FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

# ÖZNİTELİK SEÇİMİ İÇİN ÇOKLU-EBEVEYN ÇAPRAZLAMA OPERATÖRLERİNİN KARŞILAŞTIRILMASI

**YÜKSEK LİSANS TEZİ**

**NAZİF KANÇ**

**İSTANBUL, 2022**

**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
**BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

# ÖZNİTELİK SEÇİMİ İÇİN ÇOKLU-EBEVEYN ÇAPRAZLAMA OPERATÖRLERİNİN KARŞILAŞTIRILMASI

**YÜKSEK LİSANS TEZİ**

**NAZİF KANÇ**
**(180221004)**

**Danışman**
**(Dr. Öğr. Üyesi Berna Kiraz)**

**İSTANBUL, 2022**

**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**
TEZ ONAY FORMU

29/06/2022

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Yüksek Lisans programı öğrencisi 180221004 numaralı Nazif KANÇ'ın hazırladığı "Öznitelik Seçimi İçin Çoklu-Ebeveyn Çaprazlama Operatörlerinin Genetik Algoritma İçinde Kullanılması ve Performanslarının Karşılaştırılması" konulu Yüksek Lisans tezi ile ilgili Tez Savunma Sınavı, 29/06/2022 Çarşamba günü saat 15: 00'da yapılmış, sorulara alınan cevaplar sonunda adayın tezinin **Kabulüne Oy Birliği** ile karar verilmiştir.

**Düzeltme yapılması halinde:**
Adı geçen öğrencinin Tez Savunma Sınavı .../ …/ … tarihinde, saat …:… da yapılacaktır.

**Tez adı değişikliği yapılması halinde:** Tez adının "Öznitelik Seçimi için Çoklu-Ebeveyn Çaprazlama Operatörlerinin Karşılaştırılması" şeklinde değiştirilmesi uygundur.

| | **Jüri Üyesi** | **Karar** |
|---|---|---|
| 1. | (Danışman) Dr. Öğr. Üyesi Berna KİRAZ | Kabul |
| 2. | Prof. Dr. Ayşe Şima UYAR | Kabul |
| 3. | Doç. Dr. Buket DOĞAN | Kabul |
| 4. | ..... | …… |
| 5. | ..... | …… |
| 6. | (İkinci Danışman) *..... | …… |

***2. Danışman varsa doldurulması gerekmektedir.***

# ETİK BİLDİRİM

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bağlı olduğum üniversite veya bir başka üniversitedeki başka bir çalışma olarak sunulmadığını beyan ederim.

Nazif Kanç

# ÖZNİTELİK SEÇİMİ İÇİN ÇOKLU-EBEVEYN ÇAPRAZLAMA OPERATÖRLERİNİN KARŞILAŞTIRILMASI

**Nazif Kanç**

## ÖZET

Yapay zeka ve makine öğrenmesinde son yıllarda sağlanan gelişmelerle elde edilen büyük veri kümelerinin analizini daha hızlı yapmak ve veri kümelerinin boyutunu azaltarak depolama problemlerine çözüm sağlamak öznitelik seçiminin önemi ortaya koymatadır. Öznitelik seçimi ile tasarlanan modellerin başarısının artırılması hedeflenmektedir. Veri kümelerinde ilgisiz ve alakasız bilgiler bulunmaktadır. Veri kümesinin boyutunu azaltmak ve gürültüye neden olan verileri çıkarmak öznitelik seçimiyle mümkün olmaktadır. Öznitelik seçimi ile gerekli olmayan verilerin çıkarılması modelin başarımını artırmaktadır. Öte yandan, genetik algoritmalar öznitelik seçim problemlerine uygulanmış ve başarılı sonuçlar elde edilmiştir. Bu tezde genetik algoritma ile çok ebeveynli çaprazlama operatörleri kullanılarak veri kümesindeki en önemli öznitelikleri seçerek öznitelik sayısını azaltmak amaçlanmaktadır. Tek biçimli çaprazlama, oluşum tabanlı çaprazlama, uygunluk tabanlı çaprazlama ve diyagonal çaprazlama, çok ebeveynli çaprazlama operatörleri arasında yer alır. Genetik algoritmada bir aday çözümü kodlamak için farklı gösterimlerde vardır, bu tezde öznitelik seçimi için ikili gösterim kullanılmaktadır. Bu çaprazlama operatörlerinin değerlendirmesi farklı sayıda özelliğe sahip üç farklı veri kümesi üzerinde gerçekleştirildi. Algoritmaların istatistiksel karşılaştırmaları için One-way ANOVA ve Tukey HSD testleri %95 güven seviyesinde gerçekleştirilmektedir. Deneyler iki aşamada gerçekleştirildi: (1) Bu kısımda, her bir çok ebeveynli çaprazlama operatörü için ebeveyn sayısının etkisini araştırırız, (2) bu aşamada çaprazlama operatörlerinin ilgili en iyi parametre değerleri kullanılarak performansları karşılaştırılmıştır. Sonuçlar, 5 ebeveynli oluşum tabanlı

çaprazlama operatörünün diğer çaprazlama operatörlerinden daha iyi performans gösterdiğini, ancak daha fazla öznitelik seçtiğini ortaya koymaktadır.

