



**T.C.  
ONDOKUZ MAYIS ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
AKILLI SİSTEMLER MÜHENDİSLİĞİ ANA BİLİM DALI**

**ÖZELLİK SEÇME PROBLEMLERİNDE FARKLI  
METASEZGİSEL ALGORİTMALARIN KULLANILMASI**

Yüksek Lisans Tezi

**Tohid YOUSEFİ**

**DANIŞMAN  
Prof. Dr. Mehmet Serhat ODABAŞ**

**SAMSUN  
2021**

**T.C.  
ONDOKUZ MAYIS ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
AKILLI SİSTEMLER MÜHENDİSLİĞİ ANA BİLİM DALI**



**ÖZELLİK SEÇME PROBLEMLERİNDE FARKLI  
METASEZGİSEL ALGORİTMALARIN KULLANILMASI**

Yüksek Lisans Tezi

**Tohid YOUSEFİ**

Danışman

Prof. Dr. Mehmet Serhat ODABAŞ

**SAMSUN**  
2021

## TEZ KABUL VE ONAYI

**Tohid YOUSEFİ** tarafından, **Prof. Dr. Mehmet Serhat ODABAŞ** danışmanlığında hazırlanan **Özellik Seçme Problemlerinde Farklı Metasezgisel Algoritmaların Kullanılması** başlıklı bu çalışma, jürimiz tarafından 14.6.2021 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Yüksek Lisans Tezi olarak kabul edilmiştir.

	<b>Unvanı Adı Soyadı</b> <b>Üniversitesi</b> <b>Ana Bilim/Ana Sanat Dalı</b>	<b>İmza</b>	<b>Sonuç</b>
<b>Başkan</b> (Danışman)	Prof. Dr. Mehmet Serhat ODABAŞ Ondokuz Mayıs Üniversitesi Akıllı sistemler Anabilim Dalı		<input type="checkbox"/> Kabul <input type="checkbox"/> Ret
<b>Üye</b>	Dr. Öğr.Üyesi Barış ÖZKAN Ondokuz Mayıs Üniversitesi Akıllı sistemler Anabilim Dalı		<input type="checkbox"/> Kabul <input type="checkbox"/> Ret
<b>Üye</b>	Dr. Öğr.Üyesi Ertuğrul SUNAN Samsun Üniversitesi Uçak Bakım ve Onarım Anabilim Dalı		<input type="checkbox"/> Kabul <input type="checkbox"/> Ret

Bu tez enstitü yönetim kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

ONAY

.../.../2021

Prof.Dr.Ali Bolat

Enstitü Müdürü

## ETİK BEYAN

Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim

İmza  
... /... / 2021  
Tohid YOUSEFİ

### TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

**Tez Başlığı :** Özellik Seçme Problemlerinde Farklı Metasezgisel Algoritmaların Kullanılması

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından ..... tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 11

Tek kaynak oranı : % 1 çıkmıştır.

İmza  
... /... / 2021  
Prof. Dr. Mehmet Serhat ODABAŞ

## ÖZET

### ÖZELLİK SEÇME PROBLEMLERİNDE FARKLI METASEZGİSEL ALGORİTMALARIN KULLANILMASI

Tohid YOUSEFİ

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Akıllı Sistemler Mühendisliği Ana Bilim Dalı

Yüksek Lisans, Haziran/2021

Danışman : Prof. Dr. Mehmet Serhat ODABAŞ

Günümüz dünyasında çeşitli uygulamalar tarafından üretilen veriler gittikçe artmaktadır. Ayrıca günümüzde sistemler ve internet ağları, büyük miktarda veriler üretmekte ve çok yakın bir gelecekte internetin dünya çapında yaşayan herkesin beyin kapasitesini aşacağı tahmin edilmektedir. Verilerin bu kadar hızlı bir şekilde artmasının nedeni dijital sensörlerin, iletişimin, bilgi işleminin ve depolamanın ilerlemesinden kaynaklanmaktadır. Ancak gelecekte araştırmacılar büyük veri yığınlarıyla karşı karşıya kalacaklar. Bundan dolayı araştırmacılar veri hacminden, kendi yaptığı işiyle ilgili uygun ve anlamlı özellikleri çıkartma konusunda bir öneri fikri bulması gerekmektedir. Yapılan işle alakası olmayan ve fazlalık olan özellikleri veri kümesinden çıkartmaya özellik seçme işlemi denilmektedir. Özellik seçme büyük miktarda veriler için evrensel olarak kullanılan bir veri ön işleme yöntemidir. Özellik seçimi, veri kümelerini basitleştirmede, performans ve hesaplama verimliliğini arttırmada ve birçok nedenden dolayı hem denetimli hem de denetimsiz öğrenme metodlarıyla kullanılmaktadır. Genel olarak özellik seçme sorunun kesin bir çözüm yolu yoktur ve şimdiye kadarda kesin bir çözüm yolu önerilmemiştir. Ayrıca özellik seçimi için farklı yaklaşımlar literatürde mevcuttur, ama yanıtlarının kalitesi genellikle uygun değildir. Bundan dolayı akıllı optimizasyon teknikleri bu sorunlara çok daha iyi çözümler sunabilir. Bu yüzden, özellik seçimini ve ilgili sorunları çözenin en etkili ve yapıcı yollarından birisi metasezgisel yöntemlerini kullanmaktır. Bu çalışmada öncelikle yapay sinir ağı ve birden çok kez çalıştırılmalı yapay sinir ağı ile veri setinin tüm özellikleri kullanarak vücut yağ tahmini yapıldı ve korelasyon değerleri sırasıyla 0.89028 ve 0.90355 olarak bulundu. Daha sonrasında 6 tane metasezgisel algoritması (ikili genetik, karınca koloni optimizasyonu, benzetimli tavlama, parçacık sürüsü optimizasyonu, diferansiyel evrim, baskın olmayan sıralı genetik algoritması II) özellik seçimi için kullanıldı ve 8 özellikle en iyi maliyet değerine ulaşıldı. Ayrıca her biri için korelasyon değeri sırasıyla 0.88629, 0.89347, 0.89279, 0.89124, 0.90041, 0.89998 olarak bulundu. Sonuçlara bakıldığında, önemli özellikleri kullanarak daha az maliyet ve zamanla aynı performansa ulaşılabilir.

**Anahtar kelimeler:** Özellik seçme, Boyut indirgeme, Özellik çıkarma, Metasezgisel algoritmalar, Optimizasyon, İkili genetik algoritması, Karınca koloni optimizasyonu algoritması, Benzetimli tavlama algoritması, Parçacık sürüsü optimizasyonu algoritması, Diferansiyel evrim algoritması, Baskın olmayan sıralı genetik algoritması II

# ABSTRACT

## USING DIFFERENT META-HEURISTIC ALGORITHMS IN FEATURE SELECTION PROBLEMS

Tohid YOUSEFI

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Intelligent Systems Engineering

Master, June/2021

Supervisor: Prof. Dr. Mehmet Serhat ODABAŞ

In today's world, the data produced by various applications is increasing. In addition, systems and internet networks today produce large amounts of data and it is predicted that the internet will exceed the brain capacity of everyone living worldwide in the very near future. The reason data is increasing so rapidly is due to the advancement of digital sensors, communication, computing and storage. But in the future, researchers will be faced with huge data piles. Therefore, researchers need to come up with a proposal idea to extract appropriate and meaningful features related to their work from the data volume. The process of removing redundant and irrelevant features from the dataset is called feature selection. Feature selection is a universally used data preprocessing method for large amounts of data. Feature selection is used with both supervised and unsupervised learning methods to simplify datasets, improve performance and computational efficiency, and for many reasons. In general, there is no definitive solution to the feature selection problem, and no definitive solution has been proposed so far. Also, different approaches for feature selection are available in the literature, but the quality of their responses is often unsuitable. Therefore, smart optimization techniques can offer much better solutions to these problems. Therefore, one of the most effective and constructive ways to solve feature selection and related problems is to use metaheuristics. In this study, body fat was estimated using artificial neural network and multiple run artificial neural network and all the features of the data set, and correlation values were found as 0.89028 and 0.90355, respectively. Then, 6 metaheuristic algorithms (binary genetics, ant colony optimization, simulated annealing, particle swarm optimization, differential evolution, non-dominant sequential genetic algorithm II) were used for feature selection and the best cost value was achieved with 8 features. In addition, the correlation values for each of them were found as 0.88629, 0.89347, 0.89279, 0.89124, 0.90041, 0.89998, respectively. Looking at the results, the same performance can be achieved in less cost and time by using important features.

**Keywords:** Feature selection, Dimension reduction, Feature extraction, Metaheuristic algorithms, Optimization, Binary genetic algorithm, Ant colony optimization algorithm, Simulated annealing algorithm, Particle swarm optimization algorithm, Differential evolution algorithm, Non-dominanted sorting genetic algorithm II-

## ÖNSÖZ VE TEŞEKKÜR

Akademik eğitim sürecimin bir üst noktası olan yüksek lisans tez çalışmalarım boyunca yardım ve desteğini benden esirgemeyen danışman hocam Sayın Prof. Dr. Mehmet Serhat Odabaş'a teşekkürü bir borç bilirim.

Beni bugünlere getirmek için hiçbir fedakârlıktan kaçınmayan anneme, babama ve kardeşlerime sonsuz teşekkür ederim.



Tohid YOUSEFİ

# İÇİNDEKİLER

ÖZET .....	iii
ABSTRACT.....	iv
ÖNSÖZ VE TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
ŞEKİLLER DİZİNİ.....	viii
TABLolar.....	xi
1. GİRİŞ .....	1
2. LİTERATÜR ARAŞTIRMASI .....	4
2.1. Özellik Seçimi ile İlgili Literatür Çalışmaları .....	4
2.2. Metasezgisel Algoritmaları ile Özellik Seçimi Konusunda Literatür Çalışmaları .....	8
3. YÖNTEM.....	12
3.1. Özellik Seçimine Genel Bakış .....	12
3.1.1. Özellik Seçiminin Adımları .....	14
3.1.2. Özellik Seçmede Etiket Bilgisi .....	19
3.1.3. Özellik Seçmede Arama Stratejisi .....	23
3.1.3.1. Sarmal Yöntemleri .....	23
3.1.3.2. Filtre Yöntemleri.....	29
3.1.3.3. Gömülü Yöntemleri .....	50
3.1.3.4. Hibrit Yöntemleri.....	58
3.1.3.5. Topluluk Yöntemleri.....	61
3.2. Sınıflandırma .....	64
3.2.1. Sınır Ağlarına Genel Bakış .....	64
3.3. Optimizasyon'a Genel Bakış .....	66
3.3.1. Amaç Fonksiyonu .....	68
3.3.2. Uygulanabilir Çözümler ve Mümkün Olmayan Çözümler .....	68
3.3.3. En İyi Çözüm .....	69
3.3.4. Zor Kısıtlamalar ve Yumuşak Kısıtlamalar .....	69
3.3.5. Yerel ve Küresel Optimizasyon .....	70
3.3.6. Kesikli ve Sürekli Optimizasyon .....	71
3.3.7. Problem Zorluğu .....	71
3.4. Metasezgisellere Genel Bakış.....	74
3.4.1. Keşif ve Sömürü .....	76
3.4.2. Yerel ve Küresel Arama.....	77
3.4.3. Tek ve Popülasyon Bazlı .....	77
3.5. Metasezgisel Algoritmaları ile Özellik Seçimi Yapmak .....	78
3.5.1. Kodlamak.....	78
3.5.2. Uygunluk Fonksiyonu.....	79
3.5.3. Prosedür .....	79
3.6. Tezde Kullanılan Metasezgisel Algoritmalara Genel Bakış.....	79
3.6.1. Genetik Algoritmasına Genel Bakış .....	80
3.6.1.1. Seçilim .....	81
3.6.1.2. Çaprazlama .....	82
3.6.1.3. Mutasyon .....	83
3.6.2. Parçacık Sürüsü Optimizasyonu Algoritmasına Genel Bakış.....	84
3.6.2.1. Parçacık Sayısı.....	86
3.6.2.2. Yineleme Sayısı .....	87
3.6.2.3. Hız Bileşenleri .....	87

3.6.2.4. İvme Katsayıları.....	87
3.6.3. Benzetimli Tavlama Algoritmasına Genel Bakış.....	87
3.6.4. Diferansiyel Evrim Algoritmasına Genel Bakış .....	89
3.6.4.1. Mutasyon .....	90
3.6.4.2. Çaprazlama .....	90
3.6.4.3. Seçilim .....	91
3.6.5. Karınca Koloni Optimizasyonu Algoritmasına Genel Bakış .....	91
3.6.6. NSGA II Algoritmasına Genel Bakış.....	93
<b>4. UYGULAMA.....</b>	<b>96</b>
4.1. Veri Seti.....	96
4.1.1. Vücut Yağ Tahmini Veri Seti .....	96
4.2. Bulgular .....	96
4.2.1. Yapay Sinir Ağı ile Vücut Yağ Tahmini .....	97
4.2.2. Birden Çok Kez Çalıştırmalı Yapay Sinir Ağı ile Vücut Yağ Tahmini.....	99
4.2.3. Seçili Özellik Sayısını Bilmeden Özellik Seçimi Yapmak .....	101
4.2.3.1. Yapay Sinir Ağı ve İkili Genetik Algoritmayı Kullanarak Özellik Seçimi Yapmak.....	102
4.2.4. Daha Önceden Belirlenmiş Özellik Sayısını Bilerek Özellik Seçimi Yapmak .....	104
4.2.4.1. Yapay Sinir Ağı ve Karınca Koloni Optimizasyonu Algoritmasını Kullanarak Özellik Seçimi Yapmak .....	105
4.2.4.2. Yapay Sinir ve Benzetimli Tavlama Algoritmasını Kullanarak Özellik Seçimi Yapmak .....	107
4.2.4.3. Yapay Sinir Ağı ve Parçacık Sürüsü Optimizasyonu Algoritmasını Kullanarak Özellik Seçimi Yapmak .....	110
4.2.4.4. Yapay Sinir Ağı ve Diferansiyel Evrim Algoritmasını Kullanarak Özellik Seçimi Yapmak .....	112
4.2.5. Çok Amaçlı Özellik Seçimi Yapmak.....	114
4.2.5.1. Birden Çok Çalıştırma ile Parçacık Sürüsü Optimizasyonu Algoritması Kullanarak Özellik Seçimi Yapmak .....	115
4.2.5.2. Evrimsel Çok Amaçlı Yaklaşımları ile NSGA II Algoritması Kullanarak Çok Amaçlı Özellik Seçimi Yapmak .....	116
<b>5. SONUÇ .....</b>	<b>119</b>
<b>KAYNAKÇA.....</b>	<b>121</b>
<b>ÖZ GEÇMİŞ.....</b>	<b>145</b>

## ŞEKİLLER DİZİNİ

Şekil 3.1. Özellik alt kümesinin araştırılması .....	13
Şekil 3.2. Özellik seçim sürecindeki temel adımlar .....	14
Şekil 3.3. Özellik seçim kategorisi.....	18
Şekil 3.4. Denetimli özellik seçiminin genel çerçevesi.....	20
Şekil 3.5. Denetimsiz özellik seçiminin genel çerçevesi .....	21
Şekil 3.6. Yarı denetimli özellik seçiminin genel çerçevesi .....	22
Şekil 3.7. Sarmal özellik seçim yöntemlerinin genel çerçevesi .....	24
Şekil 3.8. Filtre özellik seçim yöntemlerinin genel çerçevesi.....	31
Şekil 3.9. Gömülü özellik seçim yöntemlerinin genel çerçevesi .....	52
Şekil 3.10. Hibrit (Sarmal ve Filter) yöntemlerinin genel çerçevesi.....	59
Şekil 3.11. Topluluk özellik seçiminin genel çerçevesi .....	62
Şekil 3.12. İleri beslemeli sinir ağı .....	65
Şekil 3.13. Tekrarlayan sinir ağı.....	66
Şekil 3.14. Genteik algoritmanın genel çerçevesi.....	81
Şekil 3.15. Seçimin temel süreci.....	82
Şekil 3.16. Çaprazlamanın temel süreci.....	83
Şekil 3.17. Mutasyonun temel süreci .....	84
Şekil 3.18. Parçacık sürüsü optimizasyonu algoritmasının genel akış şeması.....	85
Şekil 3.19. Benzetimli tavlama algoritmasının temel akış şeması .....	89
Şekil 3.20. Diferansiyel evrim algoritmasının genel akış şeması .....	90
Şekil 3.21. Karınca koloni optimizasyonu algoritmasının genel akış şeması .....	92
Şekil 3.22. NSGA II algoritmasının genel akış şeması.....	94
Şekil 4.1. Veriler üzerinde yapay sinir ağı diyagramı.....	97
Şekil 4.2. Yapay sinir ağı validation (doğrulama) grafiği.....	97
Şekil 4.3. Yapay sinir ağı test grafiği.....	98
Şekil 4.4. Yapay sinir ağı train (eğitim) grafiği .....	98
Şekil 4.5. Yapay sinir ağı tüm veriler grafiği.....	98
Şekil 4.6. Birden çok kez çalıştırılmalı yapay sinir ağı validation (doğrulama) grafiği.....	99
Şekil 4.7. Birden çok kez çalıştırılmalı yapay sinir ağı test grafiği.....	100
Şekil 4.8. Birden çok kez çalıştırılmalı yapay sinir ağı train (eğitim) grafiği .....	100
Şekil 4.9. Birden çok kez çalıştırılmalı yapay sinir ağı tüm veriler .....	100
Şekil 4.10. Birden çok kez çalıştırılmalı yapay sinir ağının 10 döngü için mse grafiği.....	101
Şekil 4.11. İkili genetik algoritma ile özellik seçimi, yapay sinir ağı diyagramı .....	102
Şekil 4.12. İkili genetik algoritma ile özellik seçimi, validation (doğrulama) grafiği .....	102
Şekil 4.13. İkili genetik algoritma ile özellik seçimi, test grafiği .....	102

Şekil 4.14. İkili genetik algoritma ile özellik seçimi, train (eğitim) grafiği.....	103
Şekil 4.15. İkili genetik algoritma ile özellik seçimi, tüm veriler grafiği .....	103
Şekil 4.16. İkili genetik algoritma ile özellik seçimi, her yinelemede maliyet grafiği .....	103
Şekil 4.17. Daha önceden belirlenmiş özellik sayısı ile özellik seçimi, yapay sinir ağı diyagramı .....	105
Şekil 4.18. Karınca koloni optimizasyonu algoritması ile özellik seçimi, validation (doğrulama) grafiği .....	105
Şekil 4.19. Karınca koloni optimizasyonu algoritması ile özellik seçimi, test grafiği.....	106
Şekil 4.20. Karınca koloni optimizasyonu algoritması ile özellik seçimi, train (eğitim) grafiği .....	106
Şekil 4.21. Karınca koloni optimizasyonu algoritması ile özellik seçimi, tüm veriler grafiği .....	106
Şekil 4.22. Karınca koloni optimizasyonu algoritması ile özellik seçimi, her yinelemede maliyet grafiği.....	107
Şekil 4.23. Benzetimli tavlama algoritması ile özellik seçimi, validation (doğrulama) grafiği .....	108
Şekil 4.24. Benzetimli tavlama algoritması ile özellik seçimi, test grafiği .....	108
Şekil 4.25. Benzetimli tavlama algoritması ile özellik seçimi, train (eğitim) grafiği .....	108
Şekil 4.26. Benzetimli tavlama algoritması ile özellik seçimi, tüm veriler grafiği.....	109
Şekil 4.27. Benzetimli tavlama algoritması ile özellik seçimi, her yinelemede maliyet grafiği .....	109
Şekil 4.28. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, validation (doğrulama) grafiği .....	110
Şekil 4.29. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, test grafiği .....	110
Şekil 4.30. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, train (eğitim) grafiği.....	111
Şekil 4.31. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, tüm veriler grafiği .....	111
Şekil 4.32. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, her yinelemede maliyet grafiği.....	111
Şekil 4.33. Diferansiyel evrim algoritması ile özellik seçimi, validation (doğrulama) grafiği .....	112
Şekil 4.34. Diferansiyel evrim algoritması ile özellik seçimi, test grafiği .....	113
Şekil 4.35. Diferansiyel evrim algoritması ile özellik seçimi, train (eğitim) grafiği .....	113
Şekil 4.36. Diferansiyel evrim algoritması ile özellik seçimi, tüm veriler grafiği .....	113
Şekil 4.37. Diferansiyel evrim algoritması ile özellik seçimi, her yinelemede maliyet grafiği .....	114
Şekil 4.38. Birden çok çalıştırma ile parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet grafiği.....	115
Şekil 4.39. Çok amaçlı yaklaşımlar ile NSGA II algoritması kullanarak özellik seçimi, yapay sinir ağı diyagramı .....	116
Şekil 4.40. NSGA II algoritması ile özellik seçimi, validation (doğrulama) grafiği .....	116

Şekil 4.41. NSGA II algoritması ile özellik seçimi, test grafiği.....	116
Şekil 4.42. NSGA II algoritması ile özellik seçimi, train (eğitim) grafiği.....	117
Şekil 4.43. NSGA II algoritması ile özellik seçimi, tüm veriler grafiği .....	117
Şekil 4.44. NSGA II algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet grafiği .....	117



## TABLolar

Tablo 3.1. Özellik seçiminin genel algoritması .....	12
Tablo 3.2. Arama stratejilerin sınıflandırılması .....	16
Tablo 3.3. Farklı arama stratejilerinin karşılaştırılması .....	17
Tablo 3.4. Sarmal yönteminin genel algoritması .....	24
Tablo 3.5. Ardışık ileri yönde seçiminin genel algoritması .....	26
Tablo 3.6. Ardışık geri yönde seçiminin genel algoritması .....	27
Tablo 3.7. Ardışık ileri yönde kayan seçiminin genel algoritması.....	27
Tablo 3.8. L ekler – R çıkar yönteminin genel algoritması .....	28
Tablo 3.9. Çift yönlü arama yönteminin genel algoritması .....	29
Tablo 3.10. Filtre yönteminin genel algoritması .....	31
Tablo 3.11. Sınırsız Özellik Seçiminin Genel Algoritması.....	44
Tablo 3.12. Hibrit Yönteminin Genel Algoritması .....	59
Tablo 3.13. Özellik değerlendirme yöntemlerinin avantajları ve dezavantajları .....	62
Tablo 4.1. Vücut yağı veri setinin özellikleri.....	96
Tablo 4.2. Vücut yağı veri seti üzerinde yapay sinir ağı değerleri.....	99
Tablo 4.3. Vücut yağı veri seti üzerinde birden çok kez çalıştırılmalı yapay sinir ağı değerleri .....	101
Tablo 4.4. Vücut yağı veri seti üzerinde yapılan İkili genetik algoritma ile özellik seçimi değerleri .....	104
Tablo 4.5. Vücut yağı veri seti üzerinde yapılan karınca koloni optimizasyonu algoritması ile özellik seçimi değerleri.....	107
Tablo 4.6. Vücut yağı veri seti üzerinde yapılan benzetimli tavlama algoritması ile özellik seçimi değerleri.....	109
Tablo 4.7. Vücut yağı veri seti üzerinde yapılan parçacık sürüsü optimizasyonu algoritması ile özellik seçimi değerleri .....	112
Tablo 4.8. Vücut yağı veri seti üzerinde yapılan diferansiyel evrim algoritması ile özellik seçimi değerleri.....	114
Tablo 4.9. Birden çok çalışma ile parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet değeri.....	115
Tablo 4.10. NSGA II algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet değeri .....	118

# 1. GİRİŞ

Modern dünyadaki veriler gittikçe daha yüksek boyutlu hale gelmektedirler. Özellikle yaygın bir örnek vermek gerekirse, biyoinformatik dünyasında veriler hızlı bir biçimde büyümektedir (Tarczy-Hornoch ve Minie, 2005). Son zamanlarda, metin madenciliği, bilgisayarla görme ve biyomedikal gibi birçok makine öğrenme uygulamasında hem örnek sayısı hem de boyutsallık açısından, veriler hızla artmaktadır (Miao ve Niu, 2016). Ayrıca gün geçtikçe elde edilen veriler, yüksek üretim teknolojilerinin büyümesi sebebiyle büyük örneklem büyüklüğü ile birlikte yüksek boyutlu bir duruma gelmektedir. Bundan dolayı veri madenciliği, büyük veri kümelerinden gizli bilgileri keşfetmek için çok önemli bir rol oynamaktadır (Jain ve Chandrasekaran, 1982). Araştırmacılar ve uygulayıcılar, herhangi bir bilgi keşfi yapmadan önce veya verilerin görselleştirme algoritmalarına sunmadan önce, veri madenciliğinin önemli bir aşamalarından birisi olan veri ön işleme tekniklerinin kullanma gereğini farketmişlerdir (Han vd, 2011; Liu ve Motoda, 1998).

Veriler katlanarak arttıkça, veri madenciliği, Örüntü Tanıma, Görüntü işleme ve diğer Makine Öğrenimi algoritmaları tarafından işlenmek için gereken verilerin kalitesi yavaş yavaş düşmeye başlar. Bellman bu senaryoya “Boyutsallığın Laneti” diye adlandırmaktadır (Bellman, 1957). Bilgisayarların gücü arttıkça ve veri toplama teknolojileri geliştikçe, bilgisayarların kullanıldığı her alanda çok miktarda veri üretilmektedir. Bilgisayar tarafından oluşturulan veriler bilgisayarlar tarafından analiz edilmelidir. Bilgi işlem teknolojilerinin yardımı olmadan, toplanmış büyük miktardaki veriler, hiçbir zaman incelenmeyeceği kesindir. Günümüzün gelişmiş bilgisayar teknolojilerinde bilgisayar tarafından üretilen verilerden bilgi keşfetmekte zorlanabilirler. En basit şekliyle veri kümesinin boyutu iki boyutta ölçülebilir, özellik sayısı (N) ve örnek sayısı (P). Hem N hem de P çok büyük olabilir. Bu büyüklük birçok veri madenciliği sisteminde ciddi problemlere neden olabilir (Liu ve Motoda, 2012). Bu problemlerin üstesinden gelmek için “Boyutsal Küçültme” teknikleri uygulanmaktadır ve bu veri ön işleme aşamasının bir bölümüdür (Tsai ve Chou, 2011). Bundan dolayı büyük miktarda olan verileri veya çok boyutlu olan verileri işlemek için boyutları küçültmemiz çok önemlidir (Sugiyama, 2015) ve ayrıca özellik seçimi, boyut indirgeme problemlerini çözmekte yardımcı olacak yöntemlerden birisidir (Liu ve Motoda, 2012).

Özellik seçimi, veri ön işlemede en sık kullanılan ve en önemli tekniklerden birisidir. Ayrıca özellik seçimi makine öğrenimi sürecinin vazgeçilmez bir bileşeni haline dönüşmüştür (Kalousis vd, 2007). Özellik seçme problemi uzun zamandır istatistik ve makine öğrenimi toplulukları tarafından incelenmektedir. Son zamanlarda veri madenciliğinde hızla artan araştırmalar nedeniyle daha fazla ilgi görmektedir (Mlambo vd, 2016) ve ayrıca son yirmi yılda özellik seçme üzerinde kapsamlı çalışmalar yapılmıştır (Cho vd, 2008; Ganster vd, 2001; Inza vd, 2000; Lee vd, 2003; Sierra vd, 2001). Özellik seçme yöntemi sadece verilerin boyutsallığını indirmekle kalmaz, aynı zamanda hesaplama maliyetinide düşürür ve iyi bir sınıflandırma performansını ortaya koyar (Hua vd, 2009). Özellik seçiminde (Shardlow, 2016) yaptığı araştırmada birkaç amacımız olduğunu vurgulamıştır :

- Algoritmamızın kullanım amacı problemimizin boyutunu, hesaplama süresini ve alanını azaltmaktır.
- Sınıflandırıcıları geliştirmek için gürültülü veya alakasız özellikler, kaldırılır veya kullanım alanları azaltılır.
- Hangi özelliklerin hangi soruna ait olduğunu anlayabilmek. Örneğin belirli bir hastalıkta hangi genlerin alakası olduğunu anlamak.

Kısacası özellik seçimi sınıflandırıcının doğruluğunu en üst seviyeye çıkartmakta, ölçüm maliyetlerini an aza indirmekte, alakasız ve gereksiz özellikleri azaltarak doğruluğu artırmakta, bir çözümün anlaşılır ve gerçekçi olma olasılığını yükseltmekte, ortak amacını karşılamakta kullanılır (Steppe ve Bauer Jr, 1997).

Gerçek dünya problemlerinde alakasız ve gereksiz veriler aşırı hesaplama karmaşıklığına neden olur ve algoritmanın verimliliğini azaltır. Veri madenciliğinin temel adımlarından biri olan özellik seçimi, veri kümelerin boyutunu azaltarak algoritmanın performansını iyileştirir; ancak özellik seçme algoritmalarının tek başına optimum bir sonuç bulamama sebebi çözüm alanında kapsamlı bir arama yapılmamasıdır (Yusta, 2009). Sezgisel yöntemler genellikle birçok problemin çözümünde iyi ve kabul edilebilir bir cevap bulmasına rağmen karmaşık problemlerin çözümü için uygun cevap bulamayabilir (Chinneck, 2006) ve ayrıca sezgisel teknikler büyük oranda model geliştiricinin deneyimine ve matematiksel bilgilerine bağlıdır (Coello vd, 2007); ancak metasezgisel tarama genellikle basit sezgisel tarama ile karşılaştırıldığında daha iyi performans göstermektedir. Temel olarak metasezgisel algoritmaları bazı randomizasyon ve yerel arama teknikleri kullanırlar (Yang, 2010a).

Metasezgisel algoritmaları daha kapsamlı bir arama yaptıklarından dolayı ve aynı zamanda diğer yöntemlerin çözüm üretilmediği karmaşık problemlere çözüm ürettikleri için hızla tercih edilen yöntemler haline gelmişlerdir (Glover ve Kochenberger, 2006). Bundan dolayı yüksek doğrulukta cevaplar bulmak için metasezgisel algoritmaları kullanılmaktadır.

Bu tezde öncelikle özellik seçme problemlerinde kullanılan algoritmalar ve yöntemler anlatılmaktadır. Daha sonra özellik seçme problemlerinde optimum sonuç elde edebilmek için kullanılan metasezgisel algoritmalar (ikili genetik, karınca koloni optimizasyonu, benzetimli tavlama, parçacık sürüsü optimizasyonu, diferansiyel evrim, baskın olmayan sıralı genetik algoritması II) bahsedilmektedir.



## 2. LİTERATÜR ARAŞTIRMASI

### 2.1. Özellik Seçimi ile İlgili Literatür Çalışmaları

Literatürde, özellik seçiminin uygulamaları birçok araştırma alanında sıklıkla kullanılmaktadır. Aşağıdaki kısımlarda her bir uygulama alanı özet olarak anlatılmıştır:

- Metin Sınıflandırması

Otomatik metin sınıflandırması, modern veri analizinde hem deneysel olarak hem de teorik olarak zor bir görevdir. Bu problem birçok internet uygulamasında önemli bir rol oynamaktadır ve bundan dolayı bilgi erişimi, makine öğrenimi ve algoritma teorisi gibi çeşitli alanlarda araştırmacıların dikkatini çekmiştir. Otomatik metin sınıflandırma ile ilgili zorluklar birçok yönden değerlendirilebilir: belgelerin temsili için uygun bir veri yapısı seçilmelidir; aşırı devreden kaçınmak ve iyi bir genelleme ve optimizasyon elde etmek için uygun bir objektif fonksiyon seçilmelidir (Dasgupta vd, 2007).

Forman 2003 yılında (Forman, 2003), metin sınıflandırması için filtre özellik seçim tekniğinin detaylı bir deneysel çalışmasını yapmıştır. Bu çalışmada Forman 229 metin sınıflandırma problemi örneği üzerinde on iki özellik seçim mesafe ölçüsünü uygulamıştır. Performansı değerlendirmek için ölçülmemiş parametrelere sahip doğrusal bir destek vektör makine sınıflandırıcısı kullanılmıştır. Sonuçlar hassasiyet, hatırlama, F-ölçümü ve doğruluk açısından analiz edilerek, bilgi kazanımının hassasiyet açısından en iyi performansı ve bahsedilen teknikte iki normal ayırmanın geri çağırma, F-ölçümü ve doğruluk için en iyi performansı gösterdiği kanıtlanmıştır.

- Metin Kümelemesi

Özellik seçim teknikleri metin sınıflandırmada başarıyla uygulanmış, ancak sınıf etiketi bilgilerinin bulunmamasından dolayı metin kümelemesine nadiren uygulanmıştır. Metin kümeleme, metin madenciliği ve bilgi alma alanındaki esas problemlerden birisidir. Metin kümelemenin görevi benzer belgeleri birlikte gruptandırmaktır. Bu yöntemde, bir metin veya belge her zaman bir kelime torbası olarak farz edilir; ancak özellik alanının yüksek boyutsallığı ve doğal veri seyrekliği nedeniyle kümeleme algoritmalarının performansı önemli ölçüde düşecektir. Bundan

dolayı, özellik alanı boyutluluğunun azaltılması son derece önemlidir(Aggarwal ve Yu, 2000).

Liu ve arkadaşları (Liu vd, 2003), metin kümeleme probleminde özellik seçim yöntemlerini kullanarak performansın ve verimliliğini artırabileceğini göstermişler. Üç filtre veri seti üzerinde beş filtre özelliği seçim yöntemi test etmişler ve denetimsiz özellik seçim yöntemlerinin, kümeleme performansını artırdığını kanıtlamışlar.

- Mikrodizi Veri Analizi

Neredeyse tüm biyoinformatik problemlerinin sayısı örnek sayısından aşırı ölçüde daha fazladır. Örnek olarak mikrodizi verilerine dayalı meme kanseri sınıflandırmasında, tüm genler hakkında bilgi gerekli olmayabilir ve bundan dolayı özellik seçim yöntemleri kullanılmaktadır (Bontempi ve Haibe-Kains, 2008).

Genellikle, mikrodizi gen seçim verileri yüzlerce ve binlerce özellikten oluşmaktadır. Bundan dolayı bu veriler, öğrenme modelleri için zorlayıcı olur. Bu yüzden verilerin boyutlarını azaltmaya ihtiyaç vardır. Ang ve arkadaşları 2015 yılında (Ang vd, 2015) denetimli, denetimsiz ve yarı denetimli tabanlı öğrenme modelleri için çeşitli gen seçim yöntemlerini açık bir şekilde anlatmışlardır. Mandal ve Mukhopadhyay 2013 yılında (Mandal ve Mukhopadhyay, 2013), gen ifade verileri için geliştirilmiş bir mRMR özellik seçim yöntemi önerdiler. Literatürde, yöntemlerin çoğu artıklık veya alaka düzeyi özellik seçim yöntemlerini kullanırken Mandal ve Mukhopadhyay önerdikleri yöntem, işten çıkarmanın ve alaka düzeylerini paralel olarak değerlendirir.

- Hata Tahmini

Güvenilirlik, işlevsellik, hata eğilimi, tekrar kullanılabilirlik, anlaşılabilirlik gibi çeşitli yazılım kalite güvence özellikleri mevcuttur. Özellik seçim yöntemleri, endüstriyel uygulamalarda çok sayıda yedek sensörlü olan bir makinenin performans hata teşhisinde kullanılır. Liu ve arkadaşları 2014 yılında yaptıkları araştırmada (Liu vd, 2014), bir hata tespitinin doğruluğunu, özellik seçimi kullanılarak iyileştirilebileceğini göstermişler. Bu araştırmada yazarlar hata teşhisinde özellik seçimi için küresel bir geometrik model ve benzerlik ölçümü kullanmayı önerdiler. Yazarlar açısız benzerlik, karşılıklı bilgi ve yapı benzerlik endeksi ile deneme yapmışlar. Aynı zamanda önerilen yaklaşım, mesafeye dayalı ve entropiye dayalı özellik seçimi, dahası destek vektör makinesi ve sinir ağı sarmalayıcıları ile

karşılaştırılmıştır. En iyi performans, önerilen geometrik benzerlik yaklaşımının bir sarmal yönteminin birleştirilmesiyle elde edildi.

- İzinsiz Giriş Tespiti

Günümüzde, bilgi paylaşımı, dağıtımı veya iletişim, ağ tabanlı bilgisayar sistemleri ile yaygın bir şekilde yapılmaktadır. Bu yüzden, sistemin güvenli olması ve iletişim ağlarını, düşmanların ve suçluların izinsiz girişine engel olabilmesi önemli bir konudur. İletişim ağlarının koruma yollarından birisi saldırı tespitidir. Özellik seçimi yöntemleri, sistem etkinliğini izinli veya izinsiz giriş olarak sınıflandırmak için önemli bir rol oynamaktadır. Bu sorunun üstesinden gelmek için yüksek güvenli bir saldırı tespit sistemi inşa edilmelidir. Bu sistemin gürültülü, gereksiz ve alakasız veriler içeren yüksek veri boyutları ile ilgilenmesi gerekmektedir. Dolayısıyla, izinsiz giriş algılama oranında bir azalmaya yol açar ve daha fazla hesaplama süresi gerektirir. Bu yüzden, yüksek tespit oranı elde etmek için özellik seçim yöntemlerine ihtiyaç duyulur. Amiri ve arkadaşları 2011 yılında (Amiri vd, 2011) saldırı tespit sistemi için karşılıklı bilgi bir filtre yöntemi yaklaşımı kullanarak bir özellik seçim yöntemi önermişler. Ayrıca Chen ve arkadaşları 2006 yılında (Chen vd, 2006) saldırı tespit sistemi için mevcut olan farklı özellik seçimleri hakkında araştırma yapmışlar. Lappas ve arkadaşları 2007 yılında (Lappas ve Pelechrinis, 2007) izinsiz giriş tespiti için veri madenciliği teknikleri ve özellik seçim yöntemlerini kullanmışlar. Bu çalışmada yazarlar kullandıkları tekniklerin, avantajları ve dezavantajları hakkında kıyaslamalı bir çalışma gerçekleştirdiler.

- Genomik Analiz

Bir organizmanın işlevini ve hastalıkların davranışını, ve özelliklerini anlamak için mikrodizi ve kütle spektrometrisi teknolojisi tarafından büyük ölçüde genomik ve proteomik veri üretilir. On binlerce gen tipik bir mikrodizi deneyinde ve kütle spektrometresi proteomik profilinde ölçülür. Mikrodizi veriler, yüksek boyutlu oldukları için özel veri analizi talep edilmektedir. Yüksek boyutsallığı analiz etme yollarından biri, verilerdeki en alakalı özelliklerin tanımlanmasıdır. Dolayısıyla, literatürde, tam mikrodizi verileri üzerinde özellik seçim yöntemleri başarıyla uygulanmıştır. (Hauskrecht vd, 2007)'de, özellik seçimi için filtre, sarmal ve gömülü yöntemler kullanılmıştır. Bu çalışmada kullanılan teknikler proteomik veriler ve genomik analiz için çok etkilidir. (Abusamra, 2013)'de, sınıflandırma görevi ve

kombinasyonları için 8 özellik seçiminin kıyaslamalı çalışmaları gen ekspresyon verilerine dayanılarak yapılmıştır. Ayrıca, sınıflandırma doğruluğunun, bir özellik seçme yöntemi kullanılarak az sayıda gen tarafından önemli ölçüde artırılacağı gösterilmiştir.

- Hiperspektral Görüntüler

Standart görüntüde sadece RGB spektral bantlar bulunurken, hiperspektral görüntülerde yüzlerce spektral bant bulunmaktadır. Dolayısıyla, her piksel nesnelere için kullanılır. Bu hiperspektral görüntüler uzaktan algılama, tıbbi görüntüleme ve benzeri uygulamalarda yaygın olarak kullanılmaktadır. Hiperspektral görüntülerde, veriler farklı uygulamalar için zengin bilgiler içerir, ancak belirli bir uygulama için tüm önlemler çok önemli değildir. Çok sayıda spektral bantın olmasından dolayı, bu bantlar arasında fazlalık varlığına yol açar. Bu sebeple, fazla ve gereksiz bantların ortadan kaldırılması için özellik seçim yöntemine ihtiyaç vardır. Bundan dolayı, özellik seçme teknikleri, ilgili özellik alt kümesini seçmek için gereklidir. Jia ve arkadaşları tarafından 2013 yılında yapılan çalışmada (Jia vd, 2013) çeşitli konular ve özellik seçiminde hiperspektral görüntü analizi için karşılaşılan zorluklar hakkında bilgi verilmektedir.

Kuo ve arkadaşları 2004 yılında (Kuo ve Landgrebe, 2004) yaptıkları çalışmada, özellik seçiminin hiperspektral görüntü sınıflandırmasında önemli bileşenler olarak vurgulamışlar. Ayrıca hiperspektral görüntüler için ön işleme teknikleri önerilmiştir. Ghosh ve arkadaşları 2013 yılında (Ghosh vd, 2013) evrimsel algoritmalar tarafından yönlendirilen özellik seçimi önerilmiştir ve özellik alt kümesi üretimi için kendinden uyarlamalı diferansiyel evrim kullanır. Üretilen özellik alt kümeleri, bulanık k-en yakın komşu sınıflandırıcısı ve sarmal yöntemi yardımıyla değerlendirilir. Zhao ve arkadaşları 2010 yılında yaptıkları çalışmada (Zhao vd, 2010a) sınıflandırmanın performansını arttırmak için hiperspektral bir görüntüden bir bant alt kümesi seçmek için yeni bir teknik önermişler.

- Görüntü Sınıflandırması

Görüntü özelliklerinin sayısı sınırlı olmadığından dolayı görüntüleri temsil etmek zor bir iş (Bins ve Draper, 2001). Özelliklerin seçimi tipik olarak hedef uygulamaya bağlıdır. Özellik örnekleri, yönlendirilmiş eğim histogramlarını, kenar yönlendirme histogramlarını, Haar dalgacıklarını, ham pikselleri, eğim değerlerini,

kenarları, renk kanallarını ve bunlara benzer örnekleri İçerir (Brkić vd, 2011). Bins ve Draper 2001 yılında (Bins ve Draper, 2001) genel görüntü sınıflandırması probleminde, filtre özellik seçim yöntemlerini üç farklı veri setinde kullanmışlar. Muştra ve arkadaşları 2012 yılında yaptıkları araştırmada (Muştra vd, 2012) mamografik görüntülerde meme yoğunluğu sınıflandırması için sarmal özellik seçim yöntemini uygulamışlar. Bu araştırmada mamografik görüntülerin iki veri seti üzerinde, üç farklı sınıflandırıcı ile birlikte beş sarmal özellik seçim yöntemi değerlendirildi ve ileri seçim ile en iyi ilk arama ve geri seçim ile en iyi ilk arama, en iyi performansı gösterdiler.

## **2.2. Metasezgisel Algoritmaları ile Özellik Seçimi Konusunda Literatür Çalışmaları**

Bazı araştırmacılar, değerlendirmeler için özellik alt kümesini oluşturmak için benzetilmiş tavlama aramasını kullanırlar. Örnek olarak, Lin ve arkadaşları öznelik alt kümelerini oluşturmak için benzetilmiş tavlama aramasını kullandılar ve bunları, daha iyi öznelik alt kümesini seçmek için denetimli öğrenme algoritması, yani geri yayılma ağı ile değerlendirdiler (Lin vd, 2008).

Meiri ve Zahavi 2006 yılında, pazarlama uygulaması için benzetilmiş tavlama tabanlı özellik seçimini kullanmışlar. Bu çalışma, özellik seçimi problemini çözmek için aşamalı regresyon ve benzetilmiş tavlama algoritmalarını karşılaştırmıştır. Benzetilmiş tavlama algoritmasının adım adım regresyon algoritmalarından biraz daha iyi olmasına rağmen, her iki algoritma da hemen hemen aynı çözümü verdiği için çıktı sonuçları arasında temelde önemli bir fark olmadığı görülmüştür. Bu modeller arasındaki en büyük fark, algoritmanın kararlılığında yatmaktadır. benzetilmiş tavlama algoritması daha kararlı olduğu ve optimizasyon parametrelerindeki değişikliklere neredeyse duyarsız olduğu bulundu (Meiri ve Zahavi, 2006).

Zhang ve Sun 2002 yılında, optimal özellik seçim problemini çözmek için tabu aramasını kullandılar. Bu yöntemde, tabu aramasıyla oluşturulan alt kümeler, daha iyi özellik alt kümesini bulmak için sınıflandırma hatası kriterleri kullanılarak değerlendirildi. Deneysel sonuçlar çok ümit vericiydi. Dolayısıyla tabu araştırmasının sadece optimal veya en uygun çözüme yakın bir çözümü elde etme olasılığına sahip olmadığını, aynı zamanda dal ve sınır yönteminden daha az hesaplama süresi

gerektirdiğini ve şu anda kullanılan diğer çoğu alt yöntemlerden daha az hesaplama süresi gerektirdiğini görüldü (Zhang ve Sun, 2002).

Tahir ve arkadaşları 2007 yılında, tabu aramasını kullanarak özellik alt kümelerini oluşturdular, daha sonra bu alt kümeler, önemli özellik alt kümesini elde etmek için değerlendirme kriteri olarak sınıflandırma hatası ile K-en yakın komşu sınıflandırıcı kullanılarak değerlendirildi. Bu yöntem, sınıflandırma doğruluğunu iyileştirmede etkili olduğu kanıtlanmış ve çeşitli veri setlerinde farklı sınıflandırıcılarla karşılaştırılmıştır. Sonuçlar, eşzamanlı özellik seçimi ve çıkarımının sadece daha yüksek sınıflandırma doğruluğu ile sonuçlanan K-NN sınıflandırıcısı için ağırlık bulma yeteneğine sahip olmadığını, aynı zamanda özellik vektörünün boyutunu azaltma yeteneğine sahip olduğunu göstermiştir. Önerilen yöntem, biyotıpta önemli bir sorun olan multispektral görüntüleri kullanarak prostat kanserinin sınıflandırma doğruluğunu geliştirmek için de kullanılmaktadır (Tahir vd, 2007).

Aghdam ve arkadaşları 2009 yılında, özellik alt kümelerini oluşturmak için karınca kolonisi optimizasyonunu kullandılar ve metasezgisel algoritmaya dayalı yeni bir optimal özellik seçme tekniği önerdiler. Aynı zamanda bu yöntemin performansını metin sınıflandırmasındaki diğer özellik seçim yöntemleriyle karşılaştırdılar. Karınca koloni optimizasyonu, hızlı bir şekilde yakınlaşma yeteneğine ve problem alanında güçlü bir arama kabiliyetine sahip olduğu görüldü; bundan dolayı minimum özellik alt kümesini verimli bir şekilde bulabildiği ortaya çıktı (Aghdam vd, 2009).

Kanan ve Faez 2008 yılında, yüz tanıma sistemi için karınca kolonisi optimizasyonunu kullanan bir özellik seçme yöntemi önerdiler. Bu yaklaşımda, en yakın komşu sınıflandırıcı, karınca kolonisi optimizasyonuna dayalı öğrenmeyi kullanarak oluşturulan alt kümeyi değerlendirmek için kabul edilir. Önerilen algoritmanın faydasını göstermek ve karınca koloni tabanlı özellik seçim yöntemleriyle karşılaştırmak için, bir yüz tanıma sisteminde iki farklı özellik seti kullanılarak iki deney seti gerçekleştirildi. Önerilen yaklaşımın, daha az sayıda özellik bileşeni ile diğer karınca koloni optimizasyonu ve genetik algoritma tabanlı özellik seçimi yaklaşımlarından tutarlı bir şekilde daha üstün performans göstermesi çok ümit verici oldu. Bu araştırma, önerilen karınca koloni optimizasyon tabanlı özellik seçimi yaklaşımının sınıflandırma doğruluğunu iyileştirdiğini ortaya koymaktadır (Kanan ve Faez, 2008).

Sivagaminathan ve Ramakrishnan 2007 yılında, tıbbi teşhis sistemi için yapay sinir ağları ile karınca kolonisi optimizasyonuna dayalı özellik seçimi geliştirdiler. Bu yöntemde, oluşturulan özellik alt kümeleri yapay sinir ağı kullanılarak doğrulanır. Dolayısıyla karınca algoritmasının, yapay sinir ağı örüntü sınıflandırıcısının tümevarımsal öğreniminde özellik alt kümesi seçim problemini çözmek için iyi bir yaklaşım olduğunu göstermektedir (Sivagaminathan ve Ramakrishnan, 2007).

Sreeja ve Sankar 2015 yılında, veri kümelerinin sınıflandırılması için örüntü eşleştirme tabanlı Sınıflandırma algoritması önerdiler. örüntü eşleştirme tabanının sınıflandırma doğruluğunu artırmak için, bu yöntemeye dayalı bir karınca kolonisi optimizasyonu tabanlı özellik seçimi algoritması önerildi. Diğer örnek tabanlı yöntemlere kıyasla bu yöntemin avantajı, yüksek performansla birlikte basit sınıflandırma prosedürüdür (Sreeja ve Sankar, 2015).

Welikala ve arkadaşları 2015 yılında, tıbbi veri setinin madenciliği için destek vektör makinesi ile genetik algoritma kullanan bir özellik seçimi sundular. Bu yöntemde parlak doku buzuntuları, koyu doku bozuntuları ve yansıma yapay dokuları verilen yanlış yanıtları azaltırken yeni damarların varlığını tespit edebilen otomatik bir sistem ortaya koyuldu. Bu yöntemde yazarlar, özellik seçimi ve destek vektör makinesi parametre seçimini gerçekleştirmek için genetik algoritmaya dayalı bir sistemin kullanılmasını önerdiler (Welikala vd, 2015).

Oreski ve arkadaşı 2014 yılında, kredi riski değerlendirmesi için sinir ağları ile genetik algoritmaya dayalı bir özellik seçme yöntemi önerdiler. Bu yöntemde, uzmanların önceki deneyimlerini ve özellik sıralaması için hızlı algoritmaların verimliliğini ve bir genetik algoritmanın optimizasyon kabiliyeti kullandılar. Arama alanı azaltma, azaltılmış özellik alt kümesinin iyileştirilmesi ve artımlı aşamaları içeren üç aşamalı bir hibrit algoritma tasarlandı ve bu tasarlanan hibrit algoritmanın verimliliğini tahmin etmek için, iki gerçek dünya kredi veri kümesinde, biri Hırvat ve diğeri Almanya olmak üzere başarıyla uygulandı (Oreski ve Oreski, 2014).

Das ve arkadaşları 2012 yılında, el yazısı rakam tanıma uygulaması için destek vektör makinesi tabanlı özellik seçimi ile genetik bir algoritma formüle ettiler. Bu yöntemde yüksek düzeyde ayırt edici bilgi içeren yerel bölgelerin optimal bir alt kümesini seçmek için, genetik algoritma tabanlı bir bölge örnekleme stratejisi kullanılmıştır. Bu çalışmada 6000 örnek içeren bir veritabanından el yazısı Bangla

basamaklı örnekler üzerinde değerlendirildi. Genetik algoritma kullanılarak, tanıma performansına önemli bir katkısı olmayan rakam desenlerinin bölgeleri yok edilebilmiştir (Das vd, 2012).

Wang ve arkadaşları 2011 yılında, veri sınıflandırma uygulamaları için özellik seçim sürecinde destek vektör makinesi ile alt küme üretimi için genetik algoritmayı uyguladılar. Bu çalışmada, 13 özellik içeren optimize edilmiş özellik alt kümesini belirlemek için önerilen yöntem başarıyla uygulanmıştır (Wang vd, 2011a).

Xue ve arkadaşları 2012 yılında, sınıflandırma için parçacık sürü optimizasyonu tabanlı bir özellik seçimi tasarladılar. Bu yöntemde, parçacık sürü optimizasyonu tarafından oluşturulan özellik alt kümeleri, denetimli öğrenme algoritması kullanılarak değerlendirildi (Xue vd, 2012).

Chen ve arkadaşları 2012 yılında, uyku bozukluğu teşhis sistemi için parçacık sürüsü optimizasyon araştırmasını kullanan bir özellik seçme yöntemi sundular. Önerilen yaklaşımın etkinliğini değerlendirmek için sekiz yaşam bilimi veri seti kullanıldı. Deneysel sonuçlar, önerilen yöntemin geri yayımlı sinir ağı, öğrenme oranı, destek vektör makinesi ve C4.5 ile karşılaştırıldığında önemli ölçüde daha iyi tahmin performansı sergilediğini gösterdi (Chen vd, 2012).

### 3. YÖNTEM

#### 3.1. Özellik Seçimine Genel Bakış

Özellik seçimi, veri madenciliğinin önemli süreçlerinden birisidir. Bu yöntem veri kümesindeki ilgili özellikleri seçmek için kullanılmaktadır. Ayrıca yüksek boyutlu verilerde ilgili özelliklerin, alakasız özelliklerden ayırma işleminde gerçekleştirir (Ramchandran ve Sangaiah, 2018). Aynı zamanda özellik seçimi boyutsallık azaltma yöntemlerinden birisidir. Alakasız veya gereksiz özellikleri kaldırmak için kullanılır. Dahası bu yöntem, sınıflandırma doğruluğunu artırır. Özellik seçiminin özellik çıkarma yönteminden farkı, özellik seçme teknikleri orijinal kümeden yeni bir geliştirilmiş özellik kümesi elde etmesidir. Seçilen özelliklerin alt kümesi, bazı objektif fonksiyonlar ve önemli kriterler nedeniyle en iyi performansı gösterirler (Dey vd, 2018). Özellik seçiminin genel algoritması Tablo 3.1’de görülmektedir:

Tablo 3.1. Özellik seçiminin genel algoritması

---

Girdiler :

X:	N tane özelliğe sahip bir veri kümesinin özellik kümesi
SG:	Yedek üretici operatörü
E:	Değerlendirme ölçüsü
O:	Durdurma kriterleri

Çıktılar :

$X_{opt}$ :	İsteğe bağlı özellik kümesi veya ağırlıklı özellikler
-------------	---

---

Başlatma:

$X' := \text{Başlangıç\_Noktası}(X);$

$X_{opt} := E'$ 'yi kullanarak  $X'$ 'in en iyisini bulmak;

---

---

Tekrarla :

$X' := \text{Arama\_Stratejisi}(X', \text{SG}(E), X)$ ;

$X_{\text{opt}} : E$ 'ye göre  $X$ 'in En İyisi ;

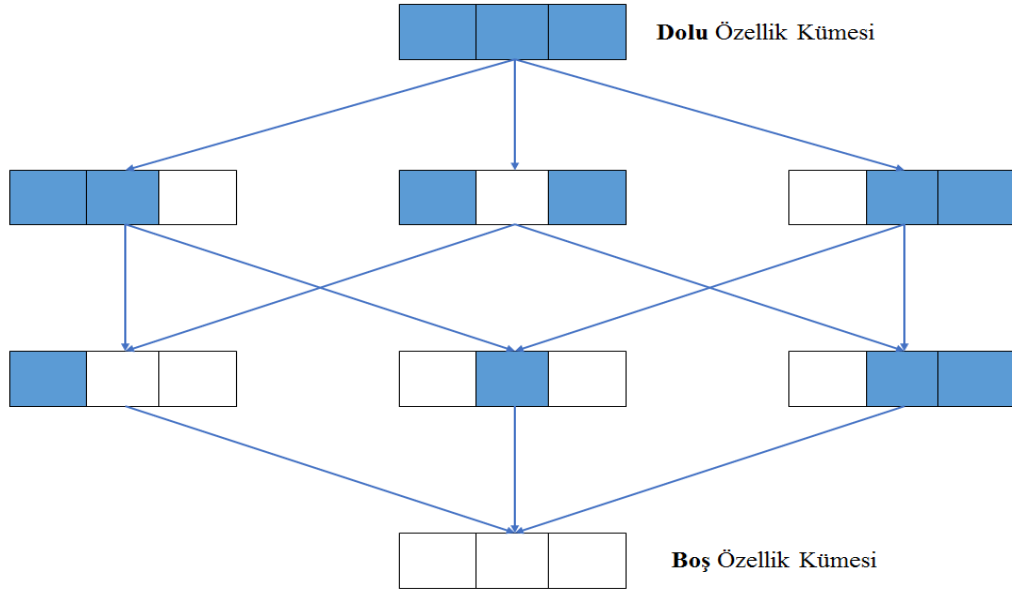
Eğer  $E(X') \geq E(X_{\text{opt}})$  ya da  $(E(X) == E(X_{\text{opt}})$  ve  $|X'| < |X_{\text{opt}}|$ )

O zaman  $X_{\text{opt}} = X'$ ;

Buraya Kadar: Durdurma ölçütleri bulunamadı;

---

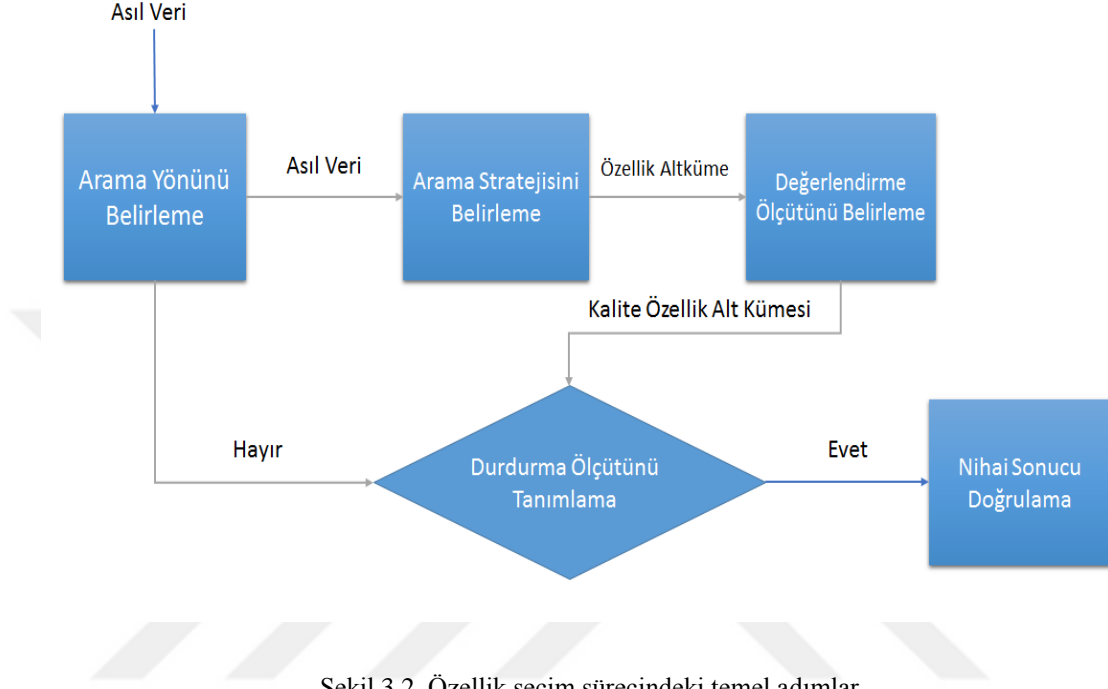
Özellik seçimi, model yapımı için Şekil 3.1'deki gibi ilgili özelliklerin bir alt kümesini seçerek, eğitim sürelerini azaltmada, modelleri basitleştirmede, modellerin yorumunu daha kolay hale getirmede, genelleme şansını artırmada ve aşırı sığmayı önlemede kullanılan bir veri madenciliği tekniğidir (Jimenez-del-Toro vd, 2017). Özellik seçiminin amacı, hesaplama çabalarını ve veri depolamasını kaydederek en önemli bilgileri bulmaktır (Nakamura vd, 2013).



Şekil 3.1. Özellik alt kümesinin araştırılması

### 3.1.1. Özellik Seçiminin Adımları

Orijinal özellik seçme işlemi Şekil 3.2'de gösterildiği gibi beş temel adımdan oluşmaktadır. Her adımda alınan karar özellik seçim performansını etkiler (Langley, 1994). Bu adımlar özet olarak aşağıdaki kısımda anlatılmıştır.



Şekil 3.2. Özellik seçim sürecindeki temel adımlar

#### Adım 1 : Arama Yönünü Belirleme

İlk adım, başlangıç noktası ve arama yönünü belirlemektir. Arama işlemi boş bir kümeyle başlatılabilir ve devamında her yinelemede art arda yeni özellikler kümeye eklenir. Bu stratejiye ileri arama denilir. Buna tersine, arama işlemi tam bir kümeyle başlatılabilir ve daha sonra özellikler her yinelemede art arda kümeden kaldırılır. Bu stratejiye geriye doğru eleme araması denilir. Başka bir alternatif, her yinelemede aynı anda özellikler ekleyip kaldırarak her iki uçla başlamaktır. Bu stratejiye çift yönlü arama denilir. Arama işlemi, alt kümeyi oluşturmak için özellikleri rastgele seçerek de ortada bir yerde başlayabilir (Ang vd, 2015).

#### Adım 2 : Arama Stratejisini Belirleme

İyi bir arama stratejisi, iyi bir küresel aramaya, en uygun çözüme hızlı bir şekilde yaklaşmaya, iyi yerel aramaya ve yüksek hesaplama verimliliğine sahip olması gerekmektedir (Gheyas ve Smith, 2010). Arama stratejileri üstel, ardışık ve rastgele

olarak üç gruba ayrılabilir (Chan vd, 2010; Kumar ve Minz, 2014a; Muni vd, 2006; Ruiz vd, 2005). Aşağıdaki kısımda her biri özet olarak anlatılmıştır.

- Üstel Arama Stratejisi

Üstel arama stratejisi (Narendra ve Fukunaga, 1977) komple arama olarak da adlandırılır ve en kapsamlı küresel arama stratejisidir. Orijinal özellik kümeden başlayarak en iyi sonucu bulmayı garanti eder. Ayrıca, bu strateji, özellikle yüksek boyutlu veri kümeleri için genellikle pratik bir seçenektir, hesaplama açısından yoğundur ve aynı zamanda az sayıda başlangıç özelliği dışında zorlayıcı bir stratejidir. Bu stratejinin bir örneği, tüm olası alt kümeleri değerlendirerek optimum alt kümeyi bulmaktır (Devijver ve Kittler, 1982). Üstel algoritmalar (Liu ve Motoda, 1998) dediklerine göre, özellik alanı boyutuyla katlanarak büyüyen bir dizi alt grubu değerlendirirler. Üstel aramanın dezavantajı,  $N$  özellikleri için  $2N$  özellik seçimi kombinasyonları gerektirmesidir. Bu dezavantajın üstesinden gelmek için araştırmacılar rastgele arama stratejileri kullanırlar (Bins ve Draper, 2001).

- Ardışık Arama Stratejisi

Ardışık arama (Guyon ve Elisseff, 2003) açgözlü tepe tırmanma araması olarak da adlandırılmaktadır. Bu strateji, her seferinde bir özellik ekler veya siler. En yaygın ardışık stratejiler ardışık ileri seçim ve ardışık geri seçimdir. Uygulanması basittir, karmaşıklığı özellik sayısına göre polinomdur ve çoklu eşzamanlılık sorunlarına karşı sağlamdır. Dahası, bu yöntemler monotonik olmayan indekslerde düşük performans gösterirler ve bir özellik eklendiğinde veya silindiğinde, silinmesine veya eklenmesine izin verilmez (Doak, 1992). Ayrıca, özellik etkileşimlerine duyarlıdırlar ve bundan dolayı çok rahat bir şekilde yerel minimaya sığabilirler (Gheyas ve Smith, 2010; Liu ve Motoda, 1998). Silinen özellikleri tekrardan seçmek ve eklenmiş özellikleri silmek için mekanizmalar sağlayarak bu problemleri aşmak için ardışık ileri kayan seçim ve ardışık geriye doğru kayan seçim geliştirilmiştir (Pudil vd, 1994). Ardışık arama stratejisinin diğer bazı örnekleri, en iyi ilk arama (Kohavi ve John, 1997) ve en iyi ilk aramanın optimize edilmiş bir hali yani ışın arama (Doak, 1992) algoritmasıdır (Ferri vd, 1994).

- Rastgele Arama Stratejisi

Rastgele arama stratejisi, özellikleri rastgele seçerek başlar ve ardından iki farklı arama stratejisiyle devam eder. Birincisi klasik ardışık veya çift yönlü aramayı, Örnek

olarak benzetimli tavlama (Doak, 1992) ve rastgele tepe tırmanışı (Skalak, 1994) algortmalarında kullanır. İkincisi, düzenli hareketleri olmayan stratejileri, örnek olarak genetik algoritma (Goldberg, 1989b), Las Vegas algoritması (Brassard ve Bratley, 1996) ve Tabu araması (Glover, 1989) algoritmalarını kullanır (Brassard ve Bratley, 2013). İkinci kullanılan strateji yerel optimadan kaçabilir ve özellikler arasındaki ilişkiyi yakalama mekanizması olmadığından dolayı yanlış sonuçlar üretebilme ihtimali daha yüksek olabilir (Ang vd, 2015). Ayrıca rastgele algoritmalar, arama prosedürlerine yerel minimadan kaçınan rastgele özellikler ekler (Liu ve Motoda, 1998).

Tablo 3.2. Arama stratejilerin sınıflandırılması

Arama stratejisi	Başka isim	Örnek çalışma
Üstel arama	Komple arama	Kapsamlı arama Dal ve sınır (Narendra ve Fukunaga, 1977)
Ardışık arama	Açgözlü tepe tırmanışı	Ardışık ileri seçim Ardışık geri seçim Ardışık ileri kayan seçim (Pudil vd, 1994) Ardışık geriye doğru kayan seçim (Pudil vd, 1994) En iyi ilk arama (Kohavi, 1995) Işın arama (Doak, 1992) Yarış arama L ekler – R çıkar Algoritması (Ferri vd, 1994)
Rastgele arama	Sezgisel arama	Beznetimli tavlama (Doak, 1992) Rastgele tepe tırmanışı (Skalak, 1994) Genetic algoritma (Goldberg, 1989a) Las vegas algoritması (Brassard ve Bratley, 2013)

		<p>Tabu arama (Glover, 1989)</p> <p>Karınca optimizasyonu (Li vd, 2008b)</p> <p>Karmaşık benzetimli tavlama (Chen ve Aihara, 1995; Nozawa, 1992)</p> <p>Gürültülü, karmaşık benzetimli tavlama (Wang vd, 2004)</p> <p>Dal ve sınır (Land ve Doig, 1960; Narendra ve Fukunaga, 1977)</p> <p>Parçacık sürüsü optimizasyonu (Poli vd, 2007)</p>
--	--	--

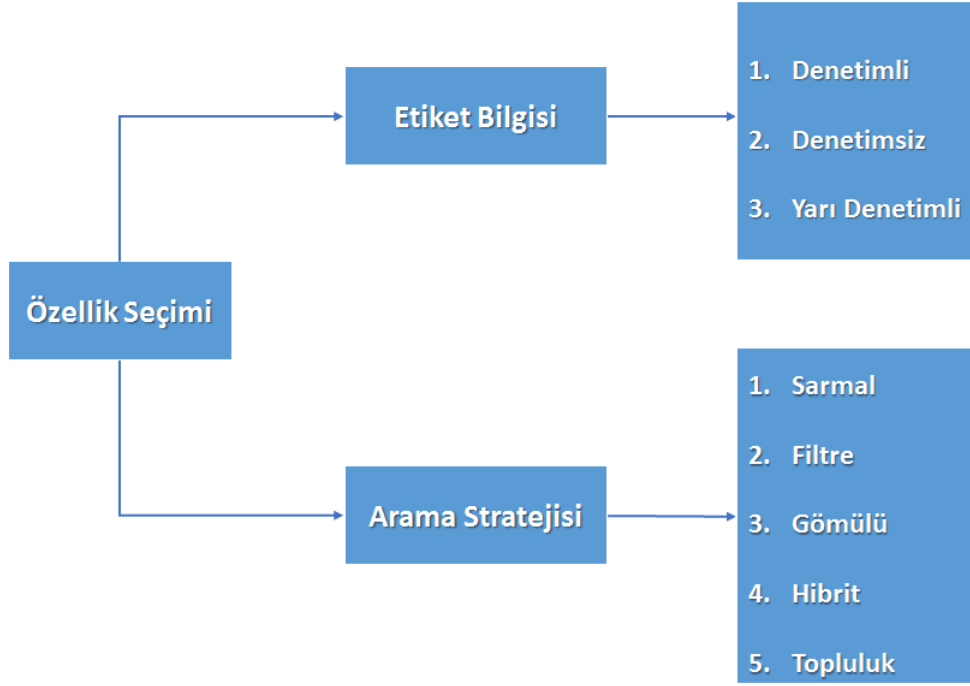
Temel olarak, bir arama stratejisinin seçiminde, optimallik ve hesaplama verimliliğine dikkat edilmelidir. Tablo 3.3, ideal bir arama stratejisini seçmek için optimumluk ve hesaplama verimliliği açısından karşılaştırılmasını göstermektedir.

Tablo 3.3. Farklı arama stratejilerinin karşılaştırılması

Arama stratejisi	Optimallik	Hesaplama verimlilik
Üstel arama	Optimal bir alt kümeye ulaşılması garanti edilir	Yavaş
Ardışık arama	Arama alanından tüm olasılıkları ziyaret etmediği için en uygun alt kümeyi bulamayabilir	Genellikle üstel aramadan daha hızlıdır
Rastgele arama	Optimizasyon, ilgili parametreler için uygun değerlerin belirlenmesine bağlıdır	Genellikle üstel aramadan daha hızlı

### Adım 3 : Değerlendirme Ölçütünü Belirleme

Geleneksel özellik seçme yöntemleri Şekil 3.3'te gösterildiği gibi, etiket bilgisi ve özellik arama stratejisi olarak iki alt başlıkta incelenmektedir.



Şekil 3.3. Özellik seçim kategorisi

#### Adım 4 : Durdurma Ölçütünü Tanımlama

Özellik seçim sürecinin ne zaman duracağını durdurma kriteri belirler . Uygun bir durdurma kriteri, aşırı uymayı önleyerek daha düşük hesaplama karmaşıklığına sahip olur ve böylece optimum özellik alt kümesinin oluşturulmasında daha faydalı bir sürece yol açar. Ayrıca önceki adımlarda verilen kararlar, durdurma kriterlerinin seçimini etkileyecektir. Durdurma kriterlerinin ortak özellikleri aşağıda belirtilmiştir (Ang vd, 2015; Kumar ve Minz, 2014a; Liu ve Yu, 2005):

- Özellik sayısı daha önceden tanımlanır.
- Yineleme sayısı daha önceden tanımlanır.
- İki ardışık yineleme adımındaki ilerleme yüzdesi elde edilir.
- Bazı değerlendirme fonksiyonlarında optimum özellik alt kümesi elde edilir.

#### Adım 5 : Nihai Sonucu Doğrulama

Sonuç doğrulamanın basit bir yolu, daha önceden verilerle alakalı ön bilgiye sahip olarak sonucu elde etmektir. Eğer sentetik verilerdeki gibi ilgili özellikleri önceden biliyorsak, seçilen özellikleri bilinen özellik grubuyla karşılaştırabiliriz. Böylece bilinen alakalı veya alakasız özellikleri, özellik seçim algoritması tarafından üretilen özellik alt kümesiyle karşılaştırarak sonuçları doğrulayabiliriz. Bununla

birlikte, alakasız veya gereksiz özellikler hakkındaki bilgi de yardımcı olabilir. Dahası, gerçek dünyadaki uygulamalarda genel olarak böyle bir ön bilgi mevcut değildir (Liu ve Motoda, 1998; Zeng vd, 2009). Dolayısıyla sonuçların doğrulanması dolaylı yöntemlerle yapılmalıdır ve bu sonuçları doğrulamak için doğrulama teknikleri kullanılır. Çapraz doğrulama, karışıklık matrisi, Jaccard benzerlik temelli ölçü ve rastgele indisi doğrulama yöntemlerinin bazılarıdır. Çapraz doğrulama en yaygın kullanılan doğrulama tekniğidir (Bolón-Canedo vd, 2013). Bu yöntemde, orijinal veri setleri, eğitim seti ve test seti olarak ikiye ayrılırlar. Eğitim seti sınıflandırıcıyı eğitmek için kullanılır ve test seti son değerlendirme için kullanılır. Çapraz doğrulamada, bir veri kümesinden alınan farklı örnekler için değerlendirme süreci tekrarlanır, dolayısıyla ortalama hata tahmini, tüm olası eşit boyutlu örneklerde, tasarlanan sınıflandırıcılar için beklenen hatayı tahim eder. Böylece, çapraz doğrulama etkili bir şekilde tarafsız bir hata tahmini üretme avantajına sahiptir. Ancak bu yöntemin en büyük dezavantajı, hata tahmininin oldukça değişken olabilmesidir (Braga-Neto vd, 2004).

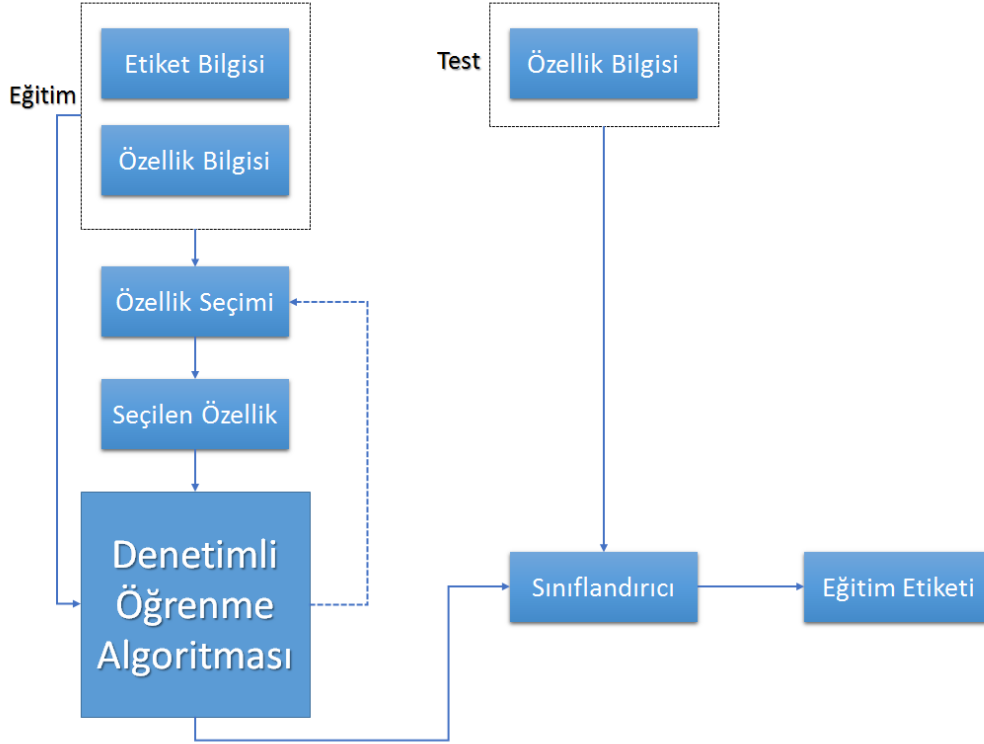
### **3.1.2. Özellik Seçmede Etiket Bilgisi**

Etiket bilgilerinin mevcudiyetine göre, özellik seçim algoritmaları genel olarak denetimli, denetimsiz ve yarı denetimli yöntemler olarak sınıflandırılabilir (Li vd, 2017). Aşağıdaki bölümde her birisi özet bir şekilde anlatılmıştır.

- **Denetimli Özellik Seçimi**

Denetimli özellik seçimi (He vd, 2012; Jian vd, 2016; Wang vd, 1997) en eski ve en yaygın uygulanan etiket bilgi tekniklerinden birisidir (Bø ve Jonassen, 2002). Ayrıca denetlenen özellik seçimi, özellik seçim modelini eğitmek için etiketli verileri kullanarak özelliklerin önemini ve alaka düzeyini ölçmek için bazı ölçütlere dayanan bir özellik alt kümesi seçme işlemidir (Ang vd, 2015). Denetimli yöntemde, özellik ve sınıf arasındaki ilişki, sınıf ve özellik arasındaki korelasyon hesaplanarak değerlendirilir (Kumar, 2014). Bu yaklaşımın en büyük problemlerinden birisi, dışarıdan alınan bilgiler vasıtasıyla verilerin etiketlenmesi sürecinin maliyetli olması ve güvenli olmamasıdır (Kalakech vd, 2011). Dolayısıyla bu uygulama, birçok ilgili özelliği istemeden kaldırarak veya alakasız özellikleri yanlışlıkla seçerek öğrenme sürecini ve aşırı uyma riskini artırır.

Denetlenen özellik seçimi genel olarak sınıflandırma veya regresyon sorunları için tasarlanmıştır. Bu yöntemin amacı, örnekleri farklı sınıflardan ayırabilen özelliklerin bir alt kümesini seçmektir. Denetlenen özellik seçiminin genel bir çerçevesi Şekil 3.4'te gösterilmektedir (Li vd, 2017). Denetimli özellik seçimiyle ilgili literatürde bazı incelemeler vardır (Ahmad vd, 2008; Alelyani vd, 2018; George ve Raj, 2011; Tang vd, 2014).

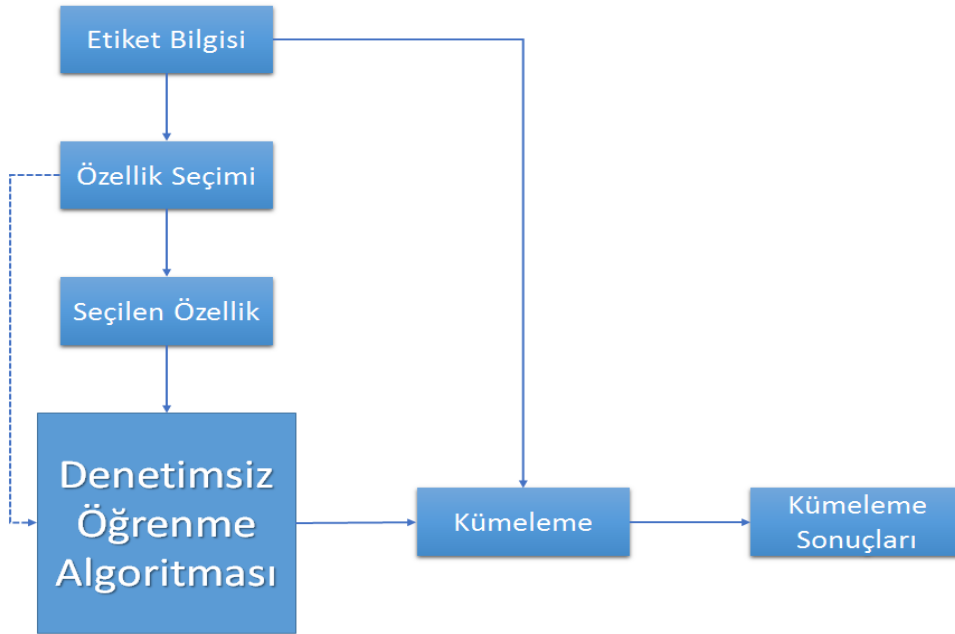


Şekil 3.4. Denetimli özellik seçiminin genel çerçevesi

- Denetimsiz Özellik Seçimi

Verilerin sınıf etiketleri mevcut olduğu zaman denetimli özellik seçim yöntemi kullanılır, aksi takdirde denetlenmeyen özellik seçim yöntemi kullanılmaktadır. Denetimsiz özellik seçimi genel olarak kümeleme problemleri için tasarlanmıştır. Etiketli verileri toplamada aşırı zaman ve çaba harcanmaktadır. Aynı zamanda daha maliyetli bir süreç olduğundan dolayı, etiketlenmemiş veriler üzerinde denetimsiz özellik seçimi uygulanması son zamanlarda büyük bir ilgi görmektedir. Özelliklerin önemini değerlendirecek etiket bilgilerinin olmaması sebebiyle, denetimsiz özellik seçim yöntemleri, alternatif kriter aramak için veri benzerliği ve yerel ayrımcı bilgiler gibi özelliklerin uygunluğunu tanımlar (Li vd, 2017).

Denetimsiz özellik seçiminin denetimli özellik seçiminden farkı, genellikle özellik seçim aşamasında bulunan tüm örneklerin kullanılmasıdır. Ayrıca denetimsiz özellik seçimi, denetimli ve yarı denetimli özellik seçimlerinden daha zordur, çünkü etiketli sınıflar tarafından desteklenmez. Aynı zamanda birkaç faydası daha vardır, örneğin, örnekleri kategorize etmek için uzmanları veya veri analistlerini kullanmaya gerek kalmadan ve ön bilgi bulunmadığında bile iyi performans gösterebilir (Ang vd, 2015). Denetimsiz özellik seçimi biyolojik verilerin keşifsel analizi için gereklidir ve hastalık tiplerinin sınıflandırılması için bilinmeyen anlamlı iç görüleri keşfetmesinde etkili bir rol oynamaktadır (Xu vd, 2010). Denetimsiz yaklaşımın temel dezavantajları, farklı özellikler arasındaki olası korelasyonu ihmal etmesidir ve prensiplerin tüm veriler için evrensel olarak geçerli olduğunu garanti etmeden bazı matematiksel prensiplere dayanmasıdır (Cai vd, 2010). Denetimsiz özellik seçiminin genel bir çerçevesi Şekil 3.5'te gösterilmiştir.



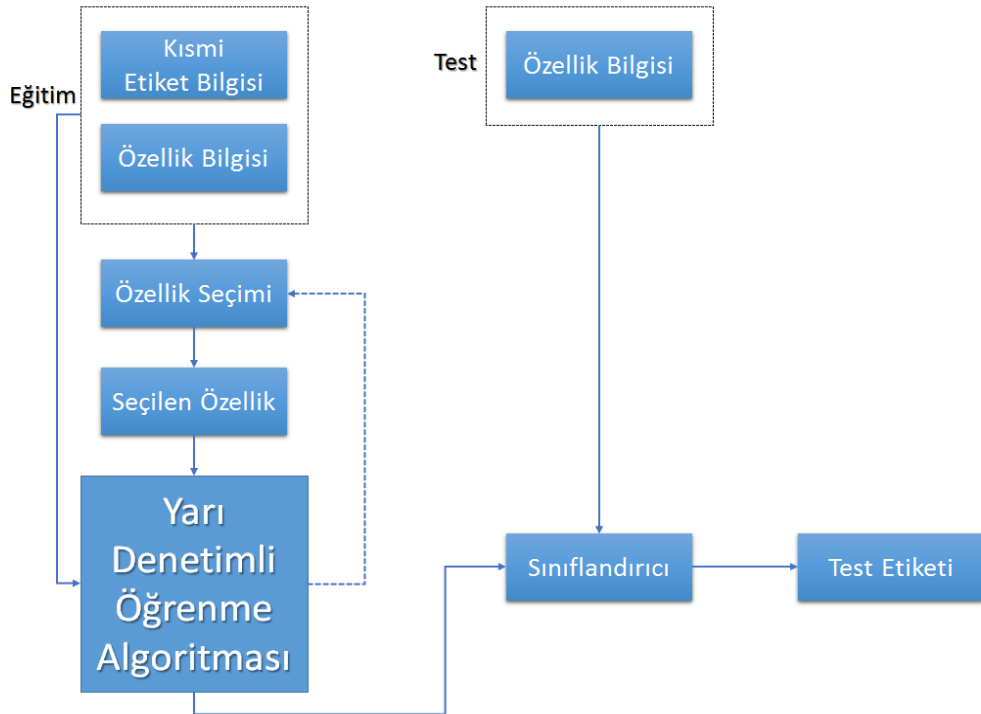
Şekil 3.5. Denetimsiz özellik seçiminin genel çerçevesi

- Yarı Denetimli Özellik Seçimi

Yarı denetimli yöntemi, makine öğrenme tekniklerinden birisidir. Bu teknik, denetimli ve denetimsiz öğrenmenin birleşmesiyle uygulanır ve aynı zamanda karma yöntemlerden birisidir (Innocent ve Kurian, 2013). Denetimli ve denetimsiz özellik seçmelerinin uzantılarından olan yarı denetimli ve yarı denetimsiz özellik seçmeleri,

hem etiketli hem de etiketlenmemiş veriler üzerinde çalışırlar. yarı denetimli verilerin çoğunluğu etiketli olduğunda ve yarı denetimsiz verilerin çoğunluğu etiketsiz olduğunda kullanılır (Zhao vd, 2008). Denetimli ve denetimsiz özellik seçme yöntemleri, farklı şekillerde özellik alaka düzeyini ölçmeyi gerektirir. Dolayısıyla yararlı bir yarı denetimli özellik seçme algoritmasının tasarlanma sebebi, bir özelliğin alaka düzeyinin hem etiketli hem de etiketsiz verilerle doğal olarak değerlendirebileceği bir çerçeve geliştirmektir (Zhao ve Liu, 2007a).

Birçok gerçek dünya uygulamasında, genel olarak az sayıda etiketlenmiş örnek ve çok sayıda etiketsiz örnek mevcuttur. Hem denetimli hem de denetimsiz özellik seçim algoritmaları bu senaryodaki tüm örneklerden tam olarak yararlanamaz. Denetimli yöntemler için, az sayıda etiketlenmiş örnek, özelliklerin korelasyon bilgilerini sağlamak için yetersiz olabilir. Denetimsiz yöntemler ise farklı sınıfları ayırt etmek için yararlı bilgiler sağlayabilecek sınıf etiketlerini tamamen görmezden gelir. Dolayısıyla, hem etiketlenmiş hem de etiketlenmemiş örnekleri kullanarak yarı denetimli yöntemler tasarlanır (Li vd, 2017). Yarı denetimli özellik seçiminin genel bir çerçevesi Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Yarı denetimli özellik seçiminin genel çerçevesi

### 3.1.3. Özellik Seçmede Arama Stratejisi

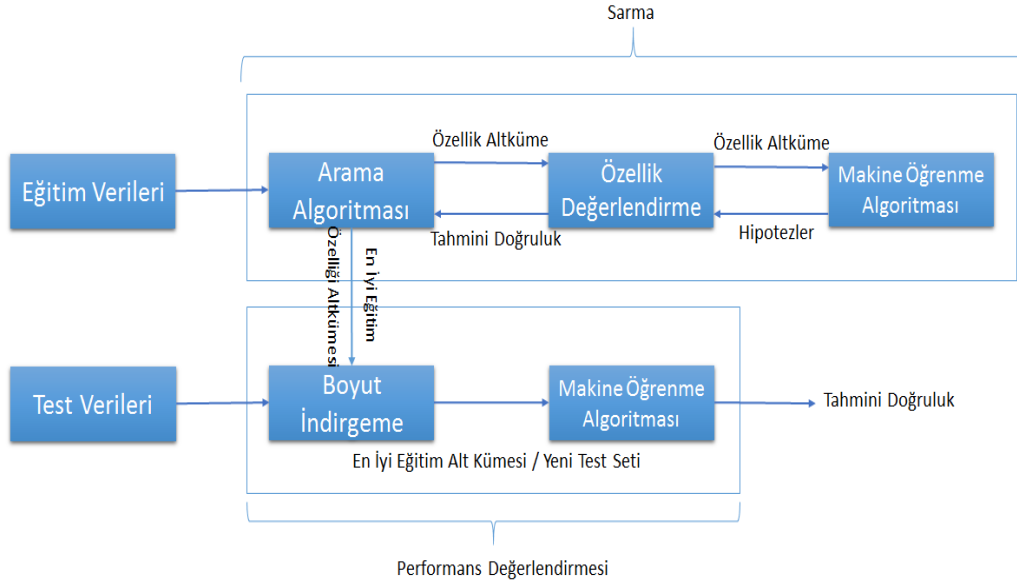
Farklı seçim stratejileri açısından, özellik seçme yöntemleri sarmal, filtre, gömülü ve hybrid yöntemler olarak kategorize edilebilir (Leung ve Hung, 2008), ancak son yıllarda topluluk özellik seçme yöntemide geliştirilmiştir (Yang vd, 2010). Aşağıdaki bölümde her birisi özet bir şekilde anlatılmıştır.

#### 3.1.3.1. Sarmal Yöntemleri

Sarıcı yöntemleri 1997'de Kohavi ve John'nun dediğine göre sarıcı yöntemindeki özellik alt kümesi bir kara kutu olarak yapılmıştır (Kohavi ve John, 1997). Sarmalayıcılar, kara kutu değerlendiricisi olarak alınan bir modelleme algoritmasındaki performansın kalitesine göre özellik alt kümelerini dikkate alırlar (Jović vd, 2015). Böylece, sınıflandırma görevleri için, bir sarıcı alt kümeleri sınıflandırıcı performansına göre (Bradley ve Mangasarian, 1998) ve ayrıca kümeleme için, bir sarıcı alt kümeleri bir kümeleme algoritmasının performansına göre değerlendirilir (Maldonado vd, 2014). Bu yöntemler, seçilen özelliklerin kalitesini değerlendirmek için daha önceden tanımlanmış bir öğrenme algoritmasının öngörücü performansına dayanmaktadır. Bununla birlikte belirli bir öğrenme algoritması göz önüne alındığında, tipik bir sarmal yöntemi özelliklerin bir alt kümesini arayarak ve seçilen özellikleri değerlendirerek gerçekleştirilir (Li vd, 2017; Liu vd, 2010b).

Sarıcı yöntemler, denetimli öğrenme problemlerinde yaygın bir şekilde kullanılmaktadır. Ayrıca bu yöntemin üstün bir alternatif olarak kabul edilme nedeni, alternatifleri değerlendirmek için tümevarımsal algoritmayı kullanarak algoritmanın belirli sapmalarını dikkate almasıdır (Talavera, 2005). Başka bir deyişle özellik alt kümesi seçim algoritması, değerlendirme işlevinin bir parçası olarak tümevarım algoritmasını kullanır ve iyi bir alt kümeyi araştırır. İndüklenen sınıflandırıcıların doğruluğu, doğruluk tahmin yöntemlerini kullanılarak tahmin edilir (Kohavi, 1995). Araştırdığımız problem, durum uzay araştırmasıdır ve sonraki bölümlerde farklı arama motorları araştırılacaktır. Sarıcı yaklaşımı muhtemel parametreler alanında bir arama yapar. Bir arama bir durum alanı, bir başlangıç durumu, bir sonlandırma koşulu ve bir arama motoru gerektirir (Ginsberg, 2012; Russell ve Norvig, 2002). Dahası bu yöntemler, bazı parametrelerin en iyi değerlerini belirlemek ve özelliklerin alt kümelerini tahmin güçlerine göre puanlamak için bir öğrenme algoritması ve bir arama tekniği kullanırlar. Özellik seçimi sınıflandırma ve regresyon problemlerini

çözebilse, uygulamaların çoğu sınıflandırma problemleriyle çalışmaktadır (Kohavi ve John, 1997).



Şekil 3.7. Sarmal özellik seçim yöntemlerinin genel çerçevesi

için arama alanı katlanarak büyüdüğü zaman kapsamlı aramayı kullanışsız hale getirmesidir. Dolayısıyla ardışık arama (Guyon ve Elisseeff, 2003), tepe tırmanma araması (Skalak, 1994), en iyi ilk arama (Kohavi ve John, 1997), dal ve sınırla arama (Narendra ve Fukunaga, 1977) ve genetik algoritmalar (Goldberg, 1989a) yerel optimum öğrenme performansı vermeyi önermektedirler. Ancak, arama alanı yüksek boyutlu veri kümeleri için son derece geniştir. Sonuç olarak sarmal yöntemleri çok az kullanılmaktadır (Li vd, 2017). Sarıci özellik seçim yöntemlerinin genel bir çerçevesi Şekil 3.7'de gösterilmiştir. Ayrıca sarmal yöntemlerde en çok kullanılan teknikler aşağıdaki bölümlerde özet olarak anlatılmıştır.

Tablo 3.4. Sarmal yönteminin genel algoritması

Girdiler :	
$D = X, L$	$n$ sayıda özelliğe sahip bir eğitim veri seti :
	$x = \{f_1, f_2, f_3, \dots, f_n\}$ ve $L$ ise etiketler
$X'$	önceden tanımlanmış ilk özellik alt kümesi ( $X' \subset X$ yada $X' = \{\emptyset\}$ )

	$\Theta$	Durdurma kriterleri
Çıktılar :	$X'_{opt}$ :	En uygun alt kümeyi bulmak
Begin :		
Initialize :		
	$X_{opt} = X'$ ;	
	$\vartheta_{opt} = E(X', A)$ ;	A araştırma algoritmasını kullanarak $X'$ i değerlendirir
do begin :		
	$X_g = \text{generate}(X)$ ;	Değerlendirme için alt küme oluşturma
	$\vartheta = E(X_g, A)$ ;	A ile $X_g$ alt kümesi değerlendirmesi
	Eğer ( $\vartheta > \vartheta_{opt}$ )	
	$\vartheta_{opt} = \vartheta$ ;	
	$X'_{opt} = X_g$ ;	
Tekrarla	$\Theta$ 'ya ulaşıncaya kadar	
end		
return	$X'_{opt}$ ;	
end		

Tablo 3.4'te görüldüğü gibi genel olarak, ardışık ileri yönde seçim özelliği kullanarak boş bir set  $X'$  ile başlar. Kardinalitenin  $k$  optimal alt kümesi için, geri kalan alt kümelerden bir özellik eklenir ve daha sonra  $k + 1$ 'in tüm olası kardinalite alt kümeleri aranır. Kardinalite  $k + 1$ 'de üretilen bir alt küme, bağımsız kriter  $I_m$  ile değerlendirilir ve önceki optimum alt kümeyle kıyaslanır. Daha sonra öğrenme algoritması  $A$ , mevcut optimal alt kümeyle uygulanır ve performans, kardinalite  $k$ 'deki

optimal alt kümenin performans  $\vartheta$  ile karşılaştırılır. Ölçüt durdurulduktan sonra, son optimal alt kümeyi döndürür (Kumar ve Minz, 2014a).

- Ardışık İleri Yönde Seçim

Sıralı özellik seçim algoritmaları, alakasız özellikleri tekrarlamayla ortadan kaldırırlar ve en iyi özellikleri toplayarak etkili bir özellik alt kümesi ararlar (Somol vd, 2006). Bu yöntemlerden birisi olan ardışık ileri yönde seçim, 1971 yılında Whitney tarafından önerilmiştir (Whitney, 1971). Ardışık ileri yönde seçim, boş bir kümeden başlayarak ve değerlendirme işlevind daha fazla iyi sonuçlar sağlanamayana kadar kademeli olarak birer birer eklenen bir denetimli ve sarmal özellik seçim yöntemidir (Inza vd, 2004). Bu yöntemde kümeye bir özellik eklendiğinde, ardışık ileri yönde seçim özelliği en iyi genelleştiren öğrenme yapısına yerleştirir. Öğrenme yapısına bir özellik eklendiğinde, ardışık ileri yönde seçim bunu kaldıramaz. Değerlendirme fonksiyonunun amacı, tahmin için ortalama kare hatasını en aza indirmektir. ardışık ileri yönde seçiminde en çok karşılaşılan problemlerden birisi, her zaman bir değişken eklediği için birbirine bağımlı özellikler içermemesidir (Karagiannopoulos vd, 2007) ancak, bu yöntem iç içe olma probleminden etkilenmektedir. Bu, tekrarlı sürecin bir adımında yer alan bir özelliğin daha sonraki bir adımda dışta tutulamayacağı anlamına gelir. Böylece, sonuçlar optimal değildir (Pratama vd, 2012). Dolayısıyla ardışık ileri yönde seçim küçük veri kümelerinde daha çok uygulanabilir (Pohjalainen vd, 2015). Ardışık ileri yönde seçim yönteminin genel algoritması Tablo 3.5'te görülmektedir.

Tablo 3.5. Ardışık ileri yönde seçiminin genel algoritması

---

Adım 1 :	Boş veri kümesi ile başla $Y_0 = \emptyset$
Adım 2 :	Bir sonraki en iyi özelliği seç $x^+ = \arg_{x \notin Y_k} \text{Max } J(Y_k + x)$
Adım 3 :	Eğer $J(Y_k + x^+) > J(Y_k)$ ise o zaman güncelle : $Y_{k+1} = Y_k + x^+$ ; $k = k + 1$
Adım 4 :	İkinci adıma geri dön

---

- Ardışık Geri Yönde Seçim

Ardışık geri yönde seçim 1963 yılında Marill, Thomas ve Green tarafından önerilmiştir (Marill ve Green, 1963). Bu yöntem ardışık ileri yönde seçimin aksine, tüm özellik kümesiyle başlatılır ve objektif fonksiyonun değerini en aza indiren özelliği kaldırarak özellik kümesini günceller. Ardışık geri yönde seçim tüm

özelliklerle başladığından dolayı, erken değerlendirmeleri nispeten maliyetlidir (Ladha ve Deepa, 2011; Liu ve Motoda, 2012). Bu yöntemin önemli dezavantajlarından birisi, bir özellik kaldırıldıktan sonra asla tekrar değerlendirilmemesidir (Clark ve Boswell, 2000). Ardışık geri yönde seçim yönteminin genel algoritması Tablo 3.6'da görülmektedir.

Tablo 3.6. Ardışık geri yönde seçiminin genel algoritması

---

Adım 1 :	Tüm veri kümesi ile başla $Y_0 = X$
Adım 2 :	En kötü özelliği kaldır $x^- = \arg_{x \in Y_k} \text{Max } J(Y_k - x)$
Adım 3 :	Eğer $J(Y_k - x^-) > J(Y_k)$ ise o zaman güncelle : $Y_{k+1} = Y_k - x^-$ ; $k = k + 1$
Adım 4 :	İkinci adıma geri dön

---

- Ardışık İleri Yönde Kayan Seçim

Ardışık ileri yönde kayan seçim 1994 yılında Pudil ve arkadaşları tarafından önerilmiştir (Pudil vd, 1994). Ardışık ileri yönde seçim ve ardışık geri yönde seçim bir kerede bir özellik ekleyerek veya kaldırarak bir yönde çalışırlar, ancak ardışık kayan seçimi her iki yönde değişken ekleme veya kaldırma üzerinde çalışır ve bundan dolayı ardışık kayan seçimi son özellik alt kümesinin güvenilirliğini artırır. İki farklı ardışık kayan seçimi yöntemi vardır: birincisi ardışık ileri yönde kayan seçim ve ikincisi ise ardışık geri yönde kayan seçimdir. Ardışık ileri yönde kayan seçim, ardışık ileri yönde seçiminin yaptığı gibi boş kümeyle başlatılır; aynı zamanda, her ileri adımdan sonra, bu yöntem objektif fonksiyon artana kadar geriye doğru adımlar gerçekleştirir (Pudil vd, 1994). Ardışık ileri yönde kayan seçim yönteminin genel algoritması Tablo 3.7'de görülmektedir.

Tablo 3.7. Ardışık ileri yönde kayan seçiminin genel algoritması

---

Adım 1 :	Boş veri kümesi ile başla $Y_0 = \emptyset$
Adım 2 :	Bir sonraki en iyi özelliği seç $x^+ = \arg_{x \in Y_k} \text{Max } J(Y_k + x)$
Adım 3 :	Eğer $J(Y_k + x^+) > J(Y_k)$ ise o zaman güncelle : $Y_{k+1} = Y_k + x^+$ ; $k = k + 1$
Adım 3.1 :	En kötü özelliği kaldır $x^- = \arg_{x \in Y_k} \text{Max } J(Y_k - x)$

---

---

Adım 4 : Eğer  $J(Y_k - x^-) > J(Y_k)$  ise o zaman güncelle :  $Y_{k+1} = Y_k - x^-$  ;  $k = k + 1$

Adım 4.1 : Adım 3.1'e geri dön

Adım 4.2 : Değilse, o zaman Adım 2'ye geri dön

---

- Ardışık Geri Yönde Kayan Seçim

Ardışık geri yönde kayan seçim, aynı ardışık ileri yönde kayan seçim gibi 1994 yılında Pudil ve arkadaşları tarafından önerilmiştir (Pudil vd, 1994). Ardışık kayan seçim, L ekle ve R çıkar yönteminin alternatifidir. Bu kayan yöntemler, L ve R değerlerini sabitlemek yerine, değerlerin verilerden belirlenmesini sağlar. Bu yöntem ardışık ileri yönde kayan seçim ile aynı özelliklere sahiptir, ancak tersi yönde hareket etmektedir. Ardışık geri yönde kayan seçim tüm özellik kümesiyle başlatılır ve her bir geriye doğru adımdan sonra, objektif fonksiyon arttığı sürece ileri adımlar gerçekleştirir (Mlambo vd, 2016; Pudil vd, 1994).

- L Ekler – R Çıkar

L ekle ve R çıkar yöntemi, ardışık ileri yönde seçimi ve ardışık geri yönde seçiminin genelleştirilmesidir. Bu yöntem 1976'da Streamns tarafından önerilmiştir (Streamns, 1976). Bu yöntemde  $L > R$  ise, boş özellik kümesinden başlatılır ve devamlı L özelliklerini ekler ve R özelliklerini çıkarır. Aynı zamanda  $L < R$  ise, tüm özellik kümesiyle başlatılır ve R özelliklerini ve ardından L eklemelerini tekrar tekrar çıkarır. Bu yöntemin en önemli dezavantajı, L ve R'nin optimal değerlerini tahmin etmek için bir teoremin olmamasıdır (Hall, 1999; Ladha ve Deepa, 2011). L ekle – R çıkar yönteminin genel algoritması Tablo 3.8'de görülmektedir.

Tablo 3.8. L ekler – R çıkar yönteminin genel algoritması

---

Adım 1 : Eğer  $L > R$  ise, o zaman boş veri kümesi ile başla  $Y_0 = \emptyset$

Adım 1.1 : Değilse, o zaman tüm veri kümesi ile başla  $Y_0 = X$  ; Adım 3'e git

Adım 2 : L kadar tekrar et  $x^+ = \text{arg}_{x \in Y_k} \text{Max } J(Y_k + x)$

Adım 2.1 : Eğer  $J(Y_k + x^+) > J(Y_k)$  ise o zaman güncelle :  
 $Y_{k+1} = Y_k + x^+$  ;  $k = k + 1$

---

---

Adım 3 : R kadar tekrar et  $x^- = \mathit{arg}_{x \in Y_k} \mathit{Max} J(Y_k - x)$

Eğer  $J(Y_k - x^-) > J(Y_k)$  ise o zaman güncelle :

Adım 3.1 :

$$Y_{k+1} = Y_k - x^- ; k = k + 1$$

Adım 4 : Adım 2'ye geri dön

---

- Çift Yönlü Arama

Çift yönlü arama algoritması ilk olarak 1971 yılında Ira Pohl tarafından önerilmiştir (Pohl, 1971). Bu algoritmanın amacı, ardışık iler yönde seçim ve ardışık geri yönde seçiminin aynı çözüme yaklaşmasını sağlamaktır. Dolayısıyla, ardışık iler yönde seçim tarafından seçilen özellikler ardışık geri yönde seçim tarafından kaldırılmamalı ve ardışık geri yönde seçim tarafından kaldırılan özellikler ardışık iler yönde seçim tarafından eklenmemelidir. Çift yönlü arama yönteminin genel algoritması Tablo 3.9'da görülmektedir.

Tablo 3.9. Çift yönlü arama yönteminin genel algoritması

---

Adım 1 : Ardışık ileri yönde seçimi boş küme ile başla  $Y_F = \emptyset$

Adım 2 : Ardışık geri yönde seçimi tüm veri küme ile başla  $Y_B = X$

Adım 3 : En iyi özelliği seç  $x^+ = \mathit{arg}_{x \in Y_{F_k}, x \in Y_{B_k}} \mathit{Max} J(Y_{F_k} + x)$

Eğer  $J(Y_{F_k} + x^+) > J(Y_{F_k})$  ise o zaman güncelle :

Adım 3.1 :

$$Y_{F_{k+1}} = Y_{F_k} + x^+ ; k = k + 1$$

Adım 4 : En kötü özelliği kaldır  $x^- = \mathit{arg}_{x \in Y_{B_k}, x \in Y_{F_{k+1}}} \mathit{Max} J(Y_{B_k} - x)$

Eğer  $J(Y_{B_k} - x^-) > J(Y_{B_k})$  ise o zaman güncelle :

Adım 4.1 :

$$Y_{B_{k+1}} = Y_{B_k} - x^- ; k = k + 1$$

Adım 5 : Üçüncü adıma geri dön

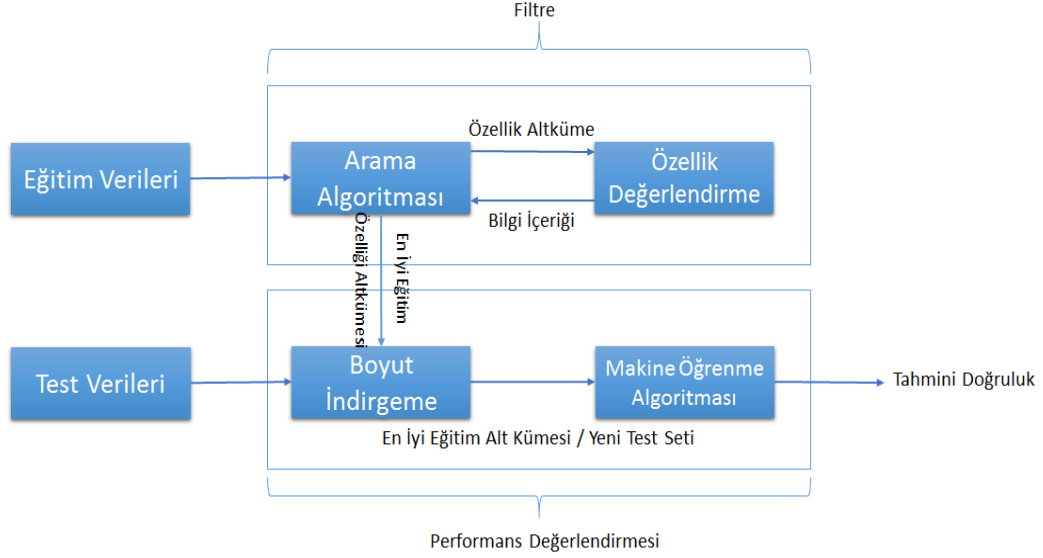
---

### 3.1.3.2. Filtre Yöntemleri

Filtre yöntemleri (Chandrashekar ve Sahin, 2014) istatistiksel yöntemlere dayanmaktadır ve bu yöntem her özelliği bağımsız olarak değerlendirir. Bu yöntem

özellikleri önemlerine göre tahmin ederek sıralarlar. Aynı zamanda, özellik ve hedef değişken arasındaki korelasyon derecesinde önemlidir (Kullback ve Leibler, 1951). Filtre tekniği öğrenme algoritmalarının hepsinden bağımsızdır. Özelliklerin önemini değerlendirmek için verilerin belirli özelliklerine dayanılır. Bu yöntem tipik olarak sarmalayıcı yöntemlerinden daha avantajlıdır. Ayrıca, özellik seçim aşamasında öncülük yapan belirli bir öğrenme algoritmasının mevcut olmama sebebiyle, öğrenme algoritmaları için seçilen özellikler uygun olmayabilir. Tipik bir filtre yöntemi iki aşamadan oluşmaktadır. İlk aşamada, özellik önemi bazı özellik değerlendirme kriterlerini kıyaslayarak bir özellik puanına göre sıralanır. Özellik önemi değerlendirme süreci tek değişkenli veya çok değişkenli olabilir. Tek değişkenli şemada, her bir özellik diğer özelliklerden bağımsız olarak ayrı ayrı sıralanırken, çok değişkenli şema birden çok özelliği toplu olarak sıralar. İkinci aşamada, düşük sıralama özellikleri filtrelenir ve kalan özellikler seçilir (Li vd, 2017).

Bu yöntem, değerlendirilecek verilerin genel benzersizliğine ve herhangi bir madencilik algoritması içermeyen özellik alt kümesini seçmeye dayanmaktadır (Bolón-Canedo vd, 2013). Filtre yöntemleri açık döngü olarak adlandırılır ve mesafe, bilgi, bağımlılık ve tutarlılığı içeren kesin değerlendirme ölçütlerini kullanmaktadır (Dash ve Liu, 1997; Liu vd, 2010b). Kısacası bir filtre yöntemi, verilerin genel özelliklerini inceler ve bazı ayırt edici ölçütlere göre özelliklerin uygunluğunu değerlendirir (Bhavani vd, 2008). Son yıllarda, filtre yöntemleri için birçok farklı değerlendirme kriteri önerilmiştir. Örnekleri ayırmak için özellik ayrımcılığı (Kira ve Rendell, 1992b; Robnik-Šikonja ve Kononenko, 2003), özellik korelasyonunu (Guyon ve Elisseeff, 2003; Koller ve Sahami, 1996), karşılıklı bilgiler (Peng vd, 2005; Yu ve Liu, 2003), ve orijinal verileri yeniden yapılandırma özelliği (Farahat vd, 2011; Masaeli vd, 2010)



Şekil 3.8. Filtre özellik seçim yöntemlerinin genel çerçevesi

Filtre işlemi sarmal yönteminden daha hızlı bir yöntemdir. Filtre yönteminin avantajı, yüksek boyutlu veri kümeleriyle kolayca hesaplanabilmesi, hesaplamalı olarak basit olması, işlemde hızlı ve veri madenciliği algoritmasından bağımsız olmasıdır. Dolayısıyla, özellik seçimi sadece bir kez yapılacak ve değerlendirme süreci için farklı sınıflandırıcılar kullanılacaktır (Beniwal vd, 2012; Huang vd, 2008). Bu yöntemin temel dezavantajı, sınıflandırıcılar arasındaki etkileşimi görmezden gelmesi ve en çok önerilen tekniklerin her bir özelliğini ayrı ayrı ele almasıdır. Dolayısıyla bu yöntem sarmal yöntemlerinin performansına göre sınıflandırma performansının düşmesine sebep olur (Ladha ve Deepa, 2011; Peng vd, 2010; Xie vd, 2013). Filtre özellik seçim yöntemlerinin genel bir çerçevesi Şekil 3.8'de gösterilmiştir.

Tablo 3.10. Filtre yönteminin genel algoritması

Girdiler :	
$D = X, L$	$n$ sayıda özelliğe sahip bir eğitim veri seti : $x = \{f_1, f_2, f_3, \dots, f_n\}$ ve $L$ ise etiketler
$X'$	önceden tanımlanmış ilk özellik alt kümesi ( $X' \subset X$ yada $X' = \{\emptyset\}$ )
$\Theta$	Durdurma kriterleri

---

Çıktılar :	$X'_{opt}$ :	En uygun alt kümeyi bulmak
------------	--------------	----------------------------

---

Begin :

Initialize :

$$X_{opt} = X' ;$$

$$\vartheta_{opt} = E(X', I_m) ;$$

$I_m$ 'i bağımsız bir ölçü birimi kullanarak  $X'$  i değerlendirir

do begin :

$$X_g = \text{generate}(X) ;$$

Değerlendirme için alt küme oluşturma

$$\vartheta = E(X_g, I_m) ;$$

$I_m$  ile  $X_g$  alt kümesi değerlendirmesi

$$\text{Eğer } (\vartheta > \vartheta_{opt})$$

$$\vartheta_{opt} = \vartheta ;$$

$$X'_{opt} = X_g ;$$

Tekrarla  $\vartheta$ 'ya ulaşıncaya kadar

End

Return  $X'_{opt}$  ;

End

---

Tablo 3.10'da görüldüğü gibi genellikle, boş bir set  $X'$  ile başlar. Kardinalitenin  $k$  optimal alt kümesi için, geri kalan alt kümelerden bir özellik ekleyerek  $k + 1$ 'in tüm olası kardinalite alt kümelerini arar. Kardinalite  $k + 1$ 'de üretilen bir alt küme, bağımsız kriter  $I_m$  ile değerlendirilir ve önceki optimum alt kümeyle karşılaştırılır. Daha sonra öğrenme algoritması  $A$ , mevcut optimal alt kümeyle uygulanır ve performans, kardinalite  $k$ 'deki optimal alt kümenin performans  $\vartheta$  ile karşılaştırılır. Ölçüt durdurulduktan sonra, son optimal alt kümeyi döndürür (Kumar ve Minz, 2014a).

- Fisher Skor

Fisher skor (Duda vd, 2012) tekniđi en çok kullanılan denetimli özellik seçim yöntemlerinden birisidir. Fisher skorunun ana fikri, seçilen özelliklerin kapsadığı veri alanında, farklı sınıflardaki veri noktaları arasındaki mesafelerin mümkün olduğunca büyük, aynı sınıftaki veri noktaları arasındaki mesafelerin mümkün olduğu kadar küçük bir özellik kümesi bulmaktır. Bununla birlikte, her bir özelliđi, Fisher kriteri altındaki puanlarına göre bağımsız olarak seçer ve bundan dolayı özelliklerin yetersiz bir alt kümesine yol açar. Bu yöntemde N örneğinin sınıf etiketlerinin  $y = \{y_1, y_2, \dots, y_n\}$  'nin c sınıflarından geldiđini varsayalım, Fisher skor özellikleri, aynı sınıftaki örneklerin özellik değerlerinin küçük, farklı sınıflardan örneklerin özellik değerlerinin büyük olacağı şekilde seçer. Her bir özelliđin fisher puanı aşağıdaki gibi değerlendirilir (Gu vd, 2012):

$$\text{fisher\_skor}(f_i) = \frac{\sum_{j=1}^c n_j (\mu_{i,j} - \mu_i)^2}{\sum_{j=1}^c n_j \sigma(i, j)^2} \quad (3.1)$$

Burada  $n_j$ ,  $\mu_i$ ,  $\mu_{i,j}$  ve  $\sigma_{i,j}^2$  sırasıyla j sınıfındaki örneklerin sayısını,  $f_i$  özelliklerinin ortalama değerini, j sınıfındaki örnekler için  $f_i$  özelliklerinin ortalama değeri, j sınıfındaki örnekler için  $f_i$  özelliđinin varyans değerini göstermektedirler. Bu özellik seçme yöntemi çoğunlukla ikili sınıflandırma amaçları için kullanılır (Gu vd, 2012).

- T-Skor

T-skoru ikili sınıflandırma problemleri için kullanılan bir yöntemdir. Her  $f_i$  özelliđi için,  $\mu_1$  ve  $\mu_2$ 'nin birinci sınıf ve ikinci sınıftan örnekler için ortalama özellik değerleri olduğunu varsayalım,  $\sigma_1$  ve  $\sigma_2$  ise karşılık gelen standart sapma değerleri,  $n_1$  ve  $n_2$  değerleride bu iki sınıftan örnek sayısını gösterir (Chandra ve Gupta, 2011).  $f_i$  özelliđi için t-skoru aşağıdaki formülü kullanarak hesaplanır (Carey ve Delaney, 2010) :

$$t\_skor(f_i) = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (3.2)$$

T-skorunun temel fikri, ortalama fark ile iki sınıfın varyansı arasındaki oranı hesaplayarak, özelliđin iki sınıfın anlamını istatistiksel olarak farklı hale getirip getiremeyeceđini değerlendirmektir. Genellikle, t puanı ne kadar yüksek olursa özellik

o kadar önemlidir (Faulkner, 2005). T-skoru, filtre yöntemleri arasında en sık kullanılan t-puanına dayanmaktadır. Yukarıda bahsedildiği gibi, her sınıf için özelliklerin örnek boyutu, ortalama ve standart sapma değerleri kullanılarak bir ilişki puanı hesaplanır ve yüksek puanlı özellikler t-skoru yönteminde alt kümeye eklenir (Budak ve Taşabat, 2016).

- Welch T-İstatistiği

Welch t-istatistiği (Welch, 1947), iki grubun ortalamaları arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için kullanılan bir istatistiksel yöntemdir. Bu teknik, öz-zellik seçim yöntemlerinde yaygın kullanılan bir istatistiksel testtir. Ayrıca bu yöntem ilişki puanını hesaplamak için, t-skor yöntemi gibi özelliklere ait ortalama, standart sapma değerleri ve sınıf örnek sayılarını kullanır.  $f_i$  özelliği için welch t-skoru aşağıdaki formülü kullanarak hesaplanır (Gliozzo, 2017) :

$$\text{welchT\_istatistiği}(f_i) = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{(\sigma_1^2)^2}{n_1} + \frac{(\sigma_2^2)^2}{n_2}}} \quad (3.3)$$

Ortalama karşılaştırması, birçok klasik istatistiksel yöntemlerin ana temasıdır. T-test genellikle eşit varyanslara sahip bağımsız iki aracın eşitliğini test etmek için kullanılırken, varyanslar eşit olmadığında Welch'in t testi genellikle tercih edilmektedir(Delacre vd, 2017). Zhang ve arkadaşları 2015 yılında yaptıkları çalışmada Alzheimer hastalığına bağlı insanların ve beynin bölgelerinin teşhisinde welch t-istatistiği ve üç boyutlu MRI taramaları kullanmışlar(Zhang vd, 2015).

- Ki-Kare Testi

Ki-kare testi (Kass, 1980), filtre yönteminde kullanılan özellik seçim yöntemlerinden biridir. Bu istatistiksel yöntem, iki olay arasındaki bağımsızlığı kontrol eder (Esmael vd, 2015; Kumar, 2014; Yang ve Pedersen, 1997) ve özelliğin sınıf etiketinden bağımsız olup olmadığını değerlendirmek için bağımsızlık testini kullanır. R farklı özellik değerlerine sahip belirli bir özellik verildiğinde, bu özelliğin ki-kare puanı aşağıdaki formülü kullanarak hesaplanabilir (Inc, 2007; Jin vd, 2006; Witten vd, 2005):

$$ki\_kare(f_i) = \sum_{j=1}^r \sum_{s=1}^c \frac{(n_{js} - \mu_{js})^2}{\mu_{js}} \quad (3.4)$$

Burada  $n_{js}$ ,  $f_i$  özelliği verilen  $j$  özellik değerine sahip örnek sayısıdır ve  $\mu_{js} = \frac{n_j * n_s}{n}$  beklenen birim sayısıdır. Ayrıca daha yüksek ki-kare puanı, özelliğin nispeten daha önemli olduğunu gösterir (Ugoni ve Walker, 1995). Ki-kare puanı, gözlemlenen birim sayısı ile her kategoride beklenen birim sayısı arasındaki fark hesaplanarak elde edilir. Bu fark karedirilir ve bu kategoride beklenen birim sayısına bölünür. Bu değerler daha sonra tüm kategoriler için eklenir ve toplam, ki kare değeri olarak adlandırılır (Bryant ve Satorra, 2012).

- Bilgi Kazancı

Kullback-Leibler (Kullback ve Leibler, 1951) mesafesi olarak da bilinen bilgi kazancı (Cover ve Thomas, 2012; Hoque vd, 2014; Liu vd, 2002; Quinlan, 1986; Raileanu ve Stoffel, 2004; Yang ve Pedersen, 1997), belirli bir veri/rasgele değişken üzerinde gerçekleştirilen işlemler sebebiyle kazanılan entropinin ölçüsüdür. Entropi aslında verilerdeki varyasyonun ölçüsüdür, varyasyon ne kadar az olursa entropi o kadar az olur ve veriler o kadar büyük korelasyon gösterir. Başka bir deyişle, bilgi kazancı daha büyükse bir özellik daha önemlidir (Hall ve Smith, 1998; Swingle, 2012). Bilgi kazancı, tüm özelliklere bağımsız olarak davranır ve bu ölçek, özelliklerin bilgi kazanımına dayanmaktadır. Bir özelliğin bilgi kazancı, önceki belirsizlik ile beklenen sonraki belirsizlik arasındaki fark olarak tanımlanır. Eşit olası sınıflar için bilgi kazancı en üst seviyededir ve belirsizlik en düşük seviyededir. Shannon entropisi belirsizlik önlemleri için yaygın olarak kullanılmaktadır (Shannon ve Weaver, 1949). Setiono ve Liu (Setiono ve Liu, 1996) tarafından önerilen normalize edilmiş bilgi kazancı, aşağıdaki gibi hesaplanmaktadır (Kraskov vd, 2004):

İlk kavram entropi olarak adlandırılır ve farklı bir rastgele değişkenin belirsizliğinin bir ölçüsüdür. Farklı bir rasgele değişken  $X$ 'in entropisi aşağıdaki gibi tanımlanır:

$$H(X) = - \sum_{x_i \in X} P(x_i) \log(P(x_i)) \quad (3.5)$$

Burada  $x_i$ , rastgele deęişken X'in belirli bir deęerini belirtir,  $P(x_i)$ , verilerden tahmin edilebilecek X'in olası tüm deęerleri üzerindeki  $x_i$  olasılıęını belirtir.

İkinci kavram, bir başka Farklı rasgele deęişken Y verildięinde X'in koşullu entropisidir:

$$H(X|Y) = \sum_{y_i \in Y} P(y_i) \sum_{x_i \in X} P(x_i|y_i) \log(P(x_i|y_i)) \quad (3.6)$$

Burada  $P(y_i)$   $y_i$ 'nin önceki olasılıęı,  $P(x_i | y_j)$  ise  $y_j$  verilen  $x_i$ 'nin koşullu olasılıęıdır. Koşullu entropinin ölçüsü, başka bir farklı rastgele deęişken Y verildięinde X'in belirsizlięini gösterir.

X ve Y arasındaki bilgi kazancı kavramı entropi ve koşullu entropiye baęımlılıklarını ölçmek için kullanılır. X ve Y tarafından paylaşılan bilgi miktarını birlikte ölçtüęü için, genellikle aőağıdaki gibi hesaplanan karşılıklı bilgi olarak adlandırılır:

$$\text{bilgi\_kazancı} = H(X) - H(X|Y) \quad (3.7)$$

burada  $H(X)$  entropidir ve  $H(X | Y)$  şartlı entropidir. Bu deęer, X'in deęerleri biliniyorsa veya daha basit bir şekilde belirtilirse, Y deęişkeninin ne kadar düzenli hale geldięini gösterir (Lee vd, 2011).

- Kazanç Oranı

Kazanç oranı, bilgi kazancı tarafından sergilenen daha büyük deęerlere sahip özelliklere yönelik sapmaları çözen ve aynı zamanda bilgi kazancı özellik seçim yönteminin bir ilerlemesidir. Kazanç oranı, bir özellik seçerken dal sayısını ve boyutunu dikkate alır. Bir bölünmenin gerçek bilgilerini dikkate alarak bilgi kazancını düzenler. İçsel bilgi, örneklerin dallara dağıtılmasının entropisidir. Başka bir deyişle, bir örneğin hangi dala ait olduęunu söylemek için ne kadar bilgiye ihtiyaç vardır. İçsel bilgi büyüdükçe özellik deęeri azalır (Han vd, 2011; Witten vd, 2005). Kazanç oranı aőağıdaki formülü kullanarak hesaplanabilir :

$$\text{kazanç\_oranı} = \frac{\text{bilgi\_kazancı}}{H(X)} \quad (3.8)$$

Bu yöntem 0 ile 1 arasında değer almaktadır. Kazanç oranı 1'e eşit olduğu zaman bilginin tahmin edildiğini ve 0'a eşit olduğunda veriler arasında hiçbir ilişki olmadığını göstermektedir (Novaković, 2016). Kazanç oranı özellik seçim sıralama yöntemlerinde, yüksek boyutlu veri kümelerinin özelliklerini sıralamak için kullanılan bir yöntemdir (Priyadarsini vd, 2011). Ayrıca kazanç oranı şarkı sözlerinde (Kumar ve Minz, 2013) ve şiirlerde (Kumar ve Minz, 2014a; Kumar ve Minz, 2014b) metin sınıflandırması için kullanılan bir yöntemdir.

- Simetrik Belirsizlik Katsayısı

Simetrik belirsizlik katsayısı (Hall, 1999; Huang ve Chow, 2005; Song vd, 2011; Witten vd, 2005), X ve Y'nin entropilerine bölerek bilgi kazancın dezavantajlarının üstesinden gelebilmek için geliştirilmiş bir yöntemdir (Dash ve Liu, 2003). Simetrik belirsizlik, özellik ve hedef sınıf arasındaki özelliklerin uygunluğunu hesaplamak için kullanılır. Bu yöntemde değeri yüksek olan özellik, önem kazanmaktadır. Simetrik belirsizlik katsayısı aşağıdaki formülü kullanarak hesaplanabilir (Ali ve Shahzad, 2012; Hernández-Torruco vd, 2014; Novaković, 2016; Saikhu vd, 2019; Singh vd, 2014) :

$$\text{simetrik\_belirsizlik\_katsayısı} = 2 \frac{\text{bilgi\_kazancı}}{H(X) + H(Y)} \quad (3.9)$$

Bu yöntemde tıpkı kazanç oranı gibi 0 ile 1 arasında değer almaktadır. Simetrik belirsizlik katsayısı 1'e eşit olduğu zaman bir özellik bilgisinin diğerini tahmin edildiğini ve 0'a eşit olduğunda X ve Y arasında hiçbir ilişki olmadığını göstermektedir (Mani ve Kalpana, 2016; Tang vd, 2014). Bu yöntemin temel sınırlaması sadece çift etkileşimleri dikkate almaktır ve bundan dolayı ikiden fazla özelliklerle çalışırken fazlalık tespitinde başarılı olmayabilir. Bu dezavantajın üstesinden gelebilmek için çok değişkenli bir simetrik belirsizlik katsayısı tanımlanmalıdır (Sosa-Cabrera vd, 2017). Simetrik belirsizlik kullanımının daha önceki çalışmalarda (Ding vd, 2005; Kannan ve Ramaraj, 2010; Piao vd, 2019; Piao vd, 2012; Sarhrouni vd, 2012; Singh vd, 2014; Yu ve Liu, 2004) boyutsallığın azaltılmasında faydalı olduğu ispatlanmıştır.

- Korelasyon Tabanlı Özellik Seçimi

Korelasyon tabanlı özellik seçimi (Hall, 1999), korelasyon dayalı sezgisel değerlendirme fonksiyonuna göre özellik alt kümelerini sıralayan basit bir çok değişkenli filtre algoritmasıdır (Hall, 1999; Witten vd, 2005). Başka bir deyişle bu yöntem, özellik alt kümelerini sıralar ve korelasyon tabanlı sezgisel değerlendirme işlevine göre özelliğin veya özellik alt kümesinin değerini belirler (Yildirim, 2015). Değerlendirme işlevi, sınıfla yüksek derecede ilişkili ve birbirleriyle ilişkisiz özellikler içeren alt kümelere yöneliktir. Alakasız özelliklerin göz ardı edilmesinin nedeni ise, sınıfla çok düşük korelasyonları olmasıdır. Geri kalan özellikler ise, kalan özelliklerden bir veya daha fazlası ile yüksek düzeyde korelasyon göstereceğinden taranmalıdır (Bolón-Canedo vd, 2014; Hernández-Torruco vd, 2014). Korelasyon tabanlı özellik seçimi, seçilen değişkenlerin iç korelasyonunu en aza indirmeyi ve seçilen değişkenler ile hedef arasındaki değişkenlerin bağımlılığı en üst düzeye çıkarmayı amaçlayan bir denetimli özellik seçim algoritmasıdır. Bu yöntem sadece özellik-özellik korelasyonlarının yanı sıra girdi-çıkı korelasyonlarını da ölçebilir. Bir özellik başka bir özellikle yüksek derecede ilişkiyise, alakasız olarak kabul edilir. Ancak, bir özellik hedefle güçlü bir şekilde ilişkilendirilirse, ilgili özellik olarak belirlenir (Yu ve Liu, 2003). Korelasyon tabanlı özellik seçimi, aşağıdaki formülü kullanarak hesaplanabilir (Hall, 1999; Sánchez-Marroño vd, 2007; Tyagi ve Mishra, 2013; Ziemba ve Piwowarski, 2011) :

$$\text{korelasyon\_tabanlı\_özellik\_seçimi} = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}} \quad (3.10)$$

Burada korelasyon tabanlı özellik seçimi, sınıf değişkeni ile toplanan alt küme özellikleri arasındaki korelasyon,  $k$ , alt küme özelliklerin sayısıdır,  $r_{cf}$ , sınıf değişkeni ile alt küme özellikleri arasındaki korelasyonların ortalamasıdır ve  $r_{ff}$  alt küme özellikleri arasındaki karşılıklı ilişkilerin ortalamasıdır (Hall, 1999; Sánchez-Marroño vd, 2007). Korelasyon tabanlı özellik seçimi, her bir özelliğin bireysel öngörme yeteneğini ve aralarındaki fazlalık derecesini dikkate alarak bir özellik alt kümesinin değerini belirler. Korelasyon katsayıları, özneliklerin alt kümesi ile sınıf arasındaki korelasyonu ve özellikler arasındaki korelasyonları tahmin etmek için kullanılır. Bir grup özelliğin alaka düzeyi, özellikler ve sınıflar arasındaki korelasyon ile büyümektedir ve artan korelasyon ile azalmaktadır (Hall, 1999). Bu yöntem genel olarak regresyon görevleri için düşük boyutlu veri setlerinde uygulanmıştır ve

performanslarının bir veri setinden diğerine değiştiği ve genellikle çeşitli alanlarda ortalama bir performans yaptıkları görülmektedir (Doquire ve Verleysen, 2013).

- Relief

Relief, Kira ve Rendell tarafından önerilen sınıflandırma problemleri için çok değişkenli ve çok popüler bir klasik filtre yöntemidir (Almuallim ve Dietterich, 1991; Kira ve Rendell, 1992a). Verilerden rastgele bir örnek alarak çalışan ve daha sonra aynı ve karşıt sınıftan en yakın komşusunu bulan, örnek tabanlı bir özellik seçimidir. En yakın komşuları örnek olarak karşılaştırılır ve her özelliğin alaka düzeyi puanlarını güncellemek için kullanılır (Hall ve Holmes, 2000; Sánchez-Marono vd, 2007). Başka bir deyişle relief örnekleri sınıflandırarak ve aynı ve farklı bir sınıfın en yakın örneği için geçerli özelliğin değerini karşılaştırarak, özelliklerin alaka düzeyini ölçmektedir (Esmael vd, 2015). Bu yöntemin güçlü yanları, sezgisel taramalara bağlı olmaması, düşük dereceli polinom zamanında çalıştığı ve ikili ve sürekli veriler için uygulanabilir olmasının yanı sıra gürültü verilere ve özellik etkileşimlerine karşı dayanıklı olmasıdır; ancak, fazlalık özellikler arasında ayırım yapmaz ve dolayısıyla az sayıda eğitim örneği algoritmayı kandırır (Lun Gaoa vd, 2013). İlk olarak bu yöntem sezgisel düşüncelere dayalı olarak önerildi, ancak yöntemin bazı kusurlarını tanımlayan ve ayrıntılı bir analiz sağlamak için Kononenko (Kononenko, 1994) ve Robnik-Sikonja ve Kononenko (Robnik-Šikonja ve Kononenko, 2003) daha sonra yöntemi geliştirdiler. Relief puanı aşağıdaki formülü kullanarak hesaplanabilir (Li vd, 2017; Miao ve Niu, 2016; Nilsson, 2007) :

$$\text{relief\_puanı} = \frac{1}{2} \sum_{j=1}^l d(X(j, i) - X(NM(j), i)) - d(X(j, i) - X(NH(j), i)) \quad (3.11)$$

Burada NM(j) ve NH(j), sırasıyla aynı sınıf etiketine ve farklı sınıfa sahip  $x_j$ 'ye en yakın veri örneklerini belirtir.  $d$  ise genellikle öklid mesafesi olarak ayarlanmış bir mesafe metriğidir(Li vd, 2017; Miao ve Niu, 2016; Nilsson, 2007).

- Relief-F

Relief-F (Kononenko, 1994) orijinal relief algoritmasının bir uzantısıdır (Kira ve Rendell, 1992a). Orijinal relief, bir örneği verilerden rastgele seçerek ve daha sonra aynı ve karşıt sınıftan en yakın komşusunu bularak çalışmaktadır. En yakın komşuların

özelliklerinin değerleri seçilen örneklerle karşılaştırılır ve her özellik için alaka düzeylerini güncellemek için kullanılır (Arauzo-Azofra vd, 2004; Hall ve Holmes, 2000; Urbanowicz vd, 2018b). Relief-F, çok sınıflı problemlerle başa çıkma yeteneğine sahiptir ve ayrıca daha sağlamdır ve eksik ve gürültülü verilerin üstesinden gelebilmektedir. Bu yöntem her durumda uygulanabilir, düşük hataya sahiptir, özellikler arasındaki etkileşimi içerir ve diğer yöntemlerin kaçırdığı yerel bağımlılıkları yakalayabilir (Bolón-Canedo vd, 2014; Chandra ve Gupta, 2011; Urbanowicz vd, 2018a). Bu yöntemin ana fikri, farklı sınıflardan kaynaklanan örnekleri ayırt edebilen özellikler seçmektir (Vora ve Yang, 2017). Relief-F algoritması bazı yayınlarda daha çok detaylanmıştır (Durgabai ve Bhushan, 2014; Kononenko vd, 1996; Kononenko vd, 1997; Robnik-Šikonja ve Kononenko, 1997; Robnik-Šikonja ve Kononenko, 2003). Relief-F puanını hesaplamak için aşağıdaki formül kullanılır (Kononenko, 1994):

$$\text{ReliefF}(f_i) = \frac{1}{c} \sum_{j=1}^1 \left( -\frac{1}{m_j} \sum_{x_r \in NH(j)} d(X(j, i) - X(r, i)) \right. \\ \left. + \sum_{y \neq y_j} \frac{1}{h_{jy}} \frac{P(y)}{1 - P(y)} \sum_{x_r \in NM(j, y)} d(X(j, i) - X(r, i)) \right) \quad (3.12)$$

Burada  $NH(j)$  ve  $NM(j, y)$ , aynı sınıfta ve farklı bir sınıf  $y$ 'de  $x_j$ 'ye en yakın veri örneklerini belirtir ve boyutları  $h_{jy}$  ve  $m_j$ 'dir.  $P(y)$  ise,  $y$  sınıf etiketi olan örneklerin oranıdır (Li vd, 2017; Vora ve Yang, 2017).

- One-R

One-R algoritması 1993 yılında Holte (Holte, 1993) tarafından önerilen ve eğitim verilerindeki her özellik için basit bir kural oluşturan ve kuralı en küçük hatayla seçen basit bir özellik seçim algoritmasıdır (Holmes ve Nevill-Manning, 1995; Mani ve Kalpana, 2016). Sayısal olarak değer verilen tüm özellikleri sürekli olarak ele alır ve değer aralığını birkaç farklı aralığa bölmek için basit bir yöntem kullanır. Eksik verileri uygun bir değer olarak değerlendirerek eksik değerleri işler (Russell ve Norvig, 2002; Yildirim, 2015).

Bu algoritma en basit yöntemlerden biridir. Sadece bir özelliğe dayalı basit kurallar üretir. Minimum bir sınıflandırıcı biçimi olmasına rağmen, diğer öğrenme

şemaları için bir referans olarak temel bir performansın belirlenmesi için yararlı olabilir (Novaković, 2016).

- En Az Fazlalık – En Çok Alaka Düzeyi Puanı

En az fazlalık-En çok alaka düzeyi puanı 2005 yılında Ding ve Peng tarafından önerilen filtre tabalı ve denetimli özellik seçim yöntemidir (Ding vd, 2005). Bu yöntem özellikler arasındaki fazlalığı en aza indirerek ve alaka düzeyini maksimum yaparak, özellikleri seçer (Chandra ve Gupta, 2011; Radovic vd, 2017; Vora ve Yang, 2017). Ayrıca bu yöntem farklı veri kümeleri için uygunluk ölçütü olarak karşılıklı bilgi ölçütünü kullanır (Bolón-Canedo vd, 2014; Peng vd, 2005; Zhang vd, 2008). En az fazlalık aşağıdaki formülü kullanarak hesaplanabilir (Ding vd, 2005; Katsov, 2017; Tang vd, 2014) :

$$w = \frac{1}{|S|^2} \sum_{i,j} c(i,j) \quad (3.13)$$

Ve en çok alaka düzeyi aşağıdaki formülü kullanarak hesaplanabilir :

$$V_F = \frac{1}{|S|} \sum_{i \in S} F(i, h) \quad (3.14)$$

Burada S bir dizi özelliktir, |S| S'deki özelliklerin sayısıdır,  $c(i, j)$  i ve j arasındaki özelliklerin korelasyonudur, h hedeftir ve  $F(i, h)$  F-statistikdir.

En az fazlalık-En çok alaka düzeyi yöntemi, hem sınıflandırmada hem de regresyonda uygulanabilen birkaç özellik seçme yönteminden biridir. Bu yöntem, özellik sayısının örnek sayısından önemli oranda fazla olduğunda ve yüksek boyutlu veri setlerinde genel olarak daha iyi performans gösterdiği görülmüştür (Ding vd, 2005; Peng ve Fan, 2015).

- Kanonik Korelasyon Analizi

Kanonik korelasyon analizi (Hotelling, 1992), iki özellik kümesi arasındaki korelasyon ilişkilerini bulmak için yaygın olarak kullanılan bir istatistiksel yöntemdir (El-Manzalawy, 2018). Başka bir deyişle Kanonik korelasyon analizi çok değişkenli iki değişken kümesi arasındaki ilişkiyi tanımlamaktadır. Bu yöntem, girdi ve çıktı değişkenleri arasındaki korelasyona göre bir özellik alt kümesi oluşturur. Ayrıca, bu yöntem hem sınıflandırma hem de sürekli hedeflerle kullanılabilir. Bu kısmen sürekli

hedeflere doğru kaymakta olan konuşma dışı iletişim analizi ve tahmindeki araştırma eğilimi göz önüne alındığında önemli bir konu haline gelmektedir (Gunes ve Schuller, 2013). Öte yandan, akustik konuşma işlemeye yönelik uygulamalarla kanonik korelasyon analizinin genişletilmesine olan ilgi giderek artmaktadır (Andrew vd, 2013; Arora ve Livescu, 2013; Kaya vd, 2014). Burada  $U_i$  ve  $V_j$  arasındaki korelasyonu hesaplamak için, bu iki değişken olan  $kov(U_i, V_j)$  arasındaki kovaryans, varyansların çarpımının kare köküne bölünür (Hardoon vd, 2004):

$$\text{korelasyon} = \frac{(kov(U_i, V_j))}{\sqrt{\text{var}(U_i, V_j)}} \quad (3.15)$$

Kanonik korelasyon analizi bu yukarıdaki korelasyonun özel bir versiyonudur. Dolayısıyla, kanonik değişken çifti,  $U_i, V_j$  arasındaki korelasyondur ve aşağıdaki formülden hesaplanabilir (Hardoon vd, 2004):

$$P_i^* = \frac{(kov(U_i, V_j))}{\sqrt{\text{var}(U_i, V_j)}} \quad (3.16)$$

Burada  $U_i$ , X için bir dizi doğrusal kombinasyon ve  $V_j$ , Y için bir doğrusal kombinasyon grubudur, kov kovaryans, p korelasyon ve var varyansı temsil etmektedir (Hardoon vd, 2004).

- Maksimum Olabilirlik Özellik Seçimi

Maksimum olabilirlik özellik seçimi (Suzuki vd, 2008), giriş hedefi bağımlılık ölçüsüne göre değişkenlere öncelik veren bir filtre, çok değişkenli ve denetimli özellik seçim yöntemidir. Tahminciler, giriş hedef arasındaki bağımlılığı ölçmek için maksimum olabilirlik karşılıklı bilgilerini (Suzuki vd, 2009) kullanırlar. Bu yöntem, yoğunluk tahminine dayanan karşılıklı bilgi tahmincisidir. Bu yöntemin yoğunluk oranı aşağıdaki formülü kullanarak hesaplanır (Suzuki vd, 2008):

$$w(x, y) = \frac{P_{xy}(x, y)}{P_x(x)P_y(y)} \quad (3.17)$$

Burada  $P_{xy}(x, y)$ , X ve Y'nin eklem yoğunluğudur,  $P_x(x)$ ,  $P_y(y)$ , sırasıyla X ve Y yoğunluklarıdır. Maksimum olabilirlik özellik seçimi, hem sınıflandırma hem de regresyon problemleri için kullanılan yöntemdir (Suzuki vd, 2008).

- En Küçük Kareler Özellik Seçimi

En küçük kareler özellik seçimi (Kanamori vd, 2009), Maksimum olabilirlik özellik seçim (Suzuki vd, 2008) yöntemine oldukça benzer; sadece tek farkı, bu yöntem kare kaybı karşılıklı bilgileri yoluyla özellikler ve hedef arasındaki bağımlılığı değerlendirir (Suzuki vd, 2009). Bu yöntemin yoğunluk oranı aşağıdaki formülü kullanarak hesaplanır:

$$r(x, y) = \frac{P(x, y)}{P(x)P(y)} \quad (3.18)$$

Burada  $P(x, y)$ , X ve Y'nin eklem yoğunluğudur,  $P(x)$ ,  $P(y)$ , sırasıyla X ve Y yoğunluklarıdır.

- Mesafe Ölçü Tabanlı Koşullu Karşılıklı Bilgi

Mesafe ölçme tabanlı koşullu karşılıklı bilgi (Latorre Carmona vd, 2012), hem tekli hem de çoklu çıktı regresyon görevlerini gerçekleştirmek için kullanılabilen denetimli ve filtre özellik seçim yöntemidir. Özelliklerin önemini belirlemek için bilgiye dayalı teknikler uygular. Bir veri kümesindeki kümeleri bulmak için Koşullu Karşılıklı Bilgiden faydalanır. Ayrıca, hedef değişkene göre en yüksek karşılıklı bilgiyi üreten bir özellik seçer. Bu yöntem, özellikle örnek sayısının özellik sayısından fazla olduğu zaman, küçük boyutlu veri setleri için iyi performans ürettiği gösterilmiştir (Latorre Carmona vd, 2012).

- Sınırsız Özellik Seçimi

Sınırsız özellik seçimi (Roffo vd, 2015) bir filtre ve denetimsiz özellik seçim yöntemidir. Bu yöntemde, her özellik bir grafikteki bir düğümle (Bonacich, 1987) temsil edilir ve özellikler merkez puanlarına göre seçilir. Tüm olası özellik alt kümeleri, grafikteki yollar olarak kabul edilir ve her özellik bu şekilde sıralanır. Sınırsız özellik seçiminin genel algoritması Tablo 3.11'de görülmektedir (Roffo vd, 2015):

Tablo 3.11. Sınırsız Özellik Seçiminin Genel Algoritması

---

Girdi :  $F = \{f^{(1)}, \dots, f^{(n)}\}, \alpha$

Çıktı : grafiği oluşturan her özellik için,  $\check{s}$  enerji puanları

for i = 1 : n do

for j = 1 : n do

$\sigma_{ij} = \max(\text{std}(f^{(i)}), \text{std}(f^{(j)}))$

$c_{ij} = 1 - |\text{Spearman}(f^{(i)}, f^{(j)})|$

$A(i, j) = \alpha \sigma_{ij} + (1 - \alpha) c_{ij}$

end for

end for

İzin yolları sınırsız olma eğilimindedir

$r = \frac{0.9}{p(A)}$

$\check{S} = (I - rA)^{-1} - I$

$\check{s} = \check{S}e$

return  $\check{s}$

---

- Laplacian Özellik Seçim Puanı

Laplacian özellik seçim puanı (He vd, 2006), temel olarak laplacian eigenmaps (Belkin ve Niyogi, 2002) ve yerellik koruma güçlerine (He ve Niyogi, 2004) dayanmaktadır; ayrıca bu yöntem özellikleri yerellik koruma güçlerine göre sıralayan grafik tabanlı, denetimsiz ve tek değişkenli özellik seçme algoritmasıdır (He vd, 2006; He ve Niyogi, 2004; Luxburg, 2007; Zhao ve Liu, 2007b). Laplacian algoritmasında özellikler bağımsız olarak değerlendirilir; dolayısıyla bu algoritma özellik fazlalığını

değerlendiremez (Zhao vd, 2010b). Bir özelliğin laplacian puanı aşağıdaki formülleri kullanarak hesaplanabilir (He vd, 2006; Li vd, 2017):

$$\text{Lap}(f_i) = \frac{\tilde{f}_i' L f_i'}{\tilde{f}_i' D f_i'} \quad (3.19)$$

$$f_i = f_i - \frac{f_i' D 1}{1' D 1} 1, \quad 1 = [1, 1, \dots, 1]' \quad (3.20)$$

$$D(i, j) = \sum_{j=1}^n S(i, j) \quad (3.21)$$

$$S(i, j) = \frac{e^{-\|x_i - x_j\|^2}}{t} \quad (3.22)$$

$$\text{Laplacian\_matrisi}(L) = D - S \quad (3.23)$$

Laplacian grafiğinin oluşturması ve aynı zamanda özelliklerin sayısının çok fazla olduğu durumda, hesaplama açısından maliyetli olduğu bilinmektedir (He vd, 2006).

- Spektral Regresyon Özellik Seçimi

Spektral regresyon özellik seçimi (Zhao ve Liu, 2007b), laplacian özellik seçim puanının bir uzantısı olarak düşünülebilir (Dy ve Brodley, 2004). Örnek olarak denetimsiz yöntemde, etiket bilgisi olmadan, spektral regresyon özellik seçimi iki nokta  $(x_i, x_j)$  arasındaki benzerliği hesaplamak için RBF çekirdek fonksiyonunu kullanır (Li vd, 2017; Zhao ve Liu, 2012):

$$S(i, j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (3.24)$$

Denetlenen yöntemde, etiket bilgileri kullanılarak, veri benzerliği aşağıdaki şekilde tanımlanabilir (Li vd, 2017):

$$S(i, j) = \begin{cases} 1 & \text{eğer } y_i = y_j = 1 \\ \frac{1}{n_1} & \\ 0 & \text{aksi takdirde} \end{cases} \quad (3.25)$$

Burada  $n_l$ , l sınıfındaki veri örneklerinin sayısıdır. Aşağıda derece matrisi(D), çapraz bir matristir. Laplacian matrisi ve derece matrisini hesaplamak için şu formüller kullanılır (Chung ve Graham, 1997):

$$D(i, i) = \sum_{j=1}^n W(i, j) \quad (3.26)$$

$$\text{Laplacian\_matrisi}(L) = D - W; \quad L = D^{\frac{-1}{2}} L D^{\frac{-1}{2}} \quad (3.27)$$

Spektral regresyon özellik seçiminin amacı, hem denetimli hem de denetimsiz özellik seçiminin bazı yapısal özelliklerini araştırmak ve spektral grafik teorisi üzerine inşa edilmiş birleşik bir çerçeve geliştirmektir (Zhao ve Liu, 2007b). Bu yöntem laplacian özellik seçim puanına benzer bir şekilde her özelliği bağımsız olarak değerlendirdiği için özellik fazlalığını değerlendiremez. Dolayısıyla, bu yöntemde özellikler arasındaki korelasyon dikkate alınmaz (Zhao vd, 2010b). Ayrıca Sınıflandırma problemleri için özellik seçimi söz konusu olduğunda, bu yöntem diğer yöntemlere göre ortalama bir metot olduğu gösterilmiştir (Hou vd, 2013). Dahası, spektral regresyon özellik seçimi çeşitli çalışmalarda regresyon görevleri için etkinliğini göstermiştir (Samsudin vd, 2015; Sarac vd, 2015, 2016).

- Karşılıklı Bilgi

Karşılıklı bilgi (Cover ve Thomas, 2012; Peng vd, 2005; Yu ve Liu, 2003), ilk olarak 1948 yılında Shannon tarafından önerilmiştir (Shannon, 1948). Bu yöntem tek değişkenli ve denetimli bir özellik ağırlıklandırma yöntemidir. Dahası, her bir özellik ile hedef sınıf etiketi arasında karşılıklı bilgileri hesaplar, daha sonra özellikler buna göre sıralanır ve en iyi özellikler seçilir (Mani ve Kalpana, 2016). Başka bir deyişle bu yöntem, iki rastgele değişkenin birbiri hakkında taşıdığı bilgi miktarını tanımlamaktadır. Ayrıca simetrik yapıya sahiptir ( $I(X; Y) = I(Y; X)$ ) ve değişkenler arasındaki doğrusal olmayan ilişkileri tespit edebilmektedir. Dolayısıyla çok popüler bir kriter haline gelmiştir (Battiti, 1994; François vd, 2007; Peng vd, 2005; Rossi vd, 2006). Nedeni ise, karşılıklı bilgi diğer yöntemler gibi sadece doğrusal bağımlılıkları işlemez (Doquire ve Verleysen, 2011). Karşılıklı bilgi yöntemi, hem bir özellik alt kümesinin hedef değişkenin öngörülmesindeki uygunluğunu hem de diğer değişkenlere göre fazlalığı değerlendirmek için filtre özellik seçme yöntemlerinde

başarıyla benimsenmiştir (Beraha vd, 2019). Karşılıklı bilgi aşağıdaki formülü kullanarak hesaplanabilir (Cover ve Thomas, 2012):

$$\text{karşılıklı\_bilgi}(X, Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x) * p(y)} \quad (3.28)$$

Burada X ve Y iki rastgele özellik veya değişkendir,  $p(x)$  ile  $p(y)$  olasılık yoğunluk fonksiyonları ve  $p(x, y)$  ise ortak olasılık yoğunluk fonksiyonudur (Kannan ve Ramaraj, 2010; Vergara ve Estévez, 2014). Karşılıklı bilgi, iki özellik arasında ne kadar bilgi ilişkilendirildiğini değerlendirmek için temel bir yöntemdir. Marjinal entropilerin toplamı ile eklem entropileri arasındaki fark olarak tanımlanır. Tamamen bağımsız iki nesne için karşılıklı bilgi her zaman sıfırdır (Singh vd, 2014). Bir tahmin veya sınıflandırma modeli, çıktındaki belirsizliği, bağımlı değişkeni azaltmayı amaçlamaktadır. Yukarıda belirtildiği gibi, bir dizi özelliklerin uygunluğunu değerlendirmek için iyi bir kriter, basitleştirilmiş bir tahmin modelidir. Doğal olarak, girdilerin bilinmesi nedeniyle, çıktının belirsizliğini ölçmektir (Rossi vd, 2006; Verleysen vd, 2009).

- Öklid Uzaklığı

Öklid uzaklığı, verileri arasındaki mesafeyi ölçmek için en yaygın kullanılan yöntemdir. Matematikte öklid mesafesi veya öklid uzaklığı iki nokta arasındaki ordinal mesafedir (Suebsing ve Hiransakolwong, 2009). Bu yöntem özellikler arasındaki korelasyonu hesaplar.  $A = (x_1, x_2, x_3, \dots, x_n)$  ve  $B = (y_1, y_2, y_3, \dots, y_n)$  noktaları arasındaki mesafe aşağıdaki formülü kullanarak hesaplanır (Kumar, 2014; Ladha ve Deepa, 2011) :

$$\text{öklid\_uzaklığı} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.29)$$

Yukarıdaki formülü kullanarak, özellikler arasındaki mesafe çok büyük ise, ilgisiz özellik olarak kabul edilir.

- Pearson Korelasyon Katsayısı

Pearson korelasyon katsayısı sadece sayısal verilerle çalışan, hem denetimli hem de denetimsiz, tek değişkenli bir yöntemdir. Bu yöntem değişkenler arasındaki

doğrusal ilişkinin varlığını, p değeri ile ve gücünü, r ile (-1 ve +1 arasında verilen r katsayısı ile) ölçer. Pearson korelasyon katsayısını hesaplamak için aşağıdaki formül kullanılır (Benesty vd, 2009; Goswami ve Chakrabarti, 2014; Sedgwick, 2012):

$$\text{pearson\_korelasyon\_katsayısı}(r) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} \quad (3.30)$$

Burada  $\bar{x}$  ve  $\bar{y}$  ortalama,  $\sigma_x$  ve  $\sigma_y$  ise  $x_i$  ve  $y_i$ 'nin standart sapmasıdır.

- Cramer's V Testi ile Özellik Seçimi

Ki-kare testi, değişkenler arasındaki ilişkiyi tanımlamak için yaygın olarak kullanılan yöntemlerden biridir ve ayrıca özellik seçiminde etkili olduğu kanıtlanmıştır (Lu ve Weng, 2007). Ancak ki-kare testinin örneklem büyüklüğüne çok hassas olduğu bilinmektedir (Agresti ve Agresti, 1970). Cramer'in V testi, değişkenler arasındaki ilişkinin gücünü ölçmek için kullanılan en popüler nominal yöntemdir (Garson, 2004). Örneklem büyüklüğünden etkilenmeme avantajına sahiptir ve dolayısıyla değişkenler arasında herhangi bir anlamlı ilişkiden ziyade istatistiksel olarak anlamlı bir ki-kare büyük bir örneklem büyüklüğünün sonucu olduğundan şüphelenilen durumlarda çok faydalıdır (Garson, 2004). Bu nedenle, hedef ve değişkenler arasındaki ilişkiyi ölçmek için Cramer'in V testi kullanılır. Örnek olarak, uzaktan algılanan görüntülerin sınıflandırma doğruluğunu geliştirmek için Cramer'in V testi kullanılır (LeKhac vd, 2013; Wu vd, 2013). S sütunu çapraz tablolama ile bir r satırı göz önüne alındığında, Cramer'in V doğrudan ki-kare istatistiğinden aşağıdaki gibi türetilir (Martínez Casanovas vd, 2008):

$$V = \sqrt{\frac{x^2}{N \min\{(r - 1), (s - 1)\}}} \quad (3.31)$$

Burada N toplam gözlem sayısıdır. Cramer'in V değeri 0 ile 1 arasında değişmektedir. V değeri büyükse, ilk değişkenin belirli sınıflarının ikinci değişkenin belirli sınıflarıyla ilişkilendirilme eğilimi vardır (LeKhac vd, 2013; Zhao vd, 2010b). Pratikte, bir Cramer's V'nin 0.1 değerinin, iki değişken arasında anlamlı bir ilişki olduğunu söylemek için iyi bir minimum eşik sağladığı vurgulanmıştır (Martínez Casanovas vd, 2008).  $x^2$  ise aşağıdaki gibi hesaplanır (Zhao vd, 2010b):

$$\chi^2 = \frac{\sum_{k=1}^c n_k (u_k^j - u_j)^2}{\sigma_j^2} \quad (3.32)$$

Burada k sınıfı ve i,j ise özelliği belirtmektedir (Mani ve Kalpana, 2016).

- Markov Battaniye Filtresi

Markov battaniyesi ilk olarak 1988’de Pearl’in çalışmalarında ortaya çıkmıştır (Pearl, 1988). Ancak Koller ve Sahami 1996’da ilk olarak, belirli bir hedef değişken T'nin Markov battaniyesinin, T'nin değerini tahmin etmek için teorik olarak en uygun özellik kümesi olduğunu gösterdiler (Koller ve Sahami, 1996). Belirli bir hedef değişken T'nin battaniyesi, T'nin değerini tahmin etmek için teorik olarak en uygun özellik kümesidir. Markov battaniyesinin tam bilgisinin T'nin olasılık dağılımını belirlemek için yeterli olduğunu ve diğer tüm değişkenlerin değerlerinin gereksiz hale gelmesinden dolayı, markov battaniyesi özellik seçimi prosedürüdür (Koller ve Sahami, 1996; Tan ve Liu, 2013; Tsamardinos vd, 2003a; Tsamardinos vd, 2003b). Bir hedef değişkenin markov battaniyesi, diğer tüm değişkenlerin koşullu olarak hedef değişkenden bağımsız olduğu küçük bir değişken kümesi içerir. Araştırma çalışmalarının çoğu yeni bir markov battaniye algoritmasını keşfetmeyi hedefliyor ve daha sonra nihai markov battaniye sınıflandırıcısını oluşturuyor (Aliferis vd, 2003; Margaritis ve Thrun, 2000; Tsamardinos vd, 2003b).

$$P(F_i|F_k, S) = P(F_i|S) \quad (3.33)$$

$$\forall F_k \in F - (M_i \cup \{F_i\}) \text{ s.t } P(F_i|M_i, F_k) = P(F_i|M_i) \quad (3.34)$$

S özellik uzayından alakasız özellikleri kaldırmak için, bu filtre her bir  $F_i$  özelliğinin Markov battaniyesinin  $M$ 'sini  $S$ 'den hesaplar.  $M$ , markov’un  $F_i$  battaniyesi ise,  $F_i$  şartlı olarak  $S$ ,  $M$  ve  $F_i$  ,alt kümesinden bağımsızsa,  $F_i$ ,  $S$  özellik alanından kaldırılabilir (Koller ve Sahami, 1996; Kumar, 2014; Shen vd, 2008; Yu vd, 2016).

- Kruskal Wallis Testi

Kruskal Wallis iki veya daha fazla sınıfın eşit medyana sahip olup olmadığını test eden ve buna göre bir değer sağlayan denetimli, tek değişkenli ve parametrik olmayan bir özellik ağırlıklandırma yöntemidir. Başka bir deyişle, bu yöntem hesaplama maliyeti açısından daha ucuz ve kullanımı çok basit olan özellik seçme

yöntemidir. Değer sıfıra yakınsa, özelliğin ayırıcı olduğu söylenir. Bu yöntem, ayırıcı bilgiler içeren bir özellik seçer ve diğerlerini kaldırır. Diğer istatistiksel testlere benzer şekilde, kruskal wallis testi de test istatistiğini hesaplar ve dağılım kesme noktasıyla karşılaştırır (Ali Khan vd, 2014; Daniel, 1990; Saeys vd, 2007). Kruskla wallis testini uygulamak için aşağıdaki formül kullanılır (Naik ve Rangwala, 2016) :

$$\text{kruskal\_wallis} = (N - 1) \frac{\sum_{i=1}^L n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^L \sum_{j=1}^{n_i} n_i (r_{ij} - \bar{r})^2} \quad (3.35)$$

Burada  $n_i$ ,  $i$  sınıfındaki örneklerin sayısıdır,  $r_{ij}$ ,  $i$  sınıfındaki  $j$  örneklerin sıralamasıdır ve  $\bar{r}$ , tüm örnekler arasındaki ortalama sıralamayı gösterir (Naik ve Rangwala, 2016).

- Hızlı Korelasyon Tabanlı Özellik Seçimi

Hızlı korelasyon tabanlı filtre özellik seçim yöntemi (Yu ve Liu, 2003), özellik-sınıf ve özellik-özellik korelasyonunu ölçen çok değişkenli bir algoritmadır. Bu yöntem, bilgi kazancı (denklem (3.5), (3.6) ve (3.7)) ile iki özelliğin entropisi arasındaki oran olarak tanımlanan ve simetrik belirsizliğe (denklem(3.9)) dayalı olarak sınıfla yüksek derecede korelasyona sahip bir dizi özellik seçerek başlar. Daha sonra, gereksiz özellikleri kaldıran ve sınıfla daha ilgili olan özellikleri tutan üç sezgisel tarama uygular. Hızlı korelasyon tabanlı özellik seçimi, yüksek boyutlu veriler için tasarlanmıştır ve alakasız ve gereksiz özelliklerin giderilmesinde etkili olduğu gösterilmiştir. Ancak, özellikler arasındaki etkileşimi dikkate almaz. Genel olarak bu yöntem, diğer alt küme seçim yöntemlerinden önemli ölçüde daha hızlı çalışan korelasyon tabanlı bir özellik alt küme seçim yöntemidir (Bolón-Canedo vd, 2014; Li vd, 2017; Mani ve Kalpana, 2016; Sánchez-Marroño vd, 2007; Senliol vd, 2008; Vora ve Yang, 2017).

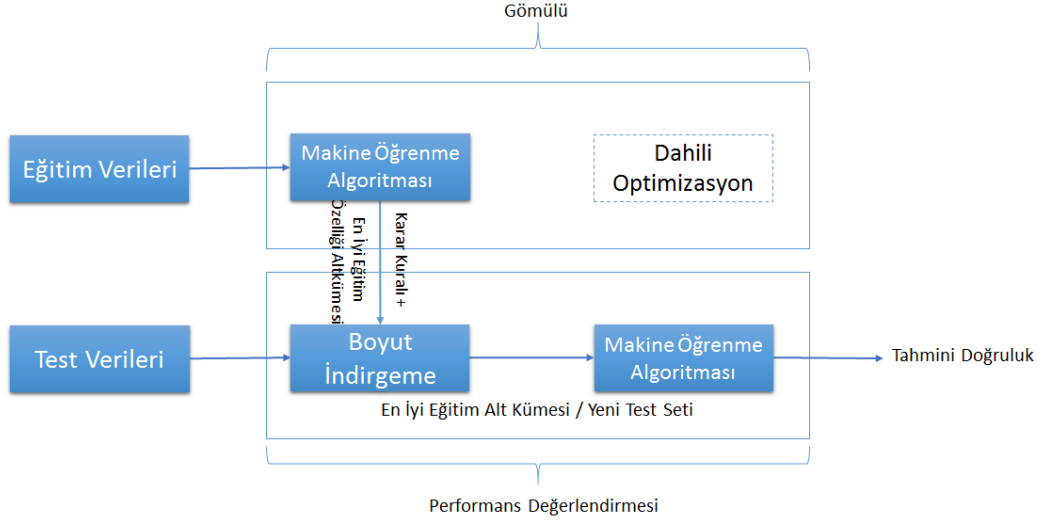
### 3.1.3.3. Gömülü Yöntemleri

Gömülü yöntemi, öğrenme algoritmasına özellik seçimini ekleyen ve özelliklerin özellik değerlendirmesini yönlendirmek için kullanılan yerleşik bir özellik seçme tekniğidir (Ang vd, 2015). Gömülü yöntem sınıflandırıcının tekrarlı yürütülmesini ve her özellik alt kümesinin incelenmesini engeller. Dolayısıyla gömülü

yöntem, sargı yönteminden daha verimli ve hesaplanabilir olarak daha uysaldır. Ayrıca, bu yöntemin takılma riski sargı yöntemine karşı daha düşüktür. Sarıcı gibi gömülü yöntemi de özellikler arasındaki bağımlılıkları hesaba katar, ancak belirli bir öğrenme algoritmasına mahsusdur (Saeys, 2004).

Bu yöntem öğrenme sırasında en iyi özelliklerin seçilmesine çalışmaktadır. Öğrenme süreci sırasında özellik seçiminin harmanlanmasını, hesaplama maliyetinin düşmesini, sınıflandırma doğruluğunun yüksek olmasını sağlamaktadır ve ayrıca her yeni özellik eklendiğinde modelin eğitilmesini önlemektedir. Bu yöntem, hesaplama karmaşıklığının üstesinden gelebilmektedir. Dahası, uygun özellik seçimi ve model öğrenme aynı anda gerçekleştirilir ve özellikler modelin eğitim aşamasında seçilir. Dolayısıyla, bu yöntemin hesaplama maliyeti sargı yönteminden daha azdır (Bolón-Canedo vd, 2013).

Gömülü yöntemler özellik seçimini ayırmaz ve sınıflandırıcıyı eğitmez. Hesaplama sürecinde bir seçim yaparlar. Bu yöntemlerin temel tekniği düzenliliktir. Düzenleme fikri, sadece hatayı değil, aynı zamanda kullanılan özellik sayısını da en aza indiren bir algoritma oluşturmaktır (Danilov vd, 2020). Gömülü yöntemleri yerleştirme hatalarını en aza indirerek ve özellik katsayılarını aynı anda küçülterek ya da tam sıfıra olmaya zorlayarak bir öğrenme modeline uymayı hedefleyen düzenlileştirme modelleridir (Li vd, 2017). Popüler özellik seçimi gömülü yöntemlerinden biri, Tikhonov düzenleme yöntemi (Wang vd, 2013) veya sırt regresyonudur (Paul ve Drineas, 2016). Gömülü özellik seçim yöntemlerinin genel bir çerçevesi Şekil 3.9'da gösterilmiştir.



Şekil 3.9. Gömülü özellik seçim yöntemlerinin genel çerçevesi

- En Küçük Mutlak Büzülme ve Seçim Operatörü

En küçük mutlak büzülme ve seçim operatörü ilk olarak 1996 yılında Robert Tibshirani tarafından ortaya çıkmıştır (Tibshirani, 1996). Bu yöntem düzenleme ve özellik seçimi görevlerini yerine getirerek güçlü bir yöntem oluşturmaktadır. Ayrıca bu yöntem, model parametrelerinin mutlak değerlerinin toplamına bir kısıtlama getirir ve bundan dolayı toplamın sabit bir değerden daha az olması gerekmektedir. Bunu yapmak için yöntem, bir kısmını sıfıra düşüren regresyon değişkenlerinin katsayılarını cezalandırdığı bir düzenleme süreci uygular. Özellik seçme işlemi sırasında, büzülme işleminden sonra hala sıfır olmayan bir katsayısı olan değişkenler modelin bir parçası olarak seçilir. Bu işlemin amacı tahmin hatasını en aza indirmektir (Fonti ve Belitser, 2017; Hara ve Maehara, 2017). En küçük mutlak büzülme ve seçim operatöründe özellik katsayısı  $w$ 'de L1 normlu düzenleme terimine dayanmaktadır (Friedman vd, 2010; Tibshirani, 1996) :

$$w = \|w\|_1 = \sum_{i=1}^d |w_i| \quad (3.36)$$

L1-norm düzenleme süresi bazı özellik katsayılarını sıfır olmaya zorlar ve bu özelliklerin ortadan kaldırılması, öğrenme performansını büyük ölçüde etkilemeyeceğinden karşılık gelen özellikler basitçe ortadan kaldırılabilir. Özellik ağırlığını ( $w$ ) bazı optimizasyon algoritmalarının yardımıyla elde ettikten sonra, özellik önemini özellik ağırlığına göre sıralamak mümkündür ve aynı zamanda özellik

ağırlığı ne kadar yüksek olursa, özellik o kadar önemlidir (Larochelle ve Bengio, 2008; Li vd, 2017). Ayrıca, bu yöntemin çözümünün verimli bir şekilde hesaplanması için çeşitli optimizasyon yazılım paketleri geliştirilmiştir (Afonso vd, 2010; Boyd vd, 2004; Combettes ve Wajs, 2005; Daubechies vd, 2004; Tomioka vd, 2011; Wright vd, 2009; Yin vd, 2008). Bu yöntem aynı zamanda genomik için yaygın olarak kullanılmaktadır (Ghosh ve Chinnaiyan, 2005).

Bu yöntem, en küçük kare çizgi regresyon için L1 norm düzeninden yararlanır ve aşağıdaki objektif işlevini en aza indirmeye çalışır (Chen vd, 2001):

$$EKMSO = \underset{\beta}{\operatorname{argmin}} \|y - \beta X\|_2^2 + \lambda \|\beta\|_1 \quad (3.37)$$

Burada rastgele yanıt değişkeni  $Y \in R$ , bir  $d$  boyutlu değişken  $X \in R^d$ 'ye bağımlıdır ve  $D = (x_i, y_i)_1^n$  eğitim verisi, sabit bir eklem dağılımı  $P_{xy}$  'den bağımsızdır ve özdeş olarak örneklendirilmiştir. Ayrıca  $\lambda$  ise normalleştirme parametresidir (Zhao ve Yu, 2006).

- Özellik Seçimi İçin en Küçük Açık Regresyonu

Özellik seçimi için en küçük açık regresyonu (Efron vd, 2004), Weisberg'in (Weisberg, 2005) dediği gibi İleri Adım Regresyon ile İleri Seçim olarak bilinen klasik model seçim yöntemiyle ilgilidir (Efron vd, 2004; Hettigoda, 2016). Bu yöntem ayrıca en küçük mutlak büzülme ve seçim operatörüne benzemektedir ve bu yöntemin aşamalı bir versiyonunda olarak düşünülebilir (Fralely ve Hesterberg, 2009).

Özellik seçimi için en küçük açık regresyonu, hedef( $y$ ) ile en büyük mutlak korelasyona sahip olan bir veri kümesinden bir  $x_i$  seçer ve sonra  $x_i$  üzerinde  $y$ 'nin basit doğrusal regresyonunu gerçekleştirir.  $x_i$ 'ye dik olan diğer öngörücüler seçilir ve daha sonra seçilen öngörücülerden faydalanılarak doğrusal bir model oluşturulur. Bu yöntem, L1 norm düzenlenmesi ile doğrusal regresyon gerçekleştirir; dolayısıyla, katkı maddelerinin çoğu sıfıra doğru küçülür ve böylece özellik seçimi gerçekleştirilir. Aynı zamanda başlangıçtaki katsayılar sıfırdır ( $\beta = 0$ ). En küçük açık regresyonu bu katsayıları artırır ve böylece tüm katsayı sıfır olmayana kadar özellikleri her adımda çıkış değişkeni ile en yüksek korelasyona sahiptir (Efron vd, 2004).

- Elastik Ağ Yöntemi ile Özellik Seçimi

En küçük mutlak büzülme ve seçim operatöründe, seçilen özelliklerin sayısı genellikle birçok uygulamada gerçekçi olmayan veri örneği sayısı ile sınırlıdır. Ayrıca, biyoinformatik, görüntü işleme ve doğal dil işleme gibi birçok uygulamada, özelliklerin bazı güçlü korelasyonları olabileceği bilinmektedir (Liu vd, 2007; Mitra vd, 2002; Segal vd, 2003). Bundan dolayı bu yöntem özellik seçimi için kullanılabilir. Ayrıca bu yöntem, yüksek derecede korelasyonlu özelliklerden sadece birini seçme eğilimindedir; Ancak her zaman en iyi seçenek olmayabilir (Waldron vd, 2011). Dolayısıyla yüksek korelasyonlu özellikleri seçmek için Zhu ve Hastie, aşağıda verilen L1 ve L2 normlarının normalleştirilmesini kullanan elastik ağ yöntemini önerdiler (Zou ve Hastie, 2005):

$$w = \sum_{i=1}^n |w_i|^\gamma + \left( \sum_{i=1}^n w_i^2 \right)^\lambda \quad (3.38)$$

Burada  $0 \leq \gamma \leq 1$  ve  $\lambda \geq 1$  ayrı ayar parametrelerdir (Cui vd, 2019). Elastik ağ yöntemi, eşzamanlı olarak özellik seçimi ve regresyonu gerçekleştiren bir azınlık tabanlı özellik seçim yöntemidir. Bundan dolayı bu yöntem eşzamanlık probleminin üstesinden gelmektedir (Chiu ve Yao, 2013; Desboulets, 2018; Li vd, 2017; Liu ve Li, 2017).

- Çoklu Küme Özellik Seçimi

Çoklu küme özellik seçimi (Cai vd, 2010), spektral regresyon (Dy ve Brodley, 2004; Ng vd, 2002) ve L1 norm düzenlenmesini (Efron vd, 2004; Friedman vd, 2001) kullanan bir dizi özellik seçen, denetimsiz ve gömülü özellik seçim yöntemlerinden birisidir (Gupta ve Begum, 2019). Bu yöntem özellikler arasındaki korelasyonu, spektral analiz kullanarak değerlendirir (Li vd, 2017). Bu yöntemle seçilen özellikler çok kümeli veri yapısını korur (Li vd, 2015) ve verilerdeki tüm geometrik etkileşimi yakalar. Spektral kümeleme ilk önce verilerin geometrik yapısını, ardından bu veri noktalarının düz çıkıntılarını çeşitli algoritmalarıyla ortaya çıkarır (Ng vd, 2002; Roweis ve Saul, 2000; Tenenbaum vd, 2000). Bu veri noktaları daha sonra geleneksel kümeleme yöntemleri kullanılarak kümelendir (Ng vd, 2002).

Mevcut seyrek öğrenme tabanlı yaklaşımların çoğu, sınıf etiketlerinin denetimi ile bir öğrenme modeli oluşturmaktadır. Özellik seçim aşaması daha sonra seyrek özellik katsayılarından türetilir. Öte yandan, etiketlenmiş verilerin elde edilmesi

maliyetli ve zaman alıcı olduğundan dolayı, denetimsiz seyrek öğrenme tabanlı özellik seçimi son yıllarda giderek daha fazla araştırma konusu olmuştur (Cai vd, 2010; Du ve Shen, 2015; Hou vd, 2013; Li vd, 2012; Liu vd, 2013; Qian ve Zhai, 2013; Yang vd, 2011b).

Çoklu küme özellik seçimi üç ana adımdan oluşmaktadır. İlk adım girdi verilerinin küme yapısını açıklamak için kullanılan spektral kümelemedir. İkinci adım seyrek ve etkili öğrenme, son adım ise özellik seçimidir. İlk adımda, verilerin küme yapısını tespit etmek için veri kümesine spektral kümeleme uygulanır (Chan vd, 1994; Ng vd, 2002). İlk olarak verinin yerel geometrik yapısını yakalamak için k-en yakın komşu grafiği oluşturur ve k, önceden tanımlanmış bir parametre olan grafik benzerlik matrisini(S) alır ve şu şekilde ifade edilebilir (Cai vd, 2010):

$$S_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}} \quad (3.39)$$

Burada  $x_i$  ve  $x_j$ , KNN grafiğindeki örneklerdir ve  $\sigma$ , önceden tanımlanmış bir parametredir. Laplacian matrisi  $L = D - S$  üzerinden hesaplanır; burada D çapraz matristir. Çoklu küme özellik seçimi, L1 normu ve spektral regresyon kullanır ve aşağıdaki işlevi en aza indirmeye çalışmaktadır (Cai vd, 2010):

$$L_z = \lambda D_z \quad (3.40)$$

Burada  $Z = [z_1, z_2, \dots, z_t]$  öz vektörleri gösterir ve t önceden belirlenmiş bir parametredir. İlgili işlevlerin bir alt kümesi, aşağıdaki işlevi en aza indirerek bulunabilir (Cai vd, 2010):

$$\min_{w_i} \|xw_i - e_i\|_2^2 + \alpha \|w_i\|_1 \quad (3.41)$$

burada  $w_i$ , i-inci gömme için özellikli vektördür. Çoklu küme özellik seçimi, seyrek regresyon problemlerini çözer ve seyrek özellik katsayıları alır,  $W = [W_1, W_2, \dots, W_t]$  ve her bir ortak veri tek bir veri gömmesine karşılık gelir. Bu yöntem, özellikleri puanlarına göre aşağıdaki formül ile hesaplanabilecek şekilde sıralar (Cai vd, 2010):

$$\text{çoklu\_küme\_puanı} = \max_i |W_{i,j}| \quad (3.42)$$

Burada  $W_{i,j}$ ,  $W$  vektörünün  $j$ -inci ögesi ve ne kadar yüksek olursa, özellik o kadar önemlidir (Cai vd, 2010; Li vd, 2017).

- Denetimsiz Ayrımcı Özellik Seçimi

Denetimsiz özellik seçim yöntemini gerçekleştirmek için, yaygın olarak kabul edilen bir kriter, verilerin çeşitli yapısını en iyi koruyan özellikleri seçmektir (Cai vd, 2010; He vd, 2006; Zhao ve Liu, 2007b). Alternatif bir yol, birçok öğrenme problemlerinde etkili olduğu kanıtlanan, verilerde kodlanan ayrımcı bilgilerin kullanılmasıdır (Fukunaga, 2013). (Yang vd, 2011b)'de, hem ayrımcı bilgileri hem de özellik korelasyonlarını kullanarak en ayrımcı özellikleri seçmek için yeni bir denetimsiz özellik seçim algoritması önermişlerdir. Denetimsiz ayrımcı özellik seçim yöntemi (Yang vd, 2011b), özellikleri seçmek için yerel ayrımcı bilgileri ve özellik korelasyonlarını birlikte kullanan seyrek öğrenme tabanlı, gömülü ve denetimsiz özellik seçim yöntemidir. Bu yöntemde Özellikler ortak bir ayrımcılık analizi ve  $L_{2,1}$  normunun en aza indirilmesi ile seçilir (Miao ve Niu, 2016; Wang vd, 2020). Denetimsiz ayrımcı özellik seçimi, aşağıdaki fonksiyonu çözmeye çalışır (Yang vd, 2011b):

$$\min_{W'W=I} \text{tr}(W'XLX'W) + \beta\|W\|_{2,1} \quad (3.43)$$

Burada  $I$  kimlik matrisidir,  $W$  bir yansıtma matrisidir,  $L = D^{-\frac{1}{2}}(D - S)D^{-1/2}$ ,  $X$  giriş verisidir,  $\text{tr}$  matrisin izidir,  $\|W\|_{2,1}$ ,  $L_{2,1}$  norm düzenini gösterir ve  $W'W$ 'nin devrik matrisidir (Fan vd, 2017; Li vd, 2017; Yang vd, 2011b).

- Güçlü Denetimsiz Özellik Seçimi

Güçlü denetimsiz özellik seçimi (Qian ve Zhai, 2013), güçlü özellik seçimi ve güçlü kümeleme gerçekleştirerek ayrımcı özellikleri seçen, gözetimsiz, seyrek öğrenme tabanlı ve gömülü bir özellik seçme algoritmasıdır (Solorio-Fernández vd, 2020; Wang vd, 2015). Bu yöntem aşağıdaki amaç işlevlerini çözmeye çalışmaktadır (Qian ve Zhai, 2013) :

$$\min_{F, G, W} \|X - GF\|_{2,1} + v\text{Tr}[G'LG] + \alpha\|XW - G\|_{2,1} + \beta\|W\|_{2,1} \quad (3.44)$$

$$G \in R_+^{n \times c}, G = Y(Y'Y)^{-1/2}, F \in R_+^{c \times d}$$

Burada  $v, \alpha, \beta \in R_+$  kullanıcı tanımlı parametrelerdir,  $G$ , sözde sınıf etiketlerini temsil eden ağırlık kümesi gösterge matrisidir,  $X$  giriş verileridir,  $W$ , yansıtma matrisidir,  $L$ , Laplacian matrisidir ve  $F$  ise, orijinal özellik alanının tamamındaki küme merkezleridir (Qian ve Zhai, 2013).

- Ortak Gömme Öğrenme ve Seyrek Regresyon

Ortka gömme öğrenme ve seyrek regresyon (Hou vd, 2013), özellik seçimini gerçekleştirmek için seyrek regresyonla gömülü öğrenmeyi birleştiren denetimsiz, seyrek öğrenme tabanlı ve gömülü bir özellik seçme yöntemidir. Bu yöntem, çoklu küme özellik seçimi ve an az fazlalık özellik seçimi yöntemlerine oldukça benzemektedir; ayrıca bu yöntem, yerel minimum yaklaşık ağırlık ve  $l_{2,1}$  norm düzenlenmesini uygulayarak yeni bir yöntem sunmaktadır (Hou vd, 2013; Li vd, 2017). Bu algoritmada, yerel çizgi yaklaşım ile ağırlığı kullanan ve optimizasyon problemini çözmek için birlikte etkili algoritmayı sağlayan normalleştirme yöntemini ekleyen bir yöntem sağlanmakta ve yakınsama analizi, hesaplama karmaşıklığı ve parametre belirleme üzerine bazı görüşlü tartışmalar yapılmaktadır (Miruthula ve Roopa, 2015). Ortka gömme öğrenme ve seyrek regresyon, aşağıdaki optimizasyon işlevini çözmeye çalışmaktadır (Hou vd, 2013; Hou vd, 2011):

$$\min_{WY} \text{Tr}(YLY') + \beta \|W'X - Y\|_2^2 + \alpha \|W\|_{2,1} \quad (3.45)$$

$$YY' = I$$

Burada  $Y$ , girdinin düşük boyutlu temsilidir,  $X$  ve  $W$  yansıtma matrisidir,  $\text{Tr}$  izdir ve  $\alpha$  ve  $\beta$  parametrelerdir (Hou vd, 2013).

- Uyarlanabilir Yapısal Öğrenme ile Denetimsiz Özellik Seçimi

Uyarlanabilir yapısal öğrenme ile denetimsiz özellik seçimi (Du ve Shen, 2015), özellik seçimi ve yapısal öğrenmeyi birlikte gerçekleştiren seyrek öğrenme tabanlı, gömülü ve denetimsiz özellik seçim yöntemidir. Çoklu küme özellik seçimi ve ortak gömme öğrenme ve seyrek regresyon gibi diğer gömülü özellik seçim yöntemlerinin aksine, daha iyi yapı öğrenimi sağlamak için yapı öğrenme prosedürünü beslemek için

özellik seçiminin çıktısını kullanır. Bu yöntem aşağıdaki optimizasyon problemini çözmeye çalışmaktadır (Du ve Shen, 2015; Luo vd, 2017):

$$\min_{W, S, P} (\|W'X - W'XS\|^2 + \alpha\|S\|_1) + \left( \sum_{ij}^n \|W'x_i - W'x_j\|^2 P_{ij} + \mu P_{ij}^2 \right) + \gamma \|W\|_{2,1} \quad (3.46)$$

$$S_{ii} = 0; \quad P1_n = 1_n; \quad P \geq 0; \quad W'XX'W = I$$

burada  $W \in R^{d \times c}$  dönüşüm matrisidir,  $\gamma$  düzenleme parametresidir,  $S \in R^{n \times n}$  aşağıdaki işlevden elde edilebilen optimal seyrek kombinasyon ağırlık matrisidir (Du ve Shen, 2015):

$$\min_S \sum_{i=1}^n \|x_i - xS_i\|^2 + \alpha \|S_i\|_1 \quad (3.47)$$

$$S_{ii} = 0$$

Burada  $\alpha$  seyreklik ve yeniden yapıma hatasını dengelemek için kullanılır.  $P_{ij} \in R^{n \times n}$  olasılıklı komşu matrisidir ve aşağıdaki formülü kullanarak hesaplanabilir (Du ve Shen, 2015):

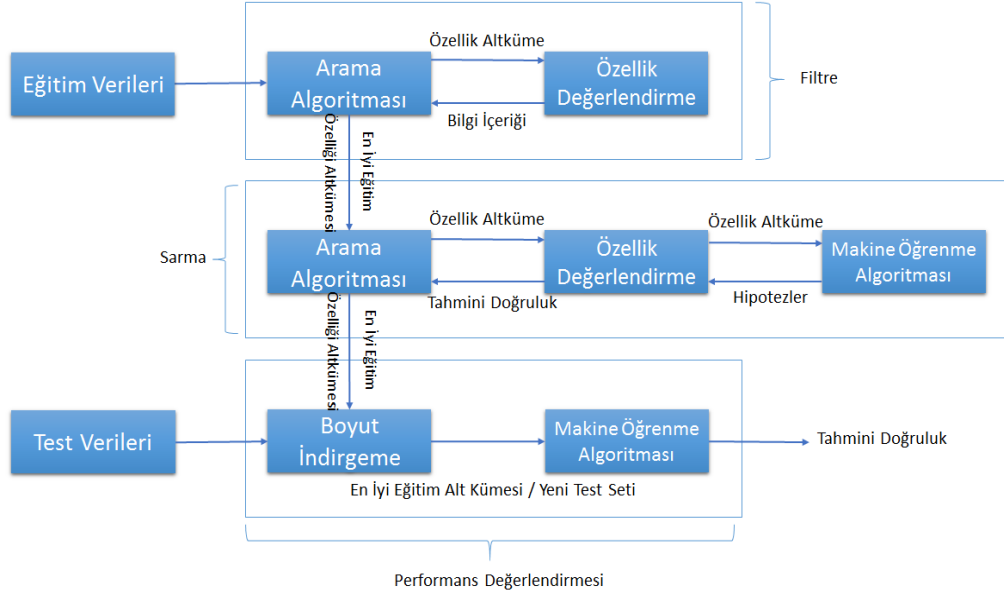
$$\min_P \sum_{i=1}^n \|x_i - x_j\|_2^2 + P_{ij} + \mu P_{ij}^2 \quad (3.48)$$

$$P1_n = 1_n; \quad P \geq 0$$

Burada  $\mu$  normalleştirme parametresi ve  $1_n$  tümüyle n-boyutlu vektördür (Du ve Shen, 2015).

### 3.1.3.4. Hibrit Yöntemleri

Hibrit yöntem, iki özellik seçim yaklaşımının birleştirilmesiyle oluşturulabilir. Bu yöntem, tamamlayıcı güçlerini birleştirerek her iki yöntemin avantajlarını miras almaya çalışmaktadır (Kabir vd, 2010). Daha iyi hesaplama performansı ile verimlilik ve tahmin performansını artırmak için farklı arama aşamalarında farklı değerlendirme kriterleri kullanır. En yaygın kullanılan hibrit yöntem, filtre ve sarmal yöntemlerinin kombinasyonudur (Peng vd, 2010) ve bu kombinasyonun genel bir çerçevesi Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Hibrit (Sarmal ve Filter) yöntemlerinin genel çerçevesi

Tablo 3.12. Hibrit Yönteminin Genel Algoritması

Girdiler :

$$D = X, L$$

$n$  sayıda özelliğe sahip bir eğitim veri seti :

$$x = \{f_1, f_2, f_3, \dots, f_n\} \text{ ve } L \text{ ise etiketler}$$

$X'$

önceden tanımlanmış ilk özellik alt kümesi  
( $X' \subset X$  yada  $X' = \{\emptyset\}$ )

$\Theta$

Durdurma kriterleri

Çıktılar :

$X'_{opt}$  :

En uygun alt kümeyi bulmak

Begin :

Initialize :

$$X_{opt} = X' ;$$

$$\mathcal{J}_{opt} = E(X', I_m) ;$$

$I_m$ 'i bağımsız bir ölçü birimi kullanarak  $X'$  i değerlendirir

$$\mathcal{J}_{opt} = E(X', A) ;$$

$A$  araştırma algoritmasını kullanarak  $X'$  i değerlendirir



$k+1$  olası tüm kardinalite alt kümelerini aranır. Kardinalite  $k+1$  ile her yeni oluşturulan alt küme, bağımsız bir  $I_m$  ölçüsü ile değerlendirilir ve bir öncekiyle kıyaslanır.  $X_g$  daha iyiye, mevcut en iyi alt küme  $X'_{opt}$  en iyi düzey  $k+1$  olur. Her tekrarlamamın sonunda,  $X'_{opt}$  üzerine en iyi  $k+1$  seviyesinde bir madencilik algoritması uygulanır ve  $k$  seviyesindeki en iyi alt kümeden çıkarılan sonucun kalitesi karşılaştırılır. Eğer  $X'_{opt}$  en iyiye, algoritma bir sonraki seviyedeki en iyi alt kümeyi bulmaya devam eder; aksi takdirde, geçerli en iyi alt kümeyi durdurur ve son en iyi alt küme olarak çıkarır (Liu ve Yu, 2005).

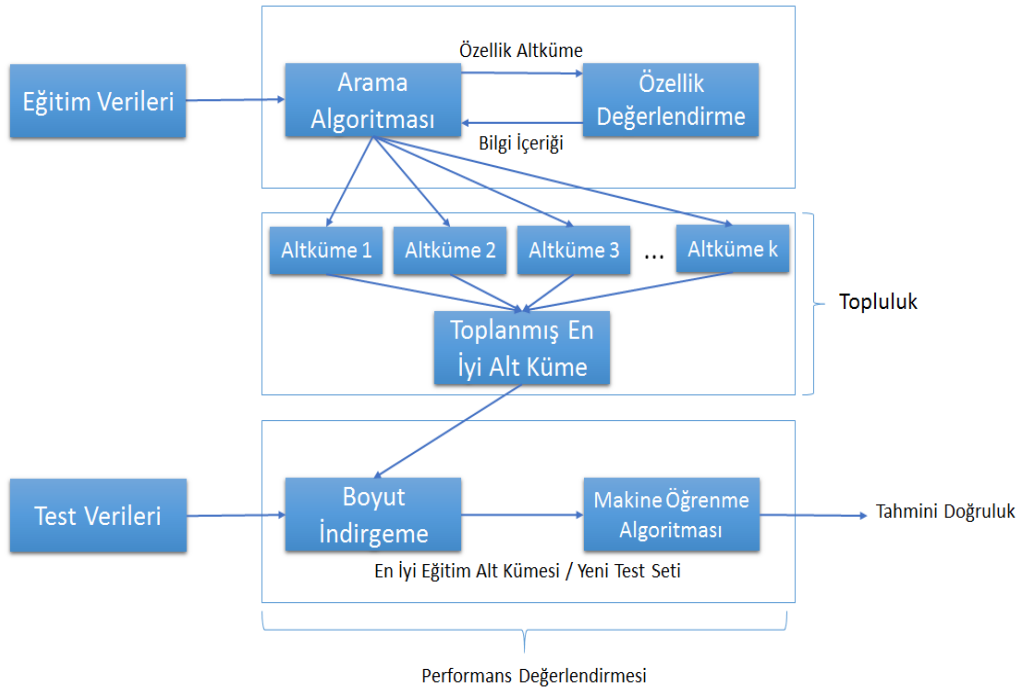
### 3.1.3.5. Topluluk Yöntemleri

Araştırmacılar özellik seçme yöntemlerinin kararlılığını artırmak için, topluluk özellik seçme yöntemleri (Abeel vd, 2010; Haury vd, 2011; Liu vd, 2010a; Saeys vd, 2008; Yang vd, 2011a; Yang vd, 2010; Yu vd, 2011) gibi yeni çerçeveler önermişlerdir. Topluluk öğrenimi makine öğrenimi için faydalı bir tekniktir. Temel fikir, farklı öğrenme modellerini bir araya getirerek daha iyi öğrenme sonuçları elde etmektir. Topluluk öğrenme yöntemleri birçok makine öğrenme yöntemi arasında, tek makine öğrenme modellerinden daha üstündür. Son zamanlarda, topluluk öğrenme fikrinden yararlanarak topluluk özellik seçiminin gelişimi daha hızlanmış. Birkaç özellik seçim yönteminden farklı olarak, yalnızca tek bir optimum özellik alt kümesi aranır. Kombine özellik seçiminin amacı, birden fazla optimum özellik elde etmektir. Entegrasyon teknolojisi sayesinde, grup özellik seçim algoritması, diğer özellik seçim algoritmalarından daha iyi kararlılığa ve sağlamlığa sahiptir (Wang vd, 2019).

Topluluk özellik seçme yöntemleri veri bozulması ve işlev bozulması olarak iki ana başlıkta incelenmektedir (He ve Yu, 2010). Veri bozulması ya da bazen homojen topluluk yaklaşımı olarak da adlandırılan yöntemde, özellik seçimi örneklerin çeşitli alt kümelerinde gerçekleştirilir. Her analiz, potansiyel olarak farklı özellik alt kümeleri oluşturur. Dolayısıyla, tüm alt kümeleri analiz etmek için aynı özellik seçim yöntemi kullanılır. Elde edilen özellik alt kümeleri daha sonra bir son özellik alt kümesine toplanır (Abeel vd, 2010; Bach, 2008; Davis vd, 2006; Seijo-Pardo vd, 2017). (Pes vd, 2017) yaptıkları çalışmada, veri bozulmasının orijinal özellik seçim yönteminin kararlılığını yükseltebileceğini göstermiştir. İşlev bozulması ya da heterojen topluluk yaklaşımı olarak da bilinen teknikte, her bir yöntemin güçlü ve zayıf taraflarını hesaba katmak için aynı eğitim seti üzerinde uygulanan çeşitli özellik seçim yöntemlerinin

sonuçlarını birleştirir. Dolayısıyla işlev bozukluğu sınıflandırma performansını koruyabilir veya geliştirebilir (Ahmed vd, 2014; Feldotto ve Graffi, 2013; Tan vd, 2009).

Topluluk yönteminin amacı, bir grup özellik alt kümesini oluşturmak ve daha sonra gruptan toplu bir sonuç elde etmektir (Shen vd, 2012). Bu yöntem birçok özellik seçim algoritmasında kararsızlık ve bozulma problemlerinin üstesinden gelmek için tasarlanmıştır. Bu yöntem, belirli bir özellik seçim yönteminin bir dizi alt örnek üzerinde çalıştırıldığı ve elde edilen özelliklerin daha kararlı bir alt küme oluşturmak için birleştirildiği farklı alt örnekleme stratejilerine dayanmaktadır. Özellik seçiminin performansı artık seçilen tek bir alt kümeyle bağlı değildir, dolayısıyla yüksek boyutlu verilerle çalışırken daha esnek ve sağlamdır. Ayrıca, topluluk yöntemi, çeşitli özellik seçicilerinin çıktılarını toplayarak, özelliklerin alt kümesine veya sıralamasına daha iyi bir yaklaşım sağlar. Topluluk özellik seçiminin genel bir çerçevesi Şekil 3.11’de gösterilmiştir.



Şekil 3.11. Topluluk özellik seçiminin genel çerçevesi

Tablo 3.13. Özellik değerlendirme yöntemlerinin avantajları ve dezavantajları

Yöntem ismi	Avantajları	Dezavantajları
-------------	-------------	----------------

Sarmal	<ul style="list-style-type: none"> <li>• Sınıflandırıcı ile etkileşime girer</li> <li>• Özellikler arasındaki bağımlılığı dikkate alır</li> <li>• Filtre yönteminden daha yüksek performans doğruluğu gösterir</li> </ul>	<ul style="list-style-type: none"> <li>• Aşırı uymaya daha yatkındır</li> <li>• Belirli sınıflandırıcıya bağlıdır</li> <li>• Karmaşık hesaplamalar gerektirir</li> </ul>
Filtre	<ul style="list-style-type: none"> <li>• Sarmal yönteminden daha hızlıdır</li> <li>• Ölçeklenebilir</li> <li>• Sınıflandırıcıdan bağımsızdır</li> <li>• Sarmal yönteminden daha iyi hesaplama performansı vardır</li> <li>• Daha iyi genelleştirilebilir özelliğe sahiptir</li> </ul>	<ul style="list-style-type: none"> <li>• Sınıflandırıcılar arasındaki etkileşimi yok sayar</li> <li>• Özellikler arasındaki bağımlılığı yok sayar</li> </ul>
Gömülü	<ul style="list-style-type: none"> <li>• Sınıflandırıcı ile etkileşime girer</li> <li>• Sarmal yönteminden daha iyi hesaplama performansı vardır</li> <li>• Filtre yönteminden daha yüksek performans doğruluğu gösterir</li> <li>• Sarmal yönteminden daha az uymaya eğilimli</li> <li>• Özellikler arasındaki bağımlılığı dikkate alır</li> </ul>	<ul style="list-style-type: none"> <li>• Belirli sınıflandırıcıya bağlıdır</li> </ul>
Hibrit	<ul style="list-style-type: none"> <li>• Filtre yönteminden daha yüksek performans doğruluğu gösterir</li> <li>• Sarmal yönteminden daha iyi hesaplama performansı vardır</li> </ul>	<ul style="list-style-type: none"> <li>• Belirli sınıflandırıcıya bağlıdır</li> </ul>

	<ul style="list-style-type: none"> <li>• Sarmal yönteminden daha az uymaya eğilimli</li> </ul>	
Topluluk	<ul style="list-style-type: none"> <li>• Aşırı uymaya daha az eğilimli</li> <li>• Yüksek boyutlu verilerde daha esnek ve sağlam performans gösterir</li> </ul>	<ul style="list-style-type: none"> <li>• Sınıflandırıcılardan oluşan bir topluluğu anlamak zordur</li> </ul>

### 3.2. Sınıflandırma

Sınıflandırma, örnekleri içeren bir eğitim veri setine dayanarak, yeni bir örneğin hangi sınıfa veya alt popülasyonlara ait olduğunu belirleme problemidir (Tang vd, 2014). Ayrıca sınıflandırma, kayıt popülasyonunu genel olarak sınıflandırabilen bir model geliştirmek için bir dizi önceden sınıflandırılmış örnek kullanan, en yaygın uygulanan veri madenciliği yöntemidir (Provost ve Fawcett, 2001). Öğrenmede eğitim verileri sınıflandırma algoritması ile analiz edilir. Sınıflandırmada test verileri, sınıflandırma kurallarının doğruluğunu tahmin etmek için kullanılır. Doğruluk kabul edilebilir ise, kurallar yeni veri gruplarına uygulanabilir. Sınıflandırıcı eğitim algoritması, uygun ayırmacılık için gerekli olan parametre setini belirlemek için daha önceden sınıflandırılmış örnekleri kullanır. Algoritma daha sonra bu parametreleri sınıflandırıcı adı verilen bir modele kodlar (Goldschmidt, 2006).

Bir sınıflandırma algoritmasının ana amacı, sınıflandırma modeli ile elde edilen tahmin doğruluğunu en üst düzeye çıkarmaktır. Sınıflandırmanın görevi, her bir örneğin bir sınıfa ait olduğu denetimli bir teknik ile belirlemektir. Sınıflandırmada kullanılan çeşitli model teknikleri vardır, bunlardan bazıları : Karar Ağacı, K-En Yakın Komşu, Destek Vektör Makineleri, Naive Bayes Sınıflandırıcılar ve Sinir Ağları (Krishnaiah vd, 2013; Tomar ve Agarwal, 2013). Bu tezde sınıflandırma algoritması olarak sinir ağları kullanılmaktadır.

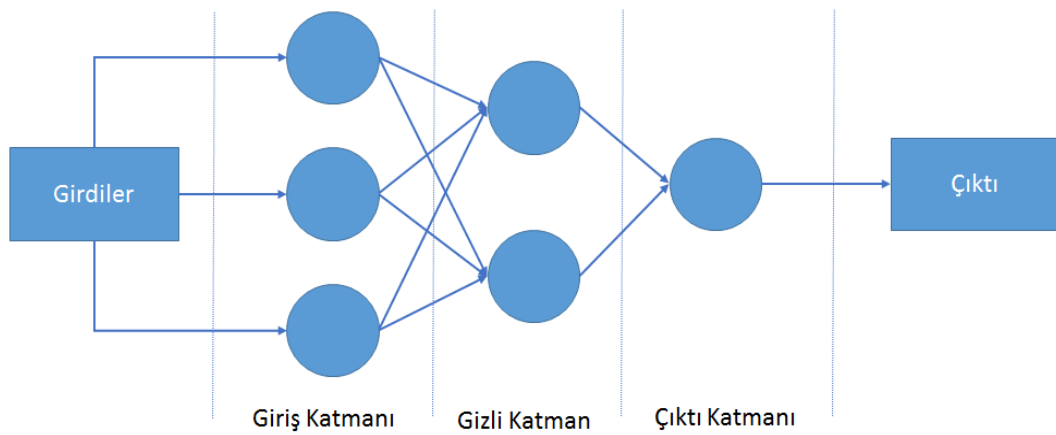
#### 3.2.1. Sinir Ağlarına Genel Bakış

Yapay sinir ağlarının tasarımlarında, nöronlar arasındaki iletişim karmaşıklığı ile tek bir nöronun hesaplama karmaşıklığı arasındaki değiş tokuş düşüncesi, önemli bir konu olmuştur. Dolayısıyla 1943'te McCulloch ve Pitts (McCulloch ve Pitts, 1943) klasik bir yapay sinir modelini önerdiler. Modelinin tanıtılmasından bu yana

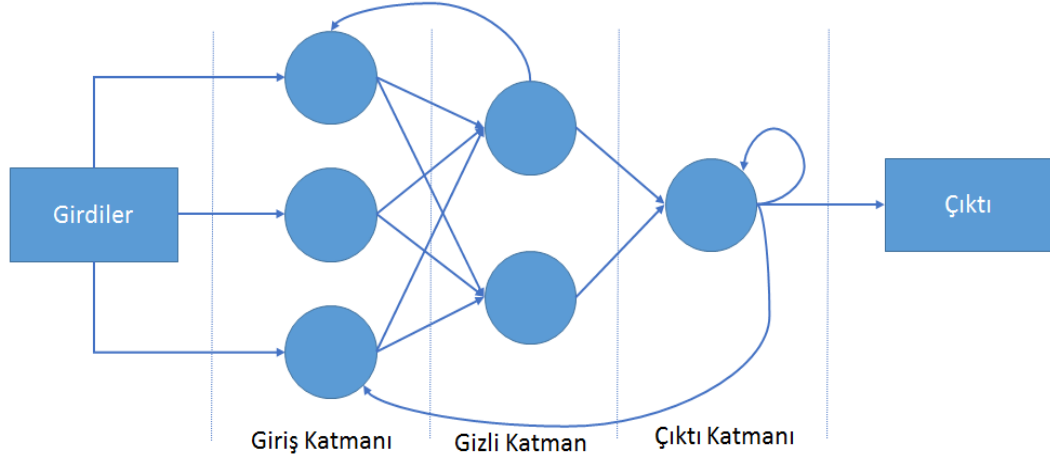
sinir bilimindeki gelişmeler, biyolojik nöronların çok daha detaylı anlaşılmasına yol açtı (Szu ve Rogers, 1992).

Yapay sinir ağı, sinir yapısını oluşturan ağırlıklarla bağlantılı, yüzlerce tek birimden ve yapay nöronlardan oluşan biyolojik olarak esinlenmiş bir hesaplama modelidir. Bilgileri işledikleri için işleme ögeleri olarak da bilinirler. Her işleme ögelerinin ağırlıklı girişleri, transfer işlevi ve bir çıkışı vardır. Dolayısıyla işleme ögeleri, esas olarak girdileri ve çıktıları dengeleyen bir denklemdir. Ayrıca yapay sinir ağları, bağlantı ağırlıkları sistemin belleğini temsil ettikleri için bağlantısal modeller olarak da adlandırılır (Zurada, 1992).

Yapay sinir ağları, giriş katmanı, çıktı katmanı ve gizli katmanlardan olmak üzere üç katmandan oluşur. Gizli katmandaki düğümlerle giriş katmanındaki düğümlerden ve çıktı katmanının düğümleriyle her gizli katman düğümünden bir bağlantı olmalıdır. Giriş katmanı, verileri ağdan alır. Gizli katman, ham bilgileri giriş katmanından alır ve işler. Daha sonrasında elde edilen değer, gizli katmandan gelen bilgileri de işleyerek, çıktıları çıktı katmanına gönderir. Katmanlar arasındaki düğümlerin birbirine bağlanması iki temel sınıfa bölünebilir, yani Şekil 3.12’de görüldüğü gibi ileri beslemeli sinir ağı ve Şekil 3.13’te görüldüğü gibi tekrarlayan sinir ağı. İleri beslemeli bölümde, girdilerden çıktıya bilgi hareketi sadece bir yöndedir. Öte yandan, tekrarlayan bölümde ise, bilgilerin bir kısmı da ters yönde hareket edebilir (Imran ve Alsuhaibani, 2019).



Şekil 3.12. İleri beslemeli sinir ağı



Şekil 3.13. Tekrarlayan sinir ağı

Yapay sinir ağları, yukarıdaki şekillerde görüldüğü gibi giriş katmanı, gizli katman ve çıktı katmanı olarak tanımlanabilir (Imran ve Alsuhaibani, 2019) :

- Giriş Katmanı: veri gruplarının ağı tanımlandığı katmandır. Girdi katmanlarındaki parametreler analizden önce seçilmelidir. Bir giriş katmanındaki nöronların sayısı, giriş verilerinin sayısına eşittir ve her giriş nöronu gizli katmana iletilir.
- Gizli Katman: ağın temel işlevidir. Bu katmanda giriş katmanından alınan veriler doğru şekilde işlenir ve ardından çıkış katmanına iletilir.

Çıktı katmanı: Öğrenme, çıktı katmanında gerçekleşir ve ağdaki son katmandır. Bu katman gizli katmandan aldığı verileri işler ve çıktı oluşturur.

### 3.3. Optimizasyon'a Genel Bakış

Optimizasyon günümüz dünyasında her yeredir ve dolayısıyla çok çeşitli uygulamalarda önemli bir rol oynamaktadır. Mühendislik ve endüstrideki neredeyse tüm uygulamalarda, maliyeti ve enerji tüketimini en aza indirmek veya karı, çıktıyı, performansı ve verimliliği en üst düzeye çıkarmak için, her zaman bir şeyler optimize edilmeye çalışılır. Optimizasyon, kısaca verilen koşullar altında mümkün olan en iyi sonucu elde etme eylemidir. Tüm bu kararların amacı ya çabayı en aza indirmek ya da faydayı en üst düzeye çıkarmaktır. Çaba veya fayda genel olarak belirli tasarım değişkenlerinin bir fonksiyonu olarak ifade edilebilir. Bundan dolayı, bir işlevin maksimum veya minimum değerini veren koşulları bulma eylemine optimizasyon denilir (Hussain vd, 2019; Koziel ve Yang, 2011; Rardin ve Rardin, 1998; Weise vd, 2009; Yang, 2010a, 2013).

Optimizasyon süreci, 'iyi' veya 'kötü' olanı ölçmek ve değiştirmek mümkünse, 'en iyi' yi elde etme sürecine denilir. Uygulamada, araştırmacı 'en fazla' veya 'maksimum' , 'en az' veya 'minimum' ister. Bundan dolayı, "optimum" kelimesi, koşullara bağlı olarak "maksimum" veya "minimum" anlamına gelir; "Optimum", nicel ölçüm anlamına gelen teknik bir terimdir ve günlük kullanım için daha uygun olan "en iyi" kelimedenden daha güçlü bir kelimedir. Benzer şekilde, optimum seviyeye ulaşmak anlamına gelen "optimize" kelimesi, "iyileştirme" den daha güçlü bir kelimedir. Optimizasyon teorisi, matematiğin ve onları bulmak için kullanılan yöntemlerin nicel çalışmasını kapsayan bir dalıdır. Optimizasyon uygulaması ise optima'yı bulmak için kullanılabilir tekniklerin, yöntemlerin, prosedürlerin ve algoritmaların toplanmasıdır (Antoniou ve Lu, 2007).

Optimizasyon yapılabilmesi için, bir dizi mevcut çözümden en iyi çözümü seçerek sınırlı kaynaklarla sonucu en üst düzeye taşınmalıdır. Örnek olarak Yang 2012 yılında (Yang, 2012) büyük bir ormanda (arama alanı) elmas arama örneği ile optimizasyonu detaylandırmıştır. Arama alanına inç-inç bakmak, çok miktarda enerji ve zaman harcayacaktır. Böyle bir senaryoda, sadece elmas bulma olasılığının daha yüksek olduğu potansiyel noktalara odaklanmak için herhangi bir optimizasyon tekniği uygulanabilir. Bu şekilde, sorun akıllıca çözülecektir. Her optimizasyon problemi birkaç karar değişkeni ile (örneğin elmas aramak için hangi arama modunun kullanılacağı ve ormanı ziyaret etmek için hangi sırayla kullanılacağı), belirli objektif işlevlerin (örneğin maksimum elmas aranması) ve bazı kısıtlamalarla (örneğin elmasın daha az zaman ve çabayla aranması) çalışmalıdır (Sörensen vd, 2012). Dolayısıyla optimizasyon yöntemleri, belirli kısıtlamalar altında objektif bir fonksiyonu optimize eden karar değişkenlerinin değerlerini elde etmek için kullanılır (Hussain vd, 2019).

Optimizasyon aslında tüm çözümler arasından mümkün olan en iyi çözümü bulma yöntemidir. Ayrıca optimizasyon, bir çözümün kalitesini temsil eden bir matematiksel değerlendirme veya amaç fonksiyonu olarak çözülecek problemi modelleme görevi olarak kabul edilmektedir ve daha sonra bu çözümü bulmak için tüm olası çözümlerin alanını araştırabileceğini belirtmektedir (Burke ve Kendall, 2005). Bir optimizasyon problemi için standart form genellikle, değişkenlerine yerleştirilen ve kısıtlamalara tabi olan amaç fonksiyonunun en aza indirilmesi olarak ifade edilir (Antoniou ve Lu, 2007; Nocedal ve Wright, 2006; Sengupta vd, 2016; Venter, 2010):

$$\min f(x) : x \in \mathbb{R}^n \quad \text{s. t.} \quad \begin{cases} g_j(x) \geq 0, & j = \{1, 2, \dots, J\} \\ h_k(x) = 0, & k = \{1, 2, \dots, K\} \end{cases} \quad (3.49)$$

$$x_i^{(L)} \leq x_i \leq x_i^{(u)} \quad i = \{1, 2, \dots, n\}$$

burada  $f(x)$  optimize edilen fonksiyon,  $g_j(x)$  J eşitsizlik kısıtlamalarıdır ve  $h_k(x)$  K eşitlik kısıtlamalarını temsil etmektedir(Sengupta vd, 2016).

### 3.3.1. Amaç Fonksiyonu

Herhangi bir optimizasyon problemi, Amaç fonksiyonu ("maliyet" veya "ödeme" fonksiyonu olarak da bilinir) ve eşitlik/eşitsizlik kısıtlamalarından oluşmaktadır. Bu kısıtlamalar sorunun "fiziği" ile tanımlanır ve fiziksel düşüncelerden türetilir. Bununla birlikte, amaç fonksiyonun tanımı herhangi bir optimizasyon probleminin en önemli parçasıdır (Efthimeros vd, 2000). Objektif bir fonksiyon, bir problemin amacının, problemle ilgili olarak verilen bir çözümün kalitesini ölçmek için kullanılan bir matematiksel fonksiyon veya hesaplama modeli olarak kodlanmasıdır. Temel olarak bir algoritma ve gerçek problem arasındaki arayüzü temsil eder ve bir algoritma olmadan yönlendirmede önemli bir rol oynar, ancak onsuz verilen bir arama alanı, üretilen çözümlerin nasıl performans gösterdiğine dair hiçbir ipucu içermez. Çok amaçlı objektif problemlerde problem hedefi veya hedefleri, amacın objektif fonksiyona göre düşük değerli çözümler üretmek olduğu bir minimizasyon hedefi veya daha yüksek değerli çözümlerin olduğu bir maksimizasyon problemi olarak tanımlanabilir (Burke ve Kendall, 2005).

### 3.3.2. Uygulanabilir Çözümler ve Mümkün Olmayan Çözümler

Karar değişkenleri için herhangi bir değer belirtilmişse, çözüm olarak adlandırılır. Uygulanabilir çözümler, problemin arama alanı içindeki problem parametreleri üzerindeki kısıtlamaları, özellikle de zor kısıtlamaları karşılayan çözümlerdir; kısaca tüm kısıtlamaların karşılandığı bir çözüme uygulanabilir çözümler denilir. Tam tersine en az bir kısıtlamanın ihlal edildiği bir çözüme, mümkün olmayan çözümler denilir (Burke ve Kendall, 2005).

### 3.3.3. En İyi Çözüm

En iyi çözüm gibi bir terimin kullanılması, hepsinin farklı değer seviyelerine sahip olduğu düşünülen birden fazla çözüm olması gerektiği anlamına gelmektedir. En iyinin kalite açısından anlamı, genellikle asıl probleme, çözüm yöntemine ve izin verilen toleranslara bağlıdır. Bazı sorun türlerinin, "en iyi" teriminin mutlak bir tanımı olduğu kesin yanıtları vardır. Optimizasyon terimleriyle, bu durumlarda "en iyi", tek bir küresel optimum sorununa işaret eder. Bununla birlikte, diğer sorunların birkaç küresel tercihi olabilir (Haupt ve Ellen Haupt, 2004).

### 3.3.4. Zor Kısıtlamalar ve Yumuşak Kısıtlamalar

Bir soruna getirilen zor kısıtlamalar, bir çözümün uygulanabilir olması için yerine getirilmesi gereken koşullardır. Ayrıca, yumuşak kısıtlamalar, memnun etmek istediğimiz ancak zorunlu olmayan sınırlamalardır; bu nedenle bir veya daha fazla yumuşak kısıtlamayı ihlal eden bir çözüm hala uygulanabilir kabul edilebilir. Bir optimizasyon problemine çözüm ararken yumuşak kısıtlamaları ele almanın yaygın bir yolu, yumuşak bir kısıtlamanın karşılanmaması durumunda çözümlerin cezalandırılmasıdır. Bu nedenle, birçok sorunun nesnel işlevi, yumuşak kısıtlamalar için verilen cezaların toplamı ile temsil edilmektedir. Bu, yumuşak kısıtlamaları ihlal etmeyen çözümlerin daha yüksek kalitede olduğu düşünülecektir (Burke ve Kendall, 2005).

Yöneylem araştırması dilinde, zor kısıtlamalar uygulanabilir çözümler kümesini belirtir ve yumuşak kısıtlamalar mümkün çözümler arasında seçim için optimize edilecek bir işlevi belirtir. Her iki kısıtlama türü de maddelerle temsil edildiğinde, tüm maddeler birleştirilerek oluşturulan formül muhtemelen tatmin edici olmayacaktır. Sıradan bir tatmin edilebilirlik prosedürü kullanarak orijinal soruna bir çözüm bulmak için, tatmin edici bir formül bulunana kadar yumuşak kısıtlamaların farklı alt kümelerini problem temsilinden çıkarmaya çalışmak gerekir. Göreceli önemlerini göz önünde bulundurarak yumuşak kısıtlamalar uzayda böyle bir arama yapmak, tüm sürecin teorik karmaşıklığı sıradan tatmin edilebilirlikle aynı olsa bile pratik anlamda karmaşık ve maliyetli olabilir (Jiang vd, 1995).

Örnek olarak  $m$  öğrenci ve  $k$  yaz kursu olduğu ilinmektedir. Her öğrenci, ders çalışmasının geliştirileceği potansiyel takım arkadaşlarına sahiptir. Kurs kaydının aşağıdaki kısıtlamalara uymasını istenmektedir (Finger vd, 2013):

- Zor Kısıtlamalar: Her ders için, öğrenciler ders çalışmalarını tek başına veya 2 kişilik takımlar halinde geliştirmeye karar vermelidir. Bir öğrenci farklı derslerde farklı takım arkadaşlarına sahip olabilir ve bir derste tek başına çalışabilir ve diğerlerinde bir takım arkadaşına sahip olabilir. Öğrenciler en az bir ve en fazla üç derse kayıt yaptırmalıdır. Ders başına öğrenci sınırı olduğu görülmektedir.
- Yumuşak Kısıtlamalar: Öğrencinin takım arkadaşı olmayan bir kursa kaydı “yumuşak ihlal” olarak görülür.

### 3.3.5. Yerel ve Küresel Optimizasyon

Küresel optimizasyon, olası tüm çözüm arasında belirli bir işlevin optimal değerini bulmayı ifade ederken, yerel optimizasyon komşu aday çözüm kümesinde en uygun değeri bulur (Vasant, 2015). Yerel ve küresel optimizasyon arasındaki fark, optimizasyon görevinin kapsamındadır. Yerel optimizasyon, yerel bir uygulanabilir bölge içinde, yani sorunlu alandaki uygulanabilir bir bölgenin en küçük objektif değer olan yerel bir optimum keşfetmek için optimize etmeye çalışır (Nocedal ve Wright, 2006). Bir komşuluk fonksiyonu  $N$ , bir  $s \in S$  çözümünün çevresini bir  $N : S \rightarrow 2^S$  eşleme olarak tanımlar. Her bir  $s$ 'ye  $N(s) \subset S$ , bir dizi çözümleri  $s' \in N(s)$ ,  $s$ 'nin komşusu olarak adlandırılır. Bu nedenle, bir  $S$  alanı ve  $N(s) \subset S$  alanının uygulanabilir bir bölgesi göz önüne alındığında, yerel optimizasyon  $f(x^*) \leq f(x) \forall x \in N$  olacak şekilde yerel bir optimum  $x^* \in N$  bulmaya çalışır. Öte yandan küresel optimizasyon, bir fonksiyonun tüm uygulanabilir alanı üzerindeki en küçük objektif değeri veya değerleri keşfetmekle ilgilidir ve genellikle 'en iyi çözüm' olarak da adlandırılan küresel bir optimumdur. Daha önce olduğu gibi, bir alan  $S$  verildiğinde, küresel optimizasyon  $f(s^*) \leq f(s) \forall s \in S$  olacak şekilde bir küresel optimum  $s^* \in S$  bulmaya çalışır. Bu nedenle optimizasyonun ana sonucu, bu tür birçok çözümün mevcut olabileceği küresel bir optimumun keşfedilmesidir. Birçok küresel optima mevcut olduğunda (her zaman bilinmemektedir), sonuç olarak alternatif çözüm seçenekleri üretmek için tüm küresel optima'yı keşfetmeye çalışmak olarak tanımlanabilir (Talbi, 2009).

### 3.3.6. Kesikli ve Sürekli Optimizasyon

Kesikli optimizasyon, girdi parametreleri tarafından kabul edilen sınırlı sayıda durum ile karakterize edilen problemlerin optimizasyonunu içermektedir. Ayrıca, sürekli problemler tipik olarak çözümlerini, parametreleri süreklilikleri olmayan, gerçek değerler kümesinin üyeleri olan, sayılamayan sonsuz bir çözüm kümesinden alanlardır. Bununla birlikte, dijital bilgisayar teknolojilerinin 'gerçek' sürekli bir çözüm için gereken hassasiyeti sağlayamadığı göz önüne alındığında, olası çözümler kümesi hala zorlukla büyük de olsa sayılamayacak kadar sınırsız olmaktan çok daha az olacaktır (Nocedal ve Wright, 2006).

Her iki problem türünün pratik optimizasyonuna benzer yöntemler kullanılarak (en azından kesin olmayan sezgisel yöntemler açısından) yaklaşılabilir de, etkili bir şekilde optimize etmek için her bir problem türü için dikkate alınması gereken noktalar vardır. Örnek olarak, sürekli optimizasyon problemlerinin çözülmesinin kesikli problemlerden daha kolay olduğu düşünülmektedir, nedeni ise tipik pürüzsüzlükleri objektif ve kısıtlama bilgilerini kullanarak fonksiyon  $x$  noktasına yakın tüm noktalarda işlev davranışını çözmeyi sağlar. Öte yandan, objektif işlev davranışı, bir miktar mesafe ölçüsüyle yakın kabul edilen noktalar arasında büyük ölçüde değişebilir (Nocedal ve Wright, 2006).

### 3.3.7. Problem Zorluğu

Gerçek dünya optimizasyonlarının etkili bir şekilde çözülmesinin çeşitli nedenleri aşağıdaki kısımda belirtilmiştir (Michalewicz ve Fogel, 2013):

- Arama alanı, kapsamlı bir aramanın kullanımına izin vermek için çok geniştir.
- Sorun o kadar karmaşıktır ki, herhangi bir cevabı herhangi bir şekilde belirtmek için problem etki alanının bu kadar basitleştirilmiş modellerinin kullanılmasını gerektirir ve dolayısıyla bulunan herhangi bir çözüm aslında işe yaramaz olur.
- Amaç fonksiyonu gürültülüdür veya zamanla değişir, bu nedenle tek bir çözüm yerine tüm çözüm serilerinin gerekli olmasını gerektirir.
- Arama alanı o kadar kısıtlıdır ki, soruna uygulanabilir bir çözüm bulmak bile zordur.

Bir algoritma ile belirli bir problemi çözenin zorluk derecesi, hesaplama karmaşıklığıyla, yani bunu yapmak için gereken zaman ve hafıza gibi kaynak miktarı ile yakından ilgilidir. Hesaplama karmaşıklığı, algoritmanın uygulanması için gereken giriş elemanlarının sayısına bağlıdır. Bu bağımlılık genel olarak Bachmann (Bachmann, 1894) tarafından yaklaşık sınırlar şeklinde ifade edilerek ortaya atılan ve Landau (Landau, 2000) tarafından popüler hale gelmiştir (Weise vd, 2009).

Karmaşıklık, bir optimizasyon probleminin çözülmesinin ne kadar zor olduğu üzerine yapılan bir çalışmadır (Burke ve Kendall, 2005). Burada problemler, optimizasyon algoritmalarının özelliklerine göre sınıflandırılır; bilinen bir hızlı çözücü yoksa “zor”, aksi taktirde “kolay” olarak sınıflandırılır (Eiben ve Smith, 2015).

Hesaplama karmaşıklığının üç önemli kavramı vardır (Eiben ve Smith, 2015):

1. Belirli bir problemin büyüklüğü, problemin boyutuna(problem parametrelerinin sayısına) dayanmaktadır. Problemin herhangi bir parametresini ayarlayabilen bir dizi değerin boyutuna bağlıdır.
2. Problemlerden daha çok algoritmaların çalışma süresi ile ilgilidir. Çalışma süresi, belirli bir probleme karşı çalışırken sonlanmadan önce bir algoritmanın gerektirdiği temel işlemlerin sayısıdır. Genel olarak, hesaplama karmaşıklığının sezgisi, daha büyük problemlerin çözülmesi için daha fazla hesaplama süresi gerektireceğidir, ancak bu her zaman doğru olmaya bilir. Bir problemin zorluğunun en iyi bilinen tanımı, çözülen problemin boyutu ile onu çözmek için kullanılacak algoritmanın en kötü durumdaki çalışma süresi arasındaki ilişkidir.
3. Tersine çevrilebilir veya geri döndürülemeyecek uygun bir haritalama yoluyla bir problemi diğerine dönüştürmenin mümkün olduğu kavramı ile ilgilidir. Buna problem azaltma denilir.

Optimizasyon problemleri yapılarına göre değişmektedir. Tek veya çok amaçlı, kısıtlı veya kısıtsız, ya da kombinasyonel optimizasyon problemleri olabilirler. Aşağıdaki kısımda bunlar özet olarak anlatılmaktadır :

- Tek veya çok amaçlı : Maksimize edilmesi veya en aza indirilmesi gereken belirli bir amaca dayanan optimizasyon problemleri, tek amaçlı optimizasyon olduğu varsayılır. Bu tür sorunların aynı zamanda, tek bir ana amaç oluşturmak için bir araya getirilmiş birkaç amacı olabilir. Tek amaçlı optimizasyon

yönteminin amacı, birkaç hedefi en iyi şekilde bir araya getirerek tek bir çözüm ya da en iyi çözümü bulmaktır. Öte yandan, birçok karmaşık endüstri ve bilimsel optimizasyon problemi, doğası gereği çok amaçlıdır. Bu problemler, aynı anda yerine getirilmesi gereken birçok çelişkili kriteri veya amacı içerir. Çok amaçlı optimizasyon problemi, belirli kısıtlamalara dayalı olarak yerine getirilmesi gereken birden fazla çelişen amaç içerir (Tan vd, 2013).

- Kısıtlı veya kısıtsız : Çoğu gerçek hayattaki optimizasyon problemleri, optimize edilmiş bir çözüm bulurken, ihlal edilemeyen belirli kısıtlamalar sahiptir. Bundan dolayı, bu sınıftaki problemler, belirli kısıtlamalara dayalı olarak minimize veya maksimize edilecek olan  $f(x)$  amaç fonksiyonunu içerir. Oysa kısıtsız optimizasyon problemleri herhangi bir kısıtlama veya sınırlama içermez ve gerçek değişkenlere bağlıdır. Bu tür problemlerin  $f$  değerini minimize veya maksimize eden nesnel işlevi vardır (Bae vd, 2012).
- Kombinasyonel optimizasyon : Amaç işlevi maksimize etmek veya en aza indirmek için bazı değişkenlerin optimal değerlerinin bulunacağı çok amaçlı optimizasyonun aksine, birçok gerçek dünya probleminin, bir dizi nesneyi, amaç işlevi maksimize ya da minimize edilir. Kombinasyonel optimizasyon, farklı matematikten türetilen optimizasyondur; burada bir dizi  $S$  elemanından alınan  $k$  büyüklüğündeki farklı elemanların sırasız koleksiyonunun farklı kombinasyonlarını denenir (Yu ve Gen, 2010).

Genel olarak, gerçek hayat problemlerinin karmaşıklığı, geleneksel matematiksel programlama yöntemlerinin çözmesi ve optimize edilmesi zorlaşacak şekilde artmaktadır. Gerçek hayattaki optimizasyon problemlerinin çoğu doğrusal olmayan, karmaşık, çok modludur ve optimum veya hatta optimal çözümler elde etme sürecinin, tek bir kolay ve doğrusal için bile çok zor bir görev olduğu uyumsuz amaç fonksiyonlarına sahiptir. Bazen, optimal çözümler hiç mevcut olmayabilir, genel olarak, gerçek hayattaki problemler için optimal bir çözüm bulmanın garantisi yoktur (Almufti vd, 2018; Almufti, 2015). Ancak optimizasyon problemlerinde tatmin edici çözümler bulmak için metasezgisel algoritmaları kullanılabilir. Metasezgisel, yaklaşık olarak çok çeşitli zorlu optimizasyon problemlerini, her probleme derinlemesine uyum sağlamak zorunda kalmadan çözmek için tasarlanmış bir algoritmadır. Ayrıca, adında yer alan yunanca "meta" daha yüksek düzey buluşsal yöntemler olduğunu belirtmek için kullanılır. Metasezgisel yöntemler, sanayide ve hizmetlerde, finanstan üretim

yönetimi ve mühendisliğe kadar değişen alanlarda karmaşık sorunları çözmek için yaygın olarak kullanılmaktadırlar (BoussaiD vd, 2013). Bir sonraki bölümde metasezgisel algoritmaları detaylı bir şekilde anlatılmıştır.

### **3.4. Metasezgisellere Genel Bakış**

Tarih boyunca, özellikle insanlık tarihinin ilk dönemlerinde, problem çözmeye yönelik temel yaklaşım her zaman sezgisel veya metasezgisel olmuştur. Pek çok önemli keşif genel olarak kazara keşfedilmiştir ki bu sezgisel anlamına gelmektedir. Aslında, günlük öğrenme deneyimimiz en azından çocukken, ağırlıklı olarak sezgiseldir (Yang, 2010a). Metasezgisel popülaritesi ve başarısı birçok nedene bağlanabilir ve temel nedenlerden biri, bu algoritmaların biyolojik sistemler, fiziksel ve kimyasal süreçler de dahil olmak üzere doğadaki en başarılı süreçleri taklit ederek geliştirilmiş olmasıdır. Çoğu algoritma için temel bileşenlerini biliyoruz, ancak bu bileşenlerin verimlilik elde etmek için tam olarak nasıl etkileşime girdiği hala büyük ölçüde bir gizem olarak kalmaktadır ve bu da daha aktif çalışmalara neden olmaktadır (Yang, 2011a; Yang, 2011b).

1980'lerin ortalarında, metasezgiseller başlangıçta kombinasyonel optimizasyon problemlerini çözmek için yeni teknikler olarak tasarlandılar. Bununla birlikte, 1990'lardan bu tarafa yeni arama kavramları ve teknikleri üzerine yapılan araştırmalar ilerledikçe, metasezgiseller genel problem çözme için güçlü araçlar olarak ortaya çıktılar. Son yıllarda, metasezgiseller çok sayıda uygulamayı çözmek için pratikte yararlı oldukları görüldü. Bugünden itibaren birçok gerçek hayat sorunu, genellikle yüksek kaliteli çözümleri hızlı bir şekilde bulmayı gerektirir. Sonuç olarak, bilim adamları ve uygulayıcılar, bu problemlerin geleneksel kesin yöntemler kullanılarak çözülmesi durumunda, günlerce veya haftalarca zaman kaybı yaşayacaklarını bildikleri için yeni yöntemler aramaya başladılar (Nesmachnow, 2014). Metasezgisel yöntemler, sezgisel veya geleneksel arama teknikleri dahil olmak üzere diğer arama yöntemleri arasındaki işbirliğini koordine etmek için üst düzey problem çözme stratejileri olarak tasarlandı. Bu yöntemler, arama, optimizasyon ve makine öğrenimi problemleri için yaklaşık çözümleri verimli ve doğru bir şekilde bulabilen bir dizi teknik tasarlamak için algoritmik çerçeveler tanımlayan genel stratejilerdir (Glover, 1986).

Metasezgisel algoritmelerde meta, "ötesinde" veya "daha yüksek seviye" anlamına gelir. Genel olarak bu algoritmalar basit buluşsal yöntemlerden daha iyi performans gösterirler. Tüm metasezgisel algoritmalar, bazı yerel arama ve küresel keşif ödünleşimlerini kullanır. Çözümlerin çeşitliliği genellikle rastgele seçim yoluyla gerçekleştirilir. Bazı araştırmacılar, buluşsal yöntemler ve metasezgiselleri birbirinin yerine kullanırlar (Gandomi vd, 2013). Metasezgiseller, karmaşık bir probleme makul ölçüde, pratik bir zamanda ve deneme yanılma yoluyla kabul edilebilir çözümler üretmenin etkili bir yolu olabilir. Bu yöntemlerin amacı, kabul edilebilir bir zaman ölçeğinde iyi uygulanabilir bir çözüm bulmaktır (Yang, 2010a, 2010b). Ayrıca metasezgisel terimi, tamamen farklı iki şey için kullanılmaktadır. Birincisi, üst düzey bir çerçeve, birbiriyle harmanlanan ve optimizasyon algoritmalarının geliştirilmesine ilişkin bir bakış açısı sunan bir dizi kavram ve stratejidir. İkincisi ise, belirli bir optimizasyon problemine bir çözüm bulmak için tasarlanmış bir çerçeveye veya farklı çerçevelerden kavramların bir kombinasyonuna dayalı bir algoritmanın özel bir uygulamasını ifade etmektedir (Sorensen vd, 2017).

Metasezgisel algoritmaların birçoğu aşağıdaki özellikleri taşımaktadırlar (Almufti, 2019; BoussaïD vd, 2013):

1. Doğadan esinlenirler(fizik, biyoloji veya etolojiden bazı ilkelere dayanır).
2. Rastgele bileşenlerden yararlanırlar(rastgele değişkenleri içerir).
3. Amaç fonksiyonun gradyanını veya Hessian matrisini kullanmazlar.
4. Eldeki probleme uydurulması gereken çeşitli parametreleri vardır.
5. Metasezgisel, arama sürecine rehberlik eden stratejilerdir.
6. Amaç, optimuma yakın çözümler bulmak için arama alanını verimli bir şekilde keşfetmektir.
7. Metasezgisel algoritmaları oluşturan teknikler, basit yerel arama prosedürlerinden karmaşık öğrenme süreçlerine kadar uzanır.
8. Metasezgisel algoritmalar yaklaşıktır ve genellikle kesin değildir.
9. Metasezgiseller probleme özgü değildir.

Metasezgisel, problem çözmeye daha genel olan veya daha yüksek düzeyli bir buluşsal yöntemdir. Öte yandan, bir hesaplama problemi kategorisinin çözümünde genel sezgisel kuralları uygulayan uyarlanabilir bir hesaplama değildir. Yukarıdaki tanımlara dayanarak, bir metasezgiselin geliştirilmiş matematiksel formülasyonu aşağıdaki gibi tanımlanabilir (Wang vd, 2011b):

$$\text{Metasezgisel} = (O, A, R^c, R^i, R^o) \quad (5.1)$$

Burada  $O$ , bir dizi metasezgisel yöntemidir (örneğin, uyarlanabilir, otomotiv, deneme yanılma, bilişsel);  $A$ , bir dizi genel algoritmadır (örneğin, genetik algoritma, parçacık sürüsü optimizasyonu);  $R^c = O \times A$ , bir iç ilişkiler kümesidir;  $R^i \subseteq A' \times A, A' \wedge A \subseteq$ , bir girdi ilişkiler kümesidir; Burada  $C'$  bir dizi dış kavramdır ve  $c$  kavram ortamıdır. Koaylık sağlamak için,  $R^i = A' \times A$  basitçe  $R^i = C' \times c$  olarak gösterilebilir.

### 3.4.1. Keşif ve Sömürü

Her arama algoritmasının bir arama alanının keşfini ve kullanımını ele alması gerekmektedir. Keşif, bir arama alanının tamamen yeni bölgelerini ziyaret etme sürecine denilir, sömürü ise, daha önce ziyaret edilen noktaların mahallesindeki bir arama alanının bu bölgelerini ziyaret etme sürecidir. Başarılı olabilmek için, bir arama algoritmasının keşif ve sömürü arasında iyi bir oran oluşturması gerekir (Črepinšek vd, 2013). Başka bir açıdan, keşif ve sömürü, sırasıyla küresel arama ve yerel aramayı açıklamaktadır (Auger vd, 2005; Li vd, 2008a). Hiç şüphesiz, bu açıklama özelliklerinin bir yönünü yansıtır, ancak keşif ve sömürü yalnızca arama kapsamına bağlı olarak ayırt etmek güvenilir değildir, çünkü yerel ölçeğini savunmak için kesin bir eşik yoktur. Kesin ama anlamsız bir önlem, araştırmayı tüm arama alanı içinde tutmak için kesin bir şekilde kısıtlamaktır ve daraltılmış bir alan içindeki herhangi bir arama, sömürü olarak kabul edilecektir. Makine öğrenimi algoritmalarında, keşif ve sömürü, sırasıyla bilginin edinilmesi ve kullanılması ile ilgilidir (Yen vd, 2002).

Keşif ve sömürü, sırasıyla çeşitlendirme ve yoğunlaştırma olarak da adlandırılmaktadır, herhangi bir optimizasyon yönteminin iki ortak ve temel özelliğidir. Öte yandan, her bir metasezgisel tarafından benimsenen arama felsefesine büyük ölçüde bağlıdır. Bu iki özellik, bir optimizasyon problemini başarıyla çözmenin temeli olarak kabul edilmektedir (Črepinšek vd, 2013). Genel olarak, çeşitlendirme, arama alanının birçok ve farklı bölgesini ziyaret etme yeteneğini ifade eder ve yoğunlaştırma, bu bölgeler içinde yüksek kaliteli çözümler elde etme yeteneğini ifade eder (Khajehzadeh vd, 2011; Lozano ve García-Martínez, 2010).

### 3.4.2. Yerel ve Küresel Arama

Yerel metasezgisel arama, mevcut çözüm adı verilen tek bir çözümde yinelemeli değişiklikler yaparak iyi çözümler bulmaktadır. Bu değişikliklere hareket adı verilir ve tipik olarak küçüktür, dolayısıyla bu metasezgisel sınıfın adıdır. Belirli bir çözüme tek bir hareket uygulayarak elde edilebilecek çözüm kümesine, o çözümün mahallesi denilir. Çözümün temsil edilme şekline bağlı olarak farklı hareket türleri tanımlanabilir. Her hareket türü bir mahalle yapısına yol açar. Çevresindeki herhangi bir çözümden daha iyi olan bir çözüme yerel optimum denilir. Ancak küresel bir optimumun, optimizasyon problemine mümkün olan en iyi çözümü bulur. Mevcut çözüm yerel bir optimum olduğunda, metasezgisel bir uzman bu yerel optimumdan kaçmak için bir strateji kullanacaktır. Meta sezgiselliği karakterize eden bu stratejidir ve genellikle metasezgiselliğin adı bundan türetilir (Trabelsi vd, 2010).

küresel arama yöntemleri doğası gereği daha araştırıcıyken (örneğin, Simüle Tavlama (Khachaturyan vd, 1981), Genetik Algoritmalar (Holland, 1992b) ve Parçacık Sürü Optimizasyonu (Eberhart ve Kennedy, 1995)), Yerel arama optimizasyon algoritmaları genel olarak daha sömürü edici yöntemlerdir (örneğin, Tabu Arama (Glover, 1989), Açgözlü Rastgele Uyarlamalı Arama Prosedürü (Feo ve Resende, 1989) ve Yinelenen Yerel Arama (Stützle, 1998)). Yerel arama algoritmalarının yerel arama kabiliyetini global arama/popülasyon tabanlı metasezgisellerde bir iyileştirme mekanizması olarak birleştiren birçok hibrit yöntem de vardır (Li ve Tian, 2015; Sahli vd, 2014).

### 3.4.3. Tek ve Popülasyon Bazlı

Tek ve popülasyon bazlı metasezgiselle karşı arama sürecinde taşınacak çözümlerin sayısı, metasezgiselliğin tek bir çözüm veya popülasyon tabanlı bir algoritma olup olmadığını belirler. Belirli bir optimizasyon problemi için bir metasezgisel seçmek için, önce yörünge veya popülasyon tabanlı bir algoritma kullanılıp kullanılmayacağına karar verilir. Genel olarak, temel tek çözüme dayalı algoritmalar daha çok sömürü odaklıdır, oysa temel popülasyon tabanlı metasezgiseller doğası gereği daha araştırıcıdır (Boussaid vd, 2013). Yörünge yöntemleri her seferinde bir çözüm kullanır ve tek bir ilk çözümle başlar. Yinelemeler sırasında, bu algoritmalar arama alanında bir yörünge oluşturur. Burada çözümün

mevcut çözümün mahallesine ait olabileceği ya da olmayabileceği dikkate alınır. Popülasyon tabanlı bir algoritma için, başlangıçta birden çok çözümden oluşan bir popülasyon oluşturulur ve daha sonra, her yinelemede, daha iyi arama alanlarına yönelik çözümler bulmak için bir dizi çözüm manipüle edilir. Bu algoritmalar ya birden çok çözümün yeniden kombinasyonunu yapar ya da arama alanının araştırılmasını ve kullanılmasını sağlamak için benimsenen strateji yoluyla her birini değiştirir (Blum ve Roli, 2003).

Tek tabanlı algoritmalarda sadece bir başlangıç noktası ile arama tamamlanır ve çözümlerini iyileştirmek için çevresini bir dizi hareketle keşfedebilir (Roeva vd, 2014). Ancak, popülasyon tabanlı algoritma, paralel bir yaklaşımla birden çok başlangıç noktasıyla arama sürecini tamamlayabilir (Beheshti ve Shamsuddin, 2013). Popülasyona dayalı algoritma, keşif için arama alanını etkili bir şekilde sağlayabileceği avantajına sahiptir. Bu yöntem, küresel arama için uygundur; çünkü küresel keşif ve yerel sömürü yeteneğine sahiptir (Samsuddin vd, 2018).

### **3.5. Metasezgisel Algoritmaları ile Özellik Seçimi Yapmak**

Özellik seçimi karmaşık bir problemdir, dolayısıyla bu problemi çözmek için yapay zeka yöntemleri kullanılmaktadır. Bununla birlikte, araştırmacılar yıllardır bu bilimsel alanda doğruluğu ve verimliliği arttırmanın en iyi yollarını bulmaya çalışıyorlar. Bilim adamlarının, özellik seçme sürecinin iyileştirilmesi için şu anda üzerinde çalışmaya çalıştıkları araştırmanın amacı, öğrenme modeli ile başka bir algoritma kullanmaktır. Öğrenme modeliyle birlikte kullanılacak bu algoritmalarından biri, optimize edici olarak metasezgisel yöntemleridir (Yusta, 2009). Son yıllarda yapılan araştırmalar, bu algoritmaların çok yüksek doğrulukta sonuçlar verme ve öznelik seçme sürecini iyileştirme becerisini göstermektedir. Özellik seçme probleminde kullanılan birçok metasezgisel algoritması vardır. Bu algoritmaların birçoğu bu görevde kullanılsa da, bu soruna çözüm getirecek yeni algoritmalar bulmak için araştırmalar devam etmektedir. Yeni algoritmaları bulma sebebi ise, daha az seçilmiş özellik bulmak veya doğruluğu yükseltmektir (Talbi vd, 2008).

#### **3.5.1. Kodlamak**

Özellik seçme problemlerinin doğası göz önüne alındığında, her bitin bir özelliğe karşılık geldiği bir ikili dizi olarak temsil edilmektedir. Bir bitin değeri 1 ise, karşılık

gelen özelliğin seçildiğini gösterirken, 0 ise, özelliğin seçilen özellik alt kümesinin dışında bırakıldığını gösterir. Tüm algoritmaların ilk popülasyonu rastgele ikili sayılarla başlatılır (Wang vd, 2014).

### **3.5.2. Uygunluk Fonksiyonu**

Bir uygunluk fonksiyonu, belirli bir çözümün grup hedeflerine ulaşmaya ne kadar yakın olduğunu anlamak için kullanılan özel bir tür nesnel işlemdir. Uygunluk fonksiyonunun insan tasarlama bile, en son tasarımı ortaya çıkaran bilgisayar olsa da, uygunluk fonksiyonunu tasarlaması gereken insan tasarımcısıdır. Dolayısıyla, bu uygunluk fonksiyonu kötü bir şekilde tasarlanırsa eğer, algoritma ya uygun olmayan bir çözüm üzerinde birleşecek ya da tamamen birleşmesi zor olacaktır. Uygunluk fonksiyonu hızlı çalışmalı, yalnızca tasarımcının amacı ile yakından ilişkili olmamalıdır. Aynı zamanda, örnek olarak tipik bir genetik algoritmanın problem için uygulanabilir bir sonuç üretmesi için birçok kez yinelenmesi gerekmektedir (Haupt ve Ellen Haupt, 2004).

### **3.5.3. Prosedür**

Meta sezgisellere dayalı özellik seçiminin uygulanmasının prosedürleri aşağıdaki gibi tanımlanabilir (Wang vd, 2014):

1. Popülasyonun boyutu, maksimum nesil ve algoritmaların kontrol parametreleri gibi metasezgisel parametreleri ayarlanır.
2. Başlangıç popülasyonunu rastgele belirlenir.
3. Her çözümün uygunluğunu önceden tanımlanmış uygunluk fonksiyonuna göre hesaplanır ve en iyi çözümü ve uygunluğunu kaydedilir.
4. Algoritmaların güncelleme operatörlerini kullanarak yeni adaylar oluşturulur.
5. Yeni çözümlerin uygunluk değeri hesaplanır ve en iyi çözümü ve uygunluğu güncellenir.
6. Yinelemeli aramayı sonlandırın ve herhangi bir sonlandırma kriteri karşılanırsa en uygun çözümü çıkartılır, aksi takdirde Adım 4'e geri dönülür.

### **3.6. Tezde Kullanılan Metasezgisel Algoritmalara Genel Bakış**

Bu tezde altı tane farklı metasezgisel algoritması (Genetik Algoritması, Parçacık Sürü Algoritması, Benzetimli Tavlama Algoritması, Diferansiyel Evrim Algoritması,

Karınca Algoritması ve NSGA II) kullanılmıştır. Bu kısımda kullanılan algoritmalarla alakalı genel bilgiler verilmektedir.

### **3.6.1. Genetik Algoritmasına Genel Bakış**

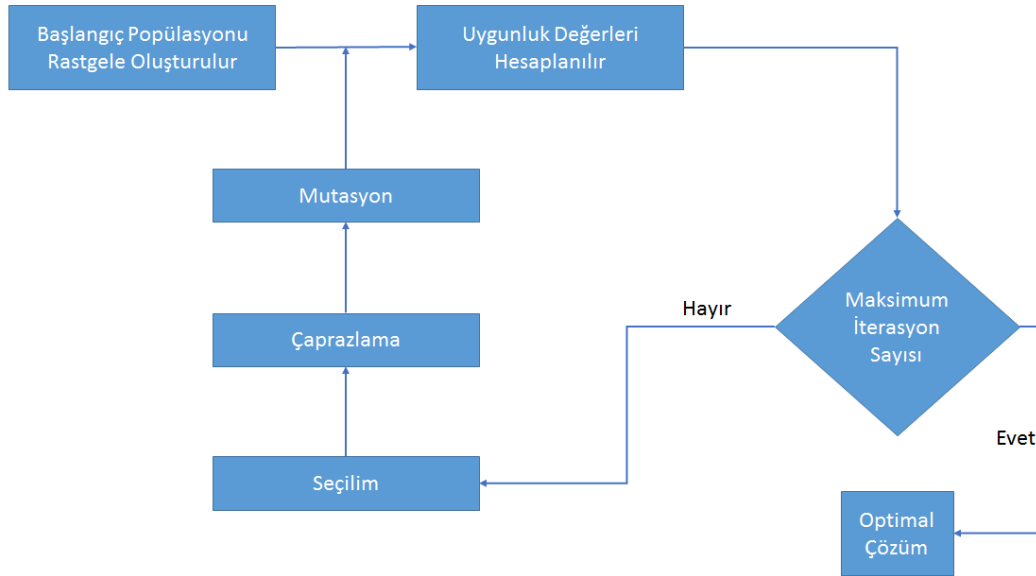
Genetik Algoritmalar, arama ve optimizasyon problemlerini çözmek için kullanılabilen yöntemlerdir. Bu algoritma biyolojik organizmaların genetik süreçlerine dayanır. Birçok nesil boyunca, doğal popülasyonlar, Charles Darwin'in Türlerin Kökeni'ndeki kitabında açıkça belirttiği, doğal seçim ve en iyinin hayatta kalması ilkelerine göre evrimleşir. Bu süreci taklit ederek, eğer genetik algoritmaları uygun şekilde kodlanmışlarsa gerçek dünya sorunlarına çözüm bulabilirler (Akbari ve Ziarati, 2011; Babatunde vd, 2014a, 2014b; Beasley vd, 1993; Goldberg, 1989a; Holland, 1992a; Mitchell, 1998; Tian vd, 2012).

Günümüzde bu yöntem pek çok farklı araştırma alanında kullanılmaktadır ancak genetik bilimlerden kaynaklanmaktadır. Bu nedenle, optimizasyon problemlerini tanımlamak için kullanılan çoğu terim, doğrudan biyolojik terimlerden miras alınır (Rejer ve Lorenz, 2013). Bugünlerde neredeyse evrensel olarak kısaltılmış olan genetik algoritma terimi, ilk olarak 1975 tarihli Doğal ve Yapay Sistemlerde Uyarılma adlı kitabı, şimdilerde orijinal genetik algoritmalarından daha çok geniş bir alana yayılan ve birçok araştırma ve uygulama alanı yaratmada etkili olan John Holland (Holland, 1992a) tarafından kullanıldı. Hollanda'nın konunun geliştirilmesindeki etkisi çok önemliydi, ancak farklı geçmişlere sahip birkaç başka bilim insanı da benzer fikirlerin geliştirilmesine dahil oldu. 1960'larda Almanya'da Ingo Rechenberg (Rechenberg, 1973) ve Hans-Paul Schwefel (Schwefel, 1977) evrim stratejisi fikrini geliştirirken, 1960'larda da Bremermann, Fogel ve ABD'deki diğerleri bu alanda kendi fikirlerini hayata geçirdiler (Reeves, 2010). Son yıllarda ise gelişmeleri kapsamak için evrimsel hesaplama (Kinnear vd, 1994) veya evrimsel algoritmalar terimi kullanılmaktadır (Reeves, 2010).

Genetik algoritmalar, evrimsel hesaplama adı verilen çalışma alanının bir kolunu temsil eder, nedeni ise, biyolojik üreme süreçlerini ve doğal seçilimi taklit ederek en uygun çözümleri çözerler. Evrimde olduğu gibi, bir genetik algoritmanın süreçlerinin çoğu rastgeledir, ancak bu optimizasyon tekniği, rasgele seçim düzeyini ve kontrol düzeyini belirlenmesine izin verir. Bu algoritmalar, rastgele arama ve kapsamlı arama

algoritmalarından çok daha güçlü ve etkilidir, ancak verilen problem hakkında ekstra bilgi gerektirmez (Carr, 2014).

Bir genetik algoritmalar, belirli bir problem için iyi bir çözüme doğru gelişen, yinelemeli olarak seçim ve değişime uğrayan bir grup bireyle çalışır. Temelde, bir genetik algoritma Şekil 3.14'teki gibi çalışmaktadır. İlk olarak, ilk birey popülasyonu rastgele oluşturulur. Ardından, her bireyin kalitesi bir uygunluk işlevi tarafından değerlendirilir. Her nesilde (yineleme), en iyi bireyler (en yüksek uygunluk değerlerine sahip olanlar) üreme için daha fazla seçilir. Seçilen bireyler, çaprazlama (yeni bir birey oluşturmak için iki bireyin parçalarını birleştiren) ve mutasyon (bir bireyin küçük bir kısmının rastgele oluşturulmuş bir değere göre değiştirildiği) gibi genetik işlemlere tabi tutulur. Üreme süreci, ebeveynlerin yerini alacak olan yavrular üretir ve önceki neslin bireylerinden daha iyi olması beklenen yeni nesil bireyler yaratır. Bu süreç, bir durdurma kriteri (örneğin, sabit sayıda nesil) karşılanana kadar tekrar edilir (da Silva vd, 2018; Mitchell, 1998).



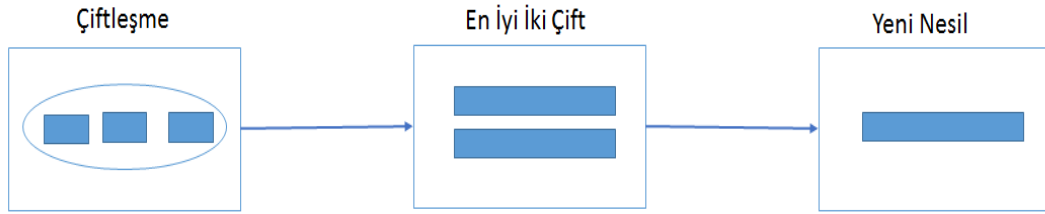
Şekil 3.14. Genetik algoritmanın genel çerçevesi

### 3.6.1.1. Seçim

Bu operatör üreme için popülasyondaki kromozomları seçer. Kromozom ne kadar iyise, çoğalmak için o kadar fazla seçilme olasılığı vardır (Mitchell, 1998). Bundan dolayı seçim, düşük uygunluk yerine yüksek uygunluğa sahip bireyleri tercih ederek algoritmayı çözüme yönlendiren bileşendir. Belirleyici bir işlem olabilir, ancak

çoğu uygulamada rastgele bileşenlere sahiptir (Bodenhofer, 2003). Seçimin temel fikri, uygunluk ile ilgili olması gerektiğidir ve uygulanmasına yönelik orijinal şema genellikle rulet çarkı yöntemi olarak bilinir. Belirli bir dizinin seçim olasılığının uygunluğuyla orantılı olduğu seçim için bir olasılık dağılımı kullanır (Reeves, 2010).

Seçim, geçiş için popülasyondan iki ebeveyn seçme sürecidir. Bu adımda, seçimin nasıl yapılacağına, yani popülasyonda gelecek nesil için yavrular yaratacak bireylerin nasıl seçileceğine ve her birinin kaç yavru yaratacağına karar verilecektir. Seçilimin amacı, çocuklarının daha yüksek uygunluğa sahip olması umuduyla, popülasyondaki sağlıklı bireyleri vurgulamaktır. Kromozomlar, üreme için ebeveyn olmak üzere ilk popülasyondan seçilir. Sorun, bu kromozomların nasıl seçileceğidir. Şekil 3.15, temel seçim sürecini gösterir (Sivanandam ve Deepa, 2007).



Şekil 3.15. Seçimin temel süreci

### 3.6.1.2. Çaprazlama

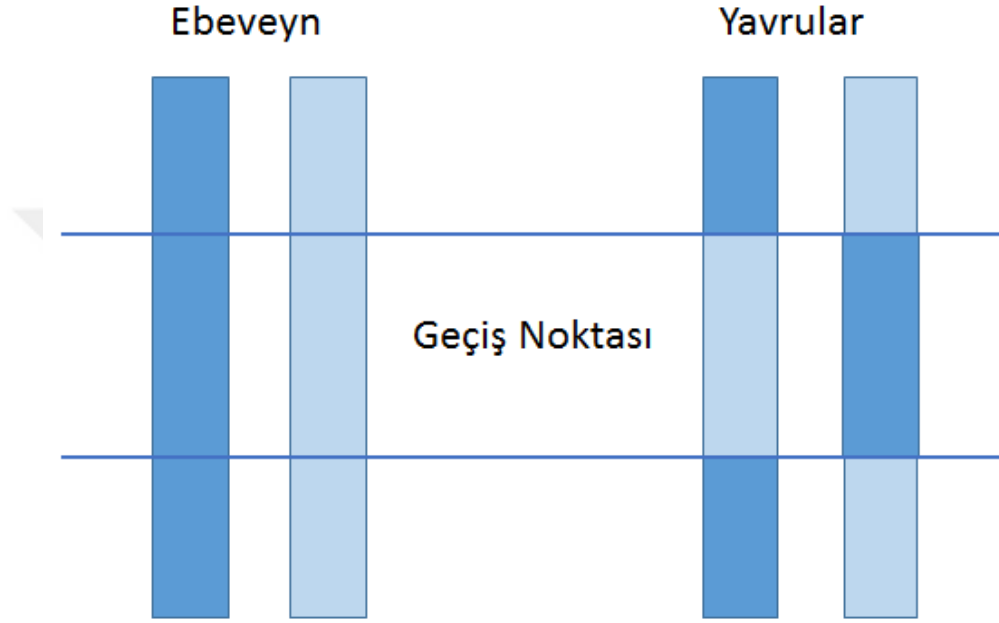
Çaprazlama, Şekil 3.16'deki gibi iki ana çözümü alıp onlardan bir çocuk üretme sürecidir. Seçilim ya da üreme sürecinden sonra nüfus daha iyi bireylerle zenginleştirilir. Üreme, iyi dizelerden klonlar oluşturur, ancak yenilerini yaratmaz. Çaprazlama operatörü, daha iyi bir yavru yaratabilmesi için çiftleşmeyi uygular (Sivanandam ve Deepa, 2007).

Çaprazlama, üç adımda ilerleyen bir operatördür (Sivanandam ve Deepa, 2007):

1. Üreme operatörü, çiftleşme için rastgele bir çift iki ayrı dizi seçer.
2. Dizi uzunluğu boyunca rastgele bir çapraz alan seçilir.
3. Son olarakta, konum değerleri çapraz kısmı izleyen iki dizi arasında değiştirilir.

Bunu yapmanın en basit yolu, rastgele bir çapraz nokta seçmek ve bu noktadan önceki her şeyi ve ilk ebeveyni kopyalamak ve sonra çapraz noktadan sonraki her şeyi

diğer ebeveynden kopyalamaktır (Sivanandam ve Deepa, 2007). Çaprazlama genellikle çiftleşme için seçilen tüm birey çiftlerine uygulanmaz. Çapraz geçişin uygulanma olasılığının tipik olarak 0.6 ile 1.0 arasında olduğu rastgele bir seçim yapılır. Çaprazlama uygulanmazsa, yavrular sadece ebeveynleri kopyalayarak üretilir. Bu, her bireye, çapraz geçişin bozulması olmadan genlerini geçirme şansı verir (Beasley vd, 1993).



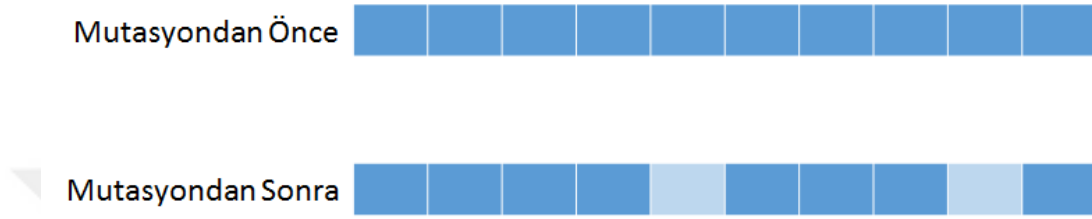
Şekil 3.16. Çaprazlamanın temel süreci

### 3.6.1.3. Mutasyon

Çaprazlamadan sonra dizeler mutasyona tabi tutulur. Mutasyon, algoritmanın yerel minimumda tutulmasını engeller ve kayıp genetik materyalleri geri kazanmanın yanı sıra jenerik bilgileri rastgele dağıtma rolünü de oynar. Mutasyon, geleneksel olarak basit bir arama operatörü gibi düşünülmüştür. Çaprazlamanın daha iyisini bulmak için mevcut çözümü kullanması bekleniyorsa, mutasyonun tüm arama alanının keşfedilmesine yardımcı olması beklenir. Mutasyon, popülasyondaki genetik çeşitliliği sürdürmek için bir arka plan operatörü olarak görülmektedir. Bazı yapı taşlarını rastgele değiştirerek popülasyonda yeni jenerik yapılar sunar. Mutasyon, yerel minimumun tuzağından kaçmaya yardımcı olur ve popülasyondaki çeşitliliği korur. Ayrıca gen havuzunu iyi stoklanmış halde tutar, dolayısıyla ergodikliği sağlar.

Herhangi bir nüfus durumundan herhangi bir çözüm üretme olasılığı sıfır olmayan bir olasılık varsa, bir arama alanının ergodik olduğu söylenir (Sivanandam ve Deepa, 2007).

Mutasyon operasyonunun amacı, yavruların genlerini değiştirmek ve popülasyon çeşitliliğini artırmaktır. Bu süreç, genetik algoritmaların erken yakınsamayı önlemek için yerel veya yetersiz çözümlerin dışına çıkmasını sağlar (Lim vd, 2017; Lynch, 2010).



Şekil 3.17. Mutasyonun temel süreci

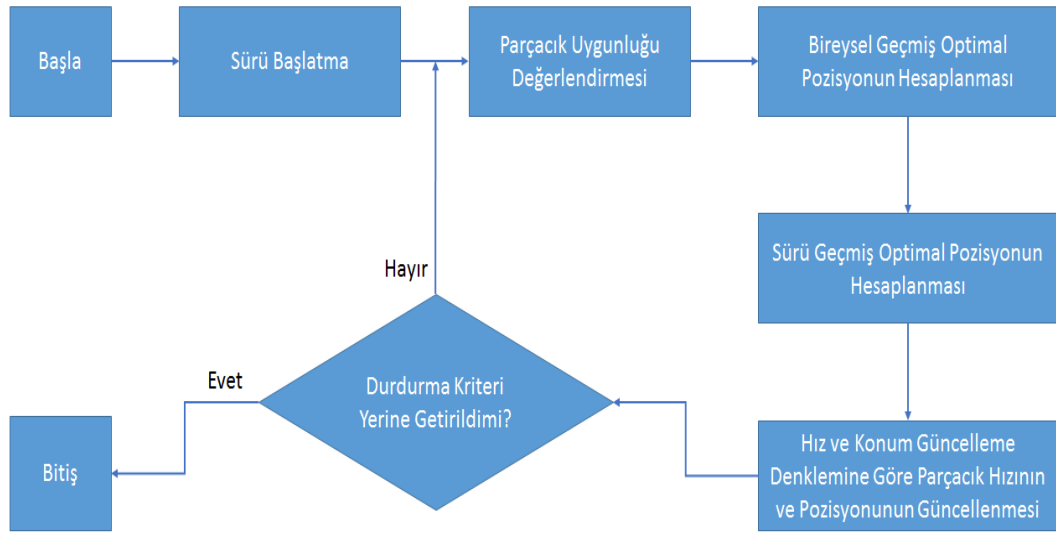
### 3.6.2. Parçacık Sürüsü Optimizasyonu Algoritmasına Genel Bakış

Parçacık sürüsü optimizasyonu, Kennedy ve Eberhart tarafından 1995 yılında önerilen evrimsel bir hesaplama yöntemidir (Eberhart ve Kennedy, 1995; Kennedy ve Eberhart, 1995). Parçacık sürüsü optimizasyonu ortaya çıktığından bu yana, optimizasyonla ilgili pratik sorunları çözmek için birçok varyasyonu geliştirilmiştir (Clerc ve Kennedy, 2002; Coello vd, 2004; Eberhart ve Shi, 2000; Li vd, 2011; Liang vd, 2006). Parçacık sürüsü optimizasyonu algoritması, böcekler, sürüler, kuşlar ve balıklar dahil olmak üzere hayvanın sosyal davranışını simüle eder. Bu sürüler yiyecek bulmak için işbirliğine dayalı bir yola girerler ve sürüdeki her üye, kendi ve diğer üyelerin öğrenme deneyimlerine göre arama modelini değiştirmeye devam eder (Wang vd, 2018).

Parçacık sürüsü optimizasyonu algoritmasının ana tasarım fikri iki araştırmayla daha çok yakından ilişkilidir: Birincisi, evrimsel algoritma gibi, evrimseldir; dolayısıyla, optimize edilmiş hedef fonksiyonunun çözüm alanında geniş bir bölgeyi eşzamanlı olarak aramasını sağlayan bir sürü modu kullanır. Diğeri ise yapay yaşam, yani yapay sistemleri yaşam özellikleriyle inceler (Wang vd, 2018).

Parçacık sürüsü optimizasyonu algoritması, D boyutlu arama alanında optimize edilmiş probleme potansiyel bir çözüm olarak her bireyin bir parçacık olarak

adlandırıldığı bir tür sürü tabanlı arama sürecidir ve sürünün ve kendisinin optimum pozisyonunun yanı sıra hızı da ezberleyebilir. Her nesilde parçacık bilgileri, her boyutun hızını ayarlamak için birleştirilir ve parçacığın yeni konumunu hesaplamak için kullanılır. Parçacıklar, dengeye veya optimal duruma ulaşana kadar veya hesaplama sınırlarının ötesine ulaşana kadar çok boyutlu arama uzayında durumlarını sürekli olarak değiştirirler. Sorunlu alanın farklı boyutları arasındaki benzersiz bağlantı, amaç işlevleri aracılığıyla sağlanır. Parçacık sürüsü optimizasyonu algoritmasının genel akış şeması Şekil 3.18’de gösterilmektedir (Wang vd, 2018).



Şekil 3.18. Parçacık sürüsü optimizasyonu algoritmasının genel akış şeması

Parçacık sürüsü optimizasyonunda her potansiyel çözüme, ağırlığı ve hacmi olmayan kuş veya parçacık denilir. Optimum çözümü bulmak için  $i$ 'inci parçacık,  $D$  boyutlu bir arama uzayında uçar. Bu parçanın konumunu gösteren bir  $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  vektörü vardır, burada  $x_{iD} \in [l_d, u_d]$ ,  $d \in [1, D]$ ,  $l_d$  ve  $u_d$   $d$  boyutunun alt ve üst sınırlarıdır.  $i$ 'inci parçacığın hızı  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  olarak temsil edilir. Herhangi bir parçacığın önceki en iyi konumu, kişisel en iyi konum olarak adlandırılan  $pbest$  olarak kaydedilir ve şimdiye kadar tüm sürü tarafından ziyaret edilen en iyi çözüm ise,  $gbest$  olarak adlandırılır. Sürü, rastgele çözümlerden oluşan bir popülasyonla başlatılır.  $pbest$  ve  $gbest$  'e göre, algoritma aşağıdaki formülleri kullanarak parçacıkların konumlarını ve hızlarını güncelleyerek en iyi çözümü arar (Tran vd, 2014) :

$$v_{id}^{t+1} = w * v_{id}^{t+1} + c_1 * r_1 * (p_{id} - x_{id}^t) + c_2 * r_2 * (p_{gd} - x_{id}^t) \quad (3.50)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (3.51)$$

burada  $t$ , algoritmanın  $t$ 'inci yinelemeye devam ettiği anlamına gelir.  $c_1$  ve  $c_2$  hızlanma sabitleridir.  $r_1$  ve  $r_2$  0 ile 1 arasında eşit olarak dağıtılan rastgele değerlerdir.  $p_{id}$ ,  $p_{best}$ 'i gösterirken,  $p_{gd}$   $g_{best}$ 'i sunar.  $w$ , ilk olarak Shi ve Eberhart (Shi ve Eberhart, 1998) tarafından sunulan atalet ağırlığıdır. Ayrıca  $w$ , parçacık sürüsü optimizasyonunun performansını artırmak için genel arama ile yerel arama arasında bir denge kurabilir. Hız  $v_{id}^t$ , maksimum hız ile sınırlıdır,  $v_{id}^{t+1} \in [-v_{max}, v_{max}]$  ve  $v_{max}$  çözülecek probleme göre önceden tanımlanmıştır (Tran vd, 2014).

Parçacık sürüsü optimizasyonu, kolay, ucuz hesaplamalı olarak uygulanabilir, düşük bellek ve CPU için düşük hız gerektiren bir algoritmadır (Eberhart, 1996). Ayrıca gradyan gibi amaç işlevi hakkında herhangi bir bilgiye ihtiyaç duymaz, sadece değerine ihtiyaç duyar. Parçacık sürüsü optimizasyonunun birçok küresel optimizasyon problemini çözmek için etkili bir yöntem olduğu ve bazı durumlarda diğer evrimsel algoritmanın karşılaştığı zorluklarla karşılaşmadığı kanıtlanmıştır (El-Shorbagy ve Hassanien, 2018).

Parçacık sürüsü optimizasyonu algoritmasında, performansını etkileyebilecek bazı parametreler vardır. Herhangi bir optimizasyon problemi için, bu parametrelerin bazı değerleri ve seçenekleri Parçacık sürüsü optimizasyonu yönteminin verimliliği üzerinde büyük etkiye sahiptir ve diğer parametrelerin etkisi çok azdır veya hiç yoktur. Temel parçacık sürüsü optimizasyonu parametreleri, sürü boyutu veya parçacık sayısı, yineleme sayısı, hız bileşenleri ve ivme katsayılarıdır (Talukder, 2011).

### 3.6.2.1. Parçacık Sayısı

Sürü boyutu, sürüdeki parçacık sayısıdır. Büyük bir sürü, her yinelemede kapsanacak arama alanının daha büyük parçalarını oluşturmaktadır. Çok sayıda parçacık, iyi bir optimizasyon sonucu elde etmek için ihtiyaç duyulan çalışma sayısını azaltabilir. Buna karşılık, büyük miktarda parçacık, yineleme başına hesaplama karmaşıklığını artırır ve daha fazla zaman alır (Talukder, 2011).

### 3.6.2.2. Yineleme Sayısı

İyi bir sonuç elde etmek için yineleme sayısı da sorunun kendisine bağlıdır. Çok az sayıda yineleme, arama sürecini vaktinden önce durdurabilirken, çok büyük yinelemeler, gereksiz ek hesaplama karmaşıklığı ve daha fazla zaman gerekmesi sonucunu doğurur (Engelbrecht, 2007).

### 3.6.2.3. Hız Bileşenleri

Hız bileşenleri, parçacığın hızının güncellenmesi için çok önemlidir ve üç farklı türü vardır (Talukder, 2011) :

1.  $v_{id}^t$  : yakın geçmişte hareket anlamına gelen önceki uçuş yönünün bir belleğini sağlayan atalet bileşeni olarak adlandırılır. Bu bileşen, parçacıkların yönünü büyük ölçüde değiştirmeyi ve mevcut yöne doğru şiddetle önleyen bir hızı temsil eder.
2.  $c_1 * r_1 * (p_{id} - x_{id}^t)$  : parçacıkların performansını geçmiş performanslara göre ölçen bilişsel bileşen olarak adlandırılır. Bu bileşen, parçacık için en iyi olan konumun ayrı bir belleğine benziyor. Bilişsel bileşenin etkisi, bireylerin geçmişte kendilerini en çok tatmin eden pozisyonlara dönme eğilimini temsil eder.
3.  $c_2 * r_2 * (p_{gd} - x_{id}^t)$  : parçacıkların performansını bir grup parçacık veya komşuya göre ölçen sosyal bileşen olarak adlandırılır. Sosyal bileşenin etkisi, her bir parçacığın, çevresinde bulunan en iyi konuma uçmasıdır.

### 3.6.2.4. İvme Katsayıları

İvme ya da Hızlanma katsayıları  $c_1$  ve  $c_2$  ,rastgele değerlerle birlikte  $r_1$  ve  $r_2$  , parçacığın hızının sırasıyla bilişsel ve sosyal bileşenlerinin stokastik etkisini sürdürür. Sabit  $c_1$  , bir parçacığın kendisine ne kadar güvendiğini ifade ederken,  $c_2$  ,bir parçacığın komşularına ne kadar güvendiğini ifade etmektedir (Engelbrecht, 2007).

### 3.6.3. Benzetimli Tavlama Algoritmasına Genel Bakış

Benzetimli tavlama yöntemi ilk olarak 1983 yılında Kirkpatrick ve arkadaşları tarafından önerildi (Kirkpatrick vd, 1983) ve katılarla fiziksel tavlama sürecine benzetmesi nedeniyle benzetimli tavlama olarak adlandırıldı (Henderson vd, 2003).

Bu algoritma, yoğunlaştırılmış maddenin düşük sıcaklıklarda davranışına dayanan bir optimizasyon yöntemidir (Zomaya ve Kazman, 2010).

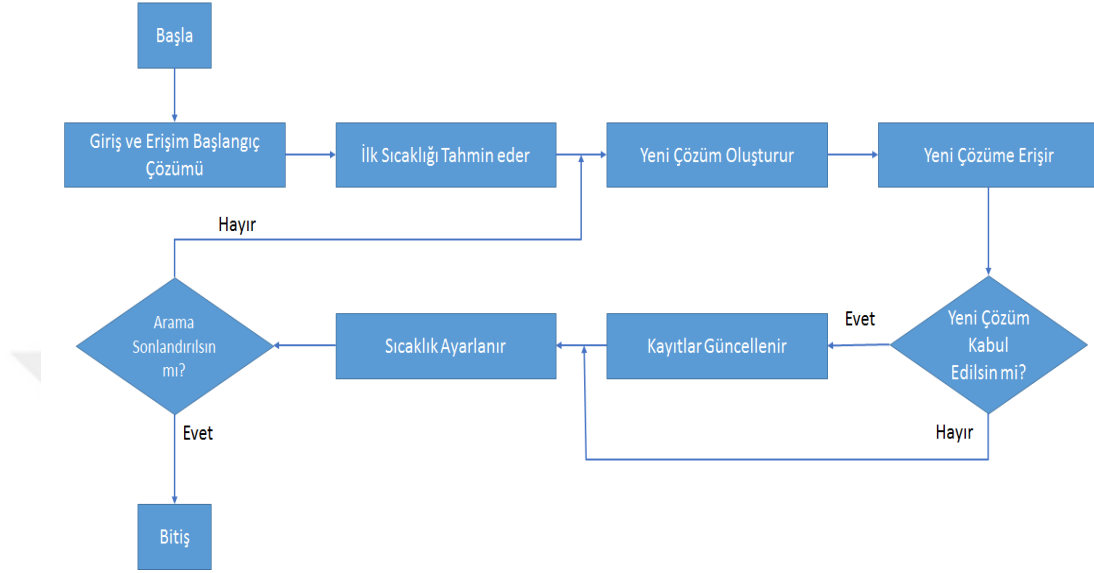
Benzetimli tavlamanın temel algoritmik özelliği, tepe tırmanma hareketlerine yani, amaç işlev değerini kötüleştiren hareketlere izin vererek yerel optimalden kaçmak için bir yol sağlamasıdır. Sıcaklık parametresi sıfıra düşürüldükçe, tırmanma hareketleri daha az sıklıkta gerçekleşir ve algoritmanın davranışını modelleyen homojen olmayan Markov zinciriyle ilişkili çözüm dağılımı, tüm olasılığın küresel olarak en uygun çözümler kümesinde yoğunlaştığı bir biçime yaklaşır (Henderson vd, 2003).

Bu algoritma, katı tavlama prosedürünü taklit etmektedir. Şekil 3.19'da gösterildiği gibi algoritma ilk olarak yüksek bir sıcaklıkta rastgele bir başlangıç noktasının seçilmesiyle başlar ve belirtilen işleve göre rastgele seçilen yeni bir noktaya hareket eder. Bu iki noktanın  $\Delta f$  fonksiyon değerleri arasındaki fark hesaplanır; Yeni bir noktanın amaç işlevi mevcut noktadan üstünse, yeni bir nokta kabul edilir, ancak belirtilen olasılık işlevine göre, algoritma, arama alanını genişletmek ve bir noktada sıkışmaktan kaçınmak için alt noktayı da kabul edebilir. Yeni bir nokta kabul edildiğinde, sıcaklık soğutma hızına, yani tavlama programına göre azaltılır. Sıcaklık düştükçe, daha düşük bir noktaya hareket etme olasılığı da azalır, böylece mükemmel bir kristalin durumunu elde etmeyi amaçlayan bir metal tavlama işleminde, yavaş bir soğutmayı taklit eden bir aramadaki rastgelelik kontrol edilir. Tipik olarak, bu dengeyi taklit etmek için, sıcaklığı düşürmeden önce belirli bir sıcaklıkta bir dizi nokta test edilir. Bir aramanın sonunda, sıcaklık neredeyse sıfır olduğunda, düşük noktaları kabul etme olasılığı çok düşüktür, dolayısıyla algoritma optimal bir çözüme yaklaşmış demektir (Spall, 2005). Boltzmann dağılımı, benzetimli tavlama yönteminin nicel bir anahtarıdır. Aşağıdaki denklem herhangi bir x durumunun gerçek olasılığını hesaplar (Kirkpatrick vd, 1983):

$$p^x = e^{\frac{-\Delta f(x)}{kT}} \quad (3.52)$$

Burada  $f(x)$  enerjinin biçimi, k Boltzmann sabitidir ve T sıcaklıktır (Rere vd, 2014). Başlangıçta, sistem yüksek bir T ile başlar, ardından belirli bir prosedüre göre T'yi yavaşça azaltarak bir soğutma işlemi uygulanır. Her T'de bir dizi rastgele yeni durum üretilir. Maliyet işlevini geliştiren durumlar kabul edilir. Daha sonrasında,

maliyet işlevini iyileştirmeyen durumları her zaman reddetmek yerine, bu durumlar, artış ve T miktarına bağlı olarak bazı sonlu olasılıklarla kabul edilebilir. Bu süreç, yinelemeli iyileştirme aşamasını rastgele hale getirir ve ayrıca yerel minimuma düşme olasılığını azaltır (Zomaya ve Kazman, 2010).



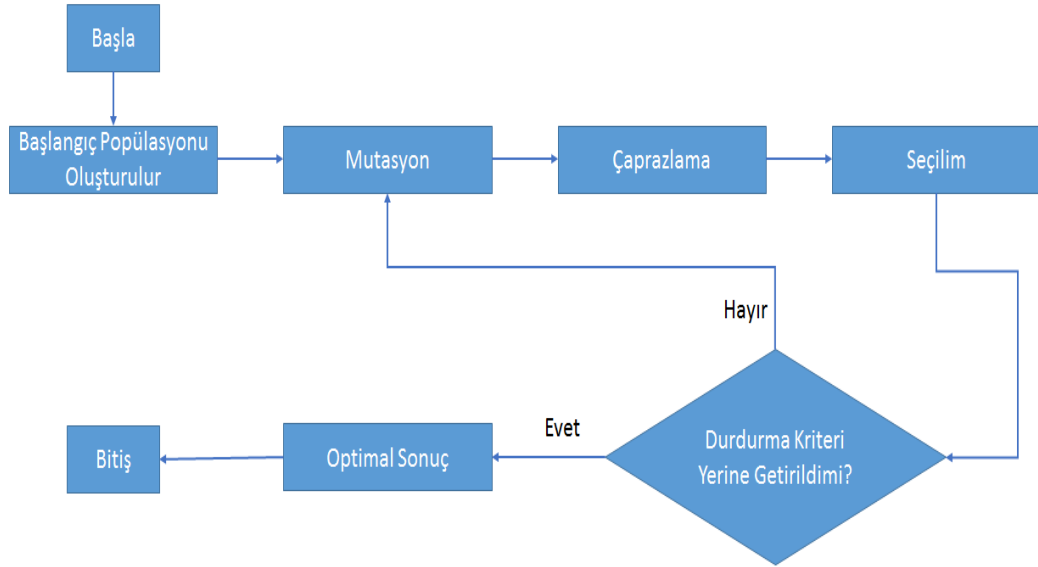
Şekil 3.19. Benzetimli tavlama algoritmasının temel akış şeması

### 3.6.4. Diferansiyel Evrim Algoritmasına Genel Bakış

Diferansiyel Evrim algoritması, Storn ve Price tarafından sunulan evrimsel bir hesaplama yöntemidir. Diferansiyel evrim algoritması, doğrusal olmayan ve türevlenemeyen işlevler gibi karmaşık işlevleri en aza indirerek küresel optimizasyonu gerçekleştirir (Price vd, 2006; Storn ve Price, 1997). Bu algoritmanın popüler hale gelmesi ve çeşitli alanlarda (Das ve Suganthan, 2010; Hattori vd, 2016) kullanma nedeni basitlik, verimlilik ve gerçek kodlama, kolay kullanım, yerel arama özelliği ve hızlılık gibi avantajlarına sahip olmasıdır (Rout vd, 2013).

Diferansiyel evrim algoritmasının süreci Şekil 3.20'deki gibi üç ana işlemle yürütülür: mutasyon, çaprazlama ve seçim. Her nesilde, mevcut popülasyonun bireyleri hedef vektörler haline getirilir. Her hedef vektör için, mutasyon işlemi rastgele seçilen iki vektör arasındaki ağırlıklı farkı üçüncü bir vektöre ekleyerek bir mutasyonlu vektör üretilir. Çaprazlama işlemi, mutasyonlu vektörün parametrelerini hedef vektörünkilerle karıştırarak deneme vektörü adı verilen yeni bir vektör oluşturur. Deneme vektörü, hedef vektörden daha iyi bir uygunluk değeri elde ederse, deneme

vektörü sonraki nesilde hedef vektörün yerini alır (Panda, 2011a; Panda, 2011b). Diferansiyel evrim ve diğer evrimsel algoritmalar arasındaki temel fark, diferansiyel evrimin yeni bir popülasyon yaratma şeklidir. Mevcut popülasyonun rastgele seçilen vektörlerinin ölçeklendirilmiş farklılıklarını kullanmaya dayanmaktadır (Das ve Suganthan, 2010).



Şekil 3.20. Diferansiyel evrim algoritmasının genel akış şeması

### 3.6.4.1. Mutasyon

Verilen bir parametre vektörü  $X_{i,G}$  için, üç vektör  $(X_{r_1,G}, X_{r_2,G}, X_{r_3,G})$ ,  $i, r_1, r_2, r_3$  indisleri farklı olacak şekilde rastgele seçilir. Verici vektör  $V_{i,G+1}$ , iki vektör arasındaki ağırlıklı farkın üçüncü vektöre aşağıdaki gibi eklenmesiyle oluşturulur (Rout vd, 2013):

$$V_{i,G+1} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \quad (3.53)$$

\* Burada F (0,2) arasında bir sabittir

### 3.6.4.2. Çaprazlama

Çapraz geçiş için üç ebeveyn seçilir ve yavru bunlardan birinin karışıklığıdır. Deneme vektörü  $U_{i,G+1}$ , hedef vektörün  $X_{i,G}$  ve verici vektörünün  $X_{i,G}$  elemanlarından geliştirilmiştir. Verici vektörünün elemanları, deneme vektörüne aşağıdaki gibi CR olasılıkla girer (Rout vd, 2013):

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{eğer } rand_{j,i} \leq CR \text{ yada } j = I_{rand} \\ X_{j,i,G+1} & \text{eğer } rand_{j,i} > CR \text{ yada } j \neq I_{rand} \end{cases} \quad (3.54)$$

$rand_{j,i} \sim U(0,1)$  ile  $I_{rand}$ ,  $(1,2, \dots, D)$ 'den rastgele bir tamsayıdır; burada  $D$ , çözümün boyutu, yani kontrol değişkenlerinin sayısıdır.

\* Burada  $I_{rand}$ ,  $V_{i,G+1} \neq X_{i,G}$  sağlanmasını garanti eder.

### 3.6.4.3. Seçim

Hedef vektör  $X_{i,G}$ , deneme vektörü  $V_{i,G+1}$  ile karşılaştırılır ve daha iyi uygunluk değerine sahip olan bir sonraki nesle kabul edilir. DE'deki seçim işlemi aşağıdaki denklemlerle temsil edilebilir (Rout vd, 2013):

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{eğer } f(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} & \text{aksi takdirde} \end{cases} \quad (3.55)$$

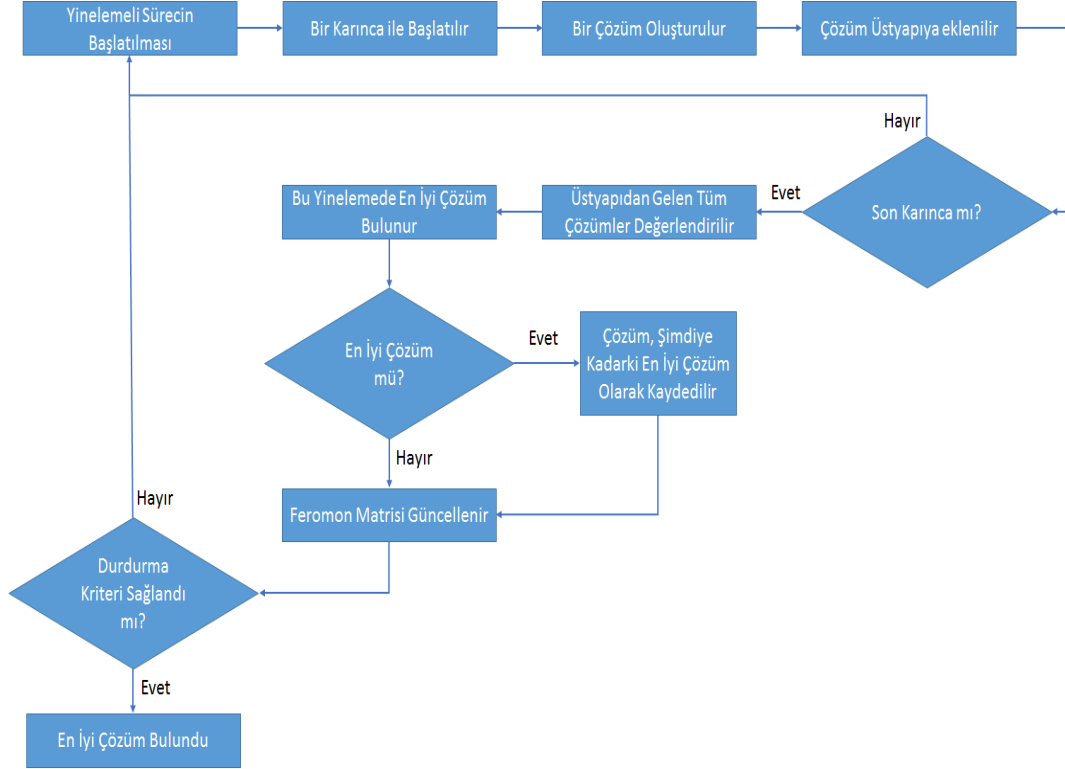
\* Burada  $i \in [1, N_p]$

### 3.6.5. Karınca Koloni Optimizasyonu Algoritmasına Genel Bakış

1990'ların başlarında, karınca koloni optimizasyonu, zor melez optimizasyon problemlerinin çözümü için doğadan esinlenen yeni bir metasezgisel algoritması olarak Dorigo ve arkadaşları tarafından 1991 yılında önerildi (Dorigo vd, 1991). Bu algoritma, bir yapay karınca kolonisinin zor ayrık optimizasyon sorunlarına iyi çözümler bulmak için işbirliği yaptığı bir metasezgisel algoritmasıdır (Dorigo, 1992; Dorigo vd, 1999; Dorigo vd, 1991; Dorigo vd, 1996; Katiyar vd, 2015). İlk olarak, bu yöntem seyyar satıcı problemini çözmek için önerildi (Balaprakash vd, 2009; Dorigo, 1992; Dorigo vd, 1996). Daha sonrasında bu yöntem çeşitli kombinasyonel optimizasyon problemleri için kullanıldı (Colorni vd, 1994; Gambardella ve Dorigo, 2000; Reimann vd, 2002; Stützle ve Dorigo, 1999).

Gerçek bir karınca kolonilerinde, Şekil 3.21'de gösterildiği gibi feromon adlı kokulu bir madde ile dolaylı bir iletişim yolu kullanılır. Bir besin kaynağı bulunduğu anda, karıncalar yolu işaretlemek için bir miktar feromon bırakırlar. Serilen feromonun miktarı besin kaynağının uzaklığına, miktarına ve kalitesine bağlıdır. Rastgele hareket eden izole bir karınca, bırakılan bir feromonu tespit ederken, büyük olasılıkla işaretlenen yoldan devam edecektir. Aynı zamanda bu karıncanın kendisinde

belli bir miktar feromon bırakacaktır ve dolayısıyla o belirli yolun feromon miktarı artacaktır. Bundan dolayı, daha fazla karıncanın kullandığı yolu izlemek daha çekici olacaktır. Diğer bir deyişle, bir karıncanın bir yol seçme olasılığı, daha önce aynı yolu seçen karıncaların sayısı ile doğru orantılıdır. Dolayısıyla bu süreç, pozitif bir geri besleme döngüsü ile tanımlanır (Dorigo ve Blum, 2005; Dorigo vd, 1996).



Şekil 3.21. Karınca koloni optimizasyonu algoritmasının genel akış şeması

Karıncalar, bir olasılık kuralı tarafından, bir soruna çözümlerini seçmeye yönlendirilirler. İki düğüm  $j$  arasındaki olasılık kuralı, sözde Rastgele Orantılı Eylem Seçim Kuralı olarak adlandırılan bu yöntem, iki faktöre dayanmaktadır: sezgisel ve metasezgisel (Katiyar vd, 2015).

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} [\eta]^{\beta}}{\sum_{h \in S} [\tau_{ij}]^{\alpha} [\eta]^{\beta}} \quad (3.56)$$

Burada  $\tau$  feromoni temsil eder ve  $\eta$  iki düğüm arasındaki mesafenin tersidir. Ayrıca her karınca, ortamı iki farklı şekilde değiştirebilir (Dorigo vd, 1991):

- Yerel iz güncelleme: Karınca, düğümler arasında hareket ettikçe kenardaki feromon miktarını aşağıdaki denklemle günceller :

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t - 1) + \rho \cdot \tau_0 \quad (3.57)$$

Burada,  $\rho$  buharlaşma sabitidir.  $\tau_0$  ise, feromon izlerinin başlangıç değeridir ve şu şekilde hesaplanabilir:

$$\tau_0 = (n/L_n) - 1 \quad (3.58)$$

Burada  $n$  düğüm sayısıdır ve  $L_n$ , yapısal keşiflerden biri tarafından oluşturulan toplam düğümler arasında gidilen toplam mesafedir.

- Küresel iz güncelleme: Tüm karıncalar en kısa yolu bulan tüm düğümleri tamamladığında, aşağıdaki denklemi kullanarak yolundaki kenarları günceller:

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t - 1) + \frac{\rho}{L^+} \quad (3.59)$$

Burada  $L^+$ , karıncalardan birinin ürettiği en iyi yolun uzunluğudur.

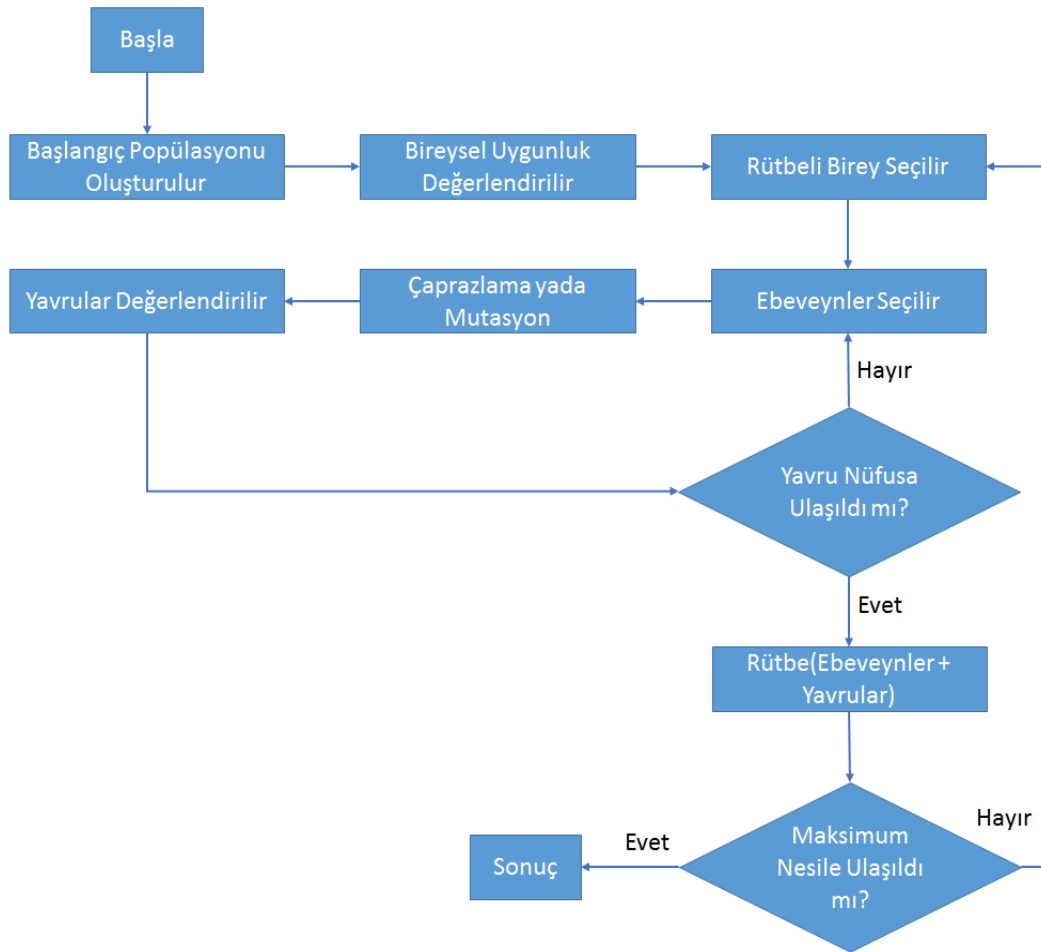
### 3.6.6. NSGA II Algoritmasına Genel Bakış

Genetik Algoritması biyolojik evrim teorisi ve genetik teori sayesinde ortaya çıkmış ve gelişmiştir. Biyolojik doğal seçilimi ve doğal genetik mekanizmayı taklit eden rastgele bir arama algoritmasıdır (Goldberg, 1989a; Holland, 1992b). Ancak genetik algoritmanın bazı geliştirilmiş versiyonlarında mevcuttur; onlardan biri ise NSGA(Non-dominated Sorting Genetic Algorithm = Baskın-olmayan Sıralama Genetik Algoritması) 1995 yılında Srinivas ve Deb tarafından önerilmiştir (Srinivas ve Deb, 1994) ve 2000 yılında geliştirilmiş versiyonu (Deb vd, 2000) ortaya konmuştur. Bu algoritmada tıkanıklık ve tıkanıklık karşılaştırma operatörü kullanılmaktadır ve elit strateji yöntemi ile iyi bireyleri gelecek nesile aktarır. Bu şekilde nüfus düzeyi hızlıca artmaktadır (Deb vd, 2002).

NSGA II algoritmasının en popüler çok amaçlı optimizasyon algoritmalarından biri olma nedeni ise, 3 tane ana özelliğe sahip olmasıdır (Deb vd, 2002) :

- baskın olmayan hızlı sıralama yaklaşımı
- hızlı kalabalık mesafe tahmin prosedürü
- basit kalabalık karşılaştırma operatörü

NSGA-II prosedürü, bir birey popülasyonu oluşturarak başlar. Daha sonra, her bireyi bir egemenlik kuralına göre sıralar. Ardından, yeni bir yavru popülasyonu bulmak için evrimsel operatörleri, yani çaprazlama ve mutasyonu uygular. İki eşit büyüklükte popülasyon oluşturulduktan sonra, yeni birleşik nüfusun yarısını Pareto cephelerinde paylaşmak için ebeveynleri ve torunları birleştirir. Cephenin çeşitliliğini sağlamak için NSGA-II, her bireye bir kalabalıklık mesafesi ekler. Bu, nüfusun çeşitliliğini garanti eder ve uygunluk ortamının keşfedilmesini geliştirir (Damak vd, 2013). Şekil 3.22, açık bir şekilde, NSGA-II algoritmasının genel akış şemasını açıklamaktadır.



Şekil 3.22. NSGA II algoritmasının genel akış şeması

Bu algoritmanın temel adımları aşağıdaki gibidir (Chang vd, 2015; Deb, 2001):

- Başlangıç popülasyonu: NSGA-II algoritmasının ilk popülasyonu, değişkenlerin boyutuna bağlı olarak seçilebilir. Herhangi bir ön bilgi olmaksızın, ilk popülasyon, varsa kısıtlamaların kullanılmasıyla rastgele oluşturulabilir.

Önceki bilgilerle, ilk popülasyon seçimi, yakınsamayı hızlandırmak ve hesaplama süresinden tasarruf etmek için kullanılabilir.

- Baskın olmayan sıralama: Bireylerin her bir yinelemede işlev değerlendirmeleri yapıldıktan sonra, bireyler uygunluk ve kalabalıklaşma mesafeleri dikkate alınarak sıralanır. Uygunluk, bir bireyin gelecek nesil için hayatta kalma ve yeniden üretme yeteneğidir; kalabalıklaşma mesafesi ise bir bireyin komşularına ne kadar yakın olduğunun bir ölçüsüdür ve her bir hedefe karşılık gelen bireysel mesafelerin toplamı olarak hesaplanır. Uygun nesnel işlev değerleriyle sonuçlanan ve iyi çeşitliliğe sahip çözümler bulmak için, bireyleri nesnel işlevlere göre ve her iki birey arasındaki kalabalık mesafesine göre sıralanır.
- Genetik operatör: Bireylerin sıralamasına bağlı olarak, ebeveyn olarak en üst sıralarda yer alan bireyleri seçilir ve yavru bireyler oluşturmak için genetik operatörü kullanılır. Ardından, uzaklaşan popülasyonlardaki her bir birey için nesnel işlevleri değerlendirilir.
- Yeniden kombine etmek ve seçim: Hem ebeveynler hem de yavrular birleştirilir, uygunluklarını değerlendirmek için baskın olmayan ayıklama işlemi uygulanır ve ardından gelecek neslin ebeveynleri olarak en üst sıradaki bireyler seçilir. Farklı hedefleri karşılaştırarak hiçbir birey diğerinden üstün değilse, iki kişiye baskın olmayan denir. Bireylerin rütbesi ya da uygunluğu, birinci cephedeki (pareto) bireylerin tamamen baskın olmayan küme, ikinci cephedeki bireylere ise yalnızca birinci cephedeki bireylerin hakim olacağı şekilde atanır. İlk paretodaki bireyler öncelikli olarak seçilir. Tasarlanan popülasyon büyüklüğüne ulaşılmazsa, ikinci paretodaki bireyler, bir sonraki yinelemeye başlamak için kişi sayısı tatmin olana kadar dikkate alınacaktır.

## 4. UYGULAMA

### 4.1. Veri Seti

Bu çalışmada hazır veri seti (Body Fat Dataset = Vücut Yağ Veri Seti) kullanıldı ve bu kullanılan veri seti, matlab programının içerisinde bulunmaktadır. Çalışmada hazır veri seti kullanmanın amaçları ise, bu tezi okuyanlar rahatlıkla veriye erişmeleri ve ayrıca denenmiş bu veri seti üzerinde sağlıklı sonuç alabilmeleridir. Aşağıdaki kısımda bu veri setinin özellikleri özet olarak anlatılmaktadır :

#### 4.1.1. Vücut Yağ Tahmini Veri Seti

Bu veri seti, anatomik ölçümlere dayalı olarak vücut yağ yüzdesini tahmin edilmesini göstermektedir. Bu veri setinde 252 farklı kişiden, 13 tane fiziksel özellik ölçümü alınarak vücut yağı tahmin edilir. Bu özelliklerin isimleri Tablo 4.1’de görülmektedir :

Tablo 4.1. Vücut yağı veri setinin özellikleri

Yaş (Yıl)	Kilo (pound)	Boy (inç)	Boyun çevresi (santimetre)
Göğüs çevresi (santimetre)	Karın çevresi (santimetre)	Kalça çevresi (santimetre)	Uyluk çevresi (santimetre)
Diz çevresi (santimetre)	Ayak bileği çevresi (santimetre)	Pazu çevresi (santimetre)	Önkol çevresi (santimetre)
Bilek çevresi (santimetre)			

### 4.2. Bulgular

Bu kısımda öncelikle veri seti üzerinde yapay sinir ağı yöntemi uygulanmaktadır. Bu analizlerin sonucunda bulunan değerler ve grafikler ayrı ayrı aşağıdaki kısımda görülmektedir. Daha sonrasında veri seti üzerinde metasezgisel algoritmaları ile üç farklı özellik seçme yöntemi uygulanır :

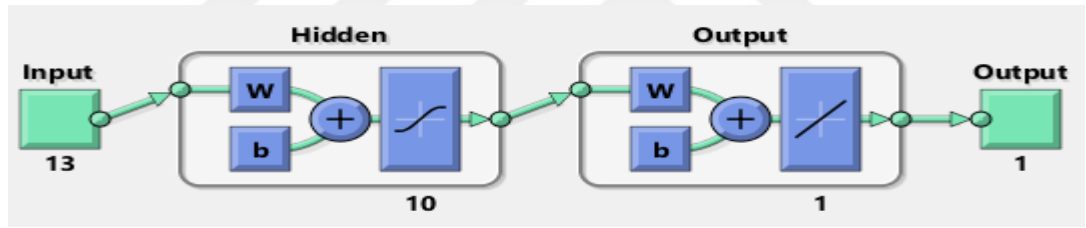
- Seçili özellik sayısını bilmeden, özellik seçimi yapmak
- Daha önceden belirlenmiş özellik sayısını bilerek, özellik seçimi yapmak
- Çok amaçlı özellik seçimi yapmak

Bu uygulanan üç farklı özellik seçme yöntemlerinin sonuçları aşağıdaki kısımlarda görülmektedir. Bulunan sonuçların değerlendirilmesi, sonuçlar bölümünde yer almaktadır.

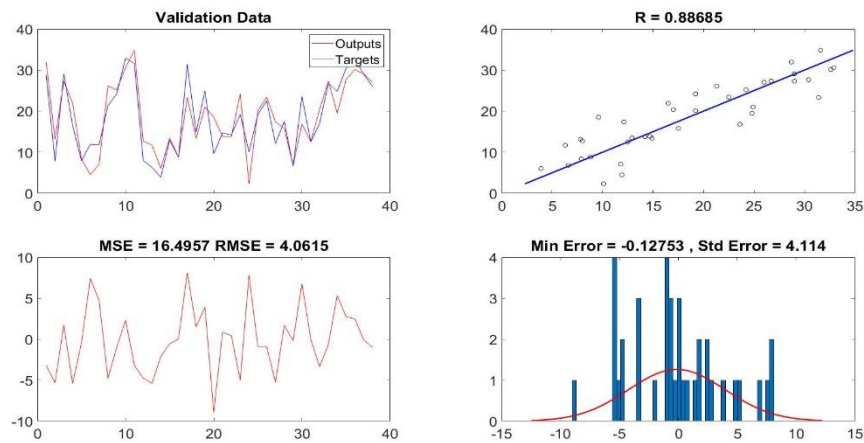
Bu bölümde ilk olarak yapay sinir ağı yöntemi ile vücut yağı tahminlenir. Daha sonrasında metasezgisel algoritmaları ile üç farklı özellik seçme yöntemi, veri seti üzerinde uygulanır.

#### 4.2.1. Yapay Sinir Ağı ile Vücut Yağ Tahmini

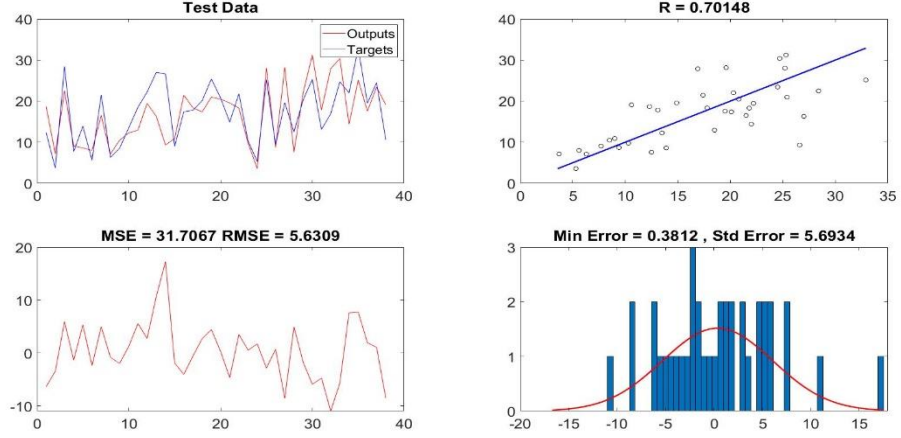
Şekil 4.1’de görüldüğü üzere yapay sinir ağı 10 tane gizli katman kullanarak çalışmaktadır ve eğitim fonksiyonu olarak Levenberg-Marquardt fonksiyonu kullanılmaktadır.



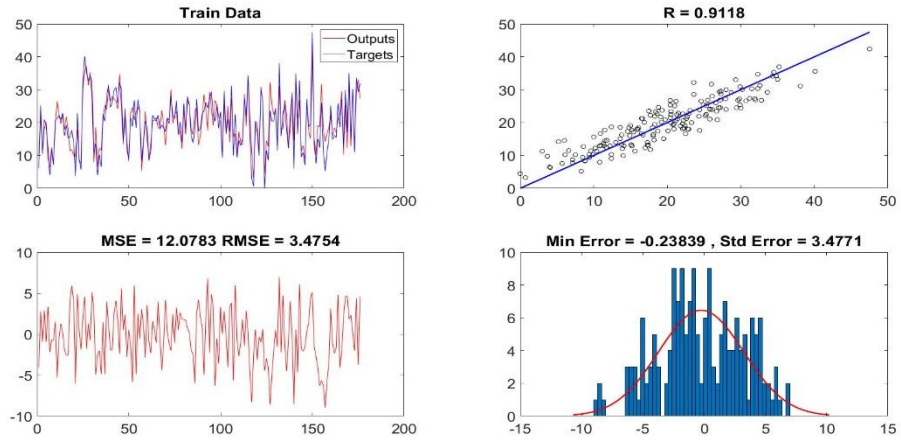
Şekil 4.1. Veriler üzerinde yapay sinir ağı diyagramı



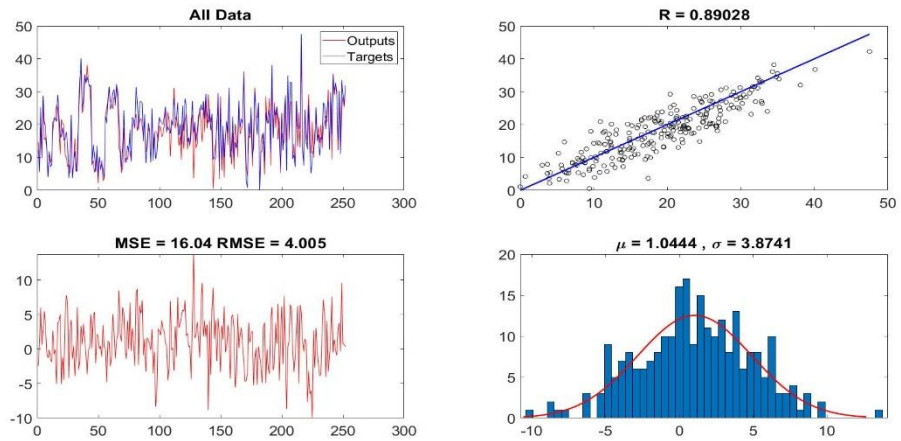
Şekil 4.2. Yapay sinir ağı validation (doğrulama) grafiği



Şekil 4.3. Yapay sinir ağı test grafiği



Şekil 4.4. Yapay sinir ağı train (eğitim) grafiği



Şekil 4.5. Yapay sinir ağı tüm veriler grafiği

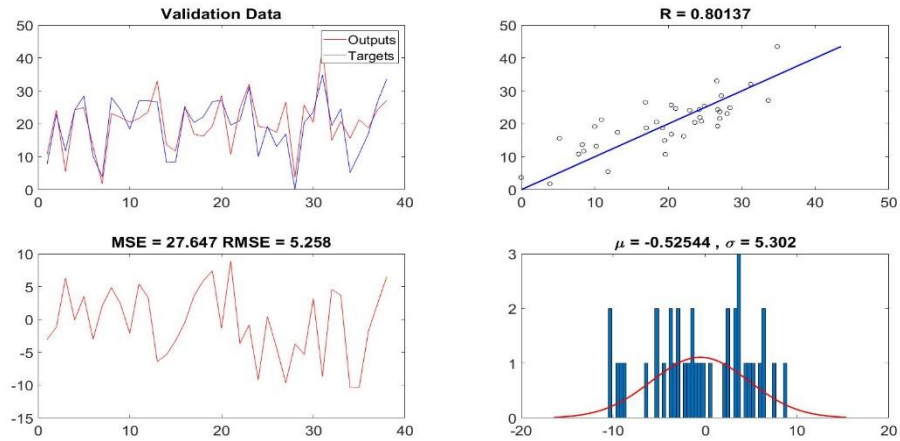
Tablo 4.2. Vücut yağı veri seti üzerinde yapay sinir ağı değerleri

	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	16.4957	4.0615	-0.12753	4.114	0.88685
Test Data	15	31.7067	5.6309	0.3812	5.6934	0.70148
Train Data	70	12.0783	3.4754	-0.23839	3.4771	0.9118
All Data	100	16.04	4.005	1.0444	3.8741	<b>0.89028</b>

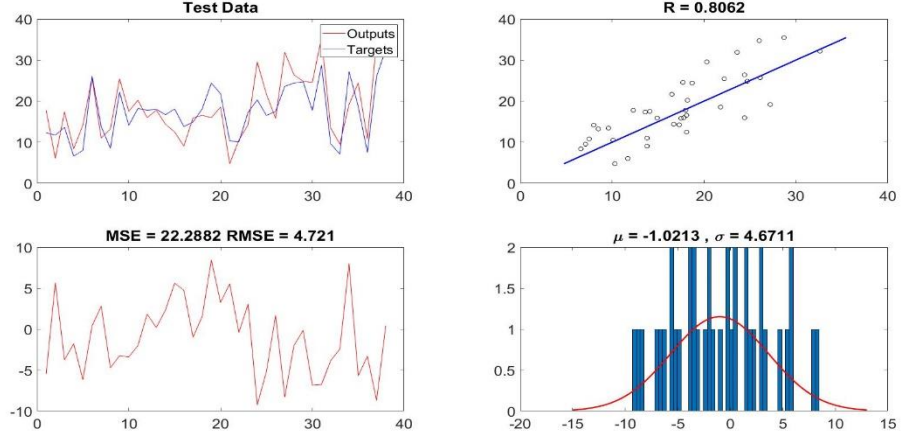
Tablo 4.2’de görüldüğü gibi normal yapay sinir ağını kullanarak tüm veriler için korelasyon değerimiz 0.89028 olarak bulundu. Bu yöntemde özellik seçimi uygulanmamıştır ve tüm özellikler tahminleme için kullanıldı. Amacımız bu tezde az özellik kullanarak aynı performansa ulaşmaktır.

#### 4.2.2. Birden Çok Kez Çalıştırılmalı Yapay Sinir Ağı ile Vücut Yağ Tahmini

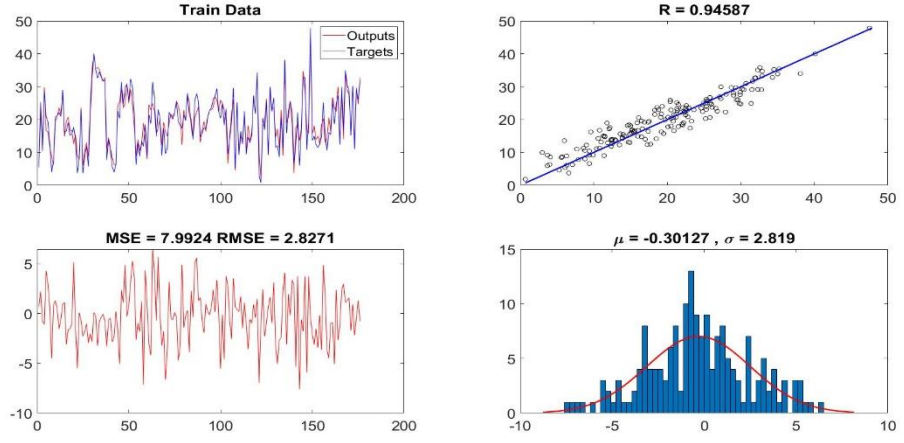
Aşağıdaki kısımda ise, veri seti üzerinde birden çok kez çalıştırılmalı yapay sinir ağı uygulanmaktadır. Bu yöntemde 10 gizli katmanı, 10 defa çalıştırılır ve sonunda bulunan tüm değerlerin ortalamasını alarak, daha gerçekçi bir sonuç üretilir.



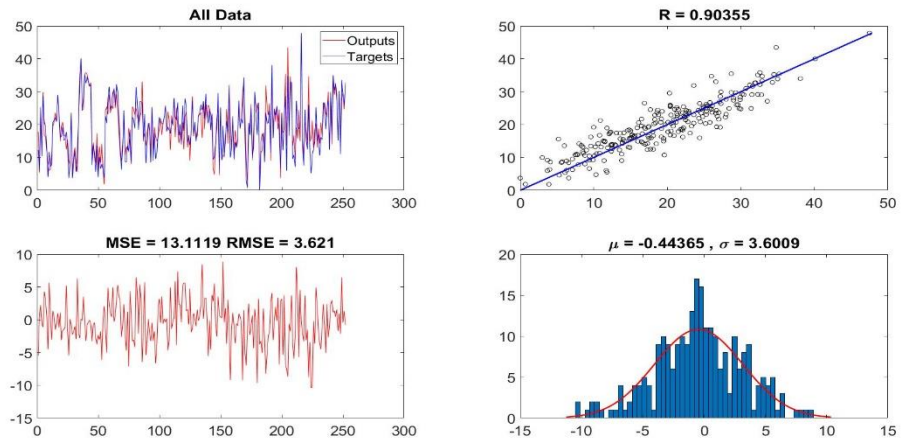
Şekil 4.6. Birden çok kez çalıştırılmalı yapay sinir ağı validation (doğrulama) grafiği



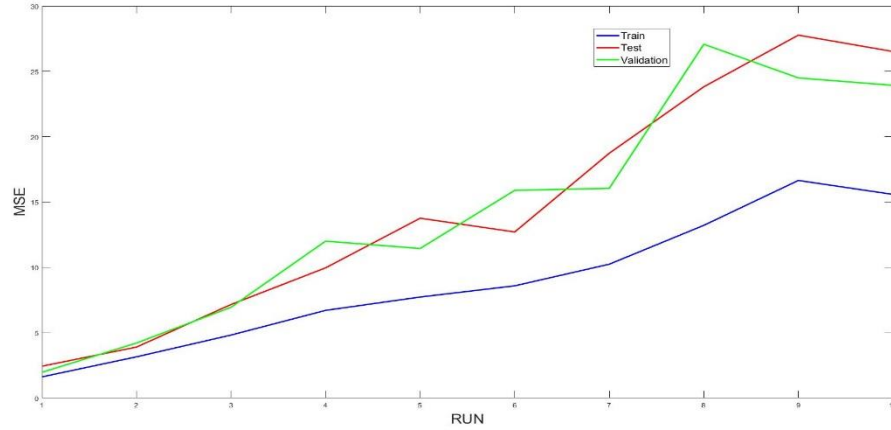
Şekil 4.7. Birden çok kez çalıştırılmalı yapay sinir ağı test grafiği



Şekil 4.8. Birden çok kez çalıştırılmalı yapay sinir ağı train (eğitim) grafiği



Şekil 4.9. Birden çok kez çalıştırılmalı yapay sinir ağı tüm veriler



Şekil 4.10. Birden çok kez çalıştırılmalı yapay sinir ağının 10 döngü için mse grafiği

Tablo 4.3. Vücut yağı veri seti üzerinde birden çok kez çalıştırılmalı yapay sinir ağı değerleri

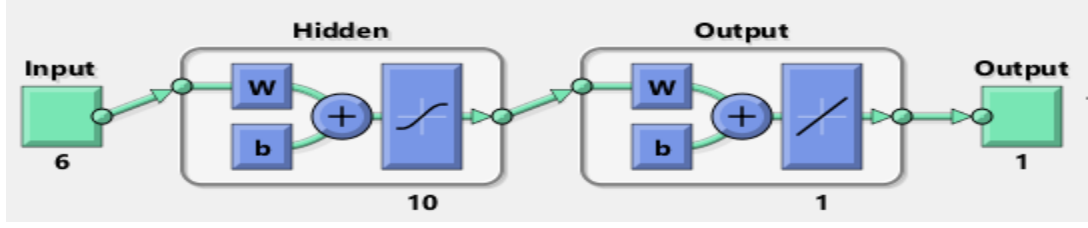
	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	27.647	5.258	-0.52544	5.302	0.80137
Test Data	15	22.2882	4.721	-1.0213	4.6711	0.8062
Train Data	70	7.9924	2.8271	-0.30127	2.819	0.94587
All Data	100	13.1119	3.621	-0.44365	3.6009	<b>0.90355</b>

Tablo 4.3’de görüldüğü gibi tüm veriler için korelasyon değerimiz 0.90355 olarak bulundu. Tablo 4.2’deki değerler ile kıyaslandığında, birden çok kez çalıştırılmalı yapay sinir ağının daha iyi bir tahminlemede bulunduğu ortaya çıkmıştır. Bundan dolayı bu tezde özellik sayısını metasezgiseller ile azaltarak, birden çok kez çalıştırılmalı yapay sinir ağının performansına yaklaşılmaktadır.

#### 4.2.3. Seçili Özellik Sayısını Bilmeden Özellik Seçimi Yapmak

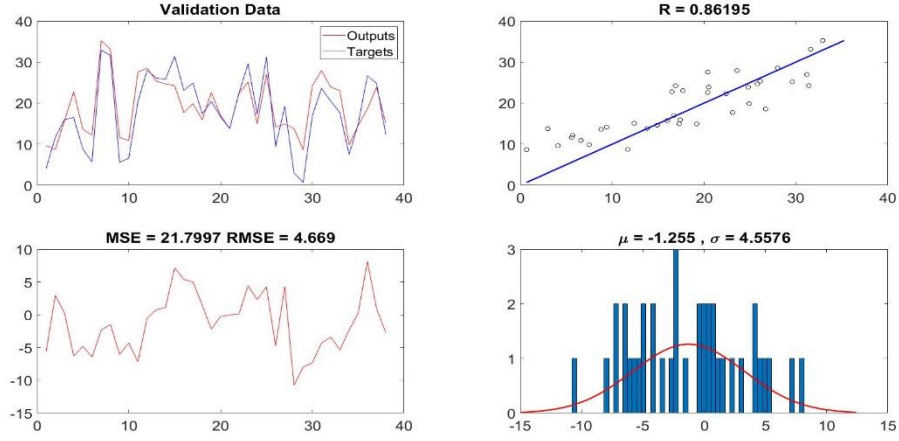
Bu bölümde özellik sayısını bilmeden, yapay sinir ağını ve ikili genetik algoritmayı kullanarak özellik seçimi yapılmaktadır.

### 4.2.3.1. Yapay Sinir Ağı ve İkili Genetik Algoritmayı Kullanarak Özellik Seçimi Yapmak

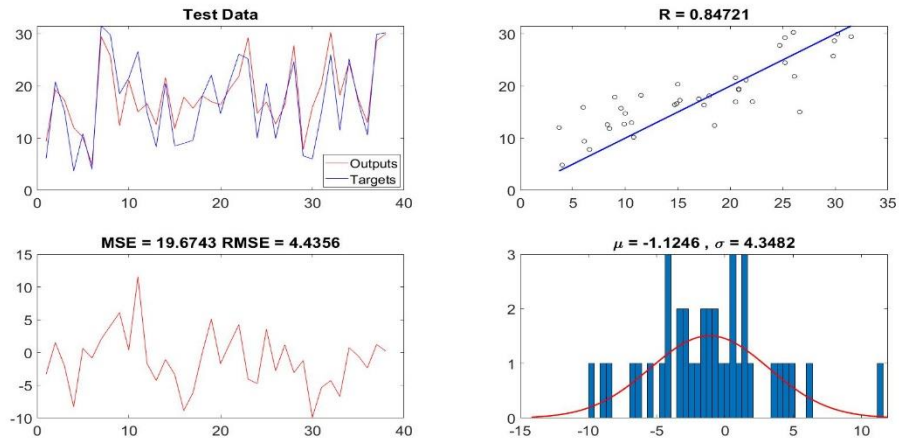


Şekil 4.11. İkili genetik algoritma ile özellik seçimi, yapay sinir ağı diyagramı

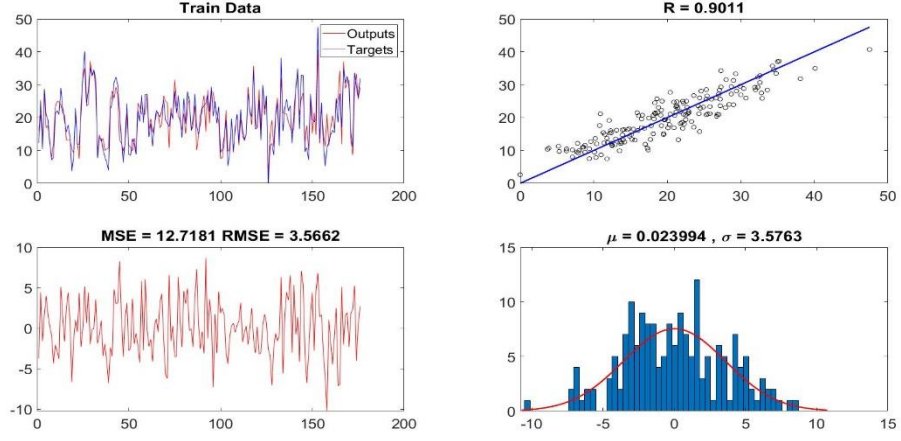
Şekil 4.11’de görüldüğü üzere ikili genetik algoritması 6 tane özelliği seçmiştir. Bu yöntemde yapay sinir ağı tüm özellikler yerine 6 tane önemli özellik ile vücut yağını tahminlemeye çalışmaktadır.



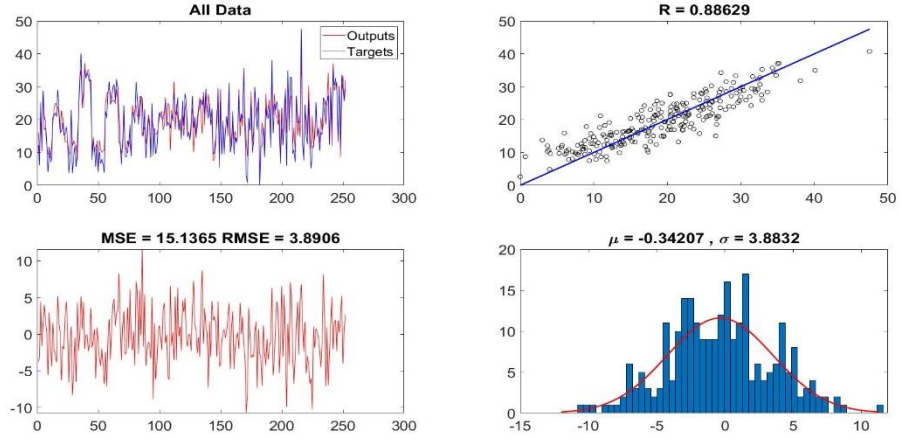
Şekil 4.12. İkili genetik algoritma ile özellik seçimi, validation (doğrulama) grafiği



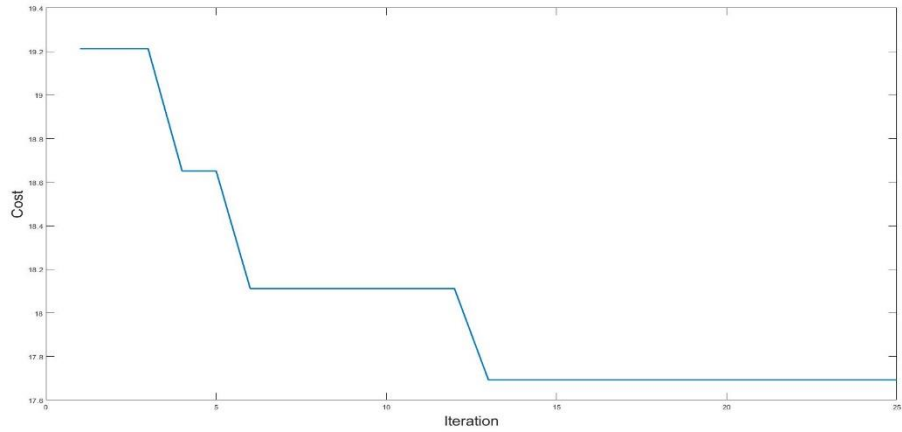
Şekil 4.13. İkili genetik algoritma ile özellik seçimi, test grafiği



Şekil 4.14. İkili genetik algoritma ile özellik seçimi, train (eğitim) grafiği



Şekil 4.15. İkili genetik algoritma ile özellik seçimi, tüm veriler grafiği



Şekil 4.16. İkili genetik algoritma ile özellik seçimi, her yinelemede maliyet grafiği

İkili genetik algoritması ile özellik seçimi, Şekil 4.16'da görüldüğü gibi 25 yinelemede çalışmaktadır ve her yinelemede maliyet değeri kaydedilmektedir. 13.üncü yinelemeden sonra maliyet değeri sabitlenmeye başlamıştır ve değişiklik görülmemektedir.

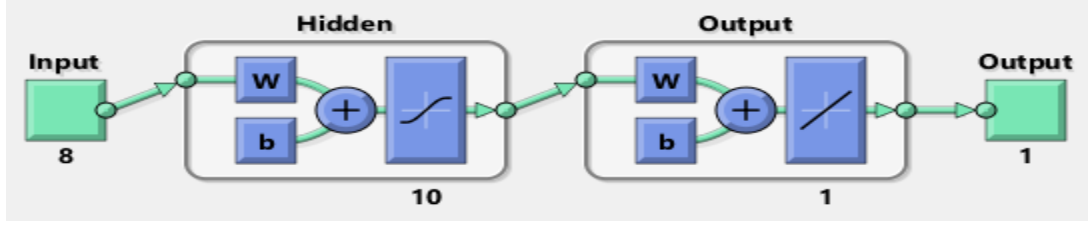
Tablo 4.4. Vücut yağı veri seti üzerinde yapılan İkili genetik algoritma ile özellik seçimi değerleri

	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	21.7997	4.669	-1.255	4.5576	0.86195
Test Data	15	19.6743	4.4356	-1.1246	4.3482	0.84721
Train Data	70	12.7181	3.5662	0.023994	3.5763	0.9011
All Data	100	15.1365	3.8906	-0.34207	3.8832	<b>0.88629</b>

Tablo 4.4'te görüldüğü gibi tüm veriler için korelasyon değerimiz 0.88629 olarak bulundu. İkili genetik algoritmasının 6 tane özellik ile, yapay sinir ağı ve birden çok kez çalıştırılmalı yapay sinir ağının değerleri ile kıyaslandığında, bunların performansına yakın bir değer bulunmuştur.

#### 4.2.4. Daha Önceden Belirlenmiş Özellik Sayısını Bilerek Özellik Seçimi Yapmak

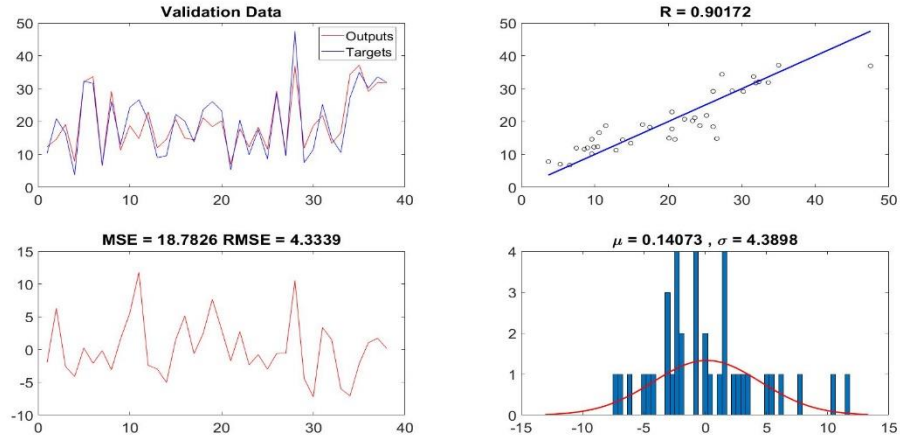
Bu bölümde daha önceden özellik sayısı Şekil 4.17'de görüldüğü gibi 8 olarak belirlenir. Ardından ayrık ve sürekli kodlama için metasezgisel algoritmaları kullanılır. Ayrık kodlama için permütasyon optimizasyon problemi kullanılmaktadır ve bu kısımda daha önceden belirlenmiş özellik sayısını bilerek yapay sinir ağını ve karınca koloni optimizasyonu algoritmasını ve benzetimli tavlama algoritmasını kullanarak, özellik seçimi yapılır. Sürekli kodlamada ise rastgele anahtar yöntemi kullanılmaktadır ve bu kısımda daha önceden belirlenmiş özellik sayısını bilerek, yapay sinir ağını ve paraçacık sürüsü optimizasyonu algoritmasını ve diferansiyel evrim algoritmasını kullanarak, özellik seçimi yapılır.



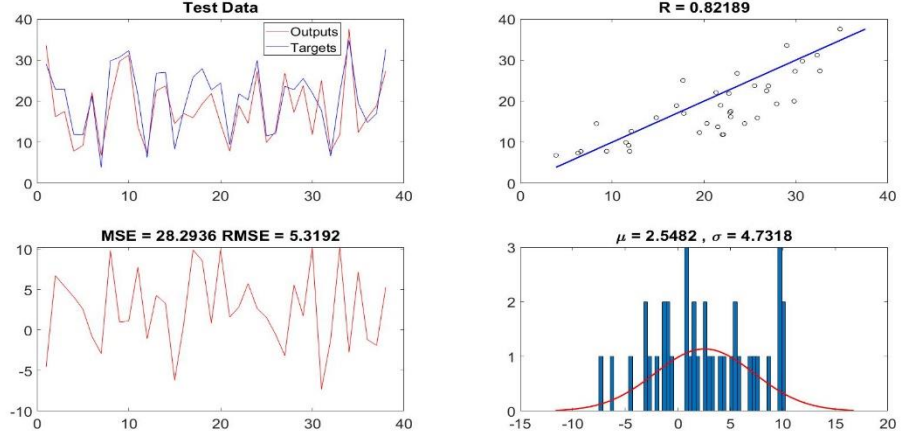
Şekil 4.17. Daha önceden belirlenmiş özellik sayısı ile özellik seçimi, yapay sinir ağı diyagramı

Daha önceden belirlenmiş özellik sayısının 8 olarak belirlenmesinin nedeni ise, ilerki kısımlarda yapılan çok amaçlı özellik seçimleri ile 8 özelliğin daha iyi bir sonuç gösterdiği görülmektedir. Dolayısıyla Şekil 4.17’de görüldüğü üzere, girdi olarak 8 tane en önemli özellik kullanılır.

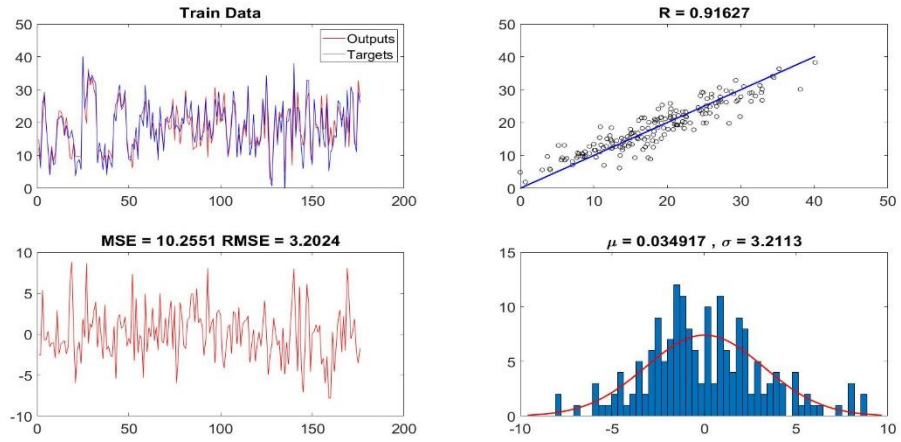
#### 4.2.4.1. Yapay Sinir Ağı ve Karınca Koloni Optimizasyonu Algoritmasını Kullanarak Özellik Seçimi Yapmak



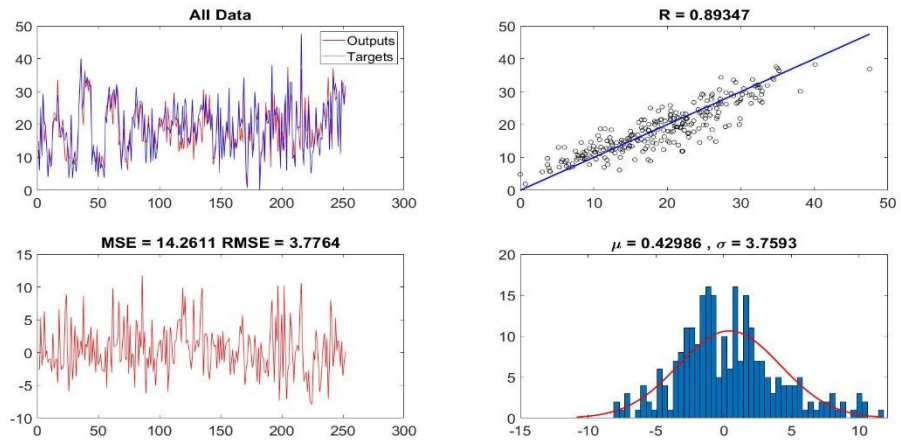
Şekil 4.18. Karınca koloni optimizasyonu algoritması ile özellik seçimi, validation (doğrulama) grafiği



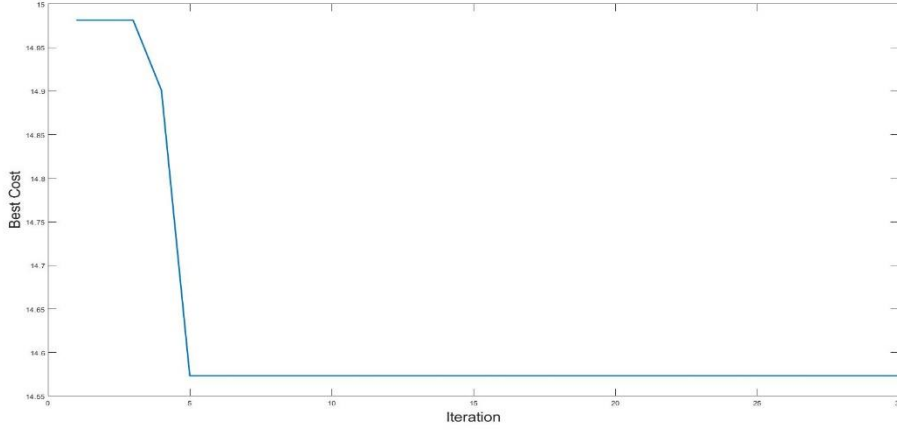
Şekil 4.19. Karınca koloni optimizasyonu algoritması ile özellik seçimi, test grafiği



Şekil 4.20. Karınca koloni optimizasyonu algoritması ile özellik seçimi, train (eğitim) grafiği



Şekil 4.21. Karınca koloni optimizasyonu algoritması ile özellik seçimi, tüm veriler grafiği



Şekil 4.22. Karınca koloni optimizasyonu algoritması ile özellik seçimi, her yinelemede maliyet grafiği

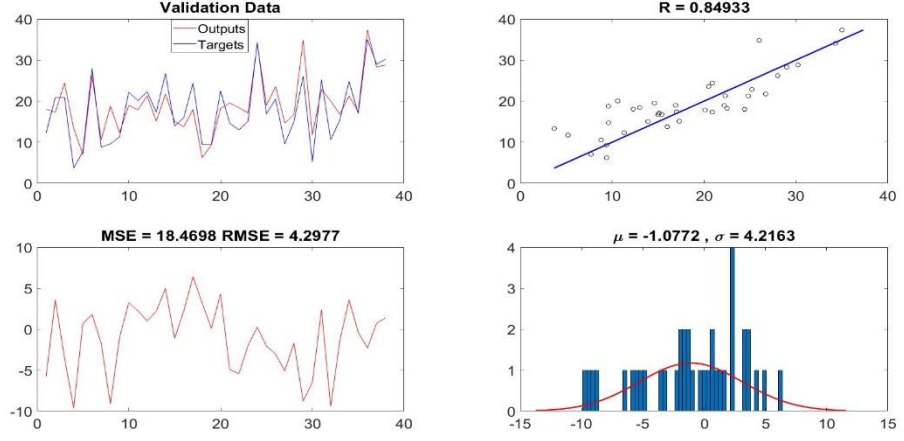
Karınca koloni optimizasyonu algoritması ile özellik seçimi, Şekil 4.22’de görüldüğü gibi 30 yinelemede çalışarak, her yinelemenin değeri kaydedilir. 5.inci yinelemeden sonra maliyet değeri sabitlenmiştir ve değişiklik göstermemektedir.

Tablo 4.5. Vücut yağı veri seti üzerinde yapılan karınca koloni optimizasyonu algoritması ile özellik seçimi değerleri

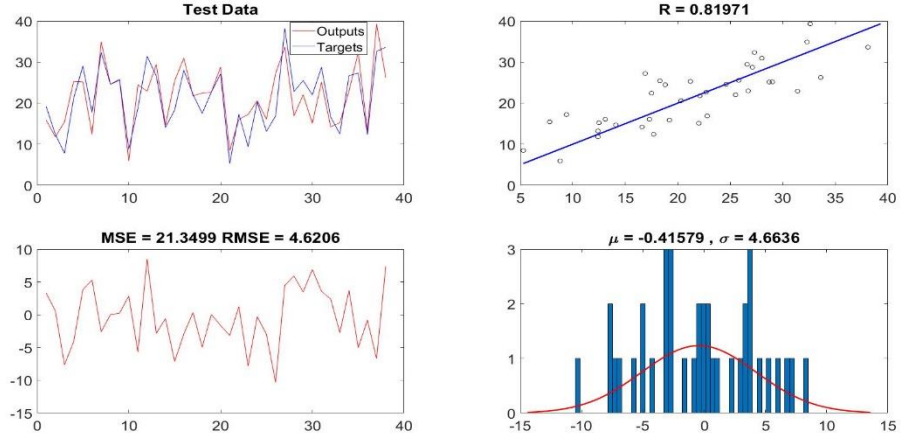
	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	18.7826	4.3339	0.14073	4.3898	0.90172
Test Data	15	28.2936	5.3192	2.5482	4.7318	0.82189
Train Data	70	10.2551	3.2024	0.034917	3.2113	0.91627
All Data	100	14.2611	3.7764	0.42986	3.7593	<b>0.89347</b>

Tablo 4.5’de görüldüğü gibi tüm veriler için korelasyon değeri 0.89347 olarak bulundu. Daha önceden özellik sayısı 8 olarak belirlenmiş karınca koloni optimizasyonu algoritması ile özellik seçimi, birden çok kez çalıştırılmalı yapay sinir ağının korelasyon değeri ile çok yakındır.

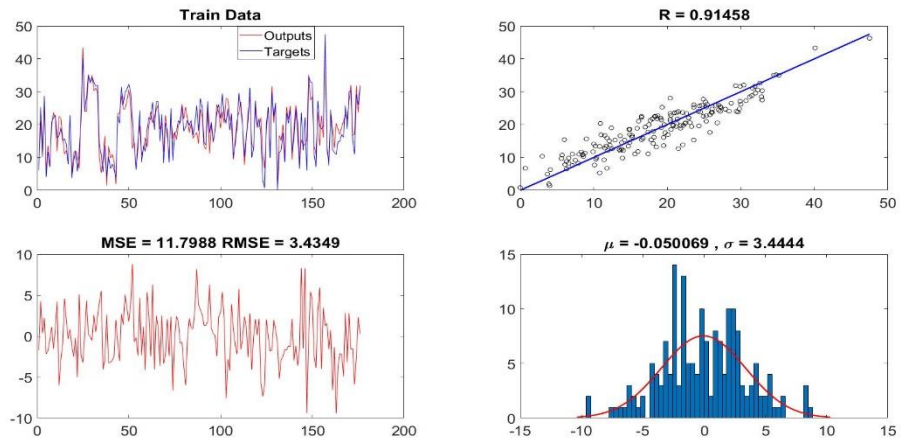
#### 4.2.4.2. Yapay Sinir ve Benzetimli Tavlama Algoritmasını Kullanarak Özellik Seçimi Yapmak



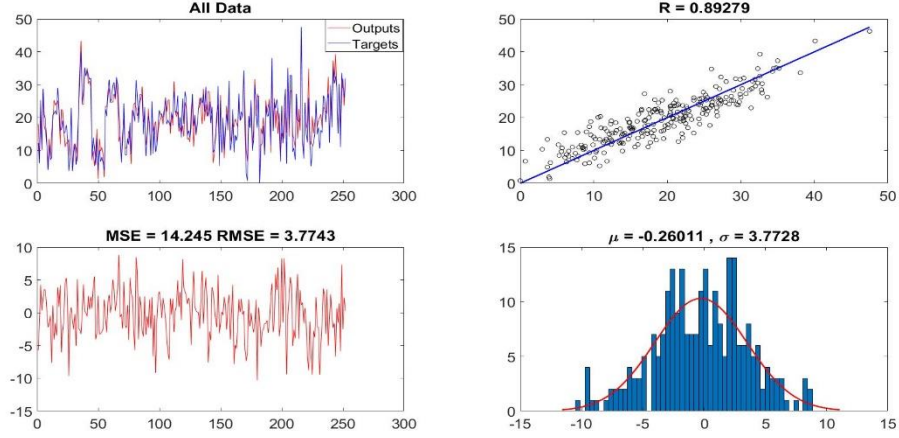
Şekil 4.23. Benzetimli tavlama algoritması ile özellik seçimi, validation (doğrulama) grafiği



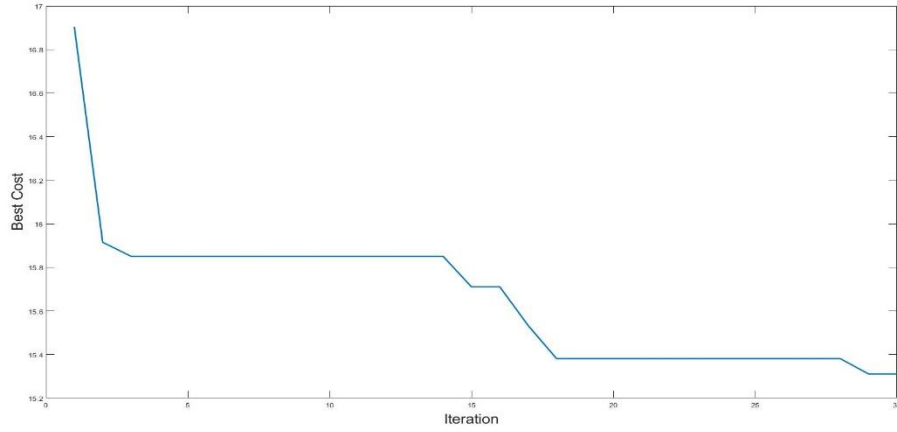
Şekil 4.24. Benzetimli tavlama algoritması ile özellik seçimi, test grafiği



Şekil 4.25. Benzetimli tavlama algoritması ile özellik seçimi, train (eğitim) grafiği



Şekil 4.26. Benzetimli tavlama algoritması ile özellik seçimi, tüm veriler grafiği



Şekil 4.27. Benzetimli tavlama algoritması ile özellik seçimi, her yinelemede maliyet grafiği

Benzetimli tavlama algoritması ile özellik seçimi, 30 yinelemede çalışmaktadır ve her yinelemenin değeri kaydedilmektedir. Şekil 4.27’de görüldüğü gibi her yinelemede maliyet değeri düşmektedir.

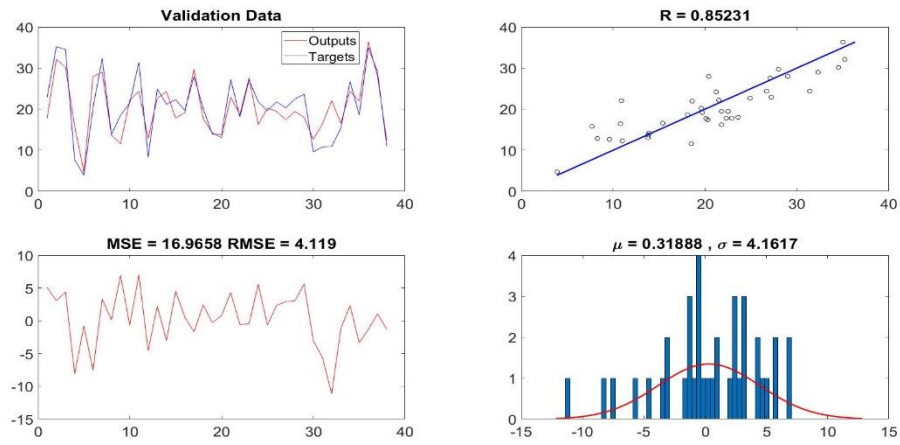
Tablo 4.6. Vücut yağı veri seti üzerinde yapılan benzetimli tavlama algoritması ile özellik seçimi değerleri

	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	18.4698	4.2977	-1.0772	4.2163	0.84933
Test Data	15	21.3499	4.6206	-0.41579	4.6636	0.81971
Train Data	70	11.7988	3.4349	-0.05006	3.4444	0.91458

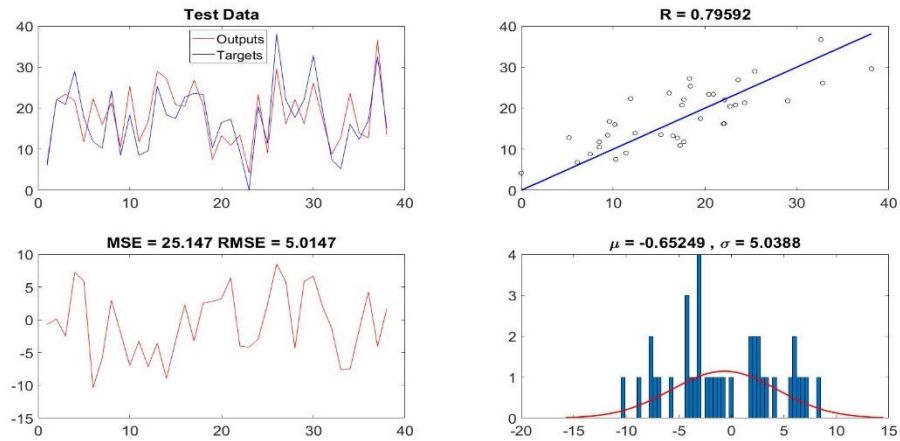
All Data	100	14.245	3.7743	-0.26011	3.7728	<b>0.89279</b>
----------	-----	--------	--------	----------	--------	----------------

Tablo 4.6’da görüldüğü gibi tüm veriler için korelasyon değeri 0.89279 olarak bulundu. Daha önceden özellik sayısı 8 olarak belirlenen benzetimli tavlama algoritması ile özellik seçiminin değerleri, yapay sinir ağı yönteminin değerleri ile hemen hemen aynıdır.

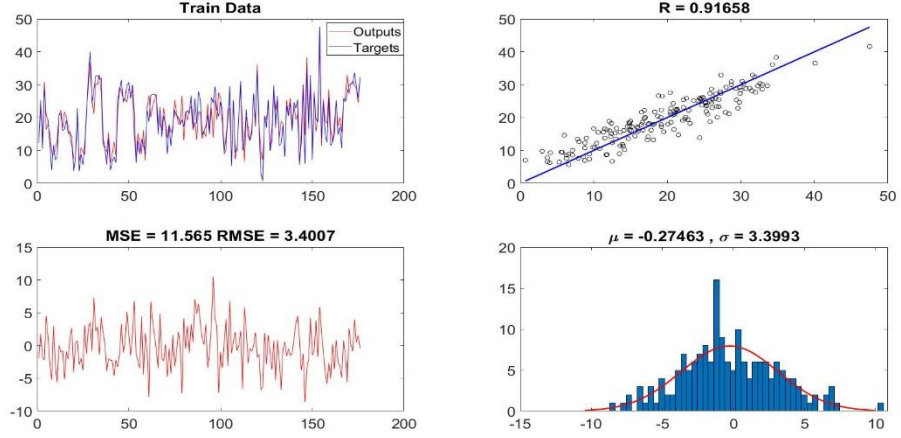
#### 4.2.4.3. Yapay Sinir Ağı ve Parçacık Sürüsü Optimizasyonu Algoritmasını Kullanarak Özellik Seçimi Yapmak



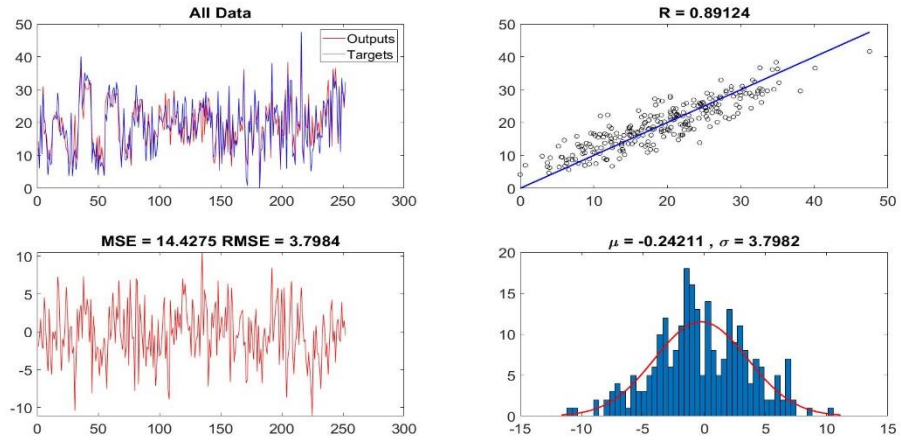
Şekil 4.28. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, validation (doğrulama) grafiği



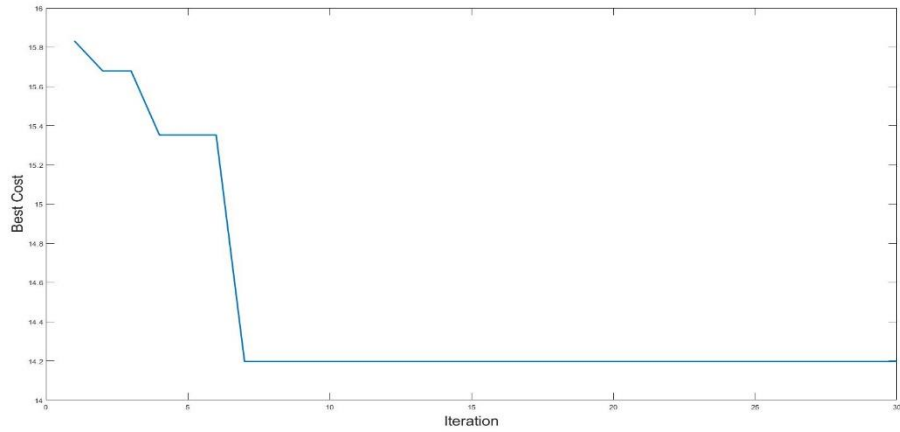
Şekil 4.29. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, test grafiği



Şekil 4.30. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, train (eğitim) grafiği



Şekil 4.31. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, tüm veriler grafiği



Şekil 4.32. Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, her yinelemede maliyet grafiği

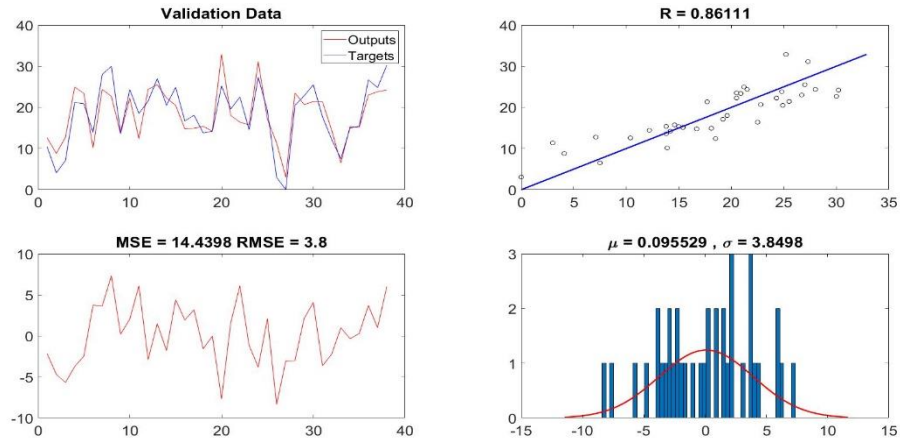
Parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, 30 yinelemede çalışmaktadır ve her yinelemenin değeri kaydedilmektedir. Şekil 4.32’de görüldüğü gibi maliyet değeri 7.inci yinelemeden sonra sabitleşmektedir.

Tablo 4.7. Vücut yağı veri seti üzerinde yapılan parçacık sürüsü optimizasyonu algoritması ile özellik seçimi değerleri

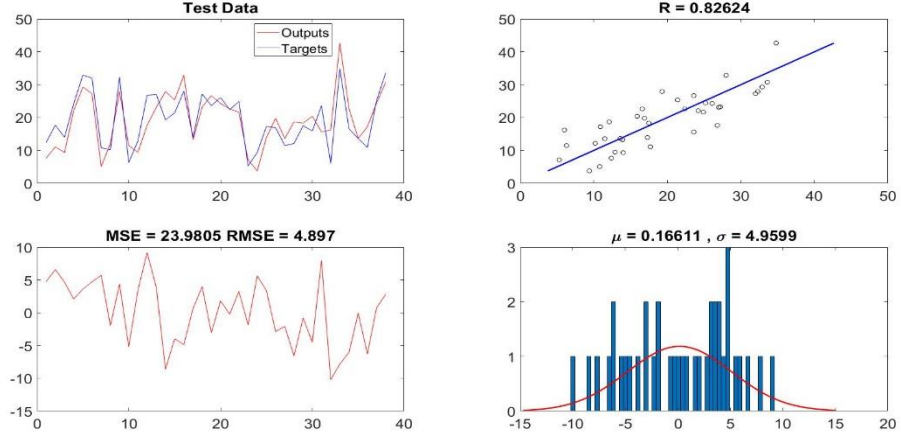
	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	16.9658	4.119	0.31888	4.1617	0.85231
Test Data	15	25.147	5.0147	-0.65249	5.0388	0.79592
Train Data	70	11.565	3.4007	-0.27463	3.3993	0.91658
All Data	100	14.4275	3.7984	-0.24211	3.7982	<b>0.89124</b>

Tablo 4.7’de görüldüğü gibi tüm veriler için korelasyon değeri 0.89124 olarak bulundu. Daha önceden özellik sayısı 8 olarak belirlenen parçacık sürüsü optimizasyonu algoritması ile özellik seçiminin değerleri, yapay sinir ağı yönteminin değerleri ile yakın ilişkileri olduğu görülmektedir.

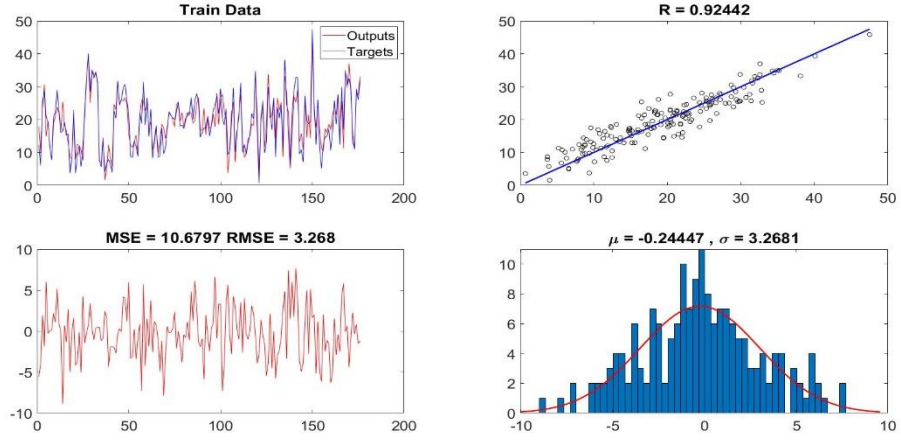
#### 4.2.4.4. Yapay Sinir Ağı ve Diferansiyel Evrim Algoritmasını Kullanarak Özellik Seçimi Yapmak



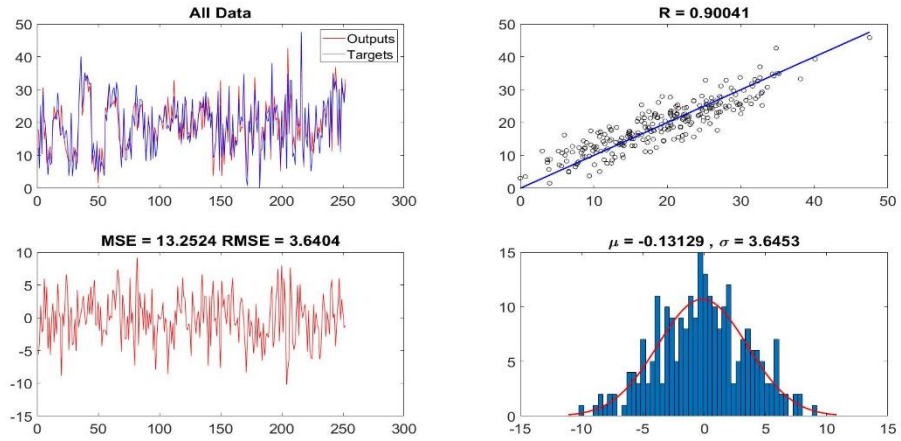
Şekil 4.33. Diferansiyel evrim algoritması ile özellik seçimi, validation (doğrulama) grafiği



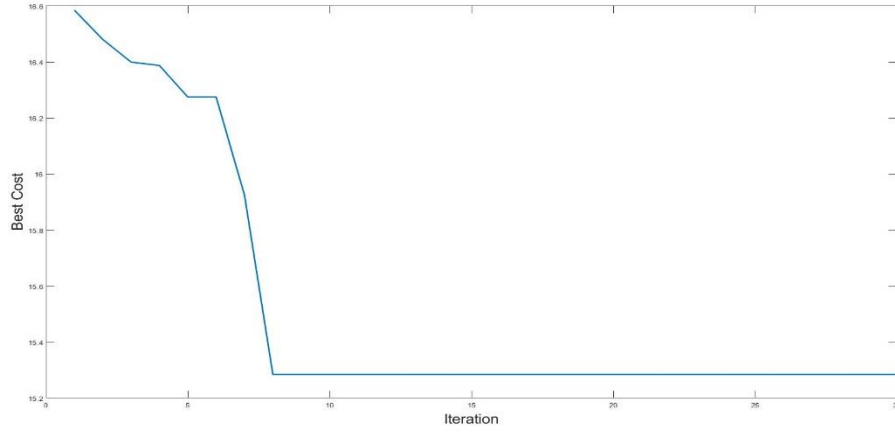
Şekil 4.34. Diferansiyel evrim algoritması ile özellik seçimi, test grafiği



Şekil 4.35. Diferansiyel evrim algoritması ile özellik seçimi, train (eğitim) grafiği



Şekil 4.36. Diferansiyel evrim algoritması ile özellik seçimi, tüm veriler grafiği



Şekil 4.37. Diferansiyel evrim algoritması ile özellik seçimi, her yinelemede maliyet grafiği

Diferansiyel evrim algoritması ile özellik seçimi, 30 yinelemede çalışmaktadır ve her yinelemenin değeri kaydedilmektedir. Şekil 4.37’de görüldüğü gibi maliyet değeri 8.inci yinelemeden sonra sabitleşmektedir.

Tablo 4.8. Vücut yağı veri seti üzerinde yapılan diferansiyel evrim algoritması ile özellik seçimi değerleri

	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	14.4398	3.8	0.095529	3.8498	0.86111
Test Data	15	23.9805	4.897	0.16611	4.9599	0.82624
Train Data	70	10.6797	3.268	-0.24447	3.2681	0.92442
All Data	100	13.2524	3.6404	-0.13129	3.6453	<b>0.90041</b>

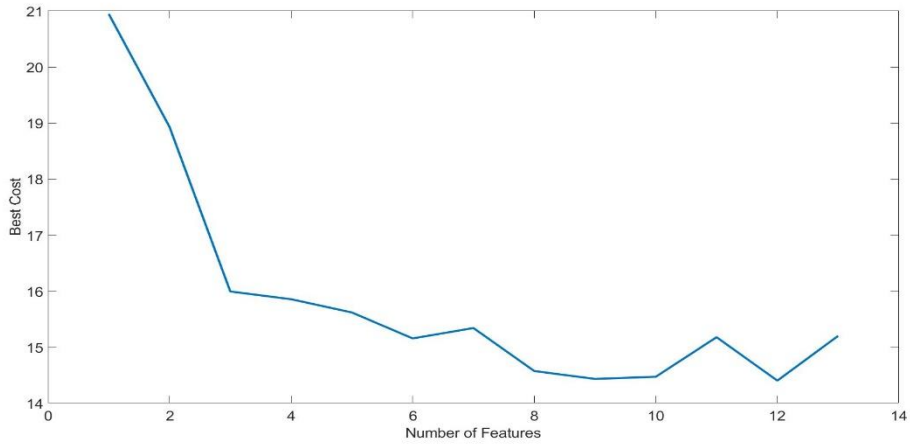
Tablo 4.7’de görüldüğü gibi tüm veriler için korelasyon değeri 0.90041 olarak bulundu. Daha önceden özellik sayısı 8 olarak belirlenen diferansiyel evrim algoritması ile özellik seçiminin değerleri, yapay sinir ağı yönteminin değerlerinden daha iyi sonuç verdiği ve birden çok kez çalıştırılmalı yapay sinir ağı yönteminin sonuçlarıyla da yakın olduğu görülmektedir.

#### 4.2.5. Çok Amaçlı Özellik Seçimi Yapmak

Bu bölümde ilk olarak belirli problem, birden çok tek amaçlı probleme ayrıştırılır ve birden çok çalıştırma ile parçacık sürüsü optimizasyonu algoritması

kullanarak, çok amaçlı bir özellik seçimi yapılmaktadır. Daha sonrasında ise evrimsel çok amaçlı yaklaşımları ile NSGA II algoritması kullanarak, çok amaçlı özellik seçimi yapılmaktadır.

#### 4.2.5.1. Birden Çok Çalıştırma ile Parçacık Sürüsü Optimizasyonu Algoritması Kullanarak Özellik Seçimi Yapmak



Şekil 4.38. Birden çok çalıştırma ile parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet grafiği

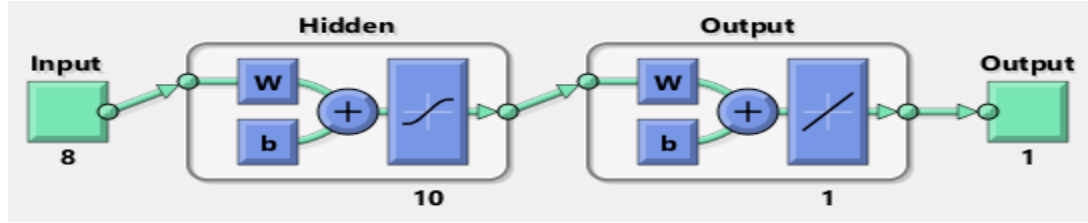
Tablo 4.9. Birden çok çalıştırma ile parçacık sürüsü optimizasyonu algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet değeri

Özellik Sayısı	Maliyet	Özellik Sayısı	Maliyet	Özellik Sayısı	Maliyet
<b>1 özellik</b>	<b>20.9471</b>	6 özellik	15.1539	11 özellik	15.1757
2 özellik	18.9293	7 özellik	15.3392	<b>12 özellik</b>	<b>14.4004</b>
3 özellik	15.9918	<b>8 özellik</b>	<b>14.5722</b>	13 özellik	15.1994
4 özellik	15.8537	9 özellik	14.4304		
5 özellik	15.6172	10 özellik	14.4695		

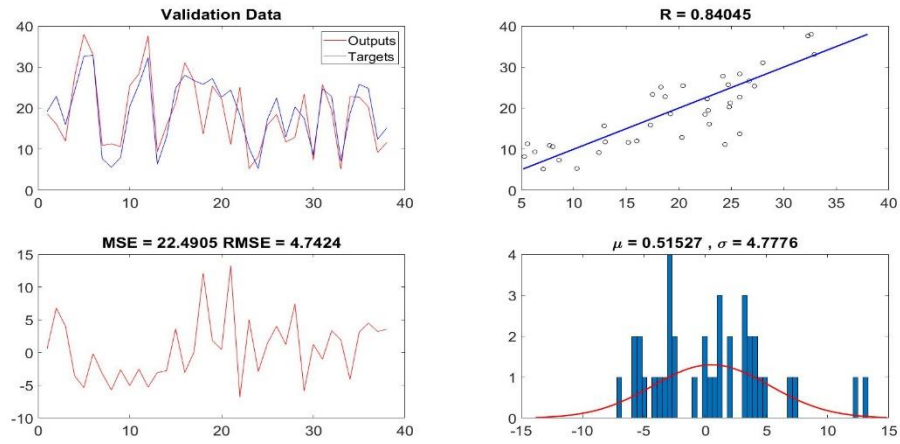
Tablo 4.9'da görüldüğü gibi, özellik sayılarının en iyi maliyet değeri hesaplanmıştır. Değerlere bakıldığında 1 özellekle çalıştığında daha maliyetli (20.9471) olduğu ve 12 özellekle çalıştığında en az maliyet (14.004) değeri olduğu

görülmektedir. Ancak özellik sayısı 8 olarak belirlenirse, hem özellik sayısı azalmış olur, hem de maliyet değeri (14.5722) düşer.

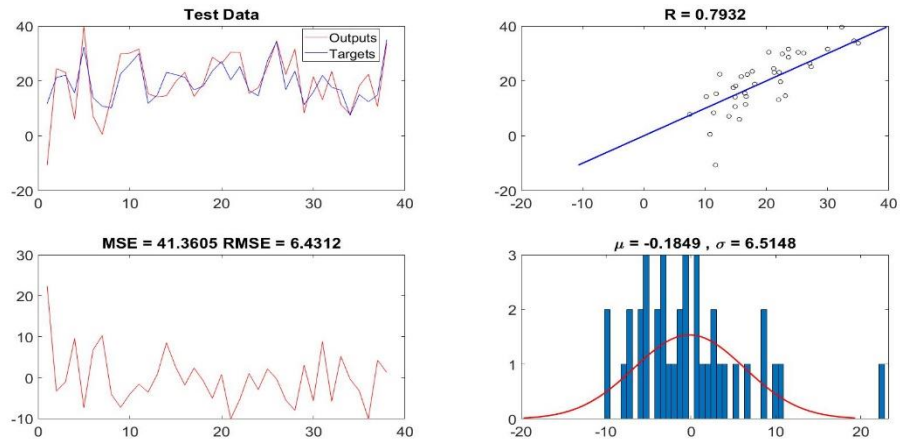
#### 4.2.5.2. Evrimsel Çok Amaçlı Yaklaşımları ile NSGA II Algoritması Kullanarak Çok Amaçlı Özellik Seçimi Yapmak



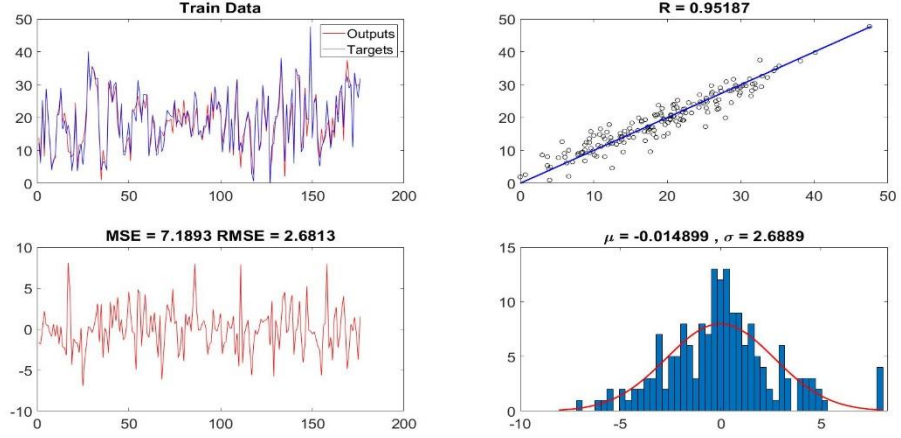
Şekil 4.39. Çok amaçlı yaklaşımlar ile NSGA II algoritması kullanarak özellik seçimi, yapay sinir ağı diyagramı



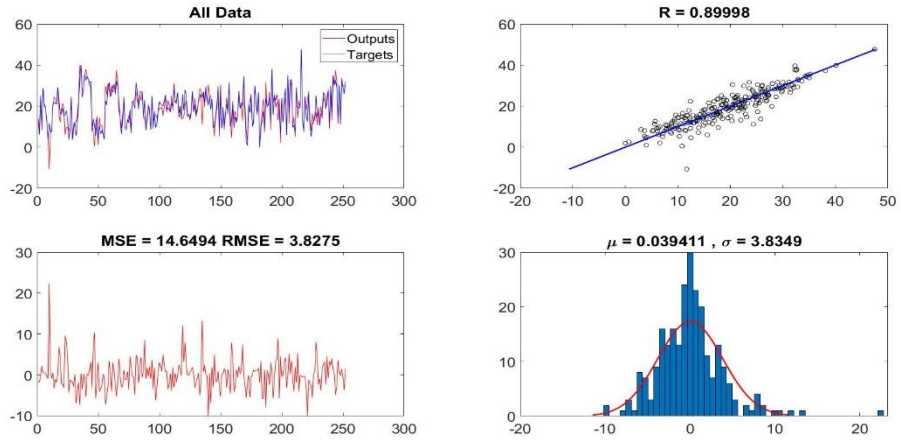
Şekil 4.40. NSGA II algoritması ile özellik seçimi, validation (doğrulama) grafiği



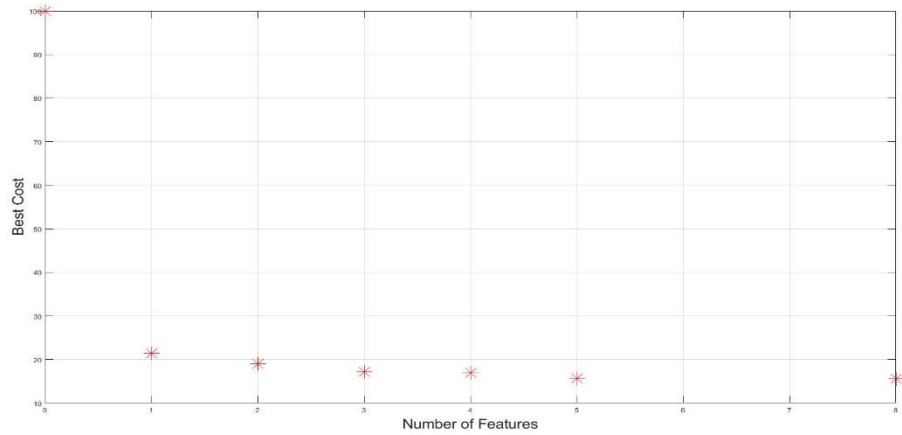
Şekil 4.41. NSGA II algoritması ile özellik seçimi, test grafiği



Şekil 4.42. NSGA II algoritması ile özellik seçimi, train (eğitim) grafiği



Şekil 4.43. NSGA II algoritması ile özellik seçimi, tüm veriler grafiği



Şekil 4.44. NSGA II algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet grafiği

Çok amaçlı NSGA II algoritması ile özellik seçimi, 30 yinelemede çalışmaktadır ve her yinelemenin en iyi maliyet değeri kaydedilmektedir. Şekil 4.44'te görüldüğü gibi en iyi maliyet değeri 8 özellekle daha iyi bir performans gösterdiği görülmektedir.

Tablo 4.10. NSGA II algoritması ile özellik seçimi, özellik sayılarının en iyi maliyet değeri

	Veri(%)	MSE	RMSE	$\mu$	$\sigma$	R
Validation Data	15	22.4905	4.7424	0.51527	4.7776	0.84045
Test Data	15	41.3605	6.4312	-0.1849	6.5148	0.7932
Train Data	70	7.1893	2.6813	-0.01489	2.6889	0.95187
All Data	100	14.6494	3.8275	0.039411	3.8349	<b>0.89998</b>

Tablo 4.10'da görüldüğü gibi tüm veriler için korelasyon değeri 0.89998 olarak bulundu. Çok amaçlı NSGA II algoritması ile özellik seçiminin değerleri, yapay sinir ağı yönteminin değerlerinden daha iyi sonuç verdiği ve birden çok kez çalıştırılmalı yapay sinir ağı yönteminin sonuçlarıyla da hemen hemen aynı olduğu görülmektedir.

## 5. SONUÇ

Günümüzde verilerle uğraşmak için özellik seçme yöntemleri, öğrenme sürecinin ayrılmaz bir parçası haline gelmiştir. Son yıllarda, birkaç büyük veri seti internette araştırmacıların kullanımına açılmıştır. Makine öğrenimi algoritmalarının büyük hacimli iş girdisi özellikleriyle uğraşması zor olduğundan dolayı, bu araştırma topluluğu için ilginç bir zorluk teşkil ediyordu. Dolayısıyla özellik seçme yöntemleri, veri madenciliğinde daha önemli rol oynamaya başladı ve bu zorluğu ortadan kaldırmak ya da daha hafifletmek için özellik indirgeme yöntemleri daha popüler hale geldi.

Bu tezde ise vücut yağını tahmin etmek için 13 özellikli bir veri seti kullanıldı. Önce normal bir yapay sinir ağı ile tüm özellikleri kullanarak korelasyon değeri 0.89028 olarak bulundu. Daha sonrasında birden çok kez çalıştırılmalı yapay sinir ağı ile tüm özellikleri kullanarak korelasyon değeri 0.90355 olarak bulundu. Bu iki değer kıyaslandığında, birden çok kez çalıştırılmalı yapay sinir ağının daha iyi bir performans gösterdiği görülmektedir.

Yapay sinir ağı ile bulunan değerlerde tüm özellikler kullanıldı; ancak daha az özellik ve daha az maliyetle aynı performansa ya da daha yakın bir sonuca ulaşmak için özellik seçme yöntemleri kullanılır. Bu çalışmada optimum bir şekilde özellik seçimi yapılması için 6 farklı metasezgisel algoritması (ikili genetik algoritması, karınca koloni optimizasyonu algoritması, benzetimli tavlama algoritması, parçacık sürüsü optimizasyonu algoritması, diferansiyel evrim algoritması, baskın olmayan sıralamalı genetik algoritması NSGA II) kullanıldı.

Çalışmada 3 farklı özellik seçme yöntemi uygulandı. Birinci yöntemde seçili özellik sayısını bilmeden, 25 yinelemeli ikili genetik algoritması ile özellik seçimi yapıldı. Bu yöntemin sonucunda, 6 özellikle korelasyon değeri 0.88629 olarak bulundu. Bu değer yapay sinir ağları ile bulunan değerler ile kıyaslandığında, yakın olduğu görülmektedir.

İkinci yöntemde daha önceden 8 olarak belirlenmiş özellik sayısını bilerek, özellik seçimi yapıldı. Ayrık kodlama için önce 30 yinelemeli karınca koloni optimizasyonu algoritması kullanıldı ve korelasyon değeri 0.89347 olarak bulundu. Daha sonrasında ise 30 yinelemeli benzetimli tavlama algoritması kullanıldı ve korelasyon değeri 0.89279 olarak bulundu. Sürekli kodlama için ise önce 30

yinelemeli parçacık sürüsü optimizasyonu algoritması kullanıldı ve korelasyon değeri 0.89124 olarak bulundu. Daha sonrasında ise 30 yinelemeli diferansiyel evrim algoritması kullanıldı ve korelasyon değeri 0.90041 olarak bulundu. Bu değerler yapay sinir ağı yöntemlerinin değerleri ile kıyaslandığında diferansiyel evrim algoritmasının daha yakın bir performans gösterdiği görülmektedir.

Üçüncü yöntemde ise çok amaçlı özellik seçimi yapıldı. İlk olarak bilinen problem, tek amaçlı problemlere ayrıştırılarak, 15 yinelemeli ve birden çok kez çalıştırılmalı parçacık sürüsü optimizasyonu algoritması ile özellik seçimi yapıldı ve hem maliyet açısından hem de daha iyi bir performansa sahip olmasından dolayı 8 özellikli bir sistem seçildi. Daha sonrasında ise 30 yinelemeli baskın olmayan sıralamalı genetik algoritması(NSGA II) kullanıldı ve 8 özellikli bir sistemde korelasyon değeri 0.89998 olarak bulundu. Bu değer yapay sinir ağı yöntemlerinin değerleri ile kıyaslandığında daha az bir özellikli hemen hemen aynı bir performansa ulaşmayı gösterdi.

Bulunan değerlere bakıldığında, veri setlerinin tüm özelliklerini kullanarak en iyi sonuca ulaşmanın iyi bir yöntem olmadığı görülmektedir. Bu veri setinde birden çok kez çalıştırılmalı parçacık sürüsü optimizasyonu algoritması ile yapılan özellik seçiminde, 13 özelliğin en iyi maliyet değeri 15.1994'iken 8 özelliğin en iyi maliyet değeri 14.5722 olarak bulundu. Dolayısıyla her zaman tüm özellikleri kullanmak iyi bir fikir olmaya bilir. Ayrıca zaman ve maliyet açısından yapılan araştırmayıda kötü etkileyebilir.

Sonuç olarak bu tezde metasezgiseller arasında hiçbir kıyaslama yapılmadığının altını çizmek isterim; çünkü hiçbir metasezgisel algoritması birbirinden üstün değildir. Sonuçlarımızı etkileyen en önemli faktörün veri seti olduğunu ve daha sonrasında her metasezgisel algoritmasının kendine ait ayarlama parametreleri olduğudur. Dolayısıyla bu çalışmada sadece özellik seçiminin günümüzde önemli rol oynadığını ve eğer veri setlerimizde uygulanırsa daha az bir zamanda, az bir maliyetle, aynı performansı veya beklediğimiz performansa daha yakın bir sonuç alacağımıza dair bir çalışma yapılmıştır.

## KAYNAKÇA

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P. ve Saeys, Y. 2010. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26:3, 392-398.
- Abusamra, H. 2013. A comparative study of feature selection and classification methods for gene expression data of glioma. *Procedia Computer Science*, 23, 5-14.
- Afonso, M. V., Bioucas-Dias, J. M. ve Figueiredo, M. A. 2010. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20:3, 681-695.
- Aggarwal, C. C. ve Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 70-81.
- Aghdam, M. H., Ghasem-Aghaee, N. ve Basiri, M. E. 2009. Text feature selection using ant colony optimization. *Expert systems with applications*, 36:3, 6843-6853.
- Agresti, A. ve Agresti, B. F. 1970. Statistical Methods for the. *Social Sciences*. CA: Dellen Publishers.
- Ahmad, F. K., Norwawi, N. M., Deris, S. ve Othman, N. H. (2008). A review of feature selection techniques via gene expression profiles. 2008 International Symposium on Information Technology, IEEE, 1-7.
- Ahmed, S., Zhang, M. ve Peng, L. 2014. Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming. *Connection Science*, 26:3, 215-243.
- Akbari, R. ve Ziarati, K. 2011. A multilevel evolutionary algorithm for optimizing numerical functions. *International Journal of Industrial Engineering Computations*, 2:2, 419-430.
- Alelyani, S., Tang, J. ve Liu, H. 2018. *Data Clustering*. Chapman and Hall/CRC, 29-60.
- Ali Khan, S., Hussain, A., Basit, A. ve Akram, S. 2014. Kruskal-Wallis-based computationally efficient feature selection for face recognition. *The Scientific World Journal*, 2014.
- Ali, S. I. ve Shahzad, W. (2012). A feature subset selection method based on symmetric uncertainty and ant colony optimization. 2012 International Conference on Emerging Technologies, IEEE, 1-6.
- Aliferis, C. F., Tsamardinos, I. ve Statnikov, A. (2003). HITON: a novel Markov Blanket algorithm for optimal variable selection. AMIA annual symposium proceedings, American Medical Informatics Association, 21.
- Almuallim, H. ve Dietterich, T. G. (1991). Learning with Many Irrelevant Features. *AAAI, Citeseer*, 547-552.
- Almufti, S., Asaad, R., Salim, B. ve Technology 2018. Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems. *International Journal of Engineering*, 7, 6109-6114.
- Almufti, S. M. 2015. U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem. Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ)
- Almufti, S. M. 2019. Historical survey on metaheuristics algorithms. *International Journal Of Scientific World*, 7:1, 1.

- Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A. ve Yazdani, N. 2011. Mutual information-based feature selection for intrusion detection systems. *Journal of Network Computer Applications*, 34:4, 1184-1199.
- Andrew, G., Arora, R., Bilmes, J. ve Livescu, K. (2013). Deep canonical correlation analysis. *International conference on machine learning*, 1247-1255.
- Ang, J. C., Mirzal, A., Haron, H. ve Hamed, H. N. A. 2015. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM transactions on computational biology bioinformatics*, 13:5, 971-989.
- Antonioni, A. ve Lu, W.-S. 2007. *Practical optimization: algorithms and engineering applications*. Springer Science & Business Media,
- Arauzo-Azofra, A., Benitez, J. M. ve Castro, J. L. (2004). A feature set measure based on relief. *Proceedings of the fifth international conference on Recent Advances in Soft Computing*, 104-109.
- Arora, R. ve Livescu, K. (2013). Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 7135-7139.
- Auger, A., Schoenauer, M. ve Teytaud, O. (2005). Local and global order 3/2 convergence of a surrogate evolutionary algorithm. *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 857-864.
- Babatunde, O. H., Armstrong, L., Leng, J. ve Diepeveen, D. 2014a. A genetic algorithm-based feature selection. *International Journal of Electronics Communication and Computer Engineering*.
- Babatunde, O. H., Armstrong, L., Leng, J. ve Diepeveen, D. 2014b. Zernike moments and genetic algorithm: Tutorial and application.
- Bach, F. R. (2008). Bolasso: model consistent lasso estimation through the bootstrap. *Proceedings of the 25th international conference on Machine learning*, 33-40.
- Bachmann, P. 1894. *Die analytische zahlentheorie*. Teubner,
- Bae, C., Yeh, W.-C., Wahid, N., Chung, Y. Y. ve Liu, Y. 2012. A new simplified swarm optimization (SSO) using exchange local search scheme. *International Journal of Innovative Computing, Information Control*, 8:6, 4391-4406.
- Balaprakash, P., Birattari, M., Stützle, T., Yuan, Z. ve Dorigo, M. 2009. Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem. *Swarm Intelligence*, 3:3, 223-242.
- Battiti, R. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5:4, 537-550.
- Beasley, D., Bull, D. R. ve Martin, R. R. 1993. An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15:2, 56-69.
- Beheshti, Z. ve Shamsuddin, S. M. H. 2013. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, 5:1, 1-35.
- Belkin, M. ve Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 585-591.
- Bellman, R. 1957. *Dynamic programming*, princeton, nj: Princeton univ. *Versity Press. Bellman Dynamic Programming*.
- Benesty, J., Chen, J., Huang, Y. ve Cohen, I. 2009. *Noise reduction in speech processing*. Springer, 1-4.

- Beniwal, S., Arora, J. ve technology 2012. Classification and feature selection techniques in data mining. *International journal of engineering research*, 1:6, 1-6.
- Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A. ve Restelli, M. (2019). Feature Selection via Mutual Information: New Theoretical Insights. 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 1-9.
- Bhavani, S. D., Rani, T. S. ve Bapi, R. S. 2008. Feature selection using correlation fractal dimension: Issues and applications in binary classification problems. *Applied Soft Computing*, 8:1, 555-563.
- Bins, J. ve Draper, B. A. (2001). Feature selection from huge feature sets. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, IEEE, 159-165.
- Blum, C. ve Roli, A. J. A. c. s. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. 35:3, 268-308.
- Bø, T. H. ve Jonassen, I. 2002. New feature subset selection procedures for classification of expression profiles. *Genome biology*, 3:4, research0017. 0011.
- Bodenhofer, U. 2003. Genetic algorithms: theory and applications: Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter.
- Bolón-Canedo, V., Sánchez-Marroño, N. ve Alonso-Betanzos, A. 2013. A review of feature selection methods on synthetic data. *Knowledge Information Systems*, 34:3, 483-519.
- Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M. ve Herrera, F. 2014. A review of microarray datasets and applied feature selection methods. *Information Sciences*, 282, 111-135.
- Bonacich, P. 1987. Power and centrality: A family of measures. *American journal of sociology*, 92:5, 1170-1182.
- Bontempi, G. ve Haibe-Kains, B. 2008. Feature selection methods for mining bioinformatics data. *Bruxelles, Belgium: ULB Machine Learning Group*.
- BoussaïD, I., Lepagnot, J. ve Siarry, P. 2013. A survey on optimization metaheuristics. *Information Sciences*, 237, 82-117.
- Boyd, S., Boyd, S. P. ve Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press,
- Bradley, P. S. ve Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. ICML, 82-90.
- Braga-Neto, U., Hashimoto, R., Dougherty, E. R., Nguyen, D. V. ve Carroll, R. J. 2004. Is cross-validation better than resubstitution for ranking genes? *Bioinformatics*, 20:2, 253-258.
- Brassard, G. ve Bratley, P. 1996. *Fundamentals of algorithmics*. Prentice Hall Englewood Cliffs,
- Brassard, G. ve Bratley, P. 2013. *Fundamentals of algorithmics*. PHI Learning, Delhi.
- Brkić, K., Pinz, A., Šegvić, S. ve Kalafatić, Z. (2011). Histogram-based description of local space-time appearance. Scandinavian Conference on Image Analysis, Springer, 206-217.
- Bryant, F. B. ve Satorra, A. 2012. Principles and practice of scaled difference chi-square testing. *Structural Equation Modeling: A Multidisciplinary Journal*, 19:3, 372-398.
- Budak, H. ve Taşabat, S. E. 2016. A modified t-score for feature selection. *Anadolu Üniversitesi Bilim Ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik*, 17:5, 845-852.

- Burke, E. K. ve Kendall, G. 2005. *Search methodologies*. Springer,
- Cai, D., Zhang, C. ve He, X. (2010). Unsupervised feature selection for multi-cluster data. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 333-342.
- Carey, J. J. ve Delaney, M. F. 2010. T-scores and Z-scores. *Clinical reviews in bone mineral metabolism*, 8:3, 113-121.
- Carr, J. 2014. An introduction to genetic algorithms. *Senior Project*, 1:40, 7.
- Chan, P. K., Schlag, M. D. ve Zien, J. Y. 1994. Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on computer-aided design of integrated circuits systems*, 13:9, 1088-1096.
- Chan, Y.-H., Ng, W. W., Yeung, D. S. ve Chan, P. P. (2010). Empirical comparison of forward and backward search strategies in L-GEM based feature selection with RBFNN. 2010 International Conference on Machine Learning and Cybernetics, IEEE, 1524-1527.
- Chandra, B. ve Gupta, M. 2011. An efficient statistical feature selection approach for classification of gene expression data. *Journal of biomedical informatics*, 44:4, 529-535.
- Chandrashekar, G. ve Sahin, F. 2014. A survey on feature selection methods. *Computers Electrical Engineering*, 40:1, 16-28.
- Chang, Y., Bouzarkouna, Z. ve Devegowda, D. 2015. Multi-objective optimization for rapid and robust optimal oilfield development under geological uncertainty. *Computational Geosciences*, 19:4, 933-950.
- Chen, L.-F., Su, C.-T., Chen, K.-H. ve Wang, P.-C. 2012. Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis. *Neural Computing Applications*, 21:8, 2087-2096.
- Chen, L. ve Aihara, K. 1995. Chaotic simulated annealing by a neural network model with transient chaos. *Neural networks*, 8:6, 915-930.
- Chen, S. S., Donoho, D. L. ve Saunders, M. A. 2001. Atomic decomposition by basis pursuit. *SIAM review*, 43:1, 129-159.
- Chen, Y., Li, Y., Cheng, X.-Q. ve Guo, L. (2006). Survey and taxonomy of feature selection algorithms in intrusion detection system. International Conference on Information Security and Cryptology, Springer, 153-167.
- Chinneck, J. 2006. Practical optimization: a gentle introduction. *Carleton University*.
- Chiu, C.-C. ve Yao, Y. 2013. Multiway elastic net (MEN) for final product quality prediction and quality-related analysis of batch processes. *Chemometrics Intelligent Laboratory Systems*, 125, 153-165.
- Cho, B. H., Yu, H., Kim, K.-W., Kim, T. H., Kim, I. Y. ve Kim, S. I. 2008. Application of irregular and unbalanced data to predict diabetic nephropathy using visualization and feature selection methods. *Artificial intelligence in medicine*, 42:1, 37-53.
- Chung, F. R. ve Graham, F. C. 1997. *Spectral graph theory*. American Mathematical Soc.,
- Clark, P. ve Boswell, R. 2000. Practical machine learning tools and techniques with java implementation: Morgan Kaufmann Publisher.
- Clerc, M. ve Kennedy, J. 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6:1, 58-73.
- Coello, C. A. C., Lamont, G. B. ve Van Veldhuizen, D. A. 2007. *Evolutionary algorithms for solving multi-objective problems*. Springer,

- Coello, C. A. C., Pulido, G. T. ve Lechuga, M. S. 2004. Handling multiple objectives with particle swarm optimization. *IEEE transactions on Evolutionary Computation*, 8:3, 256-279.
- Colorni, A., Dorigo, M., Maniezzo, V. ve Trubian, M. 1994. Ant system for job-shop scheduling. *JORBEL-Belgian Journal of Operations Research, Statistics, Computer Science*, 34:1, 39-53.
- Combettes, P. L. ve Wajs, V. R. 2005. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling Simulation*, 4:4, 1168-1200.
- Cover, T. M. ve Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons,
- Črepinšek, M., Liu, S.-H. ve Mernik, M. 2013. Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys*, 45:3, 1-33.
- Cui, L., Bai, L., Zhang, Z., Wang, Y. ve Hancock, E. R. 2019. Identifying the most informative features using a structurally interacting elastic net. *Neurocomputing*, 336, 13-26.
- da Silva, P. N., Plastino, A. ve Freitas, A. A. (2018). A novel genetic algorithm for feature selection in hierarchical feature spaces. Proceedings of the 2018 siam international conference on data mining, SIAM, 738-746.
- Damak, N., Jarboui, B. ve Loukil, T. (2013). Non-dominated Sorting Genetic Algorithm-II to solve bi-objective multi-mode resource-constrained project scheduling problem. 2013 International Conference on Control, Decision and Information Technologies (CoDIT), IEEE, 842-846.
- Daniel, W. W. 1990. Kruskal–Wallis one-way analysis of variance by ranks. *Applied nonparametric statistics*, 226-234.
- Danilov, V. V., Skirnevskiy, I. P., Manakov, R. A., Gerget, O. M. ve Melgani, F. 2020. Feature selection algorithm based on PDF/PMF area difference. *Biomedical Signal Processing Control*, 57, 101681.
- Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M. ve Basu, D. K. 2012. A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. *Applied Soft Computing*, 12:5, 1592-1606.
- Das, S. ve Suganthan, P. N. 2010. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on Evolutionary Computation*, 15:1, 4-31.
- Dasgupta, A., Drineas, P., Harb, B., Josifovski, V. ve Mahoney, M. W. (2007). Feature selection methods for text classification. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 230-239.
- Dash, M. ve Liu, H. 1997. Feature selection for classification. *Intelligent data analysis*, 1:3, 131-156.
- Dash, M. ve Liu, H. 2003. Consistency-based search in feature selection. *Artificial intelligence*, 151:1-2, 155-176.
- Daubechies, I., Defrise, M. ve De Mol, C. 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57:11, 1413-1457.
- Davis, C. A., Gerick, F., Hintermair, V., Friedel, C. C., Fundel, K., Küffner, R. ve Zimmer, R. 2006. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics*, 22:19, 2356-2363.
- Deb, K. 2001. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons,

- Deb, K., Agrawal, S., Pratap, A. ve Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *International conference on parallel problem solving from nature*, Springer, 849-858.
- Deb, K., Pratap, A., Agarwal, S. ve Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on Evolutionary Computation*, 6:2, 182-197.
- Delacre, M., Lakens, D. ve Leys, C. 2017. Why psychologists should by default use Welch's t-test instead of Student's t-test. *International Review of Social Psychology*, 30:1.
- Desboulets, L. D. D. 2018. A review on variable selection in regression analysis. *Econometrics*, 6:4, 45.
- Devijver, P. A. ve Kittler, J. 1982. *Pattern recognition: A statistical approach*. Prentice hall,
- Dey, N., Borra, S., Ashour, A. S. ve Shi, F. 2018. *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*. Academic Press,
- Ding, C., Peng, H. J. J. o. b. ve biology, c. 2005. Minimum redundancy feature selection from microarray gene expression data. 3:02, 185-205.
- Doak, J. 1992. An evaluation of feature selection methods and their application to computer security. *Techninal Report CSE*.
- Doquire, G. ve Verleysen, M. (2011). Feature selection with mutual information for uncertain data. *International Conference on Data Warehousing and Knowledge Discovery*, Springer, 330-341.
- Doquire, G. ve Verleysen, M. 2013. A graph Laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing*, 121, 5-13.
- Dorigo, M. 1992. Optimization, learning and natural algorithms. *Politecnico di Milano*.
- Dorigo, M. ve Blum, C. 2005. Ant colony optimization theory: A survey. *Theoretical computer science*, 344:2-3, 243-278.
- Dorigo, M., Caro, G. D. ve Gambardella, L. M. 1999. Ant algorithms for discrete optimization. *Artificial life*, 5:2, 137-172.
- Dorigo, M., Colorni, A. ve Maniezzo, V. 1991. Distributed optimization by ant colonies.
- Dorigo, M., Maniezzo, V. ve Colorni, A. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, Cybernetics, Part B*, 26:1, 29-41.
- Du, L. ve Shen, Y.-D. (2015). Unsupervised feature selection with adaptive structure learning. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 209-218.
- Duda, R. O., Hart, P. E. ve Stork, D. G. 2012. *Pattern classification*. John Wiley & Sons,
- Durgabai, R. ve Bhushan, Y. R. 2014. Feature selection using ReliefF algorithm. *International Journal of Advanced Research in Computer Communication Engineering*, 3:10, 8215-8218.
- Dy, J. G. ve Brodley, C. E. 2004. Feature selection for unsupervised learning. *Journal of machine learning research*, 5:Aug, 845-889.
- Eberhart, R. 1996. Computational intelligence. *Denise E. M. Penrose*.
- Eberhart, R. ve Kennedy, J. (1995). A new optimizer using particle swarm theory. MHS'95. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Ieee, 39-43.

- Eberhart, R. C. ve Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512), IEEE, 84-88.
- Efron, B., Hastie, T., Johnstone, I. ve Tibshirani, R. 2004. Least angle regression. *The Annals of statistics*, 32:2, 407-499.
- Efthimeros, G., Diamantis, Z., Photeinos, D. ve Tsahalis, D. 2000. Objective functions for optimization problems in the process industry. *Laboratory of Fluid Mechanics and Energy, Department of Chemical Engineering, University of Patras*.
- Eiben, A. E. ve Smith, J. E. 2015. *Introduction to evolutionary computing*. Springer,
- El-Manzalawy, Y. (2018). CCA based multi-view feature selection for multi-omics data integration. 2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), IEEE, 1-8.
- El-Shorbagy, M. ve Hassanien, A. E. J. I. J. o. R. S. 2018. Particle swarm optimization from theory to applications. *International Journal of Rough Sets Data Analysis*, 5:2, 1-24.
- Engelbrecht, A. P. 2007. *Computational intelligence: an introduction*. John Wiley & Sons,
- Esmael, B., Arnaout, A., Fruhwirth, R. K. ve Thonhauser, G. 2015. A statistical feature-based approach for operations recognition in drilling time series. *International Journal of Computer Information Systems Industrial Management Applications*, 5, 454-461.
- Fan, M., Chang, X. ve Tao, D. (2017). Structure regularized unsupervised discriminant feature analysis. Thirty-First AAAI Conference on Artificial Intelligence.
- Farahat, A. K., Ghodsi, A. ve Kamel, M. S. (2011). An efficient greedy method for unsupervised feature selection. 2011 IEEE 11th International Conference on Data Mining, IEEE, 161-170.
- Faulkner, K. G. 2005. The tale of the T-score: review and perspective: Springer.
- Feldotto, M. ve Graffi, K. (2013). Comparative evaluation of peer-to-peer systems using PeerfactSim. KOM. 2013 International Conference on High Performance Computing & Simulation (HPCS), IEEE, 99-106.
- Feo, T. A. ve Resende, M. G. 1989. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8:2, 67-71.
- Ferri, F., Pudil, P., Hatef, M. ve Kittler, J. 1994. *Machine Intelligence and Pattern Recognition*. Elsevier, 403-413.
- Finger, M., Le Bras, R., Gomes, C. P. ve Selman, B. (2013). Solutions for hard and soft constraints using optimized probabilistic satisfiability. International Conference on Theory and Applications of Satisfiability Testing, Springer, 233-249.
- Fonti, V. ve Belitser, E. 2017. Feature selection using lasso. *VU Amsterdam Research Paper in Business Analytics*, 30, 1-25.
- Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3:Mar, 1289-1305.
- Fraley, C. ve Hesterberg, T. 2009. Least angle regression and LASSO for large datasets. *Statistical Analysis Data Mining: The ASA Data Science Journal*, 1:4, 251-259.
- François, D., Rossi, F., Wertz, V. ve Verleysen, M. 2007. Resampling methods for parameter-free and robust feature selection with mutual information. *Neurocomputing*, 70:7-9, 1276-1288.
- Friedman, J., Hastie, T. ve Tibshirani, R. 2001. *The elements of statistical learning*. Springer series in statistics New York,

- Friedman, J., Hastie, T. ve Tibshirani, R. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33:1, 1.
- Fukunaga, K. 2013. *Introduction to statistical pattern recognition*. Elsevier,
- Gambardella, L. M. ve Dorigo, M. 2000. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12:3, 237-255.
- Gandomi, A. H., Yang, X.-S., Talatahari, S. ve Alavi, A. H. 2013. *Metaheuristic applications in structures infrastructures*, 1-24.
- Ganster, H., Pinz, P., Rohrer, R., Wildling, E., Binder, M. ve Kittler, H. 2001. Automated melanoma recognition. *IEEE transactions on medical imaging*, 20:3, 233-239.
- Garson, G. D. 2004. Nominal association: Phi, contingency coefficient, Tschuprow's T, Cramer's V, lambda, uncertainty coefficient. Retrieved November, 16, 2004.
- George, G. ve Raj, V. C. 2011. Review on feature selection techniques and the impact of SVM for cancer classification using gene expression profile. *International Journal of Computer Science & Engineering Survey*.
- Gheyas, I. A. ve Smith, L. S. 2010. Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43:1, 5-13.
- Ghosh, A., Datta, A. ve Ghosh, S. 2013. Self-adaptive differential evolution for feature selection in hyperspectral image data. *Applied Soft Computing*, 13:4, 1969-1977.
- Ghosh, D. ve Chinnaiyan, A. M. 2005. Classification and selection of biomarkers in genomic data using LASSO. *Journal of Biomedicine Biotechnology*, 2005:2, 147.
- Ginsberg, M. 2012. *Essentials of artificial intelligence*. Newnes,
- Gliozzo, J. 2017. Network-based methods for outcome prediction in the "sample space".
- Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers operations research*, 13:5, 533-549.
- Glover, F. 1989. Tabu search—part I. *ORSA Journal on computing*, 1:3, 190-206.
- Glover, F. W. ve Kochenberger, G. A. 2006. *Handbook of metaheuristics*. Springer Science & Business Media,
- Goldberg, D. E. 1989a. Genetic algorithms in search. *Evolutionary Computation Research*.
- Goldberg, D. E. 1989b. others, Genetic algorithms in search, optimization, and machine learning, vol. 412: Addison-wesley Reading Menlo Park.
- Goldschmidt, P. S. 2006. Compliance monitoring for anomaly detection: Google Patents.
- Goswami, S. ve Chakrabarti, A. 2014. Feature selection: A practitioner view. *International Journal of Information Technology Computer Science*, 6:11, 66.
- Gu, Q., Li, Z. ve Han, J. (2012). Generalized fisher score for feature selection. Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence.
- Gunes, H. ve Schuller, B. 2013. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image Vision Computing*, 31:2, 120-136.
- Gupta, A. ve Begum, S. A. 2019. Efficient multi-cluster feature selection on text data. *Journal of Information Optimization Sciences*, 40:8, 1583-1598.
- Guyon, I. ve Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of machine learning research*, 3:Mar, 1157-1182.
- Hall, M. A. 1999. Correlation-based feature selection for machine learning.

- Hall, M. A. ve Holmes, G. 2000. Benchmarking attribute selection techniques for data mining. *IEEE Transactions on Knowledge and Data Engineering*.
- Hall, M. A. ve Smith, L. A. (1998). Practical feature subset selection for machine learning. Computer Science '98 Proceedings of the 21st Australasian Computer Science Conference ACSC'98, Perth, 4-6 February, 1998(pp. 181-191). Berlin.
- Han, J., Pei, J. ve Kamber, M. 2011. *Data mining: concepts and techniques*. Elsevier,
- Hara, S. ve Maehara, T. (2017). Enumerate lasso solutions for feature selection. Thirty-First AAAI Conference on Artificial Intelligence.
- Hardoon, D. R., Szedmak, S. ve Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16:12, 2639-2664.
- Hattori, L. T., Lopes, H. S. ve Lopes, F. M. 2016. Evolutionary computation and swarm intelligence for the inference of gene regulatory networks. *International Journal of Innovative Computing Applications*, 7:4, 225-235.
- Haupt, R. L. ve Ellen Haupt, S. 2004. *Practical genetic algorithms*,
- Haury, A.-C., Gestraud, P. ve Vert, J.-P. 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, 6:12.
- Hauskrecht, M., Pelikan, R., Valko, M. ve Lyons-Weiler, J. 2007. *Fundamentals of data mining in genomics and proteomics*. Springer, 149-172.
- He, R., Tan, T., Wang, L. ve Zheng, W.-S. (2012).  $l_2, l_1$  regularized correntropy for robust feature selection. 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2504-2511.
- He, X., Cai, D. ve Niyogi, P. (2006). Laplacian score for feature selection. Advances in neural information processing systems, 507-514.
- He, X. ve Niyogi, P. (2004). Locality preserving projections. Advances in neural information processing systems, 153-160.
- He, Z. ve Yu, W. 2010. Stable feature selection for biomarker discovery. *Computational biology chemistry*, 34:4, 215-225.
- Henderson, D., Jacobson, S. H. ve Johnson, A. W. 2003. *Handbook of metaheuristics*. Springer, 287-319.
- Hernández-Torruco, J., Canul-Reich, J., Frausto-Solís, J. ve Méndez-Castillo, J. J. 2014. Feature selection for better identification of subtypes of Guillain-Barré syndrome. *Computational mathematical methods in medicine*, 2014.
- Hettigoda, S. 2016. Computation of Least Angle Regression coefficient profiles and LASSO estimates. University of Louisville Mathematics
- Holland, J. H. 1992a. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press,
- Holland, J. H. 1992b. Genetic algorithms. *Scientific american*, 267:1, 66-73.
- Holmes, G. ve Nevill-Manning, C. G. 1995. Feature selection via the discovery of simple classification rules. *Computer Science*.
- Holte, R. C. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine Learning for Computer Cyber Security*, 11:1, 63-90.
- Hoque, N., Bhattacharyya, D. K. ve Kalita, J. K. 2014. MIFS-ND: A mutual information-based feature selection method. *Expert systems with applications*, 41:14, 6371-6385.
- Hotelling, H. 1992. *Breakthroughs in statistics*. Springer, 162-190.

- Hou, C., Nie, F., Li, X., Yi, D. ve Wu, Y. 2013. Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, 44:6, 793-804.
- Hou, C., Nie, F., Yi, D. ve Wu, Y. (2011). Feature selection via joint embedding learning and sparse regression. Twenty-Second international joint conference on Artificial Intelligence.
- Hua, J., Tembe, W. D. ve Dougherty, E. R. 2009. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42:3, 409-424.
- Huang, C.-J., Yang, D.-X. ve Chuang, Y.-T. 2008. Application of wrapper approach and composite classifier to the stock trend prediction. *Expert systems with applications*, 34:4, 2870-2878.
- Huang, D. ve Chow, T. W. 2005. Effective feature selection scheme using mutual information. *Neurocomputing*, 63, 325-343.
- Hussain, K., Salleh, M. N. M., Cheng, S. ve Shi, Y. 2019. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52:4, 2191-2233.
- Imran, M. ve Alsuhaibani, S. A. 2019. *Intelligent Data Analysis for Biomedical Applications*. Elsevier, 147-172.
- Inc, S. 2007. SPSS Clementine 12.0 Algorithms Guide. *SPSS Inc, Chicago*.
- Innocent, N. ve Kurian, M. 2013. Survey on semi supervised classification methods and Feature selection.
- Inza, I., Larrañaga, P., Blanco, R. ve Cerrolaza, A. J. 2004. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial intelligence in medicine*, 31:2, 91-103.
- Inza, I., Larrañaga, P., Etxeberria, R. ve Sierra, B. 2000. Feature subset selection by Bayesian network-based optimization. *Artificial intelligence*, 123:1-2, 157-184.
- Jain, A. K. ve Chandrasekaran, B. 1982. 39 Dimensionality and sample size considerations in pattern recognition practice. *Handbook of statistics*, 2, 835-855.
- Jia, X., Kuo, B.-C. ve Crawford, M. M. 2013. Feature mining for hyperspectral image classification. *Proceedings of the IEEE*, 101:3, 676-697.
- Jian, L., Li, J., Shu, K. ve Liu, H. (2016). Multi-label informed feature selection. *IJCAI*, 1627-1633.
- Jiang, Y., Kautz, H. ve Selman, B. (1995). Solving problems with hard and soft constraints using a stochastic algorithm for MAX-SAT. 1st International Joint Workshop on Artificial Intelligence and Operations Research.
- Jimenez-del-Toro, O., Otálora, S., Andersson, M., Eurén, K., Hedlund, M., Rousson, M., Müller, H. ve Atzori, M. 2017. *Biomedical Texture Analysis*. Elsevier, 281-314.
- Jin, X., Xu, A., Bie, R. ve Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. International Workshop on Data Mining for Biomedical Applications, Springer, 106-115.
- Jović, A., Brkić, K. ve Bogunović, N. (2015). A review of feature selection methods with applications. 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), Ieee, 1200-1205.
- Kabir, M. M., Islam, M. M. ve Murase, K. 2010. A new wrapper feature selection approach using neural network. *Neurocomputing*, 73:16-18, 3273-3283.
- Kalakech, M., Biela, P., Macaire, L. ve Hamad, D. 2011. Constraint scores for semi-supervised feature selection: A comparative study. *Pattern Recognition Letters*, 32:5, 656-665.

- Kalousis, A., Prados, J. ve Hilario, M. 2007. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge Information Systems*, 12:1, 95-116.
- Kanamori, T., Hido, S. ve Sugiyama, M. 2009. A least-squares approach to direct importance estimation. *Journal of machine learning research*, 10:Jul, 1391-1445.
- Kanan, H. R. ve Faez, K. 2008. An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Applied Mathematics Computational biology chemistry*, 205:2, 716-725.
- Kannan, S. S. ve Ramaraj, N. 2010. A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems*, 23:6, 580-585.
- Karagiannopoulos, M., Anyfantis, D., Kotsiantis, S. ve Pintelas, P. 2007. Feature selection for regression problems. *Proceedings of the 8th Hellenic European Research on Computer Mathematics its Applications, Athens, Greece, 2022*.
- Kass, G. V. 1980. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C*, 29:2, 119-127.
- Katiyar, S., Ibraheem, N. ve Ansari, A. Q. (2015). Ant colony optimization: a tutorial review. *National Conference on Advances in Power and Control*, 99-110.
- Katsov, I. 2017. *Introduction to Algorithmic Marketing: Artificial Intelligence for Marketing Operations*. Ilya Katsov,
- Kaya, H., Eyben, F., Salah, A. A. ve Schuller, B. (2014). CCA based feature selection with application to continuous depression recognition from acoustic speech features. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 3729-3733.
- Kennedy, J. ve Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, 1942-1948.
- Khachatryan, A., Semenovskaya, S. ve Vainshtein, B. 1981. The thermodynamic approach to the structure analysis of crystals. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical General Crystallography*, 37:5, 742-754.
- Khajezadeh, M., Taha, M. R., El-Shafie, A. ve Eslami, M. 2011. A survey on meta-heuristic global optimization algorithms. *Research Journal of Applied Sciences, Engineering Technology*, 3:6, 569-578.
- Kinnear, K. E., Langdon, W. B., Spector, L., Angeline, P. J. ve O'Reilly, U.-M. 1994. *Advances in genetic programming*. MIT press,
- Kira, K. ve Rendell, L. A. (1992a). The feature selection problem: Traditional methods and a new algorithm. *Aai*, 129-134.
- Kira, K. ve Rendell, L. A. 1992b. *Machine Learning Proceedings 1992*. Elsevier, 249-256.
- Kirkpatrick, S., Gelatt, C. D. ve Vecchi, M. P. 1983. Optimization by simulated annealing. *American Association for the Advancement of Science*, 220:4598, 671-680.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai, Montreal, Canada*, 1137-1145.
- Kohavi, R. ve John, G. H. 1997. Wrappers for feature subset selection. *Artificial intelligence*, 97:1-2, 273-324.
- Koller, D. ve Sahami, M. (1996). *Toward optimal feature selection*. Retrieved from
- Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. *European conference on machine learning*, Springer, 171-182.

- Kononenko, I., Robnik-Sikonja, M. ve Pompe, U. 1996. ReliefF for estimation and discretization of attributes in classification, regression, and ILP problems. *Artificial intelligence: methodology, systems, applications*, 31-40.
- Kononenko, I., Šimec, E. ve Robnik-Šikonja, M. 1997. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, 7:1, 39-55.
- Koziel, S. ve Yang, X.-S. 2011. *Computational optimization, methods and algorithms*. Springer,
- Kraskov, A., Stögbauer, H. ve Grassberger, P. 2004. Estimating mutual information. *Physical review E*, 69:6, 066138.
- Krishnaiah, V., Narsimha, G. ve Chandra, N. S. 2013. Diagnosis of lung cancer prediction system using data mining classification techniques. *International Journal of Computer Science Information Technologies*, 4:1, 39-45.
- Kullback, S. ve Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22:1, 79-86.
- Kumar, S. V. K. R. 2014. Analysis of feature selection algorithms on classification: a survey. *International Journal of Computer Applications*.
- Kumar, V. ve Minz, S. (2013). Mood classification of lyrics using SentiWordNet. 2013 International Conference on Computer Communication and Informatics, IEEE, 1-5.
- Kumar, V. ve Minz, S. 2014a. Feature selection: a literature review. *SmartCR*, 4:3, 211-229.
- Kumar, V. ve Minz, S. 2014b. *Advanced Computing, Networking and Informatics-Volume 1*. Springer, 57-66.
- Kuo, B.-C. ve Landgrebe, D. A. 2004. Nonparametric weighted feature extraction for classification. *IEEE Transactions on Geoscience Remote Sensing*, 42:5, 1096-1105.
- Ladha, L. ve Deepa, T. 2011. Feature selection methods and algorithms. *International journal on computer science engineering*, 3:5, 1787-1797.
- Land, A. ve Doig, A. 1960. An automatic method of solving discrete programming problems. *Econometrica*, 28.
- Landau, E. 2000. *Handbuch der Lehre von der Verteilung der Primzahlen*. Рипол Классик,
- Langley, P. (1994). Selection of relevant features in machine learning. Proceedings of the AAAI Fall symposium on relevance, 245-271.
- Lappas, T. ve Pelechrinis, K. 2007. Data mining techniques for (network) intrusion detection systems. *Department of Computer Science Engineering UC Riverside, Riverside CA*, 92521.
- Larochelle, H. ve Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. Proceedings of the 25th international conference on Machine learning, 536-543.
- Latorre Carmona, P., Sotoca, J. M. ve Pla, F. 2012. Filter-type variable selection based on information measures for regression tasks. *Entropy*, 14:2, 323-343.
- Lee, I.-H., Lushington, G. H. ve Visvanathan, M. 2011. A filter-based feature selection approach for identifying potential biomarkers for lung cancer. *Journal of Clinical Bioinformatics*, 1:1, 11.
- Lee, S., Yang, J. ve Oh, K.-w. (2003). Prediction of molecular bioactivity for drug design using a decision tree algorithm. International Conference on Discovery Science, Springer, 344-351.

- LeKhad, N., Wu, B., Chen, C. ve Kechadi, M.-T. (2013). Feature selection parallel technique for remotely sensed imagery classification. *International Conference on Computational Science and Its Applications*, Springer, 623-634.
- Leung, Y. ve Hung, Y. 2008. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM transactions on computational biology bioinformatics*, 7:1, 108-117.
- Li, B.-B., Wang, L. ve Liu, B. 2008a. An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling. *IEEE transactions on systems, man, cybernetics-part A: systems humans*, 38:4, 818-831.
- Li, B., Wang, L. ve Song, W. (2008b). Ant colony optimization for the traveling salesman problem based on ants with memory. 2008 Fourth International Conference on Natural Computation, IEEE, 496-501.
- Li, C., Yang, S. ve Nguyen, T. T. 2011. A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, Cybernetics, Part B*, 42:3, 627-646.
- Li, H., Xiang, S., Zhong, Z., Ding, K. ve Pan, C. 2015. Multicenter spatial-spectral unsupervised feature selection for hyperspectral image classification. *IEEE Geoscience Remote Sensing Letters*, 12:8, 1660-1664.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J. ve Liu, H. 2017. Feature selection: A data perspective. *ACM computing surveys*, 50:6, 1-45.
- Li, K. ve Tian, H. 2015. A de-based scatter search for global optimization problems. *Discrete Dynamics in Nature Society*, 2015.
- Li, Z., Yang, Y., Liu, J., Zhou, X. ve Lu, H. (2012). Unsupervised feature selection using nonnegative spectral analysis. Twenty-Sixth AAAI Conference on Artificial Intelligence.
- Liang, J. J., Qin, A. K., Suganthan, P. N. ve Baskar, S. 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on Evolutionary Computation*, 10:3, 281-295.
- Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A. ve Leong, K. 2017. Crossover and mutation operators of genetic algorithms. *International Journal of Machine Learning Computing*, 7:1, 9-12.
- Lin, S.-W., Tseng, T.-Y., Chou, S.-Y. ve Chen, S.-C. 2008. A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks. *Expert systems with applications*, 34:2, 1491-1499.
- Liu, C., Jiang, D. ve Yang, W. 2014. Global geometric similarity scheme for feature selection in fault diagnosis. *Expert systems with applications*, 41:8, 3585-3595.
- Liu, H., Li, J. ve Wong, L. 2002. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome informatics*, 13, 51-60.
- Liu, H., Liu, L. ve Zhang, H. 2010a. Ensemble gene selection by grouping for microarray data classification. *Journal of biomedical informatics*, 43:1, 81-87.
- Liu, H. ve Motoda, H. 1998. *Feature extraction, construction and selection: A data mining perspective*. Springer Science & Business Media,
- Liu, H. ve Motoda, H. 2012. *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media,
- Liu, H., Motoda, H., Setiono, R. ve Zhao, Z. (2010b). Feature selection: An ever evolving frontier in data mining. *Feature selection in data mining*, 4-13.

- Liu, H. ve Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge data engineering*, 17:4, 491-502.
- Liu, T., Liu, S., Chen, Z. ve Ma, W.-Y. (2003). An evaluation on feature selection for text clustering. Proceedings of the 20th international conference on machine learning (ICML-03), 488-495.
- Liu, W. ve Li, Q. 2017. An efficient elastic net with regression coefficients method for variable selection of spectrum data. *PloS one*, 12:2, e0171122.
- Liu, X., Wang, L., Zhang, J., Yin, J. ve Liu, H. 2013. Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks Learning Systems*, 25:6, 1083-1095.
- Liu, Z., Jiang, F., Tian, G., Wang, S., Sato, F., Meltzer, S. J. ve Tan, M. 2007. Sparse logistic regression with Lp penalty for biomarker identification. *Statistical Applications in Genetics Molecular Biology*, 6:1.
- Lozano, M. ve García-Martínez, C. 2010. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers Operations research letters*, 37:3, 481-497.
- Lu, D. ve Weng, Q. 2007. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28:5, 823-870.
- Lun Gaoa, T. L., Yaob, L. ve Wenb, F. 2013. Research and application of data mining feature selection based on relief algorithm. *Journal of Software*.
- Luo, M., Nie, F., Chang, X., Yang, Y., Hauptmann, A. G. ve Zheng, Q. 2017. Adaptive unsupervised feature selection with structure regularization. *IEEE Transactions on Neural Networks Learning Systems*, 29:4, 944-956.
- Luxburg, V. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17:4, 395-416.
- Lynch, M. 2010. Evolution of the mutation rate. *TRENDS in Genetics*, 26:8, 345-352.
- Maldonado, S., Weber, R. ve Famili, F. 2014. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Information Sciences*, 286, 228-246.
- Mandal, M. ve Mukhopadhyay, A. 2013. An improved minimum redundancy maximum relevance approach for feature selection in gene expression data. *Procedia Technology*, 10:0, 20-27.
- Mani, K. ve Kalpana, P. 2016. A review on filter based feature selection. *International Journal of Innovative Research in Computer Communication Engineering*, 4:5, 9146-9156.
- Margaritis, D. ve Thrun, S. (2000). Bayesian network induction via local neighborhoods. *Advances in neural information processing systems*, 505-511.
- Marill, T. ve Green, D. 1963. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9:1, 11-17.
- Martínez Casasnovas, J. A., Klaasse, A., Nogués Navarro, J. ve Ramos Martín, M. C. 2008. Comparison between land suitability and actual crop distribution in an irrigation district of the Ebro valley (Spain). *Spanish Journal of Agricultural Research*, 6:4, 700-713.
- Masaeli, M., Yan, Y., Cui, Y., Fung, G. ve Dy, J. G. (2010). Convex principal feature selection. Proceedings of the 2010 SIAM International Conference on Data Mining, SIAM, 619-628.

- McCulloch, W. S. ve Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:4, 115-133.
- Meiri, R. ve Zahavi, J. 2006. Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171:3, 842-858.
- Miao, J. ve Niu, L. 2016. A survey on feature selection. *Procedia Computer Science*, 91, 919-926.
- Michalewicz, Z. ve Fogel, D. B. 2013. *How to solve it: modern heuristics*. Springer Science & Business Media,
- Miruthula, P. ve Roopa, S. N. 2015. Unsupervised feature selection algorithms: a survey. *Int J Sci Res*.
- Mitchell, M. 1998. *An introduction to genetic algorithms*. MIT press,
- Mitra, P., Murthy, C. ve Pal, S. K. 2002. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis machine intelligence*, 24:3, 301-312.
- Mlambo, N., Cheruiyot, W. K., Kimwele, M. W. ve Science 2016. A survey and comparative study of filter and wrapper feature selection techniques. *International Journal of Engineering*, 5:8, 57-67.
- Muni, D. P., Pal, N. R. ve Das, J. 2006. Genetic programming for simultaneous feature selection and classifier design. *IEEE Transactions on Systems, Man, Cybernetics*, 36:1, 106-117.
- Muštra, M., Grgić, M. ve Delač, K. 2012. Breast density classification using multiple feature selection. *Automatika*, 53:4, 362-372.
- Naik, A. ve Rangwala, H. (2016). Embedding feature selection for large-scale hierarchical classification. 2016 IEEE International Conference on Big Data (Big Data), IEEE, 1212-1221.
- Nakamura, R. Y. M., Pereira, L. A. M., Rodrigues, D., Costa, K. A. P., Papa, J. P. ve Yang, X.-S. 2013. *Swarm Intelligence and Bio-Inspired Computation*. Elsevier, 225-237.
- Narendra, P. M. ve Fukunaga, K. 1977. A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*:9, 917-922.
- Nesmachnow, S. 2014. An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, 3:4, 320-347.
- Ng, A. Y., Jordan, M. I. ve Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 849-856.
- Nilsson, R. 2007. *Statistical feature selection: with applications in life science*. Institutionen för fysik, kemi och biologi
- Nocedal, J. ve Wright, S. J. 2006. *Numerical optimization*. Springer, New York.
- Novaković, J. 2016. Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of Operations Research*, 21:1.
- Nozawa, H. 1992. A neural network model as a globally coupled map and applications based on chaos. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2:3, 377-386.
- Oreski, S. ve Oreski, G. 2014. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert systems with applications*, 41:4, 2052-2064.
- Panda, S. 2011a. Differential evolution algorithm for SSSC-based damping controller design considering time delay. *Journal of the Franklin Institute*, 348:8, 1903-1926.

- Panda, S. 2011b. Robust coordinated design of multiple and multi-type damping controller using differential evolution algorithm. *International Journal of Electrical Power Energy Systems*, 33:4, 1018-1030.
- Paul, S. ve Drineas, P. 2016. Feature selection for ridge regression with provable guarantees. *Neural computation*, 28:4, 716-742.
- Pearl, J. 1988. Probabilistic Reasoning in Expert Systems: Networks of Plausible Reasoning: San Mateo, CA: Morgan Kaufmann (Pubs.).
- Peng, H. ve Fan, Y. 2015. Direct  $l_1$  (2, p)-Norm Learning for Feature Selection. *Computer Vision and Pattern Recognition*.
- Peng, H., Long, F. ve Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE transactions on pattern analysis machine intelligence*, 27:8, 1226-1238.
- Peng, Y., Wu, Z. ve Jiang, J. 2010. A novel feature selection approach for biomedical data classification. *Journal of biomedical informatics*, 43:1, 15-23.
- Pes, B., Dessì, N. ve Angioni, M. 2017. Exploiting the ensemble paradigm for stable feature selection: a case study on high-dimensional genomic data. *Information Fusion*, 35, 132-147.
- Piao, M., Piao, Y. ve Lee, J. Y. 2019. Symmetrical uncertainty-based feature subset generation and ensemble learning for electricity customer classification. *Symmetry*, 11:4, 498.
- Piao, Y., Piao, M., Park, K. ve Ryu, K. H. 2012. An ensemble correlation-based gene selection algorithm for cancer classification with gene expression data. *Bioinformatics*, 28:24, 3306-3315.
- Pohjalainen, J., Räsänen, O., Kadioglu, S. ve Language 2015. Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Computer Speech*, 29:1, 145-171.
- Pohl, I. 1971. Bi-directional search. *Machine intelligence*, 6, 127-140.
- Poli, R., Kennedy, J. ve Blackwell, T. 2007. Particle swarm optimization. *Swarm Intelligence*, 1:1, 33-57.
- Pratama, S. F., Muda, A. K., Choo, Y.-H. ve Muda, N. A. 2012. A comparative study of feature selection methods for authorship invariance in writer identification. *International Journal of Computer Information Systems Industrial Management Applications*, 4, 467-476.
- Price, K., Storn, R. M. ve Lampinen, J. A. 2006. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media,
- Priyadarsini, R. P., Valarmathi, M. ve Sivakumari, S. 2011. Gain ratio based feature selection method for privacy preservation. *CTACT Journal on soft computing*, 1:4, 201-205.
- Provost, F. ve Fawcett, T. 2001. Robust classification for imprecise environments. *Machine learning for Computer Cyber Security*, 42:3, 203-231.
- Pudil, P., Novovičová, J. ve Kittler, J. 1994. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:11, 1119-1125.
- Qian, M. ve Zhai, C. (2013). Robust unsupervised feature selection. Twenty-Third International Joint Conference on Artificial Intelligence.
- Quinlan, J. R. 1986. Induction of decision trees. *Machine learning for Computer Cyber Security*, 1:1, 81-106.

- Radovic, M., Ghalwash, M., Filipovic, N. ve Obradovic, Z. 2017. Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC bioinformatics*, 18:1, 9.
- Raileanu, L. E. ve Stoffel, K. 2004. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics Artificial Intelligence*, 41:1, 77-93.
- Ramchandran, A. ve Sangaiah, A. K. 2018. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*. Elsevier, 233-251.
- Rardin, R. L. ve Rardin, R. L. 1998. *Optimization in operations research*. Prentice Hall Upper Saddle River, NJ,
- Rechenberg, I. 1973. *Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Information*: Fromman Verlag, Freiburg, Germany.
- Reeves, C. 2010. 109-139.
- Reimann, M., Doerner, K. ve Hartl, R. F. (2002). Insertion based ants for vehicle routing problems with backhauls and time windows. *International Workshop on Ant Algorithms*, Springer, 135-148.
- Rejer, I. ve Lorenz, K. 2013. Genetic algorithm and forward method for feature selection in EEG feature space. *Journal of Theoretical Applied Computer Science*, 7:2, 72-82.
- Rere, L. R., Fanany, M. I. ve Murni, A. (2014). Application of metaheuristic algorithms for optimal smartphone-photo enhancement. 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), IEEE, 542-546.
- Robnik-Šikonja, M. ve Kononenko, I. (1997). An adaptation of Relief for attribute estimation in regression. *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, 296-304.
- Robnik-Šikonja, M. ve Kononenko, I. J. M. I. 2003. Theoretical and empirical analysis of ReliefF and RReliefF. 53:1-2, 23-69.
- Roeva, O., Slavov, T. ve Fidanova, S. 2014. *Handbook of Research on Novel Soft Computing Intelligent Algorithms: Theory and Practical Applications*. IGI Global, 200-233.
- Roffo, G., Melzi, S. ve Cristani, M. (2015). Infinite feature selection. *Proceedings of the IEEE International Conference on Computer Vision*, 4202-4210.
- Rossi, F., Lendasse, A., François, D., Wertz, V. ve Verleysen, M. 2006. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics Intelligent Laboratory Systems*, 80:2, 215-226.
- Rout, U. K., Sahu, R. K. ve Panda, S. 2013. Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system. *Ain Shams Engineering Journal*, 4:3, 409-421.
- Roweis, S. T. ve Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290:5500, 2323-2326.
- Ruiz, R., Riquelme, J. C. ve Aguilar-Ruiz, J. S. (2005). Heuristic search over a ranking for feature selection. *International Work-Conference on Artificial Neural Networks*, Springer, 742-749.
- Russell, S. ve Norvig, P. 2002. *Artificial intelligence: a modern approach*. Alan Apt,
- Saeyns, Y. 2004. *Feature selection for classification of nucleic acid sequences*. Ghent University
- Saeyns, Y., Abeel, T. ve Van de Peer, Y. (2008). Robust feature selection using ensemble feature selection techniques. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 313-325.

- Saeyns, Y., Inza, I. ve Larrañaga, P. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23:19, 2507-2517.
- Sahli, Z., Hamouda, A., Bekrar, A. ve Trentesaux, D. (2014). Hybrid PSO-tabu search for the optimal reactive power dispatch problem. *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 3536-3542.
- Saikhu, A., Arifin, A. Z. ve Faticah, C. 2019. Correlation and symmetrical uncertainty-based feature selection for multivariate time series classification. *Int. J. Intell. Eng. Syst*, 12:3, 129-137.
- Samsuddin, S., Othman, M. S. ve Yusuf, L. M. 2018. a Review of Single and Population-Based Metaheuristic Algorithms Solving Multi Depot Vehicle Routing Problem. *International Journal of Software Engineering Computer Systems*, 4:2, 80-93.
- Samsudin, S. H., Shafri, H. Z., Hamedianfar, A. ve Mansor, S. 2015. Spectral feature selection and classification of roofing materials using field spectroscopy data. *Journal of Applied Remote Sensing*, 9:1, 095079.
- Sánchez-Marono, N., Alonso-Betanzos, A. ve Tombilla-Sanromán, M. (2007). Filter methods for feature selection—a comparative study. *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 178-187.
- Sarac, F., Uslan, V., Seker, H. ve Bouridane, A. (2015). Comparison of unsupervised feature selection methods for high-dimensional regression problems in prediction of peptide binding affinity. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 8173-8176.
- Sarac, F., Uslan, V., Seker, H. ve Bouridane, A. (2016). Unsupervised selection of rv144 hiv vaccine-induced antibody features correlated to natural killer cell-mediated cytotoxic reactions. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 3072-3075.
- Sarhrouni, E., Hammouch, A. ve Aboutajdine, D. J. a. p. a. 2012. Application of symmetric uncertainty and mutual information to dimensionality reduction and classification of hyperspectral images.
- Schwefel, H.-P. 1977. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie*. Springer,
- Sedgwick, P. 2012. Pearson's correlation coefficient. *Bmj*, 345, e4483.
- Segal, M. R., Dahlquist, K. D. ve Conklin, B. R. 2003. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10:6, 961-980.
- Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V. ve Alonso-Betanzos, A. 2017. Ensemble feature selection: homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118, 124-139.
- Sengupta, R. N., Gupta, A. ve Dutta, J. 2016. *Decision sciences: theory and practice*. Crc Press,
- Senliol, B., Gulgezen, G., Yu, L. ve Cataltepe, Z. (2008). Fast Correlation Based Filter (FCBF) with a different search strategy. 2008 23rd international symposium on computer and information sciences, IEEE, 1-4.
- Setiono, R. ve Liu, H. 1996. Improving backpropagation learning with feature selection. *Applied Intelligence*, 6:2, 129-139.
- Shannon, C. E. 1948. A mathematical theory of communication. *Bell system technical journal*, 27:3, 379-423.

- Shannon, C. E. ve Weaver, W. 1949. The mathematical theory of communication, 117 pp. *Urbana: University of Illinois Press.*
- Shardlow, M. 2016. An analysis of feature selection techniques. *The University of Manchester*, 1, 1-7.
- Shen, J., Li, L. ve Wong, W.-K. (2008). Markov Blanket Feature Selection for Support Vector Machines. *AAAI*, 696-701.
- Shen, Q., Diao, R. ve Su, P. 2012. Feature Selection Ensemble. *Turing-100*, 10, 289-306.
- Shi, Y. ve Eberhart, R. (1998). A modified particle swarm optimizer. 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), IEEE, 69-73.
- Sierra, B., Lazkano, E., Inza, I., Merino, M., Larrañaga, P. ve Quiroga, J. (2001). Prototype selection and feature subset selection by estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips. *Conference on Artificial Intelligence in Medicine in Europe*, Springer, 20-29.
- Singh, B., Kushwaha, N. ve Vyas, O. P. 2014. A feature subset selection technique for high dimensional data using symmetric uncertainty. *Journal of Data Analysis Information Processing*, 2:04, 95.
- Sivagaminathan, R. K. ve Ramakrishnan, S. 2007. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert systems with applications*, 33:1, 49-60.
- Sivanandam, S. ve Deepa, S. 2007. *Principles of soft computing (with CD)*. John Wiley & Sons,
- Skalak, D. B. 1994. *Machine Learning Proceedings 1994*. Elsevier, 293-301.
- Solorio-Fernández, S., Carrasco-Ochoa, J. A. ve Martínez-Trinidad, J. F. 2020. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53:2, 907-948.
- Somol, P., Novovičová, J. ve Pudil, P. (2006). Flexible-hybrid sequential floating search in statistical feature selection. *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer, 632-639.
- Song, Q., Ni, J. ve Wang, G. 2011. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on knowledge data engineering*, 25:1, 1-14.
- Sörensen, K., Maya Duque, P., Vanovermeire, C. ve Castro, M. 2012. Metaheuristics for the multimodal optimization of hazmat transports. *Security Aspects of Uni-and Multimodal Hazmat Transportation Systems*, 163-181.
- Sorensen, K., Sevaux, M. ve Glover, F. 2017. A history of metaheuristics. *Handbook of Heuristics*.
- Sosa-Cabrera, G., García-Torres, M., Gómez, S., Schaerer, C. ve Divina, F. 2017. Understanding a version of multivariate symmetric uncertainty to assist in feature selection. *Universidad Nacional de Asunción*.
- Spall, J. C. 2005. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons,
- Sreeja, N. ve Sankar, A. 2015. Pattern matching based classification using ant colony optimization based feature selection. *Applied Soft Computing*, 31, 91-102.
- Srinivas, N. ve Deb, K. 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2:3, 221-248.

- Steppe, J. ve Bauer Jr, K. 1997. Feature saliency measures. *Computers Mathematics with Applications*, 33:8, 109-126.
- Storn, R. ve Price, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:4, 341-359.
- Stearns, S. 1976. On selecting features for pattern classifiers. *Proc. ICPR*, 71-75.
- Stützle, T. 1998. Local search algorithms for combinatorial problems. 20 Phd, Darmstadt University of Technology, 214
- Stützle, T. ve Dorigo, M. 1999. ACO algorithms for the quadratic assignment problem. *New ideas in optimization*:C50, 33.
- Suebsing, A. ve Hiransakolwong, N. (2009). Feature selection using euclidean distance and cosine similarity for intrusion detection model. 2009 First Asian Conference on Intelligent Information and Database Systems, IEEE, 86-91.
- Sugiyama, M. 2015. *Introduction to statistical machine learning*. Morgan Kaufmann,
- Suzuki, T., Sugiyama, M., Sese, J. ve Kanamori, T. (2008). Approximating mutual information by maximum likelihood density ratio estimation. New challenges for feature selection in data mining and knowledge discovery, 5-20.
- Suzuki, T., Sugiyama, M. ve Tanaka, T. (2009). Mutual information approximation via maximum likelihood estimation of density ratio. 2009 IEEE International Symposium on Information Theory, IEEE, 463-467.
- Swingle, B. 2012. Rényi entropy, mutual information, and fluctuation properties of Fermi liquids. *Physical Review B*, 86:4, 045109.
- Szu, H. ve Rogers, G. (1992). Generalized McCulloch-Pitts neuron model with threshold dynamics. Int'l Joint Conf. Neural Networks.
- Tahir, M. A., Bouridane, A. ve Kurugollu, F. 2007. Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. *Pattern Recognition Letters*, 28:4, 438-446.
- Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. International Symposium on Intelligent Data Analysis, Springer, 440-451.
- Talbi, E.-G. 2009. *Metaheuristics: from design to implementation*. John Wiley & Sons,
- Talbi, E.-G., Jourdan, L., Garcia-Nieto, J. ve Alba, E. (2008). Comparison of population based metaheuristics for feature selection: Application to microarray data classification. 2008 IEEE/ACS International Conference on Computer Systems and Applications, IEEE, 45-52.
- Talukder, S. 2011. Mathematicle modelling and applications of particle swarm optimization.
- Tan, N. C., Fisher, W. G., Rosenblatt, K. P. ve Garner, H. R. 2009. Application of multiple statistical tests to enhance mass spectrometry-based biomarker discovery. *BMC bioinformatics*, 10:1, 144.
- Tan, T. G., Teo, J. ve Chin, K. O. 2013. Single-versus multiobjective optimization for evolution of neural controllers in Ms. Pac-man. *International Journal of Computer Games Technology*, 2013.
- Tan, Y. ve Liu, Z. (2013). Feature selection and prediction with a Markov blanket structure learning algorithm. *BMC bioinformatics*, Springer, A3.
- Tang, J., Alelyani, S. ve Liu, H. 2014. Feature selection for classification: A review. *Data classification: Algorithms applications*, 37.

- Tarczy-Hornoch, P. ve Minie, M. 2005. *Medical Informatics*. Springer, 63-94.
- Tenenbaum, J. B., De Silva, V. ve Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290:5500, 2319-2323.
- Tian, J., Hu, Q., Ma, X. ve Han, M. 2012. An improved KPCA/GA-SVM classification model for plant leaf disease recognition. *Journal of Computational Information Systems*, 8:18, 7737-7745.
- Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:1, 267-288.
- Tomar, D. ve Agarwal, S. 2013. A survey on Data Mining approaches for Healthcare. *International Journal of Bio-Science Bio-Technology*, 5:5, 241-266.
- Tomioka, R., Suzuki, T. ve Sugiyama, M. 2011. Super-Linear Convergence of Dual Augmented Lagrangian Algorithm for Sparsity Regularized Estimation. *Journal of machine learning research*, 12:5.
- Trabelsi, K., Sevaux, M., Coussy, P., Rossi, A. ve Sörensen, K. 2010. Advanced Metaheuristics for High-Level Synthesis.
- Tran, B., Xue, B. ve Zhang, M. (2014). Overview of particle swarm optimisation for feature selection in classification. Asia-Pacific conference on simulated evolution and learning, Springer, 605-617.
- Tsai, C.-F. ve Chou, J.-S. (2011). Data pre-processing by genetic algorithms for bankruptcy prediction. 2011 IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, 1780-1783.
- Tsamardinos, I., Aliferis, C. F. ve Statnikov, A. (2003a). Time and sample efficient discovery of Markov blankets and direct causal relations. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 673-678.
- Tsamardinos, I., Aliferis, C. F., Statnikov, A. R. ve Statnikov, E. (2003b). Algorithms for large scale Markov blanket discovery. FLAIRS conference, 376-380.
- Tyagi, V. ve Mishra, A. 2013. A survey on different feature selection methods for microarray data analysis. *International Journal of Computer Applications*, 67:16.
- Ugoni, A. ve Walker, B. F. 1995. The Chi square test: an introduction. *COMSIG review*, 4:3, 61.
- Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S. ve Moore, J. H. 2018a. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics*, 85, 189-203.
- Urbanowicz, R. J., Olson, R. S., Schmitt, P., Meeker, M. ve Moore, J. H. 2018b. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of biomedical informatics*, 85, 168-188.
- Vasant, P. 2015. *Handbook of research on artificial intelligence techniques and algorithms*,
- Venter, G. 2010. Review of optimization techniques. *Encyclopedia of aerospace engineering*.
- Vergara, J. R. ve Estévez, P. A. 2014. A review of feature selection methods based on mutual information. *Neural Computing Applications*, 24:1, 175-186.
- Verleysen, M., Rossi, F. ve François, D. 2009. *Similarity-Based Clustering*. Springer, 52-69.
- Vora, S. ve Yang, H. (2017). A comprehensive study of eleven feature selection algorithms and their impact on text classification. 2017 Computing Conference, IEEE, 440-449.

- Waldron, L., Pintilie, M., Tsao, M.-S., Shepherd, F. A., Huttenhower, C. ve Jurisica, I. 2011. Optimized application of penalized regression methods to diverse genomic data. *Bioinformatics*, 27:24, 3399-3406.
- Wang, D., Tan, D. ve Liu, L. 2018. Particle swarm optimization algorithm: an overview. *Soft Computing*, 22:2, 387-408.
- Wang, F., Chawla, S. ve Liu, W. (2013). Tikhonov or lasso regularization: Which is better and when. 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, IEEE, 795-802.
- Wang, J., Xu, J., Zhao, C., Peng, Y. ve Wang, H. 2019. An ensemble feature selection method for high-dimensional data based on sort aggregation. *Systems Science Control Engineering*, 7:2, 32-39.
- Wang, L., Li, S., Tian, F. ve Fu, X. 2004. A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Transactions on Systems, Man, Cybernetics*, 34:5, 2119-2125.
- Wang, L., Ni, H., Yang, R., Pappu, V., Fenn, M. B. ve Pardalos, P. M. 2014. Feature selection based on meta-heuristics for biomedicine. *Optimization Methods Software*, 29:4, 703-719.
- Wang, R., Bian, J., Nie, F. ve Li, X. 2020. Unsupervised Discriminative Projection for Feature Selection. *IEEE Transactions on knowledge data engineering*.
- Wang, S., Tang, J. ve Liu, H. (2015). Embedded unsupervised feature selection. Twenty-ninth AAAI conference on artificial intelligence.
- Wang, W., Yang, J. ve Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. *VLDB*, 186-195.
- Wang, Y., Chen, X., Jiang, W., Li, L., Li, W., Yang, L., Liao, M., Lian, B., Lv, Y. ve Wang, S. 2011a. Predicting human microRNA precursors based on an optimized feature subset generated by GA-SVM. *Genomics*, 98:2, 73-78.
- Wang, Y., Widrow, B. C., Zhang, B., Kinsner, W., Sugawara, K., Sun, F., Lu, J., Weise, T. ve Zhang, D. 2011b. Perspectives on the field of cognitive informatics and its future development. *International Journal of Cognitive Informatics Natural Intelligence*, 5:1, 1-17.
- Weisberg, S. 2005. *Applied linear regression*. John Wiley & Sons,
- Weise, T., Zapf, M., Chiong, R. ve Nebro, A. J. 2009. *Nature-inspired algorithms for optimisation*. Springer, 1-50.
- Welch, B. L. 1947. The generalization of ofstudent's' problem when several different population variances are involved. *Biometrika*, 34:1/2, 28-35.
- Welikala, R. A., Fraz, M. M., Dehmeshki, J., Hoppe, A., Tah, V., Mann, S., Williamson, T. H. ve Barman, S. A. 2015. Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy. *Computerized Medical Imaging Graphics*, 43, 64-77.
- Whitney, A. W. 1971. A direct method of nonparametric measurement selection. *IEEE Transactions on computers*, 100:9, 1100-1103.
- Witten, I. H., Frank, E. ve Hall, M. A. J. M. K. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 578.
- Wright, S. J., Nowak, R. D. ve Figueiredo, M. A. 2009. Sparse reconstruction by separable approximation. *IEEE Transactions on signal processing*, 57:7, 2479-2493.

- Wu, B., Zhang, L. ve Zhao, Y. 2013. Feature selection via Cramer's V-test discretization for remote-sensing image classification. *IEEE Transactions on Geoscience Remote Sensing*, 52:5, 2593-2606.
- Xie, J., Lei, J., Xie, W., Shi, Y. ve Liu, X. 2013. Two-stage hybrid feature selection algorithms for diagnosing erythemato-squamous diseases. *Health information science*, 1:1, 10.
- Xu, R., Damelin, S., Nadler, B. ve Wunsch II, D. C. 2010. Clustering of high-dimensional gene expression data with feature filtering methods and diffusion maps. *Artificial intelligence in medicine*, 48:2-3, 91-98.
- Xue, B., Zhang, M. ve Browne, W. N. 2012. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43:6, 1656-1671.
- Yang, P., Ho, J. W., Yang, Y. H. ve Zhou, B. B. 2011a. Gene-gene interaction filtering with ensemble of filters. *BMC bioinformatics*, 12:1, S10.
- Yang, P., Hwa Yang, Y., B Zhou, B. ve Y Zomaya, A. 2010. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5:4, 296-308.
- Yang, X.-S. 2010a. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons,
- Yang, X.-S. 2010b. *Nature-inspired metaheuristic algorithms*. Luniver press,
- Yang, X.-S. (2011a). Metaheuristic optimization: algorithm analysis and open problems. International Symposium on Experimental Algorithms, Springer, 21-32.
- Yang, X.-S. 2011b. Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation*, 3:2, 77-84.
- Yang, X.-S. 2012. Nature-inspired metaheuristic algorithms: Success and new challenges. *Journal of Computer Engineering & Information Technology*, 1:1.
- Yang, X.-S. 2013. *Optimization and metaheuristic algorithms in engineering*, 1-23,
- Yang, Y. ve Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. Icml, Nashville, TN, USA, 35.
- Yang, Y., Shen, H. T., Ma, Z., Huang, Z. ve Zhou, X. (2011b). L2, 1-norm regularized discriminative feature selection for unsupervised. Twenty-Second International Joint Conference on Artificial Intelligence.
- Yen, G., Yang, F. ve Hickey, T. 2002. Coordination of exploration and exploitation in a dynamic environment. *International Journal of Smart Engineering System Design*, 4:3, 177-182.
- Yildirim, P. 2015. Filter based feature selection methods for prediction of risks in hepatitis disease. *International Journal of Machine Learning and Computing*, 5:4, 258.
- Yin, W., Osher, S., Goldfarb, D. ve Darbon, J. 2008. Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing. *SIAM Journal on Imaging sciences*, 1:1, 143-168.
- Yu, K., Wu, X., Ding, W., Mu, Y. ve Wang, H. 2016. Markov blanket feature selection using representative sets. *IEEE Transactions on Neural Networks Learning Systems*, 28:11, 2775-2788.
- Yu, L., Han, Y. ve Berens, M. E. 2011. Stable gene selection from microarray data via sample weighting. *IEEE/ACM transactions on computational biology bioinformatics*, 9:1, 262-272.

- Yu, L. ve Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. *Proceedings of the 20th international conference on machine learning (ICML-03)*, 856-863.
- Yu, L. ve Liu, H. 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5:Oct, 1205-1224.
- Yu, X. ve Gen, M. 2010. *Introduction to evolutionary algorithms*. Springer Science & Business Media,
- Yusta, S. C. 2009. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters*, 30:5, 525-534.
- Zeng, X., Chen, Y.-W., Tao, C. ve van Alphen, D. (2009). Feature selection using recursive feature elimination for handwritten digit recognition. 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE, 1205-1208.
- Zhang, H. ve Sun, G. 2002. Feature selection using tabu search method. *Pattern Recognition*, 35:3, 701-711.
- Zhang, Y., Ding, C. ve Li, T. 2008. Gene selection algorithm by combining reliefF and mRMR. *BMC genomics*, 9:S2, S27.
- Zhang, Y., Dong, Z., Phillips, P., Wang, S., Ji, G., Yang, J. ve Yuan, T.-F. 2015. Detection of subjects and brain regions related to Alzheimer's disease using 3D MRI scans based on eigenbrain and machine learning. *Frontiers in computational neuroscience*, 9:9, 66.
- Zhao, J., Lu, K. ve He, X. 2008. Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71:10-12, 1842-1849.
- Zhao, P. ve Yu, B. 2006. On model selection consistency of Lasso. *Journal of machine learning research*, 7:Nov, 2541-2563.
- Zhao, Y.-Q., Zhang, L. ve Kong, S. G. 2010a. Band-subset-based clustering and fusion for hyperspectral imagery classification. *IEEE Transactions on Geoscience Remote Sensing*, 49:2, 747-756.
- Zhao, Z. ve Liu, H. (2007a). Semi-supervised feature selection via spectral analysis. *Proceedings of the 2007 SIAM international conference on data mining, SIAM*, 641-646.
- Zhao, Z. ve Liu, H. (2007b). Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th international conference on Machine learning*, 1151-1157.
- Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A. ve Liu, H. 2010b. Advancing feature selection research. *ASU feature selection repository*, 1-28.
- Zhao, Z. A. ve Liu, H. 2012. *Spectral feature selection for data mining*. Taylor & Francis,
- Ziamba, P. ve Piwowarski, M. 2011. Feature Selection Methods in Data Mining Techniques. *Prace Naukowe Uniwersytetu Ekonomicznego we Wroclawiu*:206, 213-223.
- Zomaya, A. Y. ve Kazman, R. 2010. *Algorithms and theory of computation handbook: general concepts and techniques*, 33-33.
- Zou, H. ve Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, 67:2, 301-320.
- Zurada, J. M. 1992. *Introduction to artificial neural systems*. West St. Paul,