306	27 %8.0	6, 10, 23, 25, 63, 175, 216, 253, 288, 304, 305, 306	12 %3.5
K=5 t=100 N=20	10, 15, 23, 25, 63, 175, 216, 253, 262, 270, 288, 306	12 %3.5	2, 4, 6, 10, 12, 15, 23, 25, 44, 63, 67, 86, 175, 216, 239, 253, 255, 261, 262, 270, 288, 306	22 %6.5	6, 10, 63, 175, 216, 288, 304, 305, 306	9 %2.6
K=5 t=100 N=50	6, 10, 15, 23,25, 44, 63, 175, 216, 253, 262, 270, 288, 306	14 %4.1	10, 15, 63, 175, 216, 224, 228, 239, 241, 253, 255, 261, 262, 270, 288, 306	16 %4.7	10, 15, 23, 25, 44, 253, 255, 261, 262, 288, 305, 306	12 %3.5
K=5 t=200 N=20	6, 10, 25, 63, 175, 216, 288, 304, 305, 306	10 %2.9	6, 10, 15, 23, 25, 44, 60, 63, 67, 175, 216, 239, 253, 261, 262, 270, 288, 306	18 %5.3	10, 15, 23, 25, 44, 63, 253, 255, 261, 262, 288, 305, 306	13 %3.8
K=5 t=200 N=50	6, 10, 15, 23, 25, 44, 63, 175, 216, 239, 253, 255, 61, 262, 270, 288, 306	17 %5.0	10, 15, 23, 25, 63, 175, 216, 253, 255, 261, 262, 270, 288, 306	14 %4.1	6, 10, 63, 175, 216, 288, 305, 306	8 %2.8
K=7 t=100 N=20	6, 10, 15, 25, 63, 175, 239, 262, 270, 288, 306	11 %3.2	9, 10, 13, 15, 23, 25, 32, 44, 63, 175, 216, 224, 228, 239, 241, 253, 255, 261, 262, 270, 288,306	22 %6.5	6, 10, 15, 23, 25, 63, 175, 216, 288, 306	10 %2.9
K=7 t=100 N=50	6, 10, 25, 63, 288, 306	6 %1.7	10, 63, 175, 216, 288, 306	6 %1.7	15, 23, 25, 63, 175, 216, 228, 304, 305, 306	10 %3.2
K=7 t=200 N=20	6, 10, 12, 25, 63, 216, 288, 306	8 %2.8	10, 63, 97, 175, 216, 306	6 %1.7 %	10, 15, 23, 25, 44, 63, 253, 255, 261, 262, 288, 306	12 %3.5
K=7 t=200 N=50	6, 10, 25, 23, 63, 216, 288, 306	8 %2.8	10, 15, 23, 63,107, 108, 109, 137, 175, 216, 306	11 %3.2	10, 23, 25, 44, 60, 63, 76, 80, 93, 97, 169, 175, 178, 182, 186, 189, 216, 224, 228, 241, 253, 255, 261, 262, 288, 306	26 %7.7

Bu aykırı değerlerin Şekil 4.4.'de 3 boyutlu grafiksel gösterimi bulunmaktadır. Grafiğe bakıldığında aykırı değerlerin normal verilerden ayrıldığı görülmektedir.

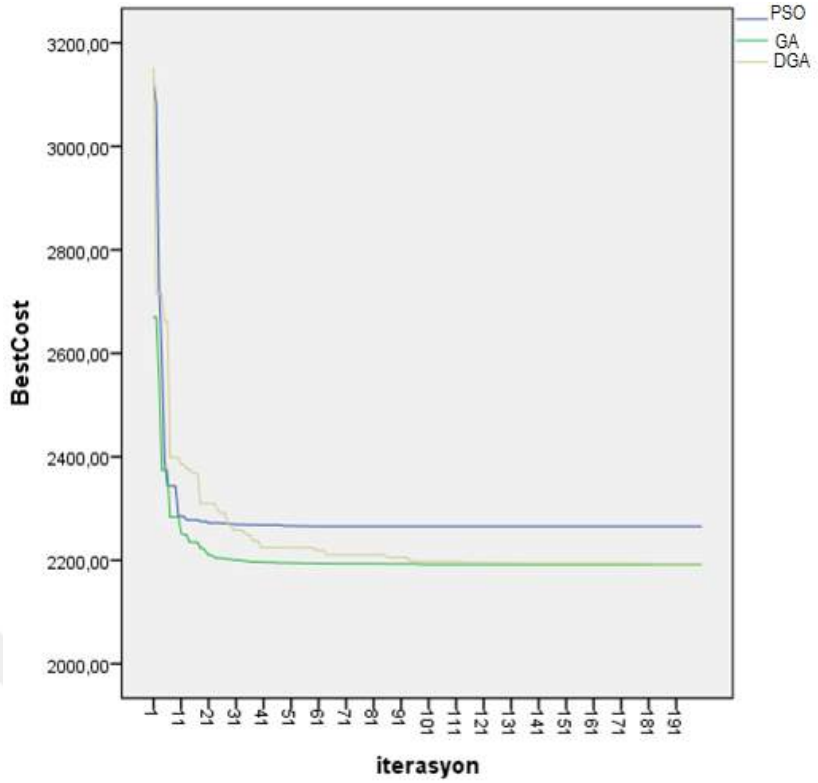


Şekil 4.4. Haberman veri seti için aykırı değerlerin grafiksel gösterimi

Şekil 4.5.'de ve Şekil 4.6.'da algoritmaların iterasyonlara bağlı elde ettikleri BestCost değerleri gösterilmektedir.



Şekil 4.5. Haberman veri seti için 100 iterasyondaki BestCost değerleri



Şekil 4.6. Haberman veri seti için 200 iterasyondaki BestCost değerleri

1484 veriden oluşan yeast veri seti için elde edilen değerler Çizelge 4.3.'de gösterildiği gibidir. 12 farklı durumda elde edilen değerler göz önünde bulundurulduğunda GA'nın elde ettiği aykırı değerlerin ortalama sayısı %1.8, DGA'nın elde ettiği aykırı değerlerin ortalama sayısı %3.5 ve PSO'nun elde ettiği aykırı değerlerin ortalama sayısı %2.7 olarak tespit edilmektedir.

Çizelge 4.3. Yeast veri seti için elde edilen aykırı değerler

Yeast	GA	Adet	DGA	Adet	PSO	Adet
K=3 t=100 N=20	5, 6, 32, 50, 60, 61, 74, 76, 85, 86, 111, 115, 116, 128, 130, 175, 176, 180, 184, 204, 208, 211, 251, 272, 320, 396, 471, 501, 502, 559, 593, 599, 610, 708, 721, 787, 989, 990, 991, 1040, 1078, 1080, 1146, 288,1389, 1390, 1453, 1455	48 %3.2	6, 32, 52, 85, 111, 115, 149, 176, 210, 251, 272, 320, 334, 396, 471, 489, 501, 502, 554, 559, 593, 599, 610, 708, 721, 842, 882, 989, 990, 991, 1080, 1146, 1253, 1288, 1389, 1390, 1455	37 %2.4	6, 32, 111, 115, 176, 210, 272, 334, 471, 502, 507, 599, 721, 989, 990, 991, 1080, 1146, 1288, 1389	20 %1.3
K=3 t=100 N=50	6, 52, 60, 74, 85, 115, 116, 128, 146, 149, 180, 204, 208, 272, 471, 501, 502, 507, 554, 708, 740, 842, 882, 989, 990, 991, 1080, 1253, 1288	29 %1.9	6, 32, 60, 74, 85, 111, 115, 176, 180, 204, 210, 220, 251, 272, 307, 320, 334, 393, 396, 434, 459, 471, 501, 502, 534, 554, 559, 588, 590, 593, 599, 610, 693, 708, 721, 740, 787, 842, 882, 989, 990, 991, 1040, 1080, 1091, 1106, 1126, 1146, 1214, 1253, 1288, 1358, 1389, 1390, 1416, 1453, 1455	57 %3.8	32, 115, 471, 502, 507, 989, 990, 991	8 %0.5
K=3 t=200 N=20	6, 32, 115, 158, 210, 272, 305, 334, 471, 489, 502, 507, 599, 740, 989, 990, 991, 1080, 1288, 1455	20 %1.3	6, 32, 85, 111, 115, 149, 176, 210, 220, 230, 251, 272, 305, 307, 320, 334, 393, 396, 434, 459, 460, 471, 489, 501, 502, 507, 534, 548, 554, 559, 582, 588, 590, 593, 599, 610, 649, 660, 693, 703, 708, 721, 733, 740, 760, 765, 787, 842, 882, 989, 990, 991, 1040, 1059, 1078, 1080, 1091, 1106, 1115, 1126, 1146, 1156, 1171, 1183, 1202, 1214, 1253, 1288, 1358, 1383, 1389, 1390, 1398, 1416, 1449, 1453, 1455, 1463	78 %5.2	32, 115, 471, 502, 507, 989 990, 991, 1080, 1146, 1288, 1389	12 %0.8
K=3 t=200 N=50	32, 60, 111, 115, 210, 272, 334, 471, 502, 507, 599, 989, 990, 991, 1080, 1288	16 %0.10	6, 32, 60, 74, 85, 111, 115, 149, 176, 180, 204, 210, 220, 230, 251, 272, 305, 307, 320, 334, 393, 396, 434, 459, 460, 471, 489, 501, 502, 507, 534, 548, 554, 559, 582, 588, 590, 593, 599, 610, 649, 660, 693, 708, 721, 733, 740, 760, 765, 787, 842, 882, 989, 990, 991, 1040, 1059, 1078, 1080, 1091, 1106, 1126, 1146, 1171, 1202, 1214, 1253, 1288, 1358, 1383, 1389, 1390, 1398, 1416, 1449, 1453, 1455, 1463	78 %5.2	32, 52, 115, 471, 502, 507, 989, 990, 991, 1080, 1146, 1288, 1455	13 %0.8

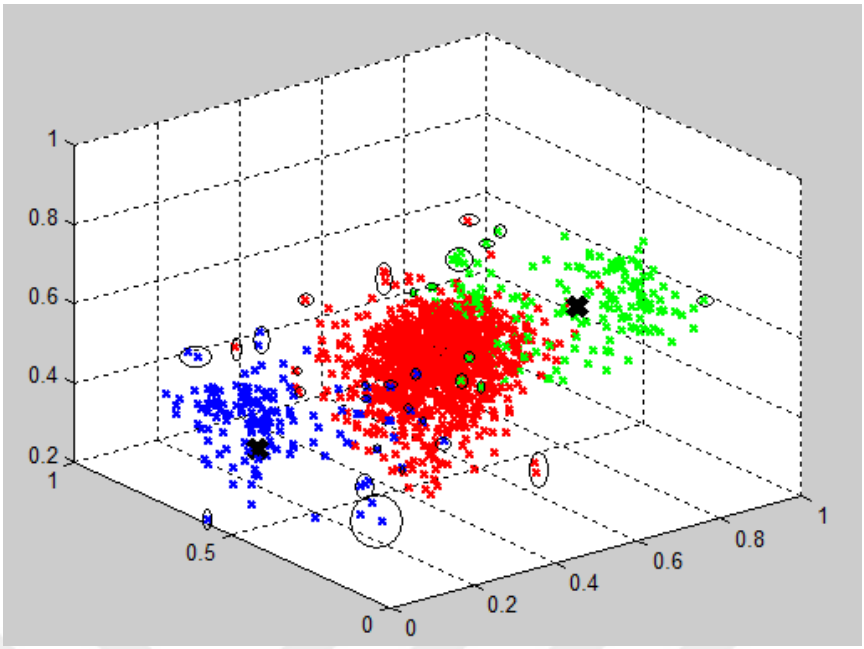
Çizelge 4.3. (**devam**)

K=5 t=100 N=20	32, 85, 115, 176, 210, 251, 272, 320, 502, 559, 593, 610, 721, 882, 989, 990, 991, 1080, 1146, 1389, 1390	21 %1.4	85, 176, 220, 251, 272, 305, 320, 334, 471, 489, 501, 502, 507, 559, 610, 721, 740, 842, 989, 990, 991, 1080, 1146, 1288, 1389, 1390, 1455	27 %1.8	32, 52, 60, 74, 85, 111, 115, 28, 149, 176, 180, 204, 208, 272, 320, 502, 559, 610, 882, 1080, 1146,1288, 1455	23 %1.5
K=5 t=100 N=50	6, 32, 52,60, 74, 85, 115, 116, 128, 149, 180, 204, 208, 272, 334, 471, 502, 507, 882, 989, 990, 991, 1080, 1253, 1288	25 %1.6	6, 7, 32, 52,60, 74, 85, 111, 115, 128, 146, 149, 158, 176, 180, 204, 210, 251, 272, 305, 320, 334, 396, 471, 489, 501, 502, 507, 554, 559, 593, 599, 610, 721, 740, 842, 882, 989, 990, 991, 1040, 1078, 1146, 1253, 1389, 1390, 1455	47 %3.1	5, 6, 32, 52, 60, 61, 74, 76, 85, 86, 104, 111, 113, 115, 116, 128, 130, 137, 146, 149, 169, 170, 175, 176, 180, 184, 204, 208, 210, 211, 220, 272, 305, 334, 396, 434, 459, 471, 489, 501, 502, 507, 554, 599, 708, 740, 787, 842, 882, 1040, 1078, 1080, 1106, 1214, 1253, 1288, 1453, 1455	58 %3.9
K=5 t=200 N=20	60, 74, 76, 85, 86, 116, 128, 180, 204, 208, 220, 272, 305, 334, 471, 501, 502, 504, 507, 554, 599, 708, 740, 989, 990, 991, 1424, 1455	28 %1.8	6, 7, 32, 52, 85, 111, 115, 149, 158, 176, 210, 220, 230, 251, 272, 305, 307, 320, 324, 334, 396, 471, 476, 489, 501, 502, 504, 507, 534, 554, 559, 593, 610, 660, 693, 708, 717, 721, 740, 760, 765, 779, 842, 876, 882, 891, 989, 990, 991, 1039, 1040, 1059, 1078, 1080, 1106, 1115, 1123, 1146, 1150, 1156, 1171, 1202, 1214, 1253, 1276, 1288, 1383, 1389, 1390, 1416, 1449, 1453, 1460	73 %4.9	6, 32, 46, 60, 85, 111, 115, 149, 176, 180, 204, 210, 220, 305, 307, 334, 393, 396, 459, 471, 489, 501, 502, 507, 534, 548, 554, 582, 590, 599, 649, 660, 693, 708, 733, 740, 765, 842, 882, 989, 990,991, 1040, 1059, 1078, 1080, 1091, 1106, 1115, 1126, 1156, 1171, 1202, 1214, 1253, 1288, 383, 1398, 1416, 1449, 1453, 455, 1463,	63 %4.2
K=5 t=200 N=50	6, 30, 32, 46, 60, 70, 74, 85,111, 115, 116, 128, 176, 180, 204, 208, 220, 305, 307, 396, 471, 489, 501, 502, 507, 554, 660, 708, 740, 765, 797, 798, 842, 989, 990, 991, 1040, 1059, 1078, 1357	40 %2.6	6, 32, 60, 85, 111, 115, 176, 210, 220, 230, 251, 272, 305, 307, 320, 393, 396, 489, 501, 502, 507, 534, 554, 559, 588, 590, 593, 599,610, 660, 693, 708, 721, 740, 760, 765, 787, 842, 882, 989, 990, 991, 1040, 1059, 1078, 1080, 1106, 1126, 1146, 1171, 1202, 1214, 1253, 1288, 1358, 1389, 1390, 1416, 1453, 1455	60 %4.0	6, 7, 32, 46, 52, 60, 74, 85, 111, 115, 149, 158, 176, 180, 204, 210, 220, 272, 305, 307, 324, 334, 396, 471, 476, 489, 501, 502, 504, 507, 534, 554 599, 660, 693, 708, 740, 765, 779, 842, 876, 882, 891, 989, 990, 991, 1039, 104, 1059, 1078, 1080, 1106, 1115, 1123, 1150, 1156, 1171, 1183, 1202, 1214, 1253,1276, 1288, 1358, 1383, 416, 1449, 1453, 1460	69 %4.6
K=7 t=100 N=20	6, 32, 85, 111, 115, 158, 176, 210, 251, 320, 334, 471, 507, 559, 593, 610, 721, 989, 990, 991, 1080, 1146, 1389, 1390	24 %1.6	32, 52, 60, 74, 111, 115, 128, 149, 176, 180, 204, 210, 251, 272, 334, 396, 471, 489, 501, 502, 507, 593, 599, 740, 882, 989, 990, 991, 1146, 1253, 1288, 1389, 1455	33 %2.2	6, 32, 60, 74, 85, 115, 116, 128, 180, 204, 208, 210, 220, 272, 305, 334, 396, 471, 489, 501, 502, 507, 554, 599, 708, 740, 787, 842, 989, 990, 991 1078, 1080, 1455	34 %2.2

Çizelge 4.3. (**devam**)

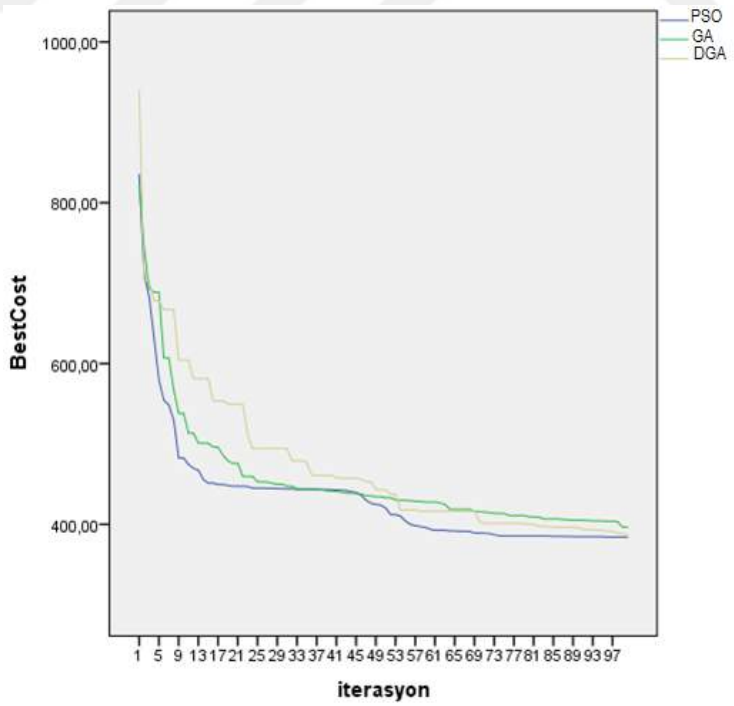
K=7 t=100 N=50	32, 115, 176, 210, 502, 507, 721, 989, 990, 991, 1146, 1389	12 %0.8	32, 115, 176, 210, 251, 272, 320, 334, 396, 471, 489, 501, 502, 507, 554, 559, 593, 599, 610, 708, 721, 740, 842, 882, 989, 990, 991, 1078, 1080, 1106, 1146, 1214, 1253, 1288, 1389, 1390, 1455	37 %2.4	5, 6, 32, 50, 60, 61, 74, 76, 80, 81, 85, 86, 91, 111, 114, 115, 116, 118, 119, 128, 130, 149, 175, 180, 184, 204, 208, 210, 211, 220, 272, 305, 307, 334, 396, 434, 459, 471, 489, 501 502, 507, 554, 590, 599, 660, 708, 740, 765, 787, 842, 989, 990, 991, 1040, 1059, 1078, 1080, 1091, 1126, 1455	61 %4.1
K=7 t=200 N=20	30, 47, 70, 83, 85, 115, 195, 220, 305, 489, 502, 504, 507, 554, 599, 708, 740, 797, 798, 882, 989, 990, 991, 1080, 1253, 1288, 1357	27 %1.8	6, 32, 60, 74, 85, 111, 115, 176, 180, 204, 210, 220, 230, 251, 272, 305, 307, 320, 334, 396, 471, 489, 501, 507, 529, 554, 559, 590, 593, 599, 610, 660, 708, 721, 740, 760, 765, 842, 989, 990, 991, 1040, 1059, 1078, 1126, 1146, 1156, 1288, 1358, 1389, 1390, 1398	52 %3.5	6, 32, 85, 111, 115, 149, 176, 210, 220, 272, 305, 307, 334, 393, 396, 434, 459, 471, 489, 501, 502, 507, 534, 554, 582, 588, 590, 599, 660, 693, 703, 708, 733, 740, 765, 787, 842, 882, 989, 990, 991, 1040, 1059, 1078, 1080, 1091, 106, 1126, 1171, 1202, 1214, 1253, 1288, 1358, 1383, 1416, 1453, 1455, 1463,	59 %3.9
K=7 t=200 N=50	6, 32, 52, 60, 74, 80, 81, 85, 86, 111, 115, 128, 149, 176, 180, 204, 210, 251, 272, 305, 320, 334, 396, 471, 489, 501, 502, 507, 554, 559, 593, 599, 610, 708, 721, 740, 842, 990, 991, 1040, 1078, 1146, 1253, 1288, 1389, 1390, 1455	47 %3.1	6, 32, 52, 60, 74, 80, 81, 85, 86, 111, 115, 128, 149, 176, 180, 204, 210, 251, 272, 305, 320, 334, 396, 471, 489, 501, 502, 507, 554, 559, 593, 599, 610, 708, 721, 740, 842, 990, 991, 1040, 1078, 1146, 1253, 1288, 1389, 1390, 1453, 1455	48 %3.2	6, 7, 32, 46, 52, 60, 74, 111, 115, 149, 158, 176, 180, 204, 210, 220, 272, 305, 307, 324, 334, 396, 471, 476, 489, 501, 502, 504, 507, 534, 554, 660, 693, 703, 708, 740, 765, 779, 842, 876, 882, 891, 989, 990, 991, 1039, 1040, 1059, 1078, 1080, 1106, 1115, 1123,1150, 1156, 1171, 1183, 1202, 1253, 1276, 1288, 1358, 1383, 1416, 1449, 1453, 1460	67 %4.5

Bu aykırı değerlerin Şekil 4.7.'de 3 boyutlu grafiksel gösterimi bulunmaktadır. Grafiğe bakıldığında aykırı değerlerin normal verilerden ayrıldığı görülmektedir.

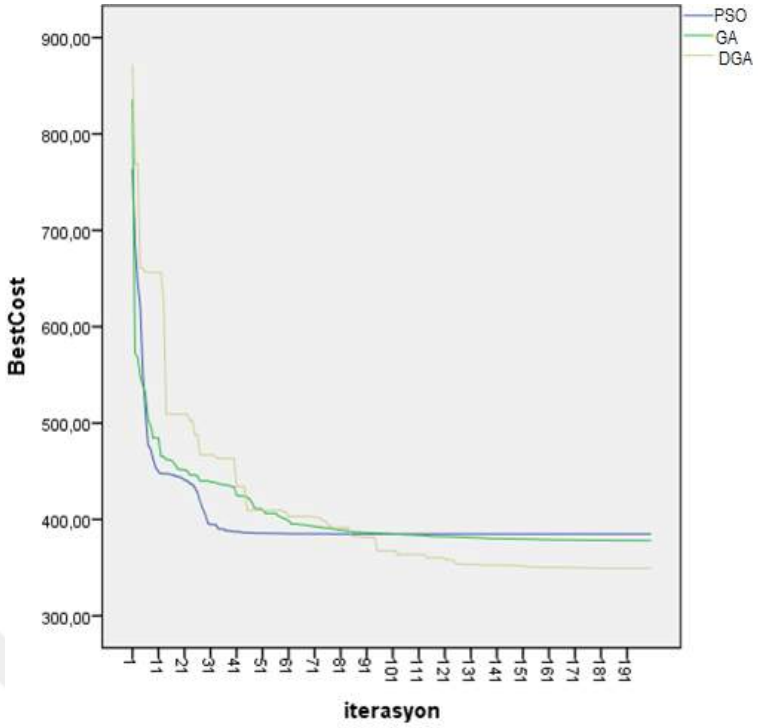


Şekil 4.7. Yeast veri seti için aykırı değerlerin grafiksel gösterimi

Şekil 4.8.'de ve Şekil 4.9.'da algoritmaların iterasyonlara bağlı elde ettikleri BestCost değerleri gösterilmektedir.



Şekil 4.8. Yeast veri seti için 100 iterasyondaki BestCost değerleri



Şekil 4.9. Yeast veri seti için 200 iterasyondaki BestCost değerleri

748 veriden oluşan blood veri seti için elde edilen değerler Çizelge 4.4.'de gösterildiği gibidir. 12 farklı durumda elde edilen değerler göz önünde bulundurulduğunda GA'nın elde ettiği aykırı değerlerin ortalama sayısı %4.57, DGA'nın elde ettiği aykırı değerlerin ortalama sayısı %5.3 ve PSO'nun elde ettiği aykırı değerlerin ortalama sayısı %4.76 olarak tespit edilmektedir.

Çizelge 4.4. Blood veri seti için elde edilen aykırı değerler

Blood	GA	Adet	DGA	Adet	PSO	Adet
K=3 t=100 N=20	1, 4, 5, 8, 10, 11, 22, 28, 37, 45, 51, 52, 54, 95,115, 116, 118,120, 132, 135, 138, 153, 184, 194, 242, 247, 248, 260, 271, 292, 342, 445, 501, 502, 503, 504, 505, 506, 507, 508, 509, 512, 513, 518, 521, 522, 529, 549, 561, 592, 594, 595, 600, 606, 630	55 %7.3	1, 4, 5, 10, 11, 13, 32, 33, 45, 56, 63, 110, 116, 176, 236, 242, 279, 342, 396, 501, 502, 503, 504, 505, 506, 507, 509, 518, 519, 520, 524, 529, 542, 564, 675, 678	36 %4.8	1, 4, 5, 10,11, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 678	26 %3.4

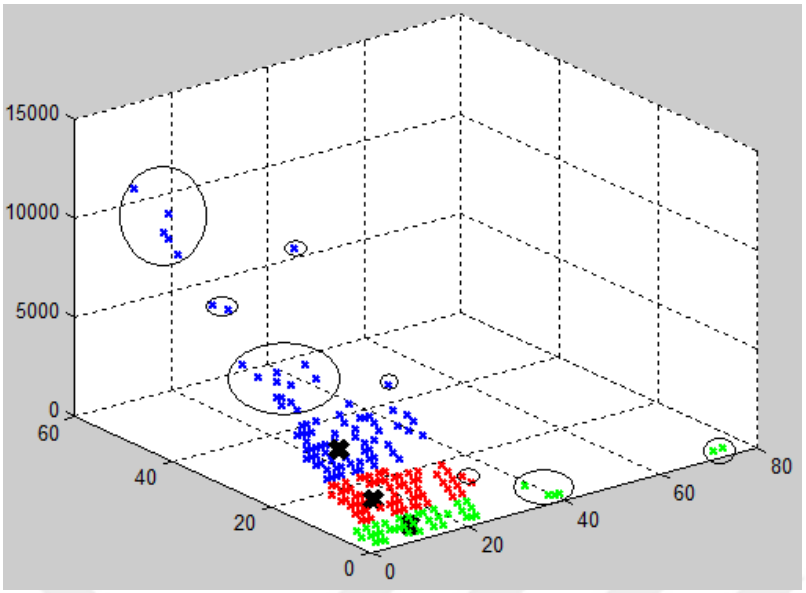
Çizelge 4.4. (**devam**)

K=3 t=100 N=50	1, 4, 5, 10, 11, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502,503, 504, 505, 506, 507, 509, 518, 529, 678	26 %3.4	1, 4, 5, 10, 11, 45, 56, 116, 242, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 678	22 %2.9	1, 4, 5, 10, 11, 32, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 628	27 %3.6
K=3 t=200 N=20	1, 4, 5, 10, 11, 13, 32, 33, 45, 56, 63, 110, 116, 176, 236, 242, 279, 342, 396, 501, 502, 503, 504, 505, 506, 507, 509, 518, 519, 520, 524, 529, 542, 564, 675,678	36 %4.8	1, 4, 5, 10, 11, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 678	26 %3.4	1, 4, 5, 10, 11, 13, 32, 33, 45, 56, 63, 110, 116, 176, 236, 242, 279, 342, 396, 501, 502, 503, 504, 505, 506, 507, 509, 518, 519, 520, 524, 529, 542, 564, 675, 678	36 %4.8
K=3 t=200 N=50	1, 4, 5, 10, 11, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 678	26 %3.4	1, 4, 5, 10, 11, 13, 32, 33, 45, 56, 63, 110, 116, 176, 236, 242, 279, 342, 396, 501, 502, 503, 504, 505, 506, 507, 509, 518, 519, 520, 524, 529, 542, 564, 675, 678	36 %4.8	1, 4, 5, 10, 11, 45, 56, 73, 97, 106, 116, 242, 244, 279, 342, 501, 502, 503, 504, 505, 506, 507, 509, 518, 529, 678	26 %3.4
K=5 t=100 N=20	1, 2, 4, 5, 10, 11, 14, 41, 45, 56, 102, 116, 242, 255, 277, 342, 364, 502, 503, 504, 505, 507, 509, 510, 518, 523, 529, 678	28 %3.7	1, 4, 5, 10, 11, 22, 45, 51,52, 56, 73, 95, 97, 106, 116, 118,120, 135, 153, 194, 242, 244, 271, 279, 501, 502, 503, 504, 505, 507, 508, 509,512, 513, 518, 521, 522, 529, 561, 592,594, 595, 600, 606, 630, 678	46 %6.1	1, 3, 4, 10, 18, 35, 45, 56, 59, 66, 73, 97, 106, 189, 244, 279, 281, 321, 328, 342, 360, 367, 501, 503, 504, 506, 507, 509, 515, 518, 528, 543, 563, 634, 636, 656, 673, 678	38 %5.0
K=5 t=100 N=50	1, 3, 4, 10, 18, 35, 45, 56, 59, 66, 73, 97, 106, 189, 244, 279, 281, 321, 328, 342, 360, 367, 501, 503, 504, 505, 506, 507, 509, 515, 518, 528, 543, 563, 634, 636, 656, 673, 678	39 %5.2	1, 4, 5, 8, 10, 11, 22, 28, 37, 45, 51, 52, 54, 56, 95, 115, 116, 118, 120, 132, 135, 138, 153, 184, 194, 242, 247, 248, 260, 271, 279, 292, 342, 445, 502, 503, 504, 505, 507, 508, 509, 512, 513, 518, 521, 522, 529, 549, 561, 592, 594, 595, 600, 606, 630, 678	56 %7.4	1, 4, 5, 10, 11, 13, 22, 32, 33, 45, 51, 52, 56, 63, 73, 95, 97, 106, 110, 116, 118, 120, 135, 153, 176, 194, 236, 242, 244, 271, 279, 342, 396, 502, 503, 505, 507, 508, 509, 512, 513, 518, 519, 520, 521, 522, 524, 529, 542, 561, 564, 592, 594, 595, 600, 606, 630, 675,678	59 %7.8
K=5 t=200 N=20	1, 4, 5, 11, 22, 32, 45, 51, 52, 56, 63, 95, 97, 106, 110, 116, 118, 120, 135, 153, 194, 242, 244, 271, 279, 342, 502, 503, 505, 506, 507, 508, 509, 512, 513, 518, 520, 521, 522, 529, 561, 592, 594, 595, 600, 606, 630, 678	48 %6.4	1, 4, 5, 11, 13, 22, 32, 33, 45, 51, 52, 56, 63, 73, 95, 97, 106, 110, 116, 118, 120, 153, 176, 194, 236, 242, 244, 271, 279, 342, 396, 502, 503, 505, 506, 507, 508, 509, 512, 513, 518, 519, 520, 521, 522, 524, 529, 542, 561, 564, 592, 594, 595, 600, 606, 630, 675, 678	58 %7.7	1, 4, 5, 11, 13, 22, 32, 33, 45, 51, 52, 56, 63, 73, 95, 97, 106, 110, 116, 118, 120, 135, 153, 176, 194, 236, 242, 244, 271, 279, 342, 396, 502, 503, 505, 507, 508, 509, 512, 513, 518, 519, 520, 521, 522, 524, 529, 542, 561, 564, 592, 594, 595, 600, 606, 630, 675,678	58 %7.7

Çizelge 4.4. (**devam**)

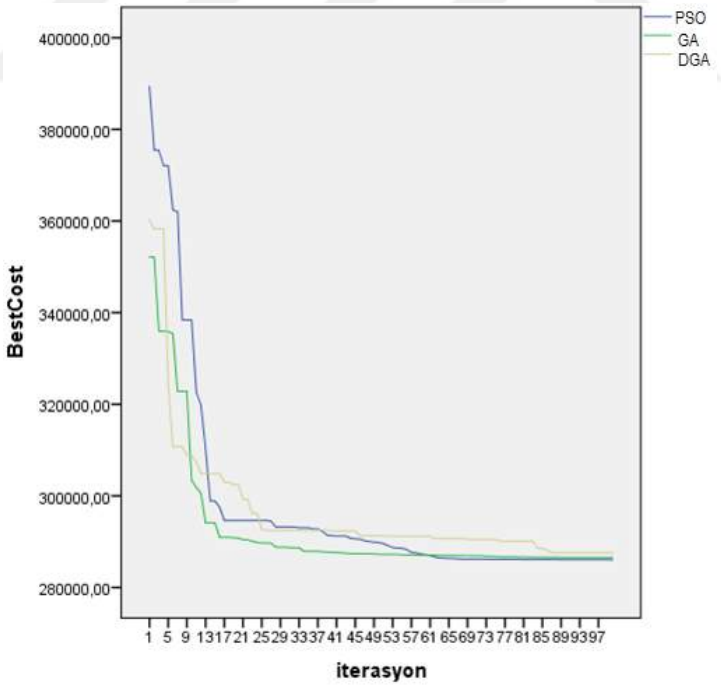
K=5 t=200 N=50	1, 2, 4, 5, 8, 10, 11, 14, 28, 37, 41, 45, 54, 56, 102, 115, 116, 132, 138, 184, 242, 247, 248, 255, 260, 277, 292, 342, 364, 445, 502, 503, 505, 507, 509, 510, 518, 523, 529, 549, 678	41 %5.4	1, 4, 5, 10, 11, 22, 32, 33, 45, 51, 52, 56, 63, 73, 95, 97, 106, 110, 116, 118, 120, 135, 153, 176, 194, 236, 42, 244, 271, 279, 342, 396, 502, 503, 505, 507, 508, 509, 512, 513, 518, 519, 520, 521, 522, 524, 529, 542, 561, 564, 592, 594, 595, 600, 606, 630, 675, 78	58 %7.7	1, 4, 5, 11, 22, 32, 33, 45, 51, 52, 56, 63, 73, 95, 97, 106, 116, 118, 120, 135, 153, 176, 194, 236, 242, 244, 271, 279, 342, 396, 502, 503, 505, 507, 508, 509, 512, 513, 518, 519, 520, 521, 522, 524, 529, 542, 561, 564, 592, 594, 595, 600, 606, 630, 675, 678, 681	57 %7.6
K=7 t=100 N=20	1, 2, 4, 5, 10, 11, 14, 17, 26, 31, 39, 41, 45, 56, 102, 116, 142, 242, 255, 262, 277, 342, 364, 439, 502, 503, 504, 505, 507, 509, 510, 514, 518, 523, 529, 545, 547, 565, 674, 678, 681	41 %5.4	1, 2, 4, 5, 8, 10, 11, 14, 28, 37, 41, 45, 54, 56, 73, 97, 102, 106, 115, 116, 132, 138, 184, 242, 244, 247, 248, 255, 260, 277, 279, 292, 342, 364, 445, 501, 502, 503, 504, 505, 507, 509, 510, 518, 523, 529, 549, 678	48 %6.4	1, 2, 4, 5, 10, 11, 14, 41, 45, 56, 102, 116, 242, 255, 277, 342, 364, 502, 505, 506, 507, 509, 510, 523, 529, 678	26 %3.4
K=7 t=100 N=50	1, 2, 4, 5, 10, 11, 14, 17, 26, 31, 39, 41, 45, 56, 102, 116, 142,242, 255, 262, 277, 279, 342, 364, 439, 502, 503, 504, 505, 507, 509, 510, 514, 518, 523, 529, 545, 547, 565, 674, 678, 681	42 %5.6	4, 5, 9, 10, 11, 25, 34, 45,46, 47, 56, 70, 98, 103, 112, 116, 141,186, 200, 205, 240, 242,279, 313,319, 342, 501, 502, 503, 504, 505, 506,507, 509, 518, 525, 529, 562, 566, 568, 578, 627, 631, 640, 678,	45 %6.0	1, 2, 4, 5, 10, 11, 14, 17, 26, 31, 39, 41, 45, 56, 102, 116, 142, 242, 255, 262, 277, 279, 342, 364, 439, 502, 503, 504, 505, 507, 509, 510, 514, 518, 523, 529, 545, 565, 674, 678, 681	41 %5.4
K=7 t=200 N=20	1, 5, 10, 56, 116, 342, 503, 505, 518, 529, 678	11 %1.4	1, 4, 5, 10, 11, 17, 26, 31, 39, 45, 56, 116, 142, 242, 262, 342, 439, 502, 503, 505, 506, 507, 509, 514, 518, 529, 545, 547, 565, 674, 678, 681	32 %4.2	1, 4, 5, 10, 11, 45, 56, 116, 242, 342, 502, 503, 505, 507, 509, 518, 529, 678	18 %2.4
K=7 t=200 N=50	1, 4, 5, 10, 11, 45, 56, 116, 242, 342, 502, 503, 505, 507, 509, 518, 529, 678	18 %2.4	1, 4, 5, 10, 11, 45, 116, 242, 342, 502, 503, 505, 507, 509, 518, 529	16 %2.1	1, 4, 5, 11, 45, 56, 116, 242, 342, 502, 505, 506, 507, 509, 529, 678	16 %2.1

Bu aykırı değerlerin Şekil 4.10.'da 3 boyutlu grafiksel gösterimi bulunmaktadır. Grafiğe bakıldığında aykırı değerlerin normal verilerden ayrıldığı görülmektedir.

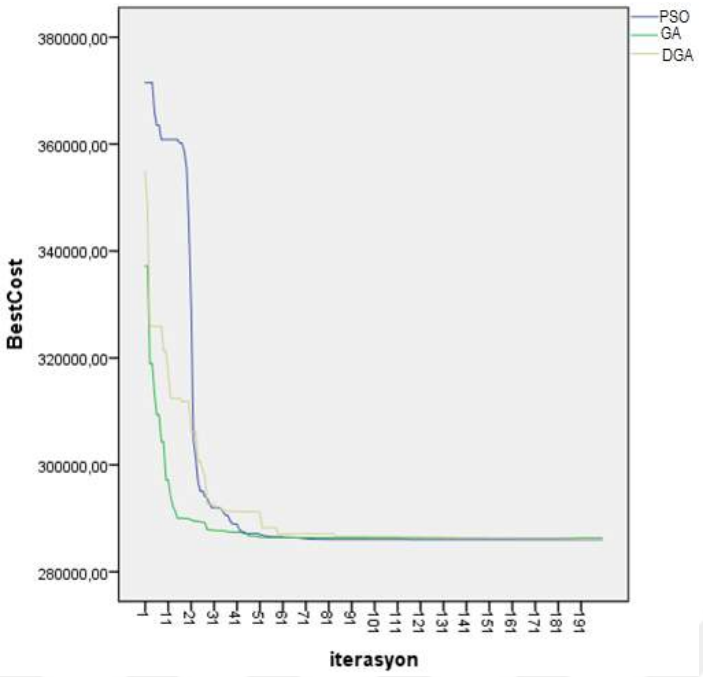


Şekil 4.10. Blood veri seti için aykırı değerlerin grafiksel gösterimi

Şekil 4.11.'de ve Şekil 4.12.'de algoritmaların iterasyonlara bağlı elde ettikleri BestCost değerleri gösterilmektedir.



Şekil 4.11. Blood veri seti için 100 iterasyondaki BestCost değerleri



Şekil 4.12. Blood veri seti için 200 iterasyondaki BestCost değerleri

345 veriden oluşan bupa veri seti için elde edilen değerler Çizelge 4.5.'de gösterildiği gibidir. 12 farklı durumda elde edilen değerler göz önünde bulundurulduğunda GA'nın elde ettiği aykırı değerlerin ortalama sayısı %3.59, DGA'nın elde ettiği aykırı değerlerin ortalama sayısı %5.04 ve PSO 'nun elde ettiği aykırı değerlerin ortalama sayısı %4.8 olarak tespit edilmektedir.

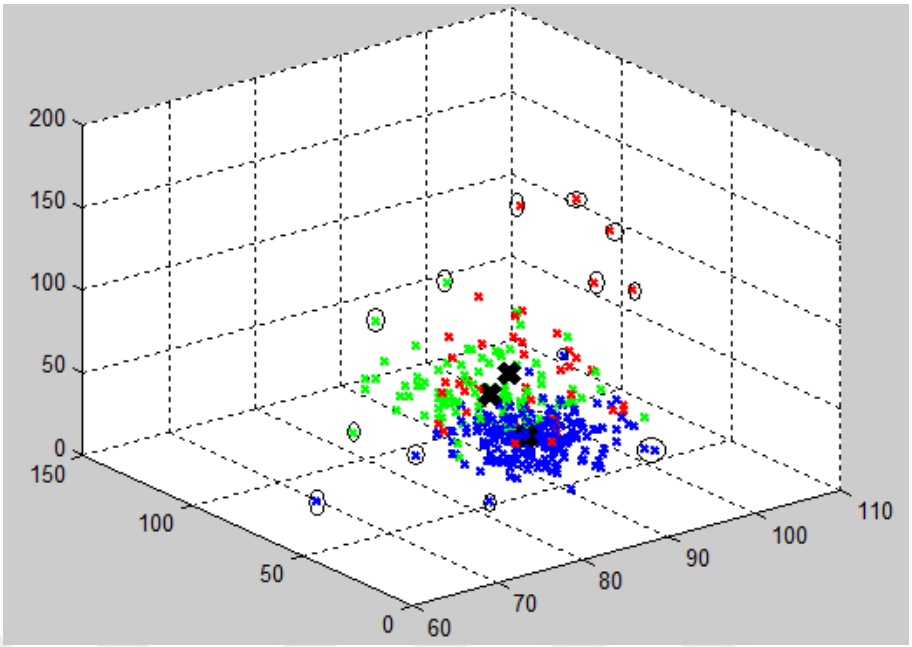
Çizelge 4.5. Bupa veri seti için elde edilen aykırı değerler

Bupa	GA	Adet	DGA	Adet	PSO	Adet
K=3 t=100 N=20	25, 36, 85, 123, 134, 182, 205, 233, 286, 300, 317, 329, 334, 335, 337	15 %4.3	25, 36, 85, 123, 134, 179, 190, 233, 286, 300, 317, 323, 331, 335, 342	15 %4.3	25, 36, 85, 115, 123, 134, 157, 179, 190, 233, 278, 286, 300, 316, 317, 323, 331, 335, 342, 345	20 %5.7
K=3 t=100 N=50	25, 36, 85, 123, 134, 205, 233, 300, 317, 335	10 %2.8	22, 25, 36, 85, 134, 157,179, 190, 233, 278,286, 300, 316, 323, 331, 342	16 %4.6	25, 36, 85, 115, 123, 134, 157, 179, 183, 190, 233, 278, 286, 300, 316, 317, 323, 326, 331, 335, 342, 345	22 %6.3
K=3 t=200 N=20	36, 85, 123, 134, 175, 182, 205, 233, 286, 300, 317, 329, 334, 335	14 %4.0	25, 36, 85, 115, 134, 179, 183, 190, 233, 278, 286, 300, 316, 317, 323, 331, 335, 342, 345	19 %5.5	25, 36, 85, 115, 123, 134, 157, 179, 190, 233, 278, 286, 300, 316, 317, 323, 331, 335, 342, 345	20 %5.7

Çizelge 4.5. (**devam**)

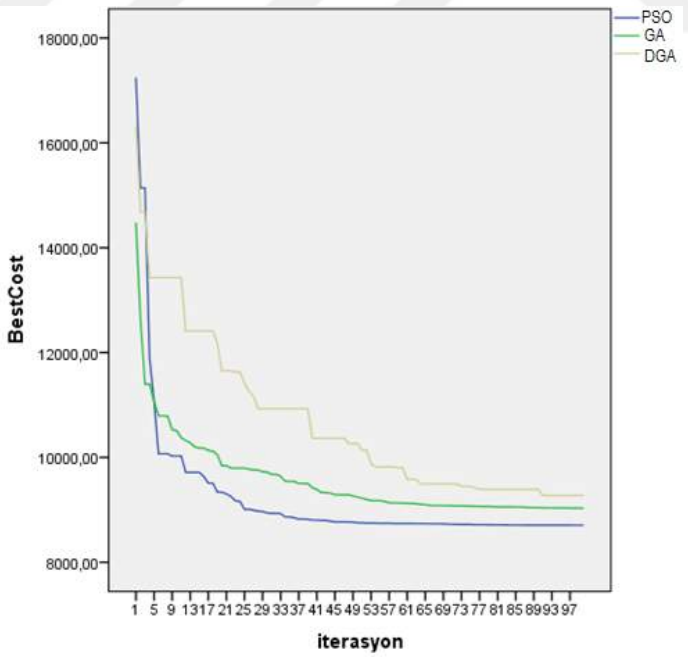
K=3 t=200 N=50	36, 85, 123, 134, 168, 175, 182, 205, 233, 286, 300, 317, 329, 334, 335, 337	16 %4.6	20, 25, 36, 85, 123, 134, 157, 179, 190, 233, 278, 286, 300, 316, 317, 323, 331, 335, 342, 345	20 %5.7	25, 36, 85, 115, 123, 134, 157, 179, 190, 233, 278, 286, 300, 316, 317, 323, 331, 335, 342, 345	20 %5.7
K=5 t=100 N=20	36, 77, 85, 115, 123, 179, 233, 286, 323, 329, 335, 337	12 %3.4	25, 36, 77, 85, 115, 183, 190, 205, 233, 278, 286, 300, 316, 317, 323, 326, 331, 335, 342, 345	20 %5.7	25, 36, 77, 85, 134, 157, 183, 233, 278, 286, 300, 317, 326, 331, 335, 345	16 %4.6
K=5 t=100 N=50	25, 36, 85, 123, 157, 183, 211, 286, 335, 337	10 %2.8	25, 36, 77, 85, 115, 123, 134, 157, 175, 233, 278, 300, 317, 331, 345	15 %4.3	25, 36, 77, 85, 123, 182, 183, 205, 278, 286, 326, 335, 345	13 %3.7
K=5 t=200 N=20	36, 77, 85, 115, 123, 134, 157, 183, 205, 233, 278, 286, 300, 317, 323, 326, 331, 335, 345	19 %5.5	25, 36, 85, 115, 134, 157, 182, 183, 205, 233, 278, 286, 300, 317, 323, 326, 331, 335, 345	19 %5.5	25, 77, 85, 123, 182, 183, 205, 278, 286, 326, 335, 345	12 %3.4
K=5 t=200 N=50	36, 77, 85, 123, 182, 183, 205, 278, 286, 326, 335	11 %3.1	25, 36, 134, 179, 183, 233, 278, 286, 300, 317, 323, 335, 345	13 %3.7	25, 36, 77, 85, 123, 182, 183, 205, 278, 286, 326, 334, 335, 345	14 %4.0
K=7 t=100 N=20	36, 77, 85, 115, 123, 134, 179, 233, 335	9 %2.6	25, 36, 77, 85, 115, 123, 134, 179, 183, 190, 205, 233, 286, 300, 316, 317, 326, 329, 335, 342, 345	21 %6.0	25, 36, 41, 77, 85, 115, 123, 134, 183, 186, 205, 233, 278, 300, 317, 323, 326, 331, 335, 345	20 %5.7
K=7 t=100 N=50	20, 36, 77, 85, 123, 233, 300, 326, 335	9 %2.6	25, 36, 85, 115, 182, 205, 233, 286, 300, 323, 331, 335, 342, 345	14 %4.0	13, 25, 36, 77, 85, 123, 220, 278, 286, 300, 326, 331, 335, 345	14 %4.0
K=7 t=200 N=20	36, 77, 85, 115, 123, 157, 183, 233, 278, 286, 300, 323, 326, 335, 345	15 %4.3	25, 36, 77, 85, 123, 134, 147,157, 183, 190, 233,278, 286, 300,317, 323, 326, 331, 345	19 %5.5	25, 36, 123, 157, 168, 175, 182, 183, 205, 278, 286, 326, 334, 335, 345	15 %4.3
K=7 t=200 N=50	36, 77, 85, 115, 123, 205, 286, 326, 335	9 %2.6	25, 36, 77, 85, 115, 134, 168, 179, 182, 205, 233, 300, 316, 317, 323, 331, 342, 345	18 %5.2	25, 36, 77, 85, 115, 123, 134, 175, 205, 220, 286, 326, 331, 335, 345	15 %4.3

Bu aykırı değerlerin Şekil 4.13.'de 3 boyutlu grafiksel gösterimi bulunmaktadır. Grafiğe bakıldığında aykırı değerlerin normal verilerden ayrıldığı görülmektedir.

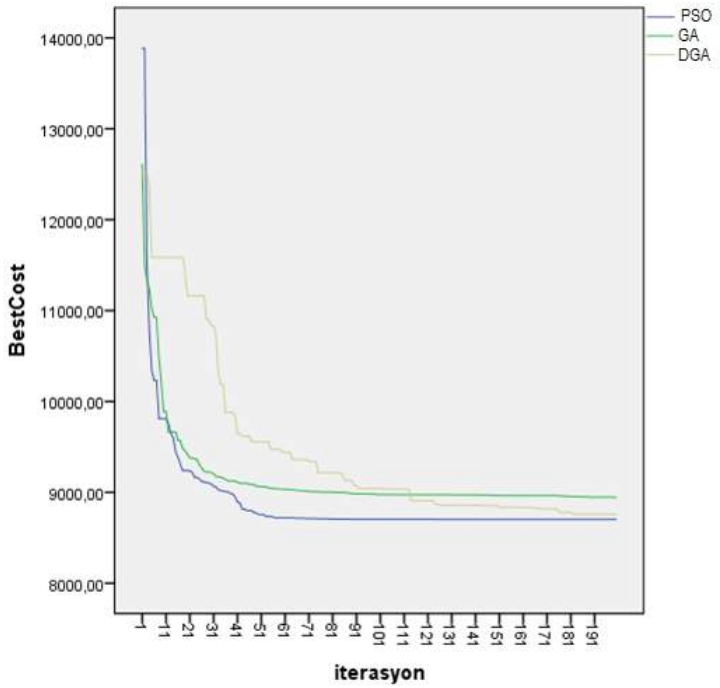


Şekil 4.13. Bupa veri seti için aykırı değerlerin grafiksel gösterimi

Şekil 4.14.'de ve Şekil 4.15.'de algoritmaların iterasyonlara bağlı elde ettikleri BestCost değerleri gösterilmektedir



Şekil 4.14. Bupa veri seti için 100 iterasyondaki BestCost değerleri



Şekil 4.15. Bupa veri seti için 200 iterasyondaki BestCost değerleri

4.2. Veri Setlerinin Elde Edilen Sınıflandırma Başarıları

Uygulamada kullanılan iris, bupa, blood, haberman ve yeast veri setlerinden kullanılan yöntemle aykırı değerler tespit edildi ve tespit edilen aykırı değerlerin veri setlerinin sınıflandırma başarısı üzerindeki etkisi bu bölümde tablolar halinde gösterilmektedir.

Çizelge 4.6.'da veri setlerinin ilk hallerinin DVM (destek vektör makineleri-support vektor machine), ÇKA (çok katmanlı algılayıcılar-multilayer perceptron) ve KNN (k en yakın komşuluk-k nearest neighbor) sınıflandırma algoritmaları kullanılarak elde edilen accuracy değerleri bulunmaktadır. Accuracy değerleri, verilerin sınıflandırması esnasında doğru olarak yapılan tahminlerin bütün tahminlere oranıdır.

Uygulamada elde edilen accuracy değerleri DVM, ÇKA ve KNN sınıflandırma algoritmalarında önemli test seçenekleri olan cross validation (çapraz doğrulama) ve percentage split (belli bir orana göre bölme) için ayrı hesaplanmaktadır.

Cross validation belirlenen değerine göre veriyi kümelere ayırmaktadır ve ilk kümeyi test için diğer kümeleri ise eğitim için kullanmaktadır. Bu çalışmada cross validation değeri için 10 kullanılmaktadır. Percentage split ise belirlenen değeri kadar veriyi test için geri kalanını ise eğitim için kullanmaktadır. Bu çalışmada percentage split değeri için %66 kullanılmaktadır.

Çizelge 4.6.. Veri setlerinin sınıflandırma başarısındaki değerleri

	İRİS	BUPA	BLOOD	HABERMAN	YEAST
DVM-C	%89.33	%58.26	%76.20	%73.52	%57.34
DVM-P	%94.11	%63.24	%74.80	%75	%55.84
ÇKA-C	%94	%71.59	%78.20	%69.28	%59.02
ÇKA-P	%96.07	%65.81	%79.92	%63.46	%60.39
KNN-C	%90.66	%62.89	%70.45	%68.30	%52.29
KNN-P	%94.11	%64.95	%70.07	%61.53	%52.27

GA, DGA ve PSO için ayrı ayrı K küme sayısı 3, 5, 7 ile T iterasyon sayısı 100 ve 200 ile npop popülasyon sayısı 20 ve 50 ile oluşan 12 farklı kombinasyonda elde edilen aykırı değerler veri setlerinin orijinal hallerinden silinerek veri setlerinin yeni halleriyle sınıflandırma yapılmaktadır. Sınıflandırmada kullanılan DVM, ÇKA ve KNN sınıflandırma algoritmaları için cross validation ve percentage split test seçeneklerine göre accuracy değerleri elde edilmektedir.

Aykırı değerlerin silinmiş olduğu veri setlerinin yeni halleriyle elde edilen accuracy değerleri her bir veri seti için aşağıda tablolar halinde gösterilmektedir.

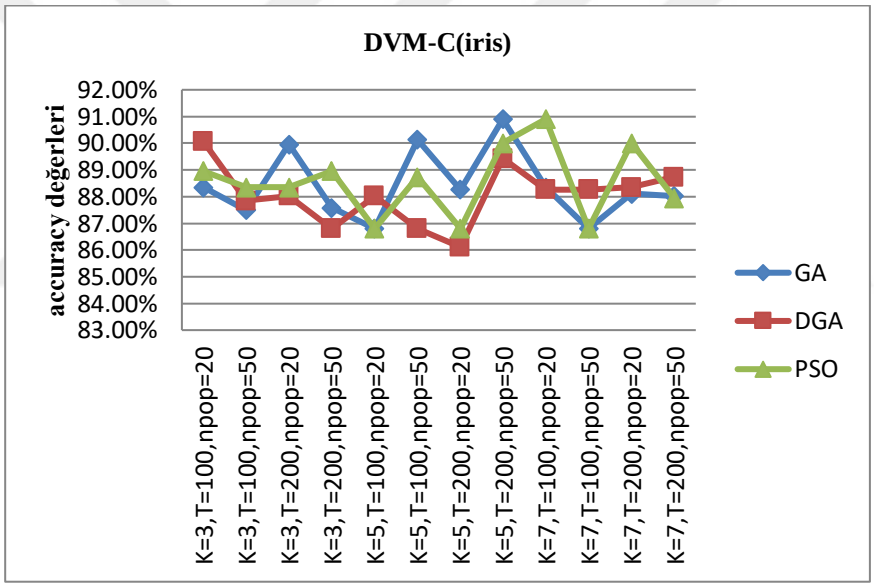
Çizelge 4.7., iris veri setinden silinen aykırı değerlerin sayısını ve bu aykırı değerler silindikten sonra veri setinin sınıflandırma algoritmalarıyla elde edilen accuracy değerlerini göstermektedir.

Çizelge 4.7. İris veri seti için elde edilen accuracy değerleri

İRİS	T/npop	T=100 npop=20			T=100 npop=50			T=200 npop=20			T=200 npop=50		
Küme	Alg.	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO
K=3	Outlier	4	9	5	6	10	4	5	8	4	5	6	5
	DVM-C	%88.35	%90.07	%88.96	%87.5	%87.85	%88.35	%89.96	%88.02	%88.35	%87.58	%86.80	%88.96
	DVM-P	%96	%85.41	%89.79	%89.79	%83.33	%96	%89.79	%95.83	%96	%89.78	%85.71	%89.79
	ÇKA-C	%95.89	%93.61	%96.55	%94.44	%94.28	%95.89	%96.55	%94.36	%95.89	%93.79	%94.44	%96.55
	ÇKA-P	%100	%91.66	%95.91	%95.91	%97.91	%100	%95.91	%97.91	%100	%97.95	%91.83	%95.91
	KNN-C	%91.78	%91.48	%93.10	%90.97	%90	%91.78	%93.10	%90.84	%91.78	%91.72	%91.66	%93.10
	KNN-P	%96	%95.83	%93.87	%91.83	%95.83	%96	%93.87	%97.91	%96	%93.87	%91.83	%93.87
K=5	Outlier	6	8	6	8	6	8	5	6	6	7	8	10
	DVM-C	%86.80	%88.02	%86.80	%90.14	%86.80	%88.73	%88.27	%86.11	%86.80	%90.90	%89.43	%90
	DVM-P	%87.85	%95.83	%87.75	%95.83	%87.85	%85.41	%89.79	%87.75	%87.75	%87.75	%95.83	%89.58
	ÇKA-C	%95.83	%94.36	%95.83	%94.36	%96.52	%94.36	%92.41	%96.52	%95.83	%95.10	%94.36	%95.71
	ÇKA-P	%93.87	%97.91	%93.87	%97.91	%91.83	%87.5	%95.91	%93.87	%93.87	%89.79	%97.91	%97.91
	KNN-C	%94.44	%90.84	%94.44	%91.54	%91.66	%91.54	%91.03	%90.97	%94.44	%92.30	%91.54	%91.42
	KNN-P	%91.83	%97.91	%91.83	%95.83	%93.87	%91.66	%93.87	%93.87	%91.83	%85.71	%97.91	%93.75
K=7	Outlier	4	5	7	6	5	6	7	4	10	8	8	9
	DVM-C	%88.35	%88.27	%90.90	%86.80	%88.27	%86.80	%88.11	%88.35	%90	%88.02	%88.73	%87.94
	DVM-P	%96	%89.79	%87.75	%87.85	%85.71	%87.75	%91.83	%96	%89.58	%95.83	%89.58	%81.25
	ÇKA-C	%95.20	%92.41	%95.10	%95.83	%95.86	%95.83	%94.40	%95.89	%95.71	%93.62	%94.36	%94.32
	ÇKA-P	%100	%95.91	%89.79	%93.87	%97.95	%93.87	%91.83	%100	%97.91	%97.91	%87.5	%91.66
	KNN-C	%92.46	%91.03	%92.30	%94.44	%93.10	%94.44	%91.60	%91.78	%91.42	%92.95	%92.25	%91.48
	KNN-P	%98	%93.87	%85.71	%91.83	%89.79	%91.83	%91.83	%96	%93.75	%97.91	%91.66	%89.58

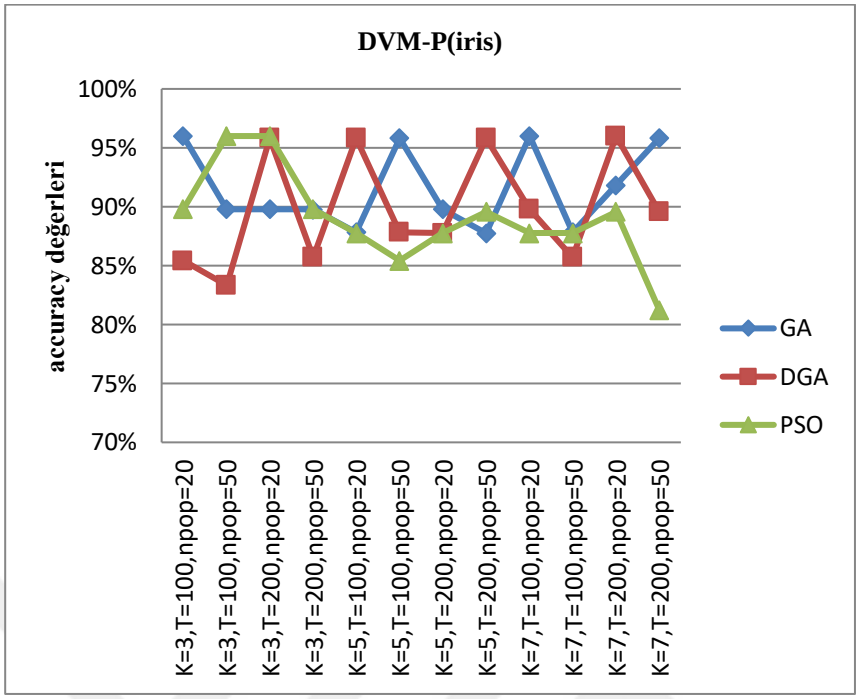
İris veri setinin ilk haliyle elde edilen Çizelge 4.6.'da bulunan accuracy değerleriyle iris veri setinin sonraki haliyle elde edilen Çizelge 4.7.'de bulunan accuracy değerleri karşılaştırıldığında aşağıdaki gibi değerlendirmeler yapılabilmektedir.

-DVM-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %89.33 iken aynı sınıflandırma algoritması, GA algoritması K=5, T=200, npop=50 için ve PSO algoritması K=7, T=100, npop=20 için elde edilen veri setleriyle %90.90 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.16.'da çizgi grafiği ile gösterilmektedir.



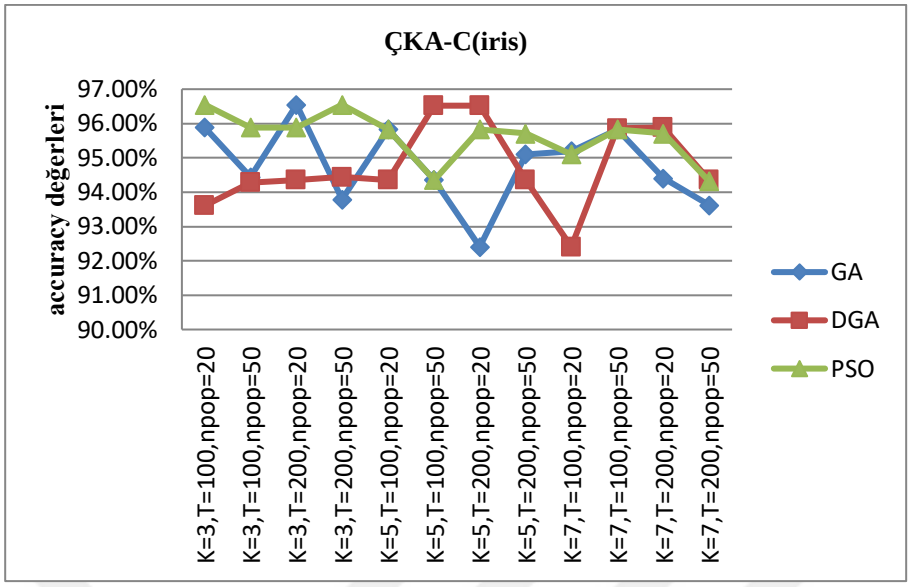
Şekil 4.16. İris veri setinin DVM-C ile elde edilen accuracy değerleri

- DVM-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %94.11 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop= 50 ve K=3, T=200, npop=20 için GA algoritması ise K=3, T=100, npop=20 ve K=7, T=100, npop=20 için elde edilen veri setleriyle %96 değeri bulunmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.17.'de çizgi grafiği ile gösterilmektedir.



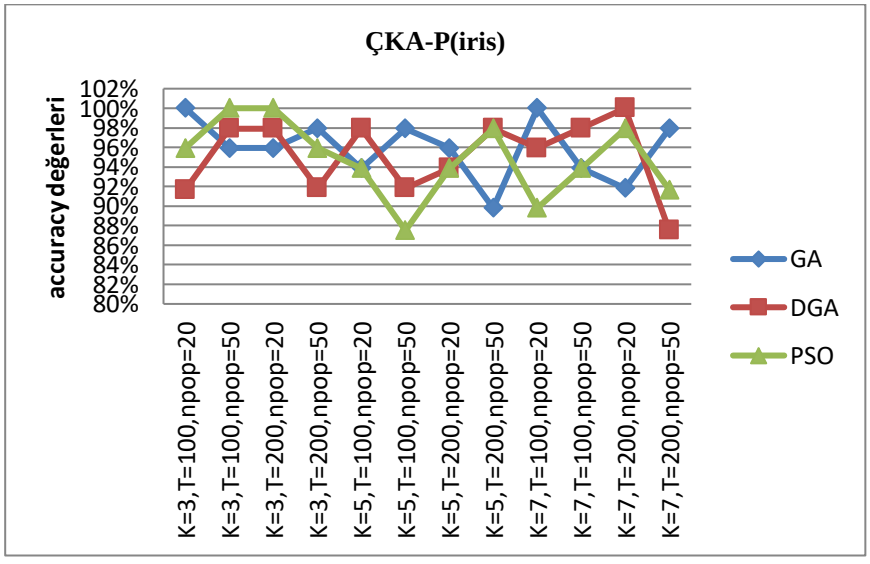
Şekil 4.17. İris veri setinin DVM-P ile elde edilen accuracy değerleri

- ÇKA-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %94 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop=20 ve K=3, T=200, npop=50 için GA algoritması da K=3, T=200, npop=20 için elde edilen veri setleriyle %96.55 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=5, T=100, npop=50 için de elde edilen veri setiyle %96.52 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.18.'de çizgi grafiği ile gösterilmektedir.



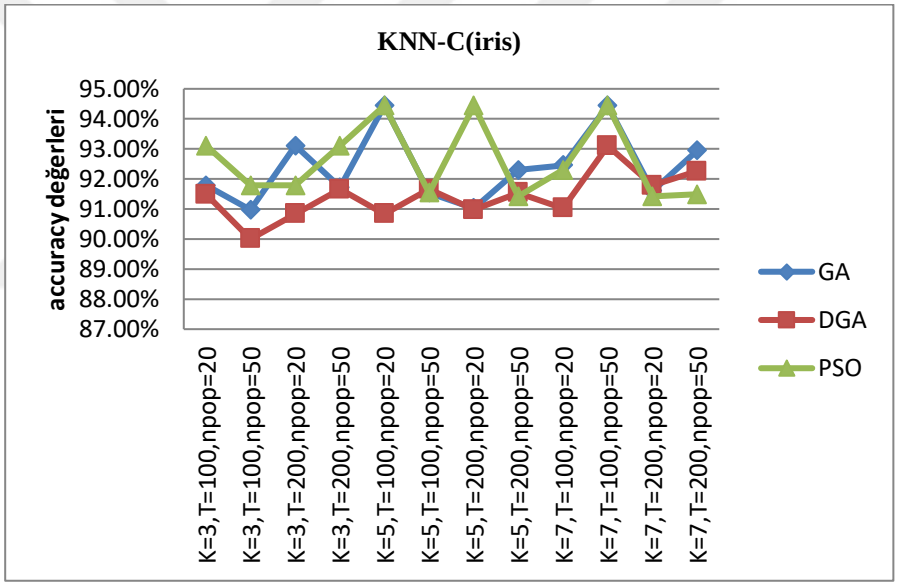
Şekil 4.18. İris veri setinin ÇKA-C ile elde edilen accuracy değerleri

-ÇKA-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %96.07 iken aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=20 ve K=7, T=100, npop=20 için PSO algoritması da K=3, T=100, npop=50 ve K=3, T=200, npop=20 için ve DGA algoritması K=7, T=200, npop=20 için elde edilen veri setleriyle %100 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.19.'da çizgi grafiği ile gösterilmektedir.



Şekil 4.19. İris veri setinin ÇKA-P ile elde edilen accuracy değerleri

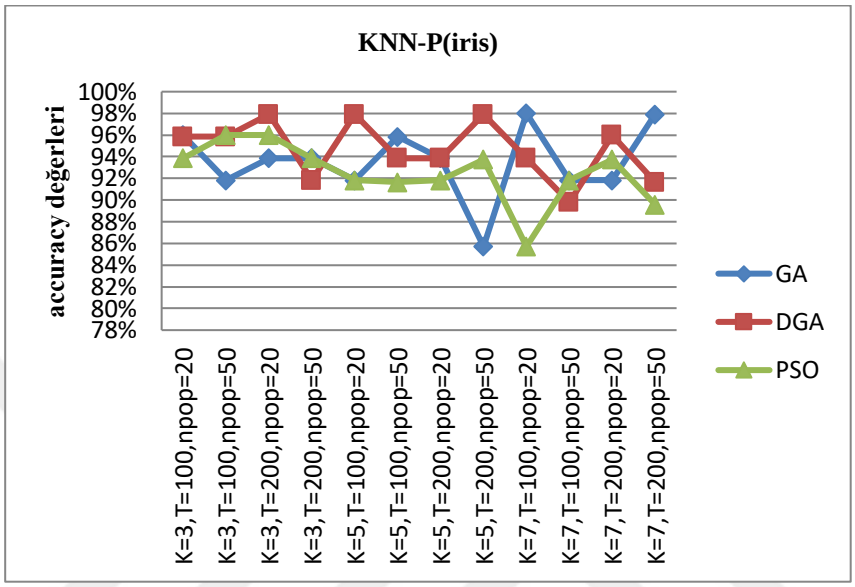
-KNN-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %90.66 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop=20 ve K=3, T=200, npop=50 için GA algoritması da K=3, T=200, npop=20 için elde edilen veri setleriyle %93.10 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=5, T=100, npop= 20 ve K=7, T=100, npop=50 için ve PSO algoritması da K=5, T=100, npop=20, K=5, T=200, npop=20 ve K=7, T=100, npop=50 için elde edilen veri setleriyle %94.44 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.20'de çizgi grafiği ile gösterilmektedir.



Şekil 4.20. İris veri setinin KNN-C ile elde edilen accuracy değerleri

-KNN-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %94.11 iken aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=20 için PSO algoritması K=3, T=100, npop=50 ve K=3, T=200, npop=20 için DGA algoritması da K=7, T=200, npop=20 için elde edilen veri setleriyle %96 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=7, T=200, npop=50 için DGA algoritması da K=3, T=200, npop=20 ve K=5, T=200, npop=50 için elde edilen veri setleriyle %97.91 değerini bulmaktadır. Aynı

sınıflandırma algoritması GA algoritması K=7, T=100, npop=20 için elde edilen veri setiyle de %98 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.21.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.21. İris veri setinin KNN-P ile elde edilen accuracy değerleri

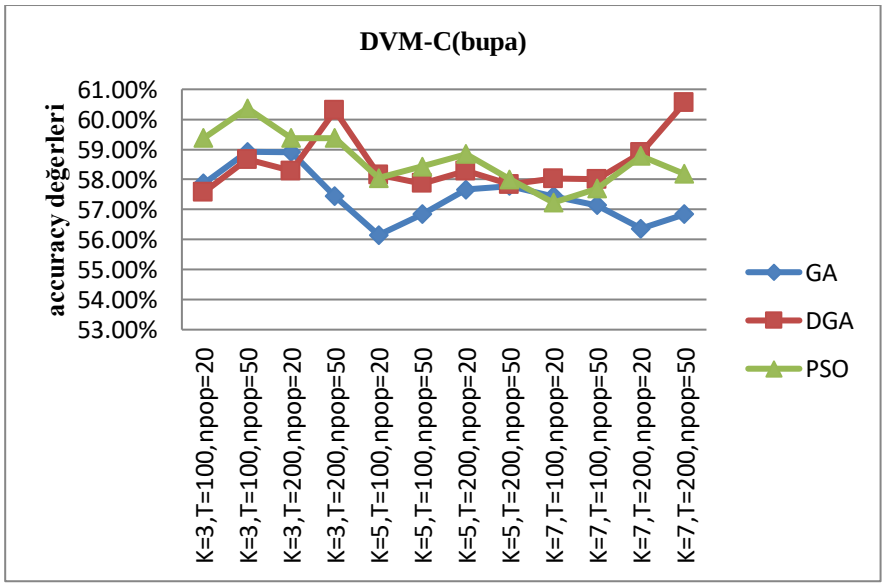
Çizelge 4.8., bupa veri setinden silinen aykırı değerlerin sayısını ve bu aykırı değerler silindikten sonra veri setinin sınıflandırma algoritmalarıyla elde edilen accuracy değerlerini göstermektedir.

Çizelge 4.8. Bupa veri seti için elde edilen accuracy değerleri

BUPA	T/npop	T=100 npop=20			T=100 npop=50			T=200 npop=20			T=200 npop=50		
Küme	Alg.	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO
K=3	Outlier	15	15	20	9	16	22	14	19	20	16	20	20
	DVM-C	%57.87	%57.57	%59.38	%58.92	%58.66	%60.37	%58.91	%58.28	%59.38	%57.44	%60.30	%59.38
	DVM-P	%60.71	%60.71	%53.63	%61.40	%51.78	%62.72	%53.98	%51.35	%53.63	%51.78	%52.72	%53.63
	ÇKA-C	%69.09	%70.60	%68.92	%66.96	%72.03	%66.56	%66.46	%70.55	%68.92	%70.51	%71.69	%68.92
	ÇKA-P	%73.21	%71.42	%73.63	%70.17	%66.96	%70.90	%64.60	%72.07	%73.63	%60.71	%71.81	%73.63
	KNN-C	%58.78	%62.12	%64.92	%58.63	%64.43	%64.39	%63.44	%65.03	%64.92	%63.52	%64	%64.92
	KNN-P	%56.25	%66.07	%67.27	%62.28	%66.07	%65.45	%59.29	%63.96	%67.27	%57.14	%67.27	%67.27
K=5	Outlier	12	20	16	10	15	13	19	19	12	11	13	14
	DVM-C	%56.15	%58.15	%58.05	%56.84	%57.87	%58.43	%57.66	%58.28	%58.85	%57.78	%57.83	%58
	DVM-P	%53.98	%55.45	%52.67	%61.40	%55.35	%61.94	%54.05	%53.15	%57.52	%62.28	%60.17	%54.86
	ÇKA-C	%67.56	%70.46	%71.42	%68.75	%69.69	%70.18	%67.17	%69.63	%70.87	%68.26	%66.26	%69.78
	ÇKA-P	%65.48	%71.81	%63.92	%66.66	%66.07	%74.33	%59.45	%65.76	%68.14	%69.29	%74.33	%71.68
	KNN-C	%60.66	%64.61	%62.31	%61.90	%60.90	%64.15	%62.88	%60.42	%60.91	%61.97	%58.13	%64.35
	KNN-P	%53.98	%65.45	%68.75	%61.40	%60.71	%55.75	%67.56	%72.07	%60.17	%59.64	%66.37	%64.60
K=7	Outlier	9	21	20	9	14	14	15	19	15	9	18	15
	DVM-C	%57.44	%58.02	%57.23	%57.14	%58	%57.70	%56.36	%58.89	58.78	%56.84	%60.55	%58.18
	DVM-P	%62.28	%55.45	%53.63	%61.40	%53.09	%54.86	%55.35	%55.85	%62.5	%62.28	%74.77	%56.25
	ÇKA-C	%70.53	%68.20	%70.15	%67.55	%67.97	%70.99	%69.39	%67.79	%73.63	%71.13	%71.55	%69.39
	ÇKA-P	%68.42	%63.63	%64.54	%71.92	%72.56	%67.75	%70.58	%65.76	%62.5	%71.92	%70.27	%65.17
	KNN-C	%60.11	%60.80	%63.38	%60.71	%63.24	%59.51	%63.93	%62.88	%63.03	%61.01	%63.60	%62.12
	KNN-P	%54.38	%70	%68.18	%62.28	%56.63	%62.83	%66.07	%67.56	%58.92	%64.03	%63.96	%54.46

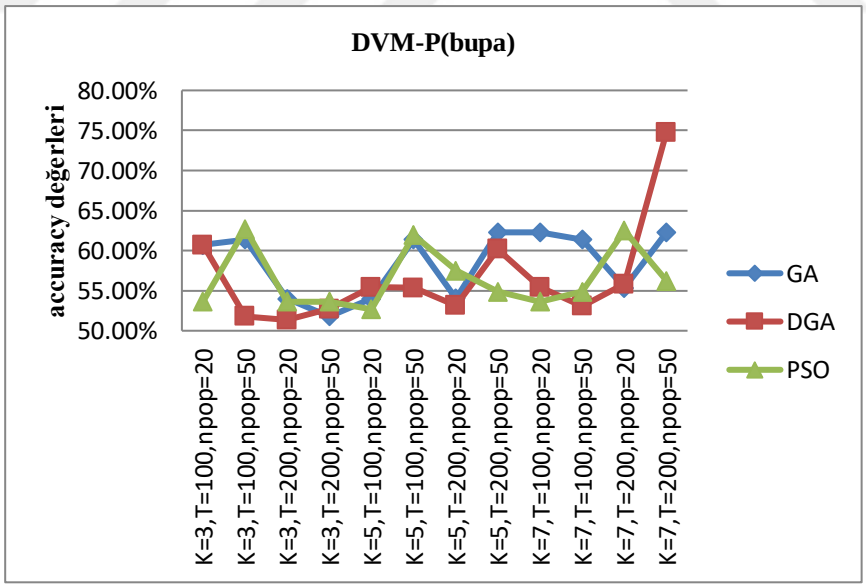
Bupa veri setinin ilk haliyle elde edilen Çizelge 4.6.'da bulunan accuracy değerleriyle bupa veri setinin sonraki haliyle elde edilen Çizelge 4.8.'de bulunan accuracy değerleri karşılaştırıldığında aşağıdaki gibi değerlendirmeler yapılabilmektedir.

- DVM-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %58.26 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=200, npop=20 , K=3, T=100, npop=20 ve K=3, T=200, npop=50 için elde edilen veri setleriyle %59.38 değerini ve K=3, T=100, npop=50 için elde edilen veri setiyle %60.37 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=200, npop=50 için elde edilen veri setiyle %60.30 değerini ve K=7, T=200, npop=50 için de elde edilen veri setiyle %60.55 değerini bulmaktadır. GA algoritmasıyla elde edilen veri setleriyle ise veri setinin orijinal halinden daha iyi bir değer elde edilmemektedir. GA algoritmasıyla 12 durum için elde edilen veri setleriyle aynı sınıflandırma algoritması ortalama %57.44 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.22.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.22. Bupa veri setinin DVM-C elde edilen accuracy değerleri

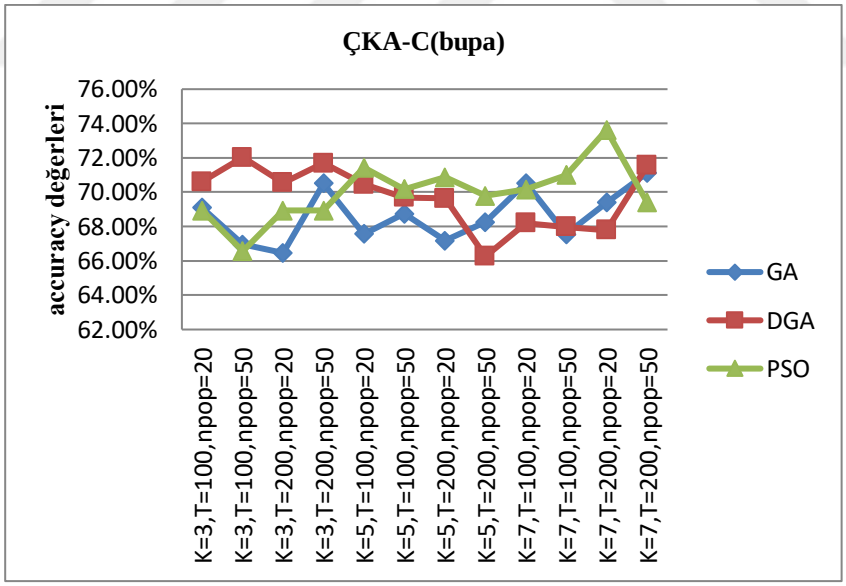
-DVM-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %63.24 iken aynı sınıflandırma algoritması DGA algoritması K=7, T=200, npop=50 için elde edilen veri setiyle %74.77 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %56.96, K=5 için 4 durumda elde edilen veri setleriyle ortalama %57.92 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %60.32 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %54.14, K=5 için 4 durumda elde edilen veri setleriyle ortalama %56.03 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %59.79 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %55.90, K=5 için 4 durumda elde edilen veri setleriyle ortalama %56.74 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %56.81 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.23.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.23. Bupa veri setinin DVM-P elde edilen accuracy değerleri

-ÇKA-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %71.59 iken aynı sınıflandırma algoritması DGA K=3, T=100,

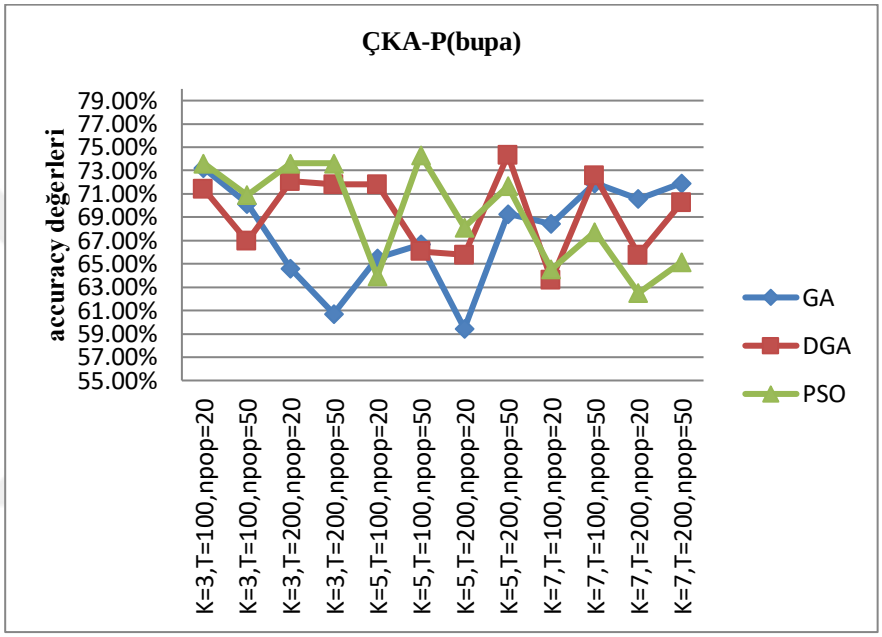
npop=50 için elde edilen veri setiyle %72.03 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=7, T=200, npop=20 için elde edilen veri setiyle %73.63 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %68.25, K=5 için4 durumda elde edilen veri setleriyle ortalama %67.93 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %69.65 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %71.21, K=5 için 4 durumda elde edilen veri setleriyle ortalama %69.01 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %68.92 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %68.33, K=5 için 4 durumda elde edilen veri setleriyle ortalama %70.56 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %71.04 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.24.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.24. Bupa veri setinin ÇKA-C elde edilen accuracy değerleri

-ÇKA-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %65.81 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop=20, K=3, T=200, npop=20 ve K=3, T=200, npop=50 için elde edilen

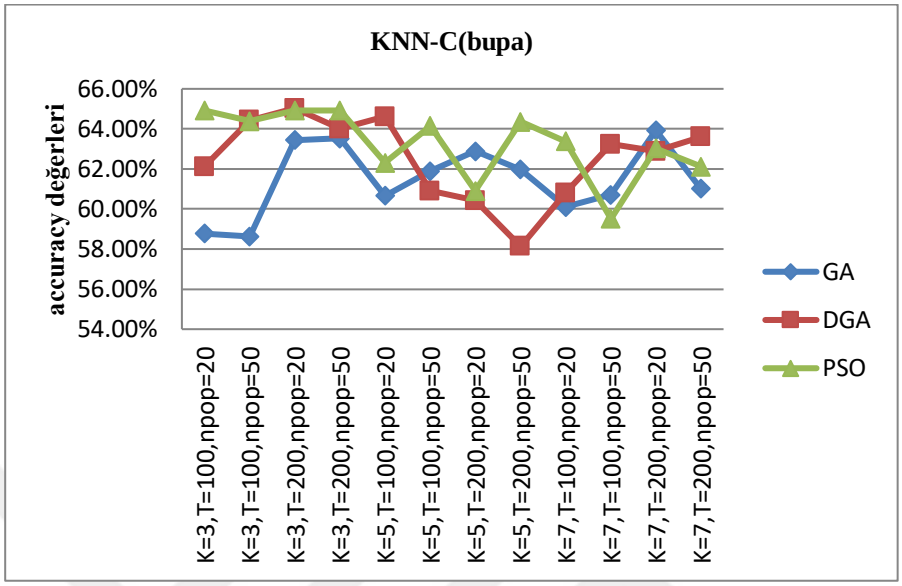
veri setleriyle %73.63 değerini ve K=5, T=100, npop=50 için elde edilen veri setiyle %74.33 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=5, T=200, npop=50 için elde edilen veri setiyle %74.33 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=20 için elde edilen veri setiyle %73.21 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.25.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.25. Bupa veri setinin ÇKA-P elde edilen accuracy değerleri

-KNN-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %62.89 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop=20, K=3, T=200, npop=20 ve K=3, T=200, npop=50 için elde edilen veri setleriyle %64.92 değerini ve K=3, T=100, npop=50 için elde edilen veri setiyle %64.39 değerini ve K=5, T=100, npop=50 için elde edilen veri setiyle %64.15 değerini ve K=5, T=200, npop=50 için de elde edilen veri setiyle %64.35 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=100, npop=50 için elde edilen veri setiyle %64.43 değerini ve K=3, T=200, npop=20 için elde edilen veri setiyle %65.03 değerini ve K=5, T=100, npop=20 için de elde edilen veri setiyle %64.61 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde

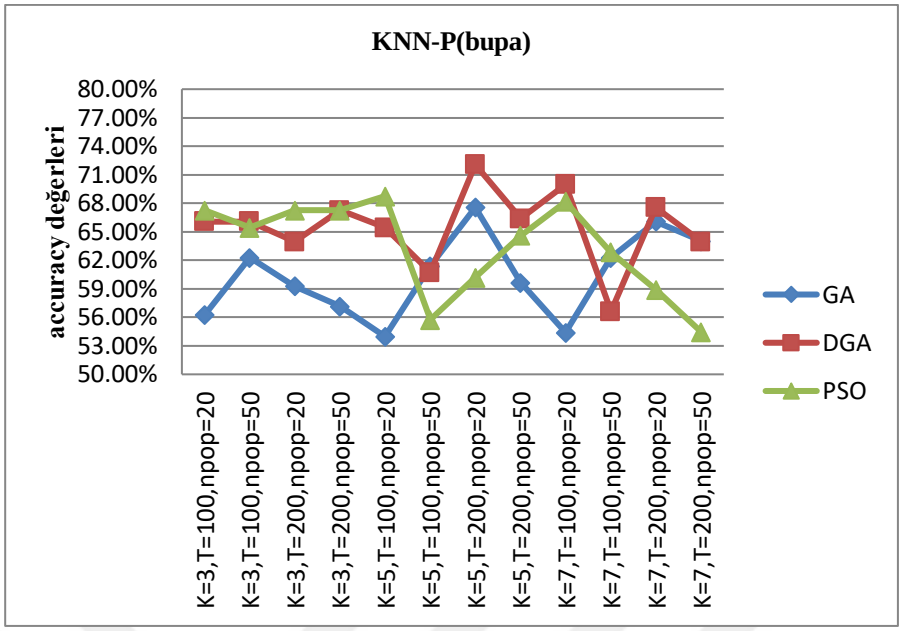
edilen bütün veri setlerinin KNN-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.26.'da çizgi grafiği ile gösterilmektedir.



Şekil 4.26. Bupa veri setinin KNN-C elde edilen accuracy değerleri

- KNN-P sınıflandırma algoritmasıyla veri setinin ilk halinde elde edilen accuracy değeri %64.95 iken aynı sınıflandırma algoritması PSO algoritması K=3, T=100, npop=20, K=3, T=200, npop=20 ve K=3, T=200, npop=50 için elde edilen veri setleriyle %67.27 değerini ve K=5, T=100, npop=20 için elde edilen veri setiyle %68.75 değerini ve K=7, T=100, npop=20 için de elde edilen veri setiyle %68.18 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=5, T=200, npop=20 için elde edilen veri setiyle %72.07 değerini ve K=7, T=100, npop=20 için de elde edilen veri setiyle %70 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.27.'de çizgi grafiği ile gösterilmektedir

.



Şekil 4.27. Bupa veri setinin KNN-P elde edilen accuracy değerleri

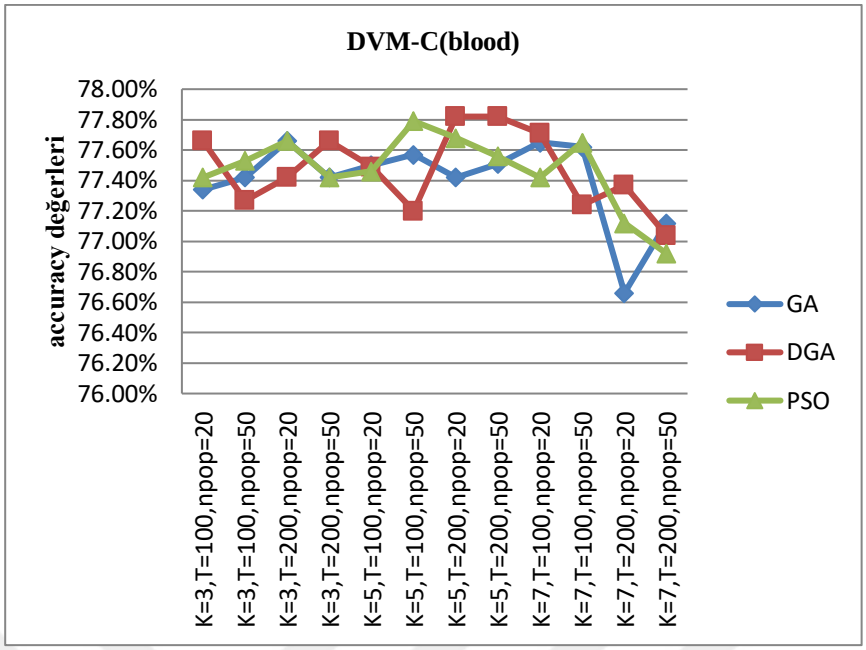
Çizelge 4.9., blood veri setinden silinen aykırı değerlerin sayısını ve bu aykırı değerler silindikten sonra veri setinin sınıflandırma algoritmalarıyla elde edilen accuracy değerlerini göstermektedir.

Çizelge 4.9. Blood veri seti için elde edilen accuracy değerleri

Blood	T/npop	T=100 npop=20			T=100 npop=50			T=200 npop=20			T=200 npop=50		
Küme	Alg.	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO
K=3	Outlier	55	36	26	26	22	27	36	26	36	26	36	26
	DVM-C	%77.34	%77.66	%77.42	%77.42	%77.27	%77.53	%77.66	%77.42	%77.66	%77.42	%77.66	%77.42
	DVM-P	%75	%78.51	%77.95	%77.95	%77.73	%76.73	%78.51	%77.95	%78.51	%77.95	%78.51	%77.95
	ÇKA-C	%80.37	%78.51	%79.08	%79.50	%78.92	%79.47	%78.51	%79.50	%78.51	%79.50	%78.51	%79.50
	ÇKA-P	%77.11	%80.16	%77.55	%78.36	%77.73	%77.55	%80.16	%78.36	%80.16	%78.36	%80.16	%78.36
	KNN-C	%72.58	%73.73	%73.26	%73.13	%72.45	%73.37	%73.73	%73.13	%73.73	%73.13	%73.23	%73.13
	KNN-P	%67.79	%76.44	%71.42	%72.24	%69.63	%75.10	%76.44	%72.24	%76.44	%72.24	%76.44	%72.24
K=5	Outlier	28	46	38	39	56	59	48	58	58	41	58	57
	DVM-C	%77.5	%77.49	%77.46	%77.57	%77.20	%77.79	%77.42	%77.82	%77.68	%77.51	%77.82	%77.56
	DVM-P	%80	%76.56	%78.00	%78.83	%75	%75.64	%74.78	%85.53	%83.40	%76.66	%82.97	%75.31
	ÇKA-C	%77.36	%79.05	%79.43	%79.26	%79.36	%78.95	%79.57	%78.98	%78.98	%79.20	%79.27	%78.87
	ÇKA-P	%77.59	%76.98	%78.42	%80.91	%75.84	%75.64	%78.15	%86.80	%85.95	%76.25	%84.68	%76.17
	KNN-C	%71.94	%71.65	%72.81	%73.62	%71.86	%71.26	%70.71	%70.72	%70.72	%71.71	%72.02	%71.49
	KNN-P	%73.87	%71.54	%74.68	%71.78	%70.76	%70.51	%71.42	%72.34	%72.34	%68.75	%71.48	%69.78
K=7	Outlier	41	48	26	42	45	41	11	32	18	18	16	16
	DVM-C	%77.65	%77.71	%77.42	%77.62	%77.24	%77.65	%76.66	%77.37	%77.12	%77.12	%77.04	%76.92
	DVM-P	%78.33	%76.89	%81.63	%78.75	%78.24	%78.33	%75.69	%74.48	%76.61	%76.61	%78.31	%77.51
	ÇKA-C	%78.64	%78.71	%79.08	%79.45	%78.23	%79.06	%78.15	%78.77	%78.76	%78.76	%78.27	%78.55
	ÇKA-P	%79.16	%78.57	%82.44	%83.75	%77.82	%78.33	%76.99	%79.01	%78.62	%78.62	%80.32	%80.32
	KNN-C	%71.42	%72.42	%71.88	%72.09	%72.97	%72.56	%71.23	%71.92	%70.68	%70.68	%71.17	%71.58
	KNN-P	%71.66	%71.42	%73.87	%72.08	%74.05	%75	%72.11	%69.13	%71.37	%71.37	%74.29	%73.49

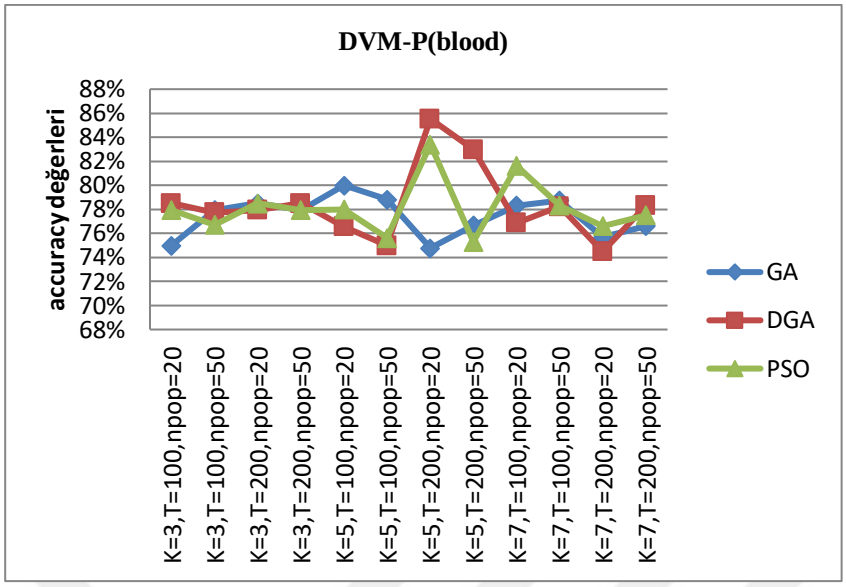
Blood veri setinin ilk haliyle elde edilen Çizelge 4.6.'da bulunan accuracy değerleriyle blood veri setinin sonraki haliyle elde edilen Çizelge 4.9.'da bulunan accuracy değerleri karşılaştırıldığında aşağıdaki gibi değerlendirmeler yapılabilmektedir.

-DVM-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %76.20 iken aynı sınıflandırma algoritması GA, DGA ve PSO için 12 farklı durumda elde edilen veri setleriyle birbirlerine yakın değerler bulmaktadırlar. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %77.46, K=5 için 4 durumda elde edilen veri setleriyle ortalama %77.5 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %77.26 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %77.50, K=5 için4 durumda elde edilen veri setleriyle ortalama %77.58 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %77.34 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %77.50, K=5 için 4 durumda elde edilen veri setleriyle ortalama %77.62 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %77.27 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.28.'de çizgi grafiği ile gösterilmektedir.



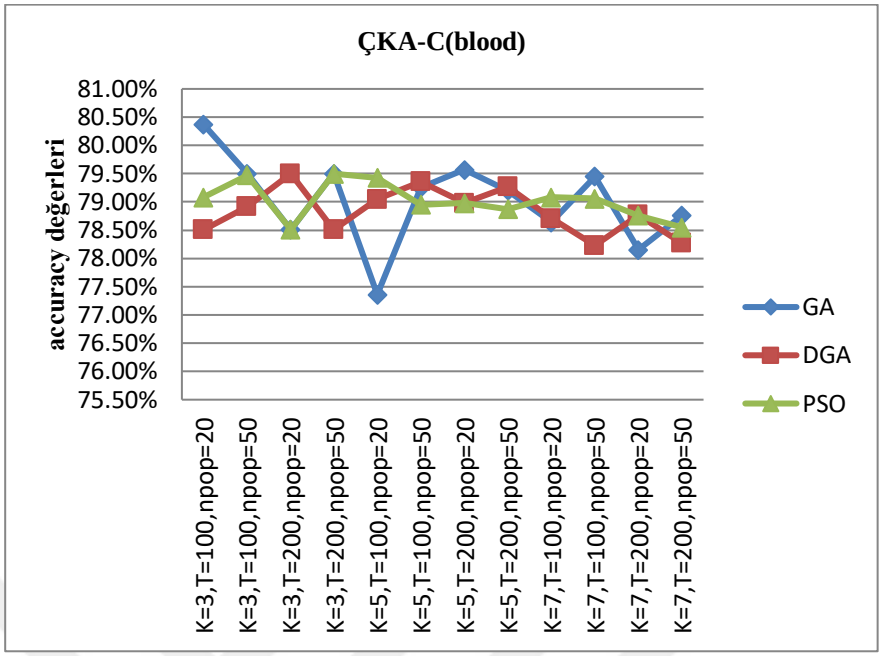
Şekil 4.28. Blood veri setinin DVM-C elde edilen accuracy değerleri

- DVM-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %74.80 iken aynı sınıflandırma algoritması GA algoritması K=5, T=100, npop=20 için elde edilen veri setiyle %80 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=5, T=200, npop=20 için elde edilen veri setiyle %85.53 değerini ve K=5, T=200, npop=50 için elde edilen veri setiyle %82.97 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=5, T=200, npop=20 için elde edilen veri setiyle %83.40 değerini ve K=7, T=100, npop=20 için elde edilen veri setiyle %81.63 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.29.'da çizgi grafiği ile gösterilmektedir.



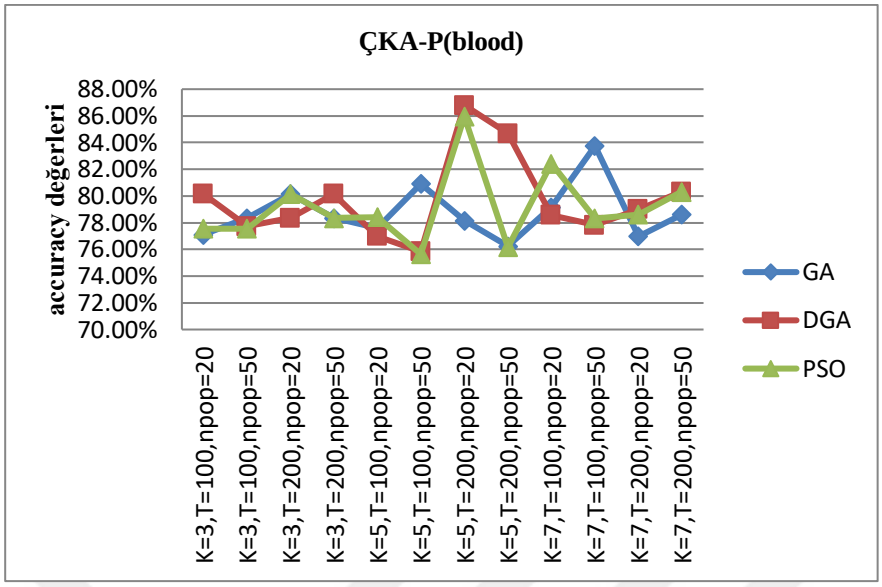
Şekil 4.29. Blood veri setinin DVM-P elde edilen accuracy değerleri

- ÇKA-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %78.20 iken aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=20 için elde edilen veri setiyle %80.37 değerini, DGA algoritması K=3, T=200, npop=20 için elde edilen veri setiyle %79.50 değerini ve PSO algoritması K=3, T=200, npop=50 için elde edilen veri setiyle de %79.50 değerini bulmaktadır. Aynı sınıflandırma algoritması her üç algoritmayla 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %79.47, K=5 için 4 durumda elde edilen veri setleriyle ortalama %78.84 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %78.75 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %78.86, K=5 için 4 durumda elde edilen veri setleriyle ortalama %79.16 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %78.37 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %79.14, K=5 için 4 durumda elde edilen veri setleriyle ortalama %79.05 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %78.86 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.30.'da çizgi grafiği ile gösterilmektedir.



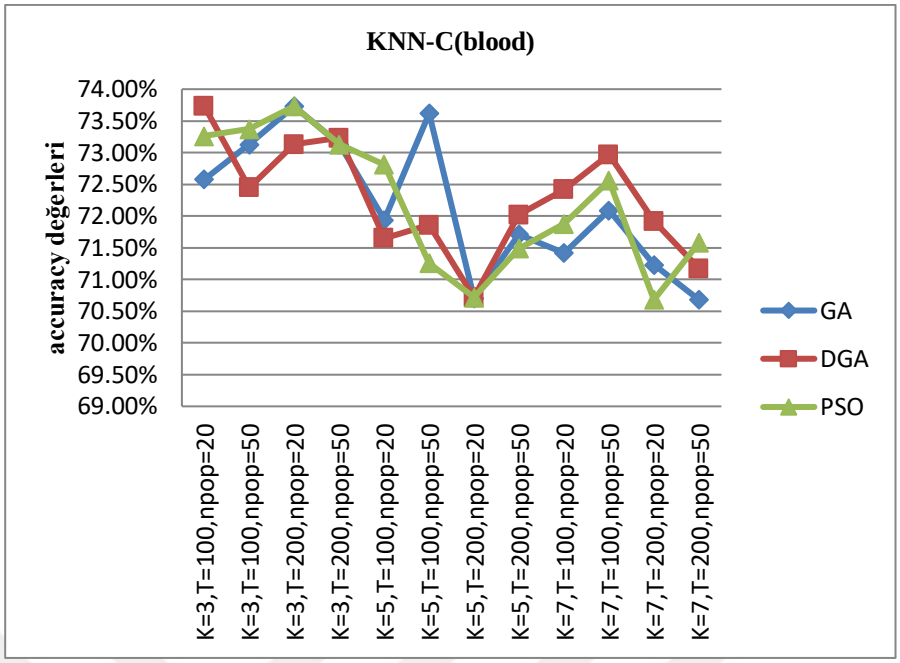
Şekil 4.30. Blood veri setinin ÇKA-C elde edilen accuracy değerleri

- ÇKA-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %79.92 iken aynı sınıflandırma algoritması GA algoritması K=7, T=100, npop=50 için elde edilen veri setiyle %83.75 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=5, T=200, npop=20 için elde edilen veri setiyle %86.80 değerini K=5, T=200, npop=50 için elde edilen veri setiyle %84.68 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=5, T=200, npop=20 için elde edilen veri setiyle %85.95 değerini ve K=7, T=100, npop=20 için elde edilen veri setiyle %82.44 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.31.'de çizgi grafiği ile gösterilmektedir.



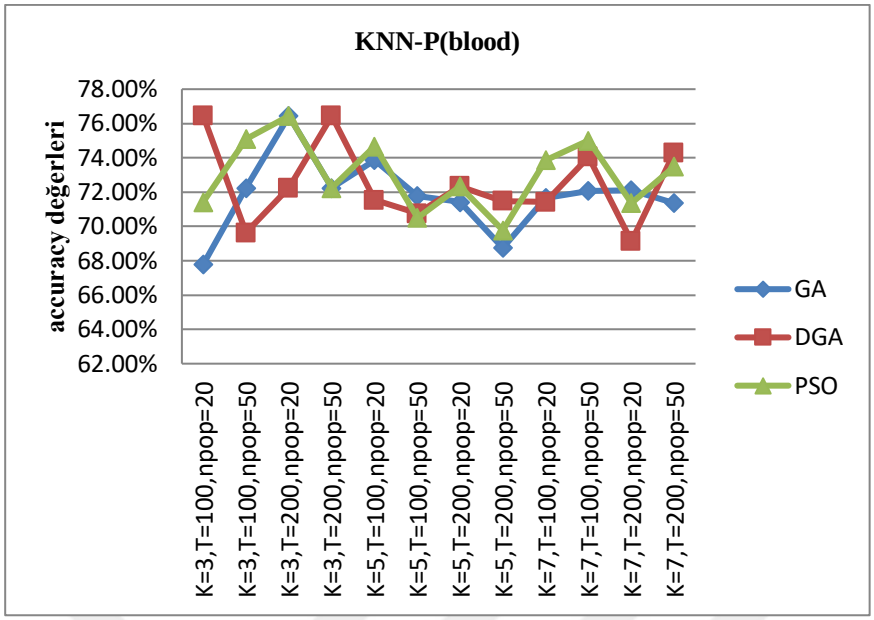
Şekil 4.31. Blood veri setinin ÇKA-P elde edilen accuracy değerleri

- KNN-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %70.45 iken aynı sınıflandırma algoritması her üç algoritma da 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %73.14, K=5 için 4 durumda elde edilen veri setleriyle ortalama %71.95 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %71.35 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %73.13, K=5 için 4 durumda elde edilen veri setleriyle ortalama %71.56 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %72.12 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %73.37, K=5 için 4 durumda elde edilen veri setleriyle ortalama %71.57 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %71.67 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.32.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.32. Blood veri setinin KNN-C elde edilen accuracy değerleri

- KNN-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %70.07 iken aynı sınıflandırma algoritması GA algoritması K=3, T=200, npop=20 için elde edilen veri setiyle %76.44 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=100, npop=20 ve K=3, T=200, npop=50 için elde edilen veri setleriyle %76.44 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3, T=200, npop=20 için elde edilen veri setiyle %76.44 değerini ve K=7, T=100, npop=50 için de elde edilen veri setiyle %75 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.33.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.33. Blood veri setinin KNN-P elde edilen accuracy değerleri

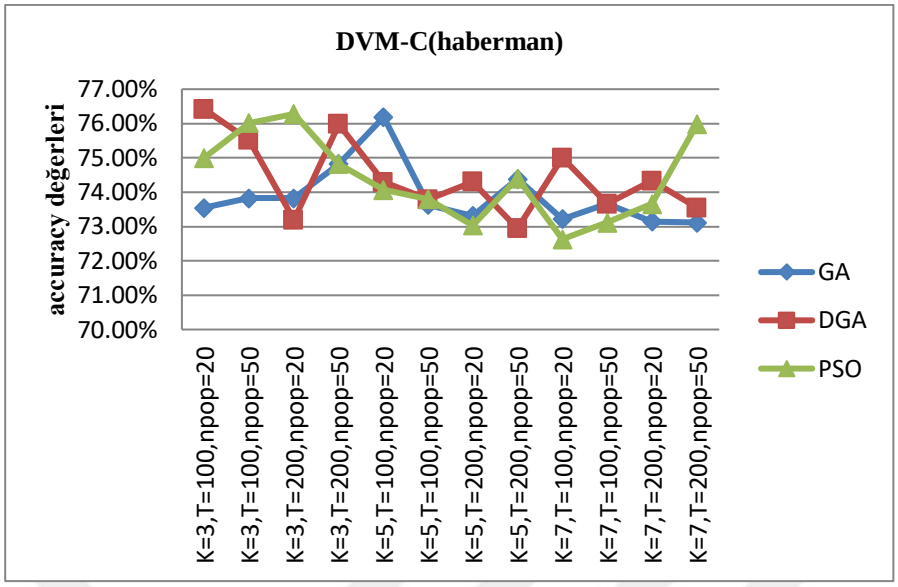
Çizelge 4.10., haberman veri setinden silinen aykırı değerlerin sayısını ve bu aykırı değerler silindikten sonra veri setinin sınıflandırma algoritmalarıyla elde edilen accuracy değerlerini göstermektedir.

Çizelge 4.10.. Haberman veri seti için elde edilen accuracy değerleri

Hab.	T/npop	T=100 npop=20			T=100 npop=50			T=200 npop=20			T=200 npop=50		
Küme	Alg.	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO
K=3	Outlier	11	26	13	27	16	14	27	28	14	12	27	12
	DVM-C	%73.55	%76.42	%75	%73.83	%75.51	%76.02	%73.83	%73.19	%76.28	%74.82	%75.98	%74.82
	DVM-P	%79	%76.84	%76.45	%81.05	%83.83	%71.64	%81.05	%75.75	%70.14	%83	%81.05	%82.35
	ÇKA-C	%70.84	%73.57	%70.98	%71.32	%68.62	%71.91	%71.32	%69.75	%72.50	%69.04	%70.25	%69.04
	ÇKA-P	%68	%68.42	%64	%80	%73.33	%76.11	%80	%74.74	%67.16	%76	%78.94	%73.52
	KNN-C	%67.79	%68	%70.98	%68.81	%66.55	%69.52	%68.81	%65.63	%69.41	%67	%68.45	%67
	KNN-P	%73	%70.52	%68.65	%71.57	%67.67	%61.19	%71.57	%66.66	%70.14	%73	%73.68	%79.41
K=5	Outlier	12	22	9	14	16	12	10	18	13	17	14	8
	DVM-C	%76.19	%74.29	%74.07	%73.63	%73.79	%73.80	%73.31	%74.30	%73.03	%74.39	%72.94	%74.40
	DVM-P	%81	%79.38	%72.05	%73.33	%77.77	%82	%75.24	%73.46	%81	%75.51	%75.75	%80
	ÇKA-C	%68.70	%68.30	%72.05	%69.86	%71.72	%74.48	%71.62	%72.56	%71.33	%71.97	%69.52	%72.35
	ÇKA-P	%76	%70.10	%63.23	%63.63	%66.66	%81	%78.21	%73.46	%77	%69.38	%69.69	%77
	KNN-C	%67.68	%69.71	%68.35	%69.52	%67.24	%71.76	%69.93	%70.83	%69.28	%69.20	%69.52	%70.64
	KNN-P	%75	%71.34	%67.32	%63.63	%71.71	%71	%71.28	%70.40	%71	%71.42	%64.64	%69
K=7	Outlier	11	22	10	6	6	10	8	6	12	8	11	26
	DVM-C	%73.22	%75	%72.63	%73.66	%73.66	%73.12	%73.15	%74.33	%73.66	%73.12	%73.55	%75.98
	DVM-P	%77	%83.50	%79.20	%72.52	%71.56	%80	%69.30	%71.56	%74.50	%81	%76	%81.05
	ÇKA-C	%73.55	%73.94	%70.27	%70.33	%66.66	%72.78	%70.13	%70.66	%72.66	%69.04	%71.18	%70.25
	ÇKA-P	%63	%63.91	%77.27	%71.56	%69.60	%74	%65.34	%74.50	%74.50	%78	%71	%78.94
	KNN-C	%69.83	%70.42	%69.93	%68.66	%68.66	%70.40	%66.77	%69.33	%69	%70.74	%67.45	%68.45
	KNN-P	%71	%72.16	%70.29	%61.76	%68.62	%69	%66.33	%66.66	%72.54	%75	%67	%73.68

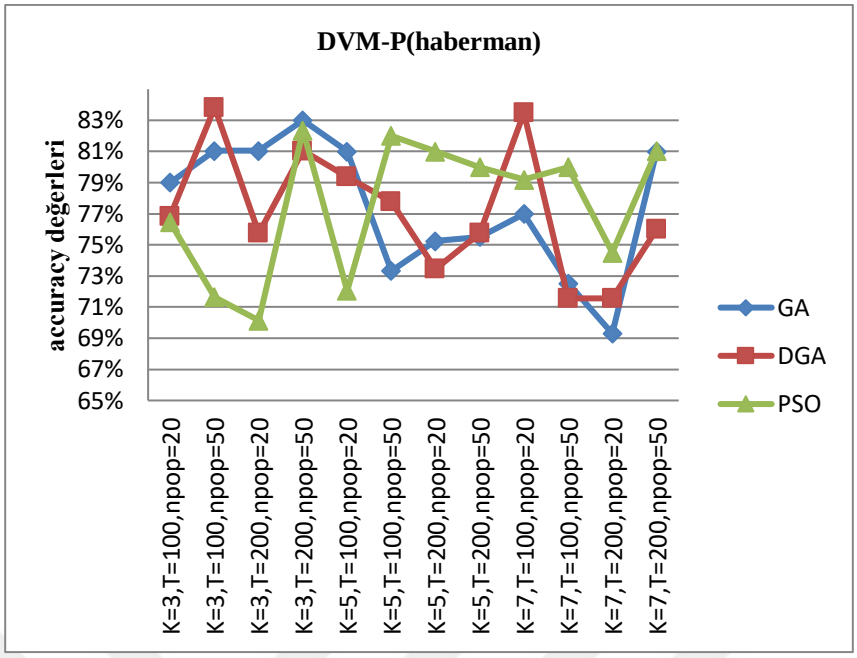
Haberman veri setinin ilk haliyle elde edilen Çizelge 4.6.'da bulunan accuracy değerleriyle haberman veri setinin sonraki haliyle elde edilen Çizelge 4.10.'da bulunan accuracy değerleri karşılaştırıldığında aşağıdaki gibi değerlendirmeler yapılabilmektedir.

- DVM-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %73.52 iken aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %74, K=5 için 4 durumda elde edilen veri setleriyle ortalama %74.38 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %73.28 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %75.27, K=5 için 4 durumda elde edilen veri setleriyle ortalama %73.83 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %74.13 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %75.53, K=5 için 4 durumda elde edilen veri setleriyle ortalama %73.82 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %73.84 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.34.'de çizgi grafiği ile gösterilmektedir.



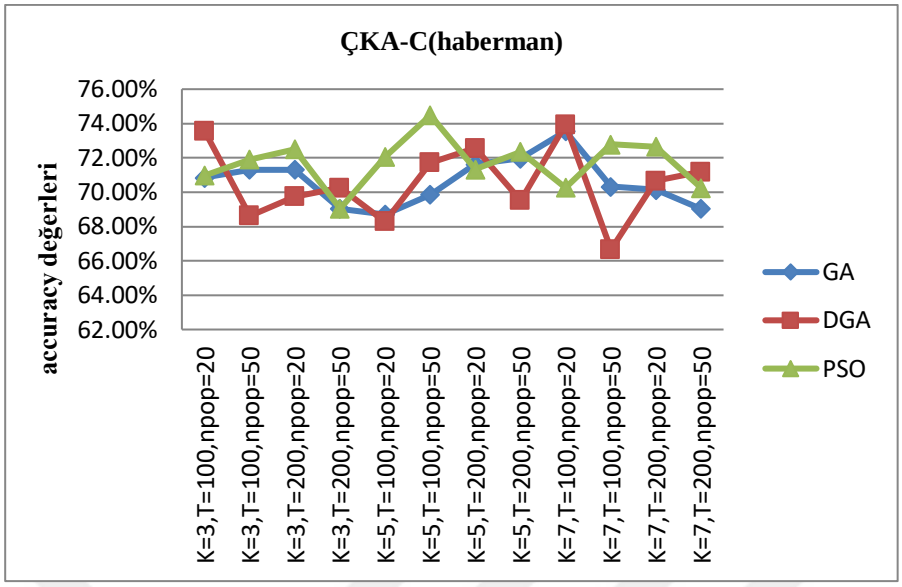
Şekil 4.34. Haberman veri setinin DVM-C elde edilen accuracy değerleri

- DVM-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %75 iken aynı sınıflandırma algoritması GA algoritması K=3, T=200, npop=50 için elde edilen veri setiyle %83 değerini K=3, T=100, npop=50 ve K=3, T=200, npop=20 için elde edilen veri setleriyle %81.05 değerini ve K=5, T=100, npop=20 için elde edilen veri setiyle %81 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=100, npop=50 için elde edilen veri setiyle %83.83 değerini ve K=3, T=200, npop=50 için de elde edilen veri setiyle %81.05 değerini K=7, T=100, npop=20 için de elde edilen veri setiyle %83.50 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3, T=200, npop=50 için elde edilen veri setiyle %82.35 değerini ve K=5, T=100, npop=50 için elde edilen veri setiyle %82 değerini ve K=5, T=200, npop=20 için de elde edilen veri setiyle %81 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.35.'de çizgi grafiği ile gösterilmektedir.



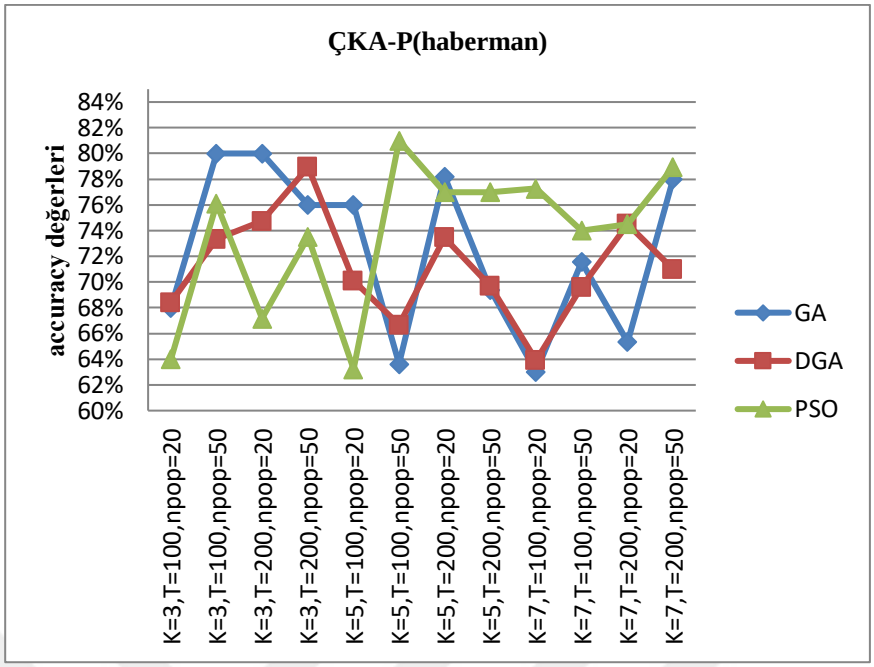
Şekil 4.35. Haberman veri setinin DVM-P elde edilen accuracy değerleri

- ÇKA-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %69.28 iken aynı sınıflandırma algoritması her üç algoritma da 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %70.63, K=5 için 4 durumda elde edilen veri setleriyle ortalama %76.27 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %70.76 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %70.67, K=5 için 4 durumda elde edilen veri setleriyle ortalama %70.52 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %70.61 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %71.10, K=5 için 4 durumda elde edilen veri setleriyle ortalama %72.55 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %71.49 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.36.'da çizgi grafiği ile gösterilmektedir.



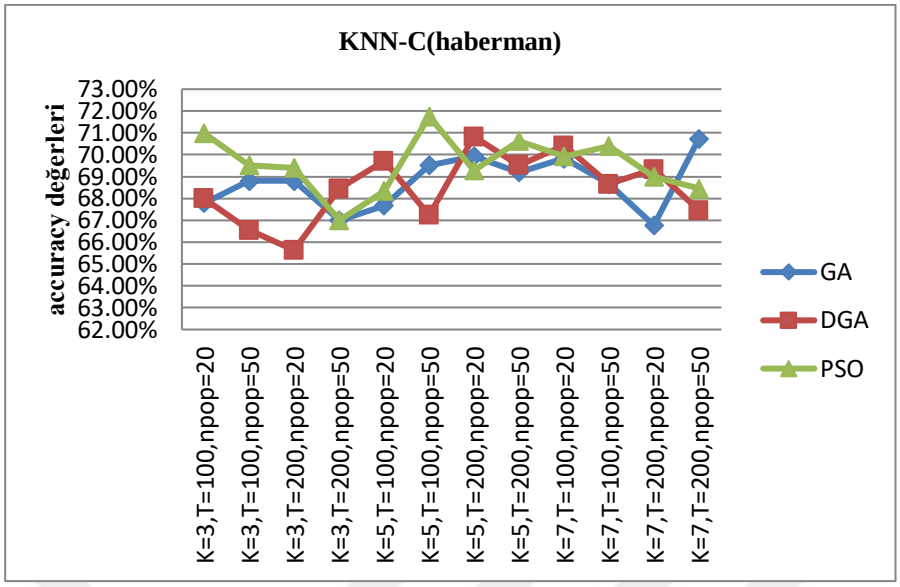
Şekil 4.36. Haberman veri setinin ÇKA-C elde edilen accuracy değerleri

- ÇKA-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %63.46 iken aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=50 için ve K=3, T=200, npop=20 için elde edilen veri setleriyle %80 değerini, K=3, T=200, npop=50 için ve K=5, T=100, npop=20 için elde edilen veri setleriyle %76 değerini ve K=5, T=200, npop=20 için elde edilen veri setiyle %78.21 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=200, npop=50 için için elde edilen veri setiyle %78.94 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=5, T=100, npop=50 için için elde edilen veri setiyle %81 değerini K=5, T=200, npop=20 için ve K=5, T=200, npop=50 için için elde edilen veri setiyle %77 değerini, K=7, T=100, npop=20 için için elde edilen veri setiyle %77.27 değerini ve K=7, T=200, npop=50 için de için elde edilen veri setiyle %78.94 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.37.'de çizgi grafiği ile gösterilmektedir.



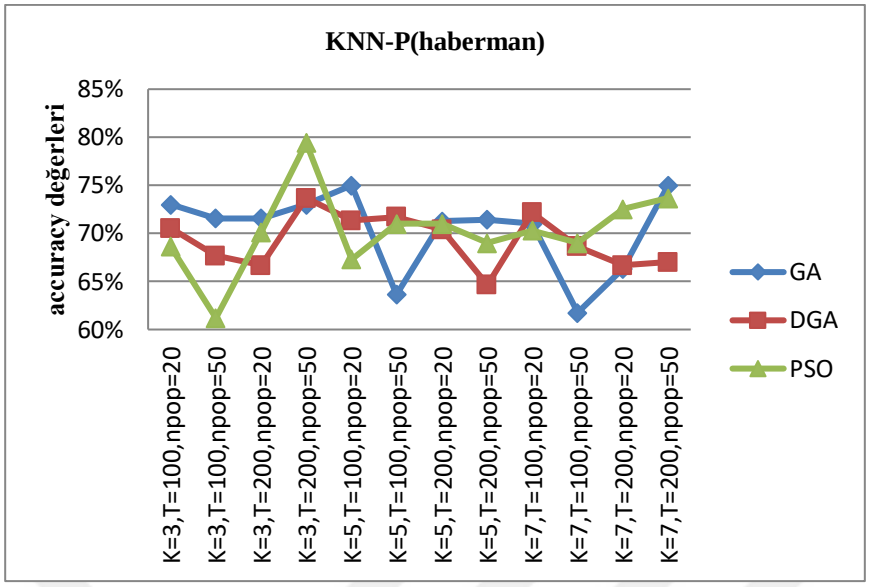
Şekil 4.37. Haberman veri setinin ÇKA-P elde edilen accuracy değerleri

- KNN-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %68.30 iken aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %68.10, K=5 için 4 durumda elde edilen veri setleriyle ortalama %69.08 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %68.75 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için4 durumda elde edilen veri setleriyle ortalama %67.15, K=5 için 4 durumda elde edilen veri setleriyle ortalama %69.32 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %68.96 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %69.22, K=5 için 4 durumda elde edilen veri setleriyle ortalama %70 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %69.44 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.38.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.38. Haberman veri setinin KNN-C elde edilen accuracy değerleri

- KNN-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %61.53 iken aynı sınıflandırma algoritması GA algoritması K=3, T=100, npop=20 ve K=3, T=200, npop=50 için elde edilen veri setleriyle %73 değerini ve K=5, T=100, npop=20 ve K=7, T=200, npop=50 için elde edilen veri setleriyle %75 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=200, npop=50 için elde edilen veri setiyle %73.68 değerini ve K=7, T=100, npop=20 için elde edilen veri setiyle %72.16 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3, T=200, npop=50 için elde edilen veri setiyle %79.41 değerini ve K=7, T=200, npop=50 için de elde edilen veri setiyle %73.68 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.39.'da çizgi grafiği ile gösterilmektedir.



Şekil 4.39. Haberman veri setinin KNN-P elde edilen accuracy değerleri

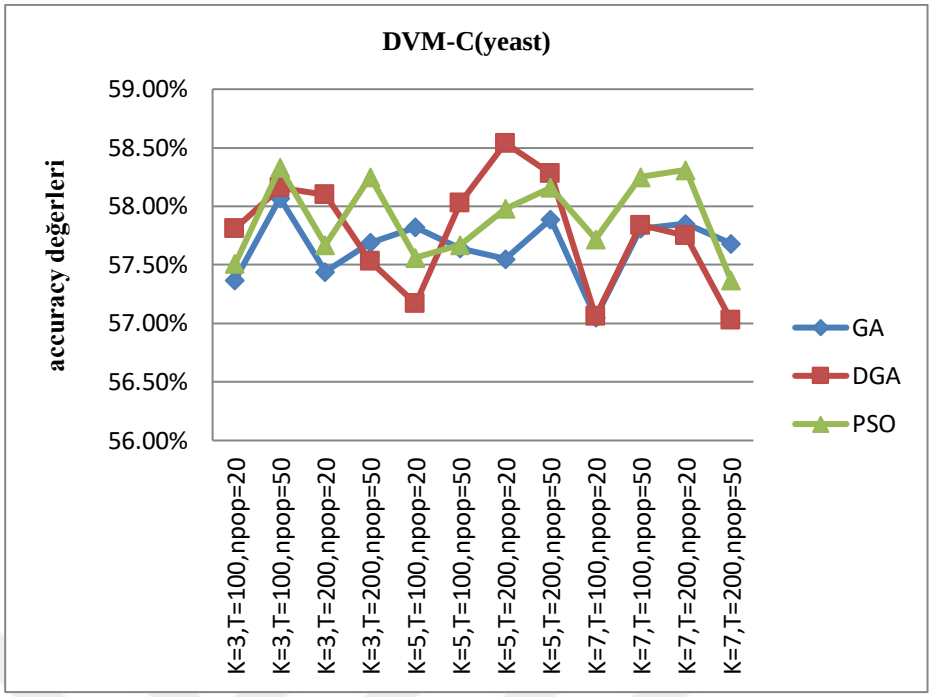
Çizelge 4.11., yeast veri setinden silinen aykırı değerlerin sayısını ve bu aykırı değerler silindikten sonra veri setinin sınıflandırma algoritmalarıyla elde edilen accuracy değerlerini göstermektedir.

Çizelge 4.11. Yeast veri seti için elde edilen accuracy değerleri

Yeast	T/npop	T=100 npop=20			T=100 npop=50			T=200 npop=20			T=200 npop=50		
Küme	Alg.	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO	GA	DGA	PSO
K=3	Outlier	48	37	20	29	57	8	20	78	12	16	78	13
	DVM-C	%57.37	%57.81	%57.51	%58.07	%58.16	%58.33	%57.44	%58.10	%57.67	%57.69	%57.53	%58.25
	DVM-P	%56.03	%57.92	%55.02	%57.37	%59.17	%55.77	%54.41	%55.02	%60.4	%54.30	%54.39	%58.4
	ÇKA-C	%57.85	%58.02	%58.24	%57.25	%59.77	%58.73	%57.17	%58.75	%59.10	%57.35	%59.74	%57.91
	ÇKA-P	%59.91	%58.13	%58.63	%57.17	%59.17	%56.57	%57.42	%55.73	%61.2	%58.51	%56.69	%60.4
	KNN-C	%52.92	%52.35	%52.86	%51.95	%53.46	%53.45	%53.21	%51.28	%53.32	%53.26	%51.77	%52.27
	KNN-P	%52.55	%51.21	%50.80	%53.53	%54.22	%51.59	%51.60	%51.46	%53.8	%50.50	%54.39	%53
K=5	Outlier	21	27	23	25	47	58	28	73	63	40	60	69
	DVM-C	%57.82	%57.17	%57.56	%57.64	%58.03	%57.67	%57.55	%58.54	%57.98	%57.89	%58.28	%58.16
	DVM-P	%57.34	%57.37	%51.10	%52.82	%56.03	%54.02	%58.98	%56.87	%55.69	%57.43	%59.29	%56.34
	ÇKA-C	%58.16	%59.57	%59.54	%57.84	%59.15	%58.09	%57.96	%58.04	%57.77	%59.27	%56.88	%57.52
	ÇKA-P	%57.74	%57.37	%58.14	%59.07	%58.07	%56.08	%58.38	%56.25	%56.72	%56.82	%57.85	%57.38
	KNN-C	%51.81	%52.23	%51.67	%52.29	%52.19	%53.18	%52.47	%52.72	%52.56	%52.35	%51.96	%53.56
	KNN-P	%54.72	%49.89	%49	%48.79	%51.53	%51.75	%52.12	%56.04	%48.44	%52.13	%51.44	%51.14
K=7	Outlier	24	33	34	12	37	61	27	52	59	47	48	67
	DVM-C	%57.05	%57.06	%57.72	%57.81	%57.84	%58.25	%57.85	%57.75	%58.31	%57.68	%57.03	%57.37
	DVM-P	%59.27	%56.79	%62.27	%57	%57.31	%57.43	%56.76	%53.38	%58.67	%56.03	%60.45	%56.63
	ÇKA-C	%59.04	%57.89	%57.79	%59.17	%57.08	%59.03	%58.68	%58.24	%57.75	%58.17	%58.42	%58
	ÇKA-P	%60.48	%58.01	%59.63	%60.6	%55.08	%60.69	%56.96	%58.93	%55.57	%62.37	%58.51	%57
	KNN-C	%53.01	%52.79	%52.27	%51.76	%51.69	%53.12	%53.12	%53	%53.54	%53.65	%52.92	%53.63
	KNN-P	%52.41	%53.95	%52.33	%52.4	%50.20	%53.30	%50.70	%54	%51.23	%52.35	%50.81	%53.11

Yeast veri setinin ilk haliyle elde edilen Çizelge 4.6.'da bulunan accuracy değerleriyle yeast veri setinin sonraki haliyle elde edilen Çizelge 4.11.'de bulunan accuracy değerleri karşılaştırıldığında aşağıdaki gibi değerlendirmeler yapılabilmektedir.

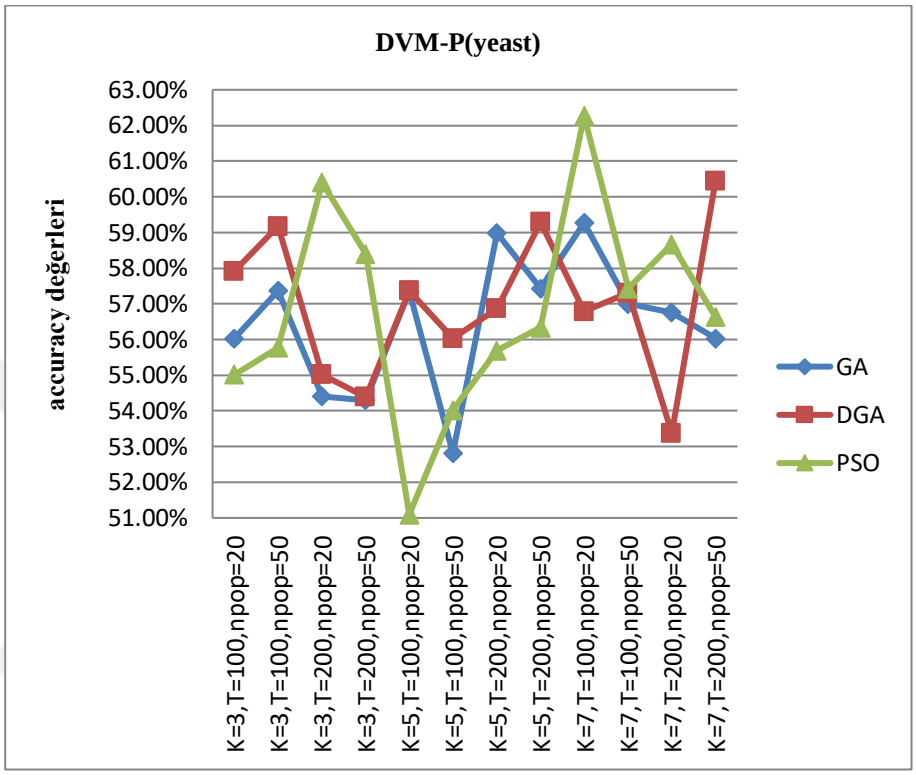
-DVM-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %57.34 iken aynı sınıflandırma algoritması K=3, T=100, npop=50 için GA algoritmasıyla elde edilen veri setiyle %58.07, DGA algoritmasıyla elde edilen veri setiyle %58.16 ve PSO algoritmasıyla elde edilen veri setiyle %58.33 değerini bulmaktadır. Aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %57.64, K=5 için 4 durumda elde edilen veri setleriyle ortalama %57.72 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %57.59 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %57.9, K=5 için 4 durumda elde edilen veri setleriyle ortalama %58 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %57.42 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %57.94, K=5 için 4 durumda elde edilen veri setleriyle ortalama %57.84 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %57.91 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.40.'da çizgi grafiği ile gösterilmektedir.



Şekil 4.40. Yeast veri setinin DVM-C elde edilen accuracy değerleri

-DVM-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri 55.84 iken aynı sınıflandırma algoritmasıyla GA algoritması K=7, T=100, npop=20 için elde edilen veri setiyle %59.27 değerini K=5, T=200, npop=20 için elde edilen veri setiyle %58.98 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3, T=100, npop=50 için elde edilen veri setiyle %59.17 değerini, K=5, T=200, npop=50 için elde edilen veri setiyle %59.29 değerini ve K=7, T=200, npop=50 için elde edilen veri setiyle %60.45 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3, T=200, npop=20 için elde edilen veri setiyle %60.4 değerini ve K=7, T=100, npop=20 için elde edilen veri setiyle %62.27 değerini bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setiyle ortalama %55.52, K=5 için 4 durumda elde edilen veri setiyle ortalama %56.64 ve K=7 için 4 durumda elde edilen veri setiyle ortalama %57.26 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setiyle ortalama %56.62, K=5 için 4 durumda elde edilen veri setiyle ortalama %57.39 ve K=7 için 4 durumda elde edilen veri setiyle ortalama %56.98 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setiyle ortalama %57.39, K=5

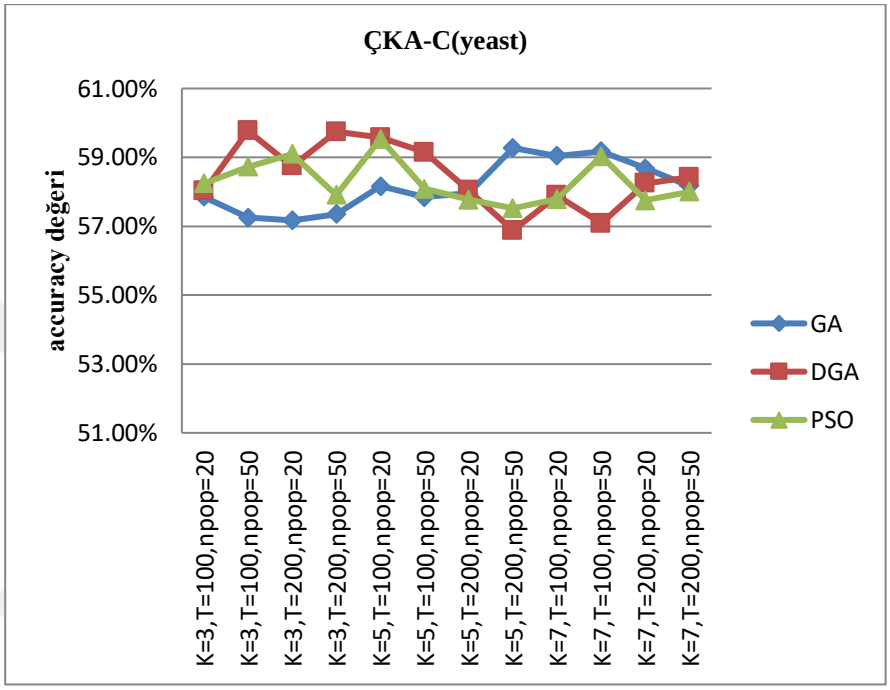
için 4 durumda elde edilen veri setiyle ortalama %54.28 ve K=7 için 4 durumda elde edilen veri setiyle ortalama %58.75 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin DVM-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.41.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.41. Yeast veri setinin DVM-P elde edilen accuracy değerleri

-ÇKA-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %59.02 iken aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %57.40, K=5 için 4 durumda elde edilen veri setleriyle ortalama %58.30 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %58.76 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %58.94, K=5 için 4 durumda elde edilen veri setleriyle ortalama %58.41 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %57.90 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %58.49, K=5

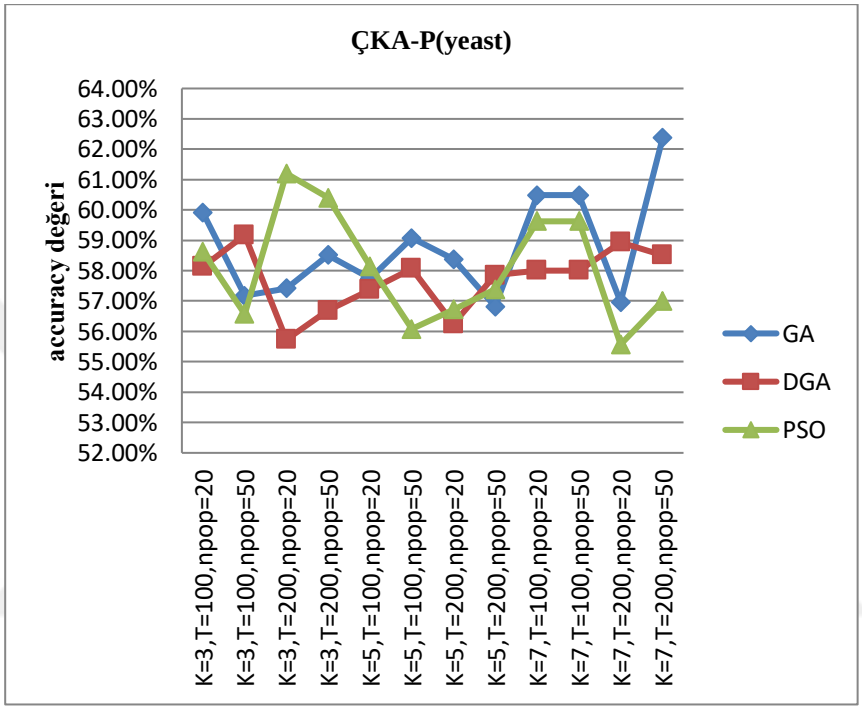
için 4 durumda elde edilen veri setleriyle ortalama %58.23 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %58.14 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.42.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.42. Yeast veri setinin ÇKA-C elde edilen accuracy değerleri

-ÇKA-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %60.39 iken aynı sınıflandırma algoritmasıyla GA K=7, T=200, npop=50 için elde edilen veri setiyle %62.37 değerini bulmaktadır. Bunun dışında aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %58.25, K=5 için 4 durumda elde edilen veri setleriyle ortalama %58 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %60.10 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %57.43, K=5 için 4 durumda elde edilen veri setleriyle ortalama %57.38 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %57.63 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri

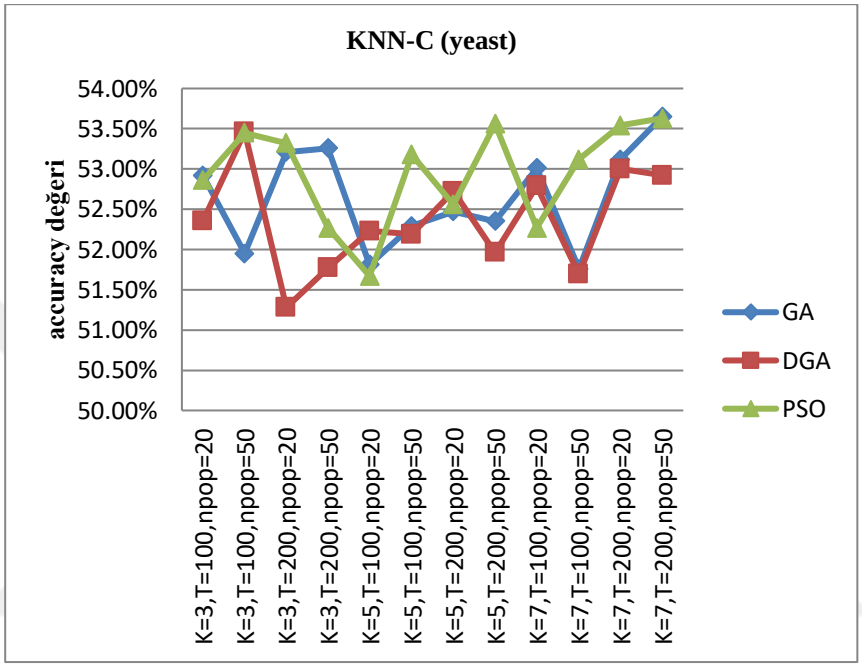
setleriyle ortalama %59.2, K=5 için 4 durumda elde edilen veri setleriyle ortalama %57.08 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %58.22 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin ÇKA-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.43.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.43. Yeast veri setinin ÇKA-P elde edilen accuracy değerleri

-KNN-C sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen accuracy değeri %52.29 iken aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.83, K=5 için 4 durumda elde edilen veri setleriyle ortalama %52.23 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %52.88 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.21, K=5 için 4 durumda elde edilen veri setleriyle ortalama %52.27 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %52.6 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.97, K=5

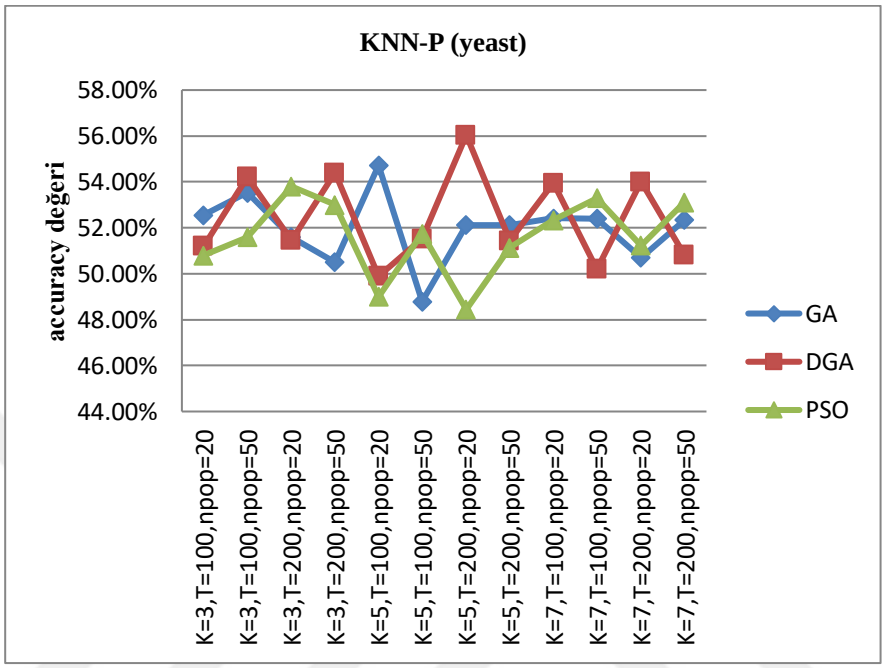
için 4 durumda elde edilen veri setleriyle ortalama %52.66 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %52.49 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-C ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.44.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.44. Yeast veri setinin KNN-C elde edilen accuracy değerleri

-KNN-P sınıflandırma algoritmasıyla veri setinin ilk halinden elde edilen değer %52.27 iken aynı sınıflandırma algoritması her üç algoritma için 12 farklı durumda elde edilen veri setleriyle birbirine yakın değerler bulmaktadır. Aynı sınıflandırma algoritması GA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.04, K=5 için 4 durumda elde edilen veri setleriyle ortalama %51.94 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %51.96 değerini bulmaktadır. Aynı sınıflandırma algoritması DGA algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.82, K=5 için 4 durumda elde edilen veri setleriyle ortalama %52.22 ve K=7 için 4 durumda elde edilen veri setleriyle ortalama %52.24 değerini bulmaktadır. Aynı sınıflandırma algoritması PSO algoritması K=3 için 4 durumda elde edilen veri setleriyle ortalama %52.29, K=5 için 4 durumda elde edilen veri setleriyle ortalama %50.08 ve K=7 için 4 durumda elde edilen veri setleriyle

ortalama %52.46 değerini bulmaktadır. Ayrıca GA, DGA ve PSO algoritmalarıyla elde edilen bütün veri setlerinin KNN-P ile sınıflandırılmasından elde edilen accuracy değerleri Şekil 4.45.'de çizgi grafiği ile gösterilmektedir.



Şekil 4.45. Yeast veri setinin KNN-P elde edilen accuracy değerleri

5. SONUÇLAR ve ÖNERİLER

5.1. Sonuçlar

İris veri seti için elde edilen değerler ile aşağıdaki gibi değerlendirmeler yapılmaktadır.

- DVM-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %89.33 iken GA ve PSO algoritmalarıyla k=5 için elde edilen veri setleriyle %91 accuracy değeri elde edilmektedir. Algoritmaların 12 farklı durumda elde ettikleri başarıları değerlendirildiğinde GA ve PSO'nun DGA'ya göre daha iyi olduğu söylenebilmektedir.
- DVM-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %94.11 iken her üç algoritmayla da farklı durumlarda elde edilen veri setleriyle %96 accuracy değeri elde edilmektedir. Algoritmaların elde ettiği başarıları değerlendirildiğinde GA ve DGA'nın PSO'ya göre daha iyi olduğu söylenebilmektedir.
- ÇKA-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %94 iken DGA algoritmasıyla k=5 durumunda elde edilen veri setiyle %96.5 accuracy değeri elde edilmektedir. GA ve PSO algoritmalarıyla k=3 durumunda elde edilen veri setleriyle %96.5 accuracy değeri elde edilmektedir. Algoritmaların elde ettiği başarıları değerlendirildiğinde PSO'nun daha iyi olduğu söylenebilmektedir.
- ÇKA-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %96.07 iken PSO algoritmasıyla k=3 durumunda elde edilen veri setiyle %100 accuracy değeri, GA ve DGA algoritmalarıyla k=7 durumunda elde edilen veri setleriyle %100 accuracy değeri elde edilmektedir.
- KNN-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %90.66 iken GA ve PSO algoritmalarıyla k=5 ve k=7 durumunda elde edilen veri setleriyle %94.5 accuracy değeri elde edilmektedir. Algoritmaların 12 farklı

durumda elde ettikleri başarıları değerlendirildiğinde PSO ve GA'nın DGA'ya göre daha iyi olduğu söylenebilmektedir.

- KNN-P sınıflandırma algoritmasıyla accuracy değeri %94.11 iken GA ve DGA algoritmalarıyla farklı durumlarda elde edilen veri setleriyle %98 accuracy değeri elde edilmektedir. Algoritmaların 12 farklı durumda elde ettikleri başarıları değerlendirildiğinde PSO'nun diğer iki algoritmaya göre geride kaldığı söylenebilmektedir.

Bupa veri seti için elde edilen değerler ile aşağıdaki gibi değerlendirmeler yapılmaktadır.

- DVM-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %58.26 iken DGA algoritmasıyla %60.55 accuracy değeri elde edilmektedir. Algoritmaların 12 farklı durumda elde ettikleri veri setleriyle başarıları değerlendirildiğinde önce DGA'nın sonra PSO'nun GA'ya göre daha iyi değerler elde ettiği söylenebilmektedir.
- DVM-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %63.24 iken DGA algoritmasıyla k=7 durumunda elde edilen veri setiyle %74.77 accuracy değeri elde edilmektedir.
- ÇKA-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %71.59 iken PSO algoritmasıyla k=7 durumunda elde edilen veri setiyle %74 accuracy değeri elde edilmektedir.
- ÇKA-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %65.81 iken PSO ve DGA algoritmalarıyla k=5 durumunda elde edilen veri setleriyle yaklaşık %75 accuracy değeri elde edilmektedir.
- KNN-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %62.89 iken PSO ve DGA algoritmalarıyla k=3 durumunda elde edilen veri setleriyle yaklaşık %65 accuracy değeri elde edilmektedir.
- KNN-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %64.95 iken DGA algoritmasıyla k=5 durumunda elde edilen veri setiyle %72.07 accuracy değeri elde edilmektedir.

Blood veri seti için elde edilen değerler ile aşağıdaki gibi değerlendirmeler yapılmaktadır.

- DVM-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %76.20 iken her üç algoritmayla da 12 farklı durumda elde elden veri setleriyle ortalama olarak %77'ye yakın accuracy değerleri elde edilmektedir.
- DVM-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %74.80 iken DGA algoritmasıyla k=5 durumunda elde edilen veri setiyle %86 accuracy değeri elde edilirken PSO algoritmasıyla da k=5 durumunda elde edilen veri setiyle %84 accuracy değeri elde edilmektedir.
- ÇKA-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %78.20 iken her üç algoritmayla da 12 durumda elde edilen veri setleriyle ortalama olarak %79 accuracy değeri elde edilmektedir.
- ÇKA-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %79.92 iken DGA ve PSO algoritmalarıyla K=5 durumunda elde edilen veri setleriyle yaklaşık %86 accuracy değeri elde edilmektedir.
- KNN-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %70.45 iken her üç algoritmayla da k=3 durumunda elde edilen veri setleriyle %74'e yakın accuracy değerleri elde edilmektedir.
- KNN-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %70.07 iken her üç algoritmayla da k=3 durumunda elde edilen veri setleriyle %76'ya yakın accuracy değerleri elde edilmektedirler.

Haberman veri seti için elde edilen değerler ile aşağıdaki gibi değerlendirmeler yapılmaktadır.

- DVM-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %73.52 iken DGA algoritmasıyla K=3 durumunda elde edilen veri setiyle %76.42 accuracy değeri elde edilmektedir.
- DVM-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %75 iken yine her üç algoritmayla da k=3 durumunda elde edilen veri setleriyle yaklaşık %83 accuracy değeri elde edilmektedir.

- ÇKA-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %69.28 iken DGA ve PSO algoritmalarıyla farklı durumlarda elde edilen veri setleriyle yaklaşık %74 accuracy değeri elde edilirken GA algoritması başarı oranı olarak DGA ve PSO algoritmalarını geriden takip etmektedir.
- ÇKA-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %63.46 iken her üç algoritmayla da farklı durumlarda elde edilen veri setleriyle yaklaşık %80 accuracy değeri elde edilmektedir.
- KNN-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %68.30 iken her üç algoritmayla da k=5 durumunda elde edilen veri setleriyle yaklaşık %71 accuracy değeri elde edilmektedir.
- KNN-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %61.53 iken PSO algoritmasıyla k=3 durumunda elde edilen veri setiyle %80 accuracy değeri elde edilmektedir.

Yeast veri seti için elde edilen değerler ile aşağıdaki gibi değerlendirmeler yapılmaktadır.

- DVM-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %57.34 iken DGA algoritmasıyla k=5 durumunda elde edilen veri setiyle %58.50 accuracy değeri elde edilmektedir.
- DVM-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %58.75 iken PSO algoritmasıyla elde edilen veri setiyle %62, DGA algoritmasıyla elde edilen veri setiyle de %60.45 accuracy değeri elde edilmektedir.
- ÇKA-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %59.02 iken burada algoritmalar ortalama olarak daha iyi bir değer elde etmemektedirler.
- ÇKA-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %60.39 iken GA algoritmasıyla k=7 durumunda elde edilen veri setiyle %62 accuracy değeri elde edilmektedir.
- KNN-C sınıflandırma algoritmasıyla elde edilen accuracy değeri %52.29 iken PSO, GA ve DGA algoritmalarıyla farklı durumlarda elde edilen veri setleriyle yaklaşık %53.50 accuracy değeri elde edilmektedir.

- KNN-P sınıflandırma algoritmasıyla elde edilen accuracy değeri %52.27 iken DGA algoritmasıyla k=5 durumunda elde edilen veri setiyle %56 accuracy değeri elde edilmektedir.

5.2. Öneriler

Literatür araştırmaları dikkate alındığında metasezgisel algoritmaların aykırı değerleri tespit etmede başarılı oldukları görülmüştür. Bunu göz önünde bulundurarak bu çalışmada aykırı değerleri tespit etmek için metasezgisel algoritmalara başvurulmaktadır.

Bu çalışmada kullanılan yöntemle her üç algoritmanın da farklı durumlarda tespit ettikleri aykırı değerlerin sınıflandırma başarısı üzerindeki etkileri incelenmektedir.

Araştırma bulguları kısmına baktığımızda her bir veri seti için uygulamada kullanılan GA, DGA ve PSO algoritmalarının farklı küme sayıları, farklı iterasyon sayıları ve farklı popülasyon sayılarıyla farklı sayıda aykırı değer tespit ettikleri görülmektedir. Bu tespit edilen aykırı değerler veri setlerinden silinerek veri setleri tekrar sınıflandırılmaktadır. Araştırma bulguları bölümünde aykırı değerlerin veri setlerinin sınıflandırma başarısı üzerinde etkisi incelendiğinde her üç algoritmanın da ortalama olarak birbirlerine yakın bir başarı elde ettikleri görülmektedir. Ancak her bir veri seti için GA, DGA ve PSO algoritmalarının 12 durumda elde ettikleri aykırı değerlerin sınıflandırma başarısı üzerine etkilerine bakıldığında bütün durumlarda veri setlerinin ilk haliyle DVM-C, DVM-P, ÇKA-C, ÇKA-P, KNN-C ve KNN-P sınıflandırma algoritmalarıyla elde edilen accuracy değerinden daha büyük bir accuracy değeri her zaman elde edilmediği görülmektedir. Daha büyük bir accuracy değerinin elde edilmemesinin nedenlerinden biri, metasezgisel algoritmalar ya yeteri kadar aykırı değer tespit edememiştir ya da aykırı olmayan değerleri aykırı değer olarak görmüştür. Bu sorunu çözmek için metasezgisel algoritmaları farklı parametre değerleriyle kullanmak bir çözüm olabilmektedir. Bu çalışmada da metasezgisel

algoritmalar farklı parametre değerleriyle kullanıldığında farklı sayıda aykırı değerler tespit edilmektedir ve tespit edilen aykırı değerlerin sınıflandırma başarısı üzerine etkileri de farklı olmaktadır. Bu yüzden metasezgisel algoritmalarla çalışıldığında parametre değerlerinin önemli olduğu görülmektedir.

Literatürde çok fazla sayıda farklı aykırı değer tespit modelleriyle PSO ve GA kullanılarak aykırı değer tespit çalışmaları yapılmıştır ancak aykırı değer tespit etmede DGA ve diğer metasezgisel algoritmalarla ilgili çok fazla çalışmaya pek rastlanılmıştır.

DGA için elde edilen sonuçlar PSO ve GA için elde edilen sonuçlar ile karşılaştırıldığında, DGA'nın bulduğu aykırı değerler ile PSO ve GA'nın bulduğu aykırı değerlerin birbirine çok yakın değerler olduğu ve bu değerlerin sınıflandırma başarısı üzerine de benzer etkileri olduğu görülmektedir. Bu yüzden DGA'nın da aykırı değerleri tespit etmede başarılı olduğu söylenebilir. DGA'nın da PSO ve GA gibi başarılı olduğu göz önünde bulundurulduğunda diğer metasezgisel algoritmalarla da aykırı değer tespit çalışmaların yapılabileceği düşünülmektedir.

PSO gibi sürü zekası temelli algoritmalardan olan yapay arı kolonisi algoritması ve karınca kolonisi algoritması gibi metasezgisel algoritmalar NP-hard (çözümü zor) problemlerinde PSO'ya benzer başarılar elde ettiğinden dolayı aykırı değer tespitinde de başarılı olacağı düşünülmektedir.

Aykırı değer tespiti veri madenciliğinin önemli bir konusu ve problemi iken aykırı değer tespitinin metasezgisel algoritmalarla başarılı bir şekilde çözülüyor olması veri madenciliğine büyük bir katkı sağlamaktadır. Bu durum göz önünde bulundurulduğunda metasezgisel algoritmaların, veri madenciliğinin diğer önemli konuları ve problemlerinde de başarılı olabileceği düşünülmektedir.

KAYNAKÇA

ABRAHAM, B., and BOX, G. E. P., 1979. Bayesian Analysis Of Some Outlier Problems İn Time Series. Biometrika, 66(2): 229–248.

AGARWAL, D., 2006. Detecting Anomalies İn Cross-Classified Streams: A Bayesian Approach. Knowl. Inf. Syst., 11(2): 29–44.

AGGARWAL, C. C., and YU, P. S., 2001. Outlier Detection For High Dimensional Data. In Proc. 2001 ACM-SIGMOD Int. Conf. Management Of Data (SIGMOD'01), 21-24 Mayıs, Cilt 30, California, s. 37–46.

AGGARWAL,C.C., 2015. Outlier Analysis. Data Mining Springer International Publishing, Switzerland, 466s.

ALAGÖZ, A., ÖGE, S., and ORTAKARPUZ, M., 2014. Bir Kurumsal Zekâ Teknolojisi Olarak Veri Madenciliği İle Muhasebe Bilgi Sistemi İlişkisi. Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi Dr. Mehmet YILDIZ Özel Sayısı, 31(1): 1-21.

ALAM MİA, M.M., BİSWAS, S.K., URMİ, M.C., and SİDDİQUE, A., 2015. An Algorithm For Training Multilayer Perceptron (MLP) For Image Reconstruction Using Neural Network Without Overfitting. Internatıonal Journal Of Scıentıfıc & Technology Research, 4(46): 271-275.

ALDENDERFER, M. S., and BLASHFİELD, R. K., 1984. Cluster Analysis. SAGE Publications, California, 88s.

ANGİULLİ, F., and PİZZUTİ, C., 2005. Outlier Mining İn Large High-Dimensional Data Sets. IEEE Trans. on Knowl. and Data Eng., 17(2): 203-215.

ANGİULLİ, F.B., and PİZZUTİ, S., 2006. Distance- Based Detection and Prediction of Outliers. IEEE Transactions on Knowledge and Data Engineering, 18(2): 145-160.

ANSCOMBE, F. J., and GUTTMAN, I., 1960. Rejection Of Outliers. Technometrics, 2(2): 123–147.

ARUNACHALAM, V., 2008. Optimization Using Differential Evolution. The Unıversıty Of Western Ontarıo Department Of Cıvıl And Envıronmental Engıneerıng Water Resources Research Report, Canada, 42s.

BAKAR, Z. A., MOHEMAD, R., AHMAD, A., and DERİS, M. M., 2006. A Comparative Study For Outlier Detection Techniques İn Data Mining. In Proc. 2006 IEEE Conf. Cybernetics And Intelligent Systems, 7-9 Haziran, Cilt 23, Bangkok, s. 1-6.

BAY, S. D., and SCHWABACHER, M., 2003. Mining Distance-Based Outliers İn Near Linear Time With Randomization And A Simple Pruning Rule. In Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery And Data Mining (KDD'03), 24-27 Haziran, Cilt 25, Washington, s. 29–38.

BEASLEY, D., BULL, D.R., and MARTIN, R.R., 1993. An Overview of Genetic Algorithms: Part 2, Research Topics. University Computing, 15(4): 170-181.

BECKMAN, R. J., and COOK, R. D., 1983. Outlier. Journal Technometrics, 25(2): 119–149.

BREUNİG, M. M., KRİEGEL, H.P., NG, R., and SANDER, J., 2000. LOF: Identifying Density-Based Local Outliers. In Proc. 2000 ACM-SIGMOD Int.

Conf. Management Of Data (SIGMOD'00), 15-18 Mayıs, Cilt 29, Texas, s. 93–104.

CHAKRABARTİ, D., KUMAR, R., and TOMKİNS, A., 2006. Evolutionary Clustering. KDD '06 Proceedings Of The 12th ACM SIGKDD İnternational Conference On Knowledge Discovery And Data Mining, 20-23 Ağustos, Cilt 20, Philadelphia, s. 554-560.

CHANDOLA, V., BANERJEE, A., and KUMAR, V., 2009. Anomaly detection: A survey. ACMComputing Surveys, 41(15): 1–72.

CLERC, M., and KENNEDY, J., 2002. The Particle Swarm- Explosion, Stability, And Convergence İn A Multidimensional Complex Space. Ieee Transactıons On Evolutıonary Computatıon, 6(1): 58 – 73.

CRAWFORD, K.D., and WAİNWRİGHT, R.L., 2000. Applying Genetic Algorithms To Outlier Detection. LOF : Identifying Density – Based Local Outliers. Morgan Kaufmann Publishers, 3(1): 546-550.

DANİEL, T. L., 2005. Discovering Knowledge in Data: An Introduction to Data Mining. John Wiley & Sons, Inc., Hoboken, New Jersey, 239s.

DASGUPTA, D., and MAJUMDAR, N. S., 2002. Anomaly Detection İn Multidimensional Data Using Negative Selection Algorithm. In Proc. 2002 Congress On Evolutionary Computation (CEC'02), 12- 17 Mayıs, Cilt 4, Washington, s.1039–1044.

DOĞAN, Y., 2009. Outlier Detection With K Nearest Neighbor Clustering. Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İzmir, 80s.

DORİGO,M., and STÜTZLE, T., 2014. Ant Colony Optimization. A Bradford Book The MIT Press Cambridge, Massachusetts Institute of Technology Massachusetts, London, 321s.

ERDOĞAN, G., 2012. Spectral Methods For Outlier Detection İn Machine Learning. Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 111s.

ESKİN, E., 2000. Anomaly Detection Over Noisy Data Using Learned Probability Distributions. In Proc. 17th Int. Conf. Machine Learning (ICML'00), 29 Haziran- 2 Temmuz, Cilt 140, San Francisco, s. 252-262.

FAWCETT,T., and PROVOST, F., 1997. Adaptive Fraud Detection. Data Mining And Knowledge Discovery, 1(3): 291–316.

FRAWLEY,GB., SHAPİRO, G.B., 1996. Knowledge Discovery İn Databases: An Overview. AI Magazine, 13(3): 57-70.

FUJİMAKİ, R., YAİRİ, T., and MACHİDA, K., 2005. An Approach To Spacecraft Anomaly Detection Problem Using Kernel Feature Space. In Proc. 2005 Int. Workshop Link Discovery (Linkkdd'05), 21-24 Ağustos, Cilt 26, Chicago, s. 401–410.

GRUBBS, F.E., 1969. Procedures For Detecting Outlying Observations İn Samples. Technometrics, 11(1): 1–21.

GUNN, S. R., 1998. Support Vector Machines For Classification And Regression. Technical Report, Unıversıty of Southampton, 342(5): 1-66.

GÜÇLÜ, M., 2012.Yapay Bağişiklik Sistem Tabanli Algoritma İle Aykiri Değer Tespiti. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 93s.

GÜRÜNLÜ ALMA, Ö., 2009. Genetic Algorithm Based Outlier Detection Using İnformation Criterion. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İzmir, 152s.

HAN, J., and KAMBER, M., 2006. Data Mining: Concepts And Techniques. The Morgan Kaufmann Series in Data Management Systems, USA, 772s.

HAN,J., KAMBER, M., and PEİ, J., 2012. Data Mining Concepts And Techniques. The Morgan Kaufmann Series in Data Management Systems Third Edition, Canada, 742s.

HAYKİN, S., 1994. Neural Networks: A Comprehensive Foundation. IEEE Press, New York, NY, USA, 823s.

HE, Z., XU, X., and DENG, S., 2003. Discovering Cluster-Based Local Outliers. Pattern Recognition Lett., 24 (9):1641–1650.

HODGE, V.J., and AUSTİN, J., 2004. A Survey Of Outlier Detection Methodologies. Artificial Intelligence Review, 22 (2): 85-126.

HOLLAND, J.H., 2005. Genetic Algorithms. Scıentıfıc Amerıcan, 267(1): 66-72.

HORN, P. S., FENG, L., Lİ, Y., and PESCE, A. J., 2001. Effect Of Outliers And Nonhealthy İndividuals On Reference İnterval Estimation. Clinical Chemistry, 47(12): 2137–2145.

JAİN, N., SRİVASTAVA, V., 2013. Data Mınıng Technıques: A Survey Paper. IJRET: International Journal of Research in Engineering and Technology, 2(1): 116-119.

JİN, W., TUNG, K. H., and HAN, J., 2001. Mining Top-N Local Outliers İn Large Databases. In Proc. 2001 ACMSIGKDD Int. Conf. Knowledge Discovery İn Databases (KDD'01), 26-29 Ağustos, Cilt 8, San Francisco, s. 293– 298.

JİN, W., TUNG, K. H., HAN, J., and WANG, W., 2006. Ranking Outliers Using Symmetric Neighborhood Relationship. In Proc. 2006 Pacific-Asia Conf. Knowledge Discovery And Data Mining (PAKDD'06), 9-12 Nisan, Cilt 10, Singapore, s.577-593.

KARACAN, H., YEŞİLBUDAK, M., 2010. Kullanıcı Merkezli İnteraktif Veri Madenciligi: Bir Literatür Taraması. Bilişim Teknolojileri Dergisi, 3(1): 17-22.

KESKİNTÜRK, T., 2006. Diferansiyel Gelişim Algoritması. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 5(9): 85-99.

KNORR, E., and NG, R., 1997. A Unified Notion Of Outliers: Properties And Computation. In Proc.1997 Int. Conf. Knowledge Discovery And Data Mining (KDD'97), 14-17 Ağustos, Cilt 55, Newport Beach, s. 219–222.

KNORR, E., NG, R. T., and TUCAKOV, V., 2000. Distance-Based Outliers: Algorithms And Applications. The VLDB J., 8(3): 237–253.

KNORR, E.M., and NG, R.T., 1998. Algorithms For Mining Distance-Based Outliers İn Large Datasets. In Proc- VLDB, 24(35): 392-403.

KRİEGEL, H.-P., SCHUBERT, M., and ZİMEK, A., 2008. Angle-Based Outlier Detection İn High dimensional Data. In Proc. 2008 ACM SIGKDD Int. Conf. Knowledge Discovery And Data Mining (KDD'08), 24-27 Ağustos, Cilt 18, Las Vegas, s. 433-446.

LAST,M. , KANDEL,A., 2001. Automated Detection of Outliers in Real-World Data. Proc. of the Second International Conference on Intelligent Technologies, 39(7): 292-301.

LAURİKKALA,J., JUHOLA, M., and KENTALA, E., 2000. Informal İdentification Of Outliers İn Medical Data. In Proc. 5th Int. Workshop On Intelligent Data Analysis İn Medicine And Pharmacology, 10(6): 1-8.

MAHONEY, M. V. ,CHAN, P. K., 2003. Learning Rules For Anomaly Detection Of Hostile Network Traffic. In Proc. 2003 Int. Conf. Data Mining (ICDM'03), 19-22 Kasım, Cilt 3, Washington, s. 81-99.

MELANİE, M., 1999. An Introduction To Genetic Algorithms. A Bradford Book The MIT Press Cambridge, Massachusetts, London, 162s.

MİCHALEWİCZ, Z.,1992. Genetic Algorithms + Data Structures = Evolution Programs. Department Of Computer Science University Of North Carolina Charlotte, USA, 340s.

MİTCHELL, T, 1997. Machine Learning. McGraw-Hill Science/Engineering/Math, California, 432s.

MOHAMED,A.W., SABRY, Z.W., and KHORSHİD,M., 2012. Orıgınal Artıcle An Alternative Differential Evolution Algorithm For Global Optimization. Journal Of Advanced Research, 3(2): 149–165.

MOHEMMED, A.W., ZHANG, M., and BROWNE,W., 2010. Particle Swarm Optimisation For Outlier Detection. Technical Report: Ecstr, 10(7): 1-8.

NOBLE, C. C., COOK, D. J., 2003. Graph-Based Anomaly Detection. In Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery And Data Mining (KDD'03), 24- 27 Ağustos, Cilt 15, Washington, s. 631–636.

OSUNA, E.E., FREUND, R., and GİROSİ, F., 1997. Support Vector Machines:Training and Applications. Massachusetts Institute of Technology and Artificial Intelligence Laboratory, Massachusetts, 42s.

OVLA, H.V., TAŞDELEN,B., 2012. Aykırı Değer Yönetimi. Mersin Üniversitesi, Sağlık Bilim Derg. 5(3): 1-8.

ÖZTEMEL, E., 2012. Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul, 231s.

PANDİAN, V., 2012. Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics and finans. Petronas University of technology, Malaysia, 734s.

PATCHA, PARK, J.-M., 2001. An Overview Of Anomaly Detection Techniques: Existing Solutions And Latest Technological Trends. Computer Networks, 51(12): 448–3470.

RAJA-DR., P.W., and BHASKARAN, V.M., 2012. An Effective Genetic Algorithm For Outlier Detection. International Journal of Computer Applications, 38(6): 30-33.

ROUSSEEUW, P.J., LEROY, A.M., 1987. Robust Regression and Outlier Detection. Wiley- Interscience, New York, 329s.

RUİZ-GAZEN, A., BERRO, A., and MARİE-SAİNTE, S., 2010. Detecting multivariate Outliers using Projection Pursuit with Particle Swarm. Proceedings of COMPSTAT'2010 , 22-27 Ağustos, Cilt 19, Paris, s. 89-98.

SHEWHART, W. A., 1931. Economic Control of Quality of Manufactured Product. D. Van Nostrand Company, Inc., Madison, 501s.

SHİ,Y., EBERHART, R., 1998. A Modified Particle Swarm Optimizer. IEEE, Conference Location: Anchorage, 4- 9 Mayıs, Cilt 98, Anchorage, s. 151-155.

SHİ,Y., EBERHART, R., 2000. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Evolutionary Computation. Proceedings of the 2000 Congress, 16-19 Temmuz, Cilt 2, La Jolla, s. 1-5.

SOLANAS, A., ROMERO, E., GÓMEZ, S., SOPENA, L.M., ALQUÉZAR, R., and J DOMİNGO, J., 2001. Feature Selection And Outliers Detection With Genetic Algorithms And Neural Networks. 8th International Conference of the Catalan Association for Artificial Intelligence CCIA 2005, 26-28 Ocak, Cilt 8, Alguer, s.1-8.

SONG, X., WU, M., JERMAİNE, C., and RANKA, S., 2007. Conditional Anomaly Detection. IEEE Trans. on Knowledge and Data Engineering, 19(5): 631–645.

STEFANSKY, W., 1972. Rejecting Outliers İn Factorial Designs. Technometrics, 14(2): 469–479.

TAO, Y., XİAO, X., and ZHOU, S., 2006. Mining Distance-Based Outliers From Large Databases İn Any Metric Space. In Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery İn Databases (KDD'06), 20-23 Ağustos, Cilt 24, New York, s. 394–403.

VAPNİK, V., 1995. The Nature Of Statistical Learning Theory. Springer Science & Business Media, New York Springerverlag, 313s.

WONG, N., RAY, P., STEPHENS, G., and LEWİS, L., 2012. Artificial İmmune Systems For The Detection Of Credit Card Fraud: An Architecture, Prototype And Preliminary Results. Information Systems Journal, 22(1): 53-76.

YAMANİSHİ, K., TAKEİCHİ, J., and WİLLİAMS, G., 2014. On-Line Unsupervised Outlier Detection Using Finite Mixtures With Discounting Learning Algorithms. Data Mining and Knowledge Discovery, 8(3): 275–300.

YE, N., CHEN, Q., 2001. An Anomaly Detection Technique Based On A Chi-Square Statistic For Detecting İntrusions İnto İnformation Systems. Quality And Reliability Engineering International, 17(2): 105–112.

ZERNİK, J., 2010. Data Mining As A Civic Duty – Online Public Prisoners Registration Systems. International Journal On Social Media: Monitoring, 10(6): 1-12.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Merve Gitmez

Uyruğu : T.C.

Doğum Yeri : Batman-Gercüş

Telefon : 0507 821 70 49

e-mail : mervegitmez@hotmail.com

EĞİTİM

Derece	**Adı, İlçe İl**	**Bitirme Yılı**
Lise	Fatih Lisesi, Batman	2006
Üniversite	Selçuk Üniversitesi, Konya	2013
Yüksek Lisans	Harran Üniversitesi, Şanlıurfa	2018