**Anahtar kelimerler;** Genetik Algoritma, Öznitelik Seçimi, Çoklu-Ebeveyn, Çaprazlama, Sınıflandırma

# COMPARISON OF MULTI-PARENT CROSSOVER OPERATORS FOR FEATURE SELECTION

**Nazif Kanç**

## ABSTRACT

With the developments in artificial intelligence and machine learning in recent years, making the analysis of large data sets obtained faster and providing solutions to storage problems by reducing the size of data sets reveal the importance of feature selection. It is aimed to increase the success of the models designed with feature selection. Datasets contain irrelevant and unrelated information. Reducing the size of the dataset and removing the data that causes noise is possible by feature selection. Removing unnecessary data with feature selection increases the performance of the model. On the other hand, genetic algorithms are to feature selection problems and successful results are obtained. In this thesis, it is aimed to reduce the number of features by selecting the most important features in the data set by using genetic algorithm and multi-parent crossover operators. Uniform crossover, occurrence-based crossover, fitness-based crossover, and diagonal crossover are among the multi-parent crossover operators.There are different representations to encode a candidate solution in genetic algorithm, in this thesis binary representation is used for feature selection. Evaluation of these crossover operators was performed on three different datasets with different numbers of features. One-way ANOVA and Tukey HSD tests are performed at 95% confidence level for statistical comparisons of algorithms. The experiments were carried out in two stages: (1) In this section, we study on the effect of the number of parents for each multi-parent crossover operator, (2) at this stage, the performances of the crossover operators were compared using the best relevant parameter values. The results reveals that the 5-parent occurrence-based crossover operator outperforms other crossover operators, but selects more features.

**Keywords;** Genetic Algorithm, Feature Selection, Multi-Parent, Crossover, Classification

# ÖNSÖZ

Bu tez kapsamında öznitelik seçimi için genetik algoritma uygulanması planlanmaktadır. Genetik algoritma içinde farklı çoklu-ebeveyn çaprazlama operatörleri kullanılarak performanslarının karşılaştırılması amaçlanmaktadır.

Yüksek lisans tezimin başından sonuna kadar bilgi ve tecrübelerinden yararlandığım, karşılaştığım tüm sorunları yardım ve yönlendirmeleri ile kolayca çözüp çalışmamın tamamlanmasını sağlayan, çalışmalarımda destek olan ve bu araştırmamın her bir aşamasında desteğini esirgemeyen danışman hocam sayın Dr. Öğr. Üyesi Berna Kiraz'a teşekkür ederim. Yüksek lisans çalışmam süreci boyunca benden maddi manevi desteklerini esirgemeyen aileme teşekkür ederim.

Haziran/2022 Nazif Kanç

# İÇİNDEKİLER

## SEMBOLLER

**N** **:** Popülasyondaki kromozom sayısı

**PN** **:** Kromozom bit değeri

**f(i)** **:** Uygunluk değeri

**P(i)** **:** Birey seçilme olasılığı

$\mathbf{S_n}$ **:** En yüksek sıra değeri

$\mathbf{P_c}$ **:** Çaprazlama oranı

$\mathbf{P_m}$ **:** Mutasyon oranı

**C** **:** Sınıflandırma karar sınırlarını ayarlama parametresi

**ER(i)** **:** Sınıflandırma hata oranı

$\alpha$ **:** Sınıflandırma hata oranını kontrol eden parametre

**K** **:** Alt küme sayısı

**d** : Veri kümesindeki toplam öznitelik sayısı

**s** **:** Seçilen özniteliklerin sayısı

## ŞEKİLLER LİSTESİ

# TABLO LİSTESİ

# KISALTMALAR

| | |
|---|---|
| **DC** | : Diyagonal Çaprazlama |
| **DVM** | : Destek Vektör Makinası |
| **FBC** | : Uygunluk Temelli Çaprazlama |
| **GA** | : Genetik Algoritma |
| **K-NN** | : K-En Yakın Komşuluk |
| **OBC** | : Oluşum Tabanlı Çaprazlama |
| **UC** | : Tek Biçimli Çaprazlama |
| **XGBoost** | : Extreme Gradient Boosting |
| **WDBC** | : Wisconsin Diagnostic Breast Cancer |

# GİRİŞ

Genetik algoritma (GA) Darwin'in doğal seçim ve evrim teorisi ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Genetik algoritmalar doğa olaylarını ve biyolojik evrimi taklit eden algoritmalardır. Evrim süreci John Holland tarafından bilgisayar ortamına aktarılarak genetik algoritmalar oluşturulmuştur [1]. Goldberg ise Genetik Algoritmaların çeşitli alanlarda kullanılabileceğini ifade etmiştir [2]. Genetik algoritmalar çeşitli çözümler arasındaki en iyi çözümü bulmaya amaçlayan popülasyon tabanlı algoritmalardır. Genetik algoritmada aday çözümün nasıl bir gösterilime sahip olacağının karar verilmesi gereken en önemli noktadır. Çözüm gösterimi belirlendikten sonra ilk popülasyon genellikle rastgele oluşturulur. Daha sonrasına ebeveyn seçme, çaprazlama ve mutasyon operatörleri kullanılarak çocuk popülasyon oluşturulur. Üretilen popülasyondan bir sonraki nesle hangi bireylerin geçeceği kısım ise seçme yöntemlerine göre belirlenir. Çıkış koşulu sağlanıncaya kadar bu süreç devam eder.

Gerçek dünyada elde edilen veri kümelerinde alakasız ve gereksiz bilgiler bulunmaktadır. Veri kümelerinden bu özelliklerin çıkarılması sınıflandırma doğruluğunu artırmaktadır. Veri kümesinde çok sayıda özellik bulunması da sınıflandırmanın doğruluğunu olumsuz etkilemekte ve daha fazla hesaplama ve alan gerektirmektedir. Veri kümesinin boyutunu azaltmak ve gürültüye neden olan verileri çıkarmak öznitelik seçimiyle mümkün olmaktadır. Öznitelik seçimi orijinal veri kümesini temsil edebilecek en iyi alt kümenin seçimi olarak tanımlanan bir ön işlemdir. Bu işlem, ele alınan problem en yararlı ve en önemli özellikleri seçerek veri kümesindeki öznitelik sayısını azaltmayı amaçlamaktadır. Öznitelik seçimi problemleri için meta-sezgisel algoritmalar sıklıkla kullanılmaktadır [3], [4].

Bu tez çalışması kapsamında genetik algoritmayla öznitelik seçimi gerçekleştirilmiştir. Genetik algoritma içinde tek biçimli çaprazlama (UC), oluşum tabanlı çaprazlama (OBC), uygunluk temelli çaprazlama (FBC) ve diyagonal

çaprazlama (DC) operatörleriyle UCI veri kümesinden elde edilen farklı veri kümeleri üzerinde 2, 3, 5 ve 10 olarak belirlenen ebeveyn değerleri ile öznitelik seçimi için testler yapıldı ve elde edilen sonuçlar doğrultusunda istatiksel analizler One-way ANOVA ve Tukey HSD testleri %95 güven seviyesinde gerçekleştirildi. Sonuçlar 5 ebeveynli çoklu-ebeveyn OBC operatöründe daha iyi sonuçlar elde edildiği fakat karakteristiği nedeniyle öznitelik sayısı performansının yeterli olmadığı gözlenmiştir.

Bu tez çalışmasının sonraki bölümlerinin organizasyonu şu şekildedir. Birinci bölüm temel kavramlar ve literatür araştırması bölümünden oluşmakta olup genetik algoritma ve temel kavramları, çaprazlama operatörleri, öznitelik seçimi, denetimli öğrenme, sınıflandırma kavramları ve literatürdeki çalışmalar hakkında bilgi verilmiştir. İkinci bölüm öznitelik seçimi ve genetik algoritmadan oluşmakta olup kullanılan genetik algoritma ve uygunluk fonksiyonu hakkında bilgi verilmiştir. Üçüncü bölümde deneysel çalışmalar yapılarak; veri setleri ve parametre atamaları, performans metriği, ebeveyn sayısı seçimi ve öznitelik sayısı belirleme aşamaları hakkında bilgi verilmiştir. Dördüncü bölüm olan bulgular ve tartışmada genetik algoritma ile dört farklı çaprazlama operatörü ve üç veri kümesi ile testler yapılmış elde edilen sonuçlar istatiksel olarak karşılaştırılmıştır. Sonuç bölümde çalışmada ulaşılan sonuçlar, karşılaşılan zorluklar ve gelecekteki çalışmalar hakkında bilgi verilmiştir.

# BİRİNCİ BÖLÜM

# 1. TEMEL KAVRAMLAR VE LİTERATÜR ARAŞTIRMASI

Bu bölümde genetik algoritmanın tarihçesi, temel kavramları, aşamaları, çaprazlama operatörleri; öznitelik seçimi ve yöntemleri ile denetimli öğrenme ve destek vektör makinaları hakkında bilgi verilmiştir.

## 1.1. GENETİK ALGORİTMA

Genetik Algoritma kavramı 1970'li yıllarda John Holland tarafından yapılan çalışmalarda ortaya çıkmıştır. John Holland biyolojik evrim sürecini taklit ederek genetik algoritmaları oluşturmuştur [5]. Holland'ın öğrencisi olan Goldberg'in 1989 yılında çıkardığı kitapta ise genetik algoritmanın çeşitli farklı konularda uygulandığı görülmektedir [2].Genetik algoritma doğal seçim ilkeleri ve biyolojik evrim sürecini taklit ederek karmaşık ve çözülmesi zor olan problemlere optimal çözümü bulmayı hedefler [6]. Genetik algoritmalar evrimsel hesaplamalarda zor ve karmaşık geleneksel problemlerin çözümünde kullanılmaktadır. GA ile bir başlangıç çözüm kümesinin geliştirilerek biyolojik evrim süreci sonunda en iyi kromozomu bulmayı amaçlamaktadır [7]. Başlangıç popülasyonun rastgele olarak oluşturulduktan sonra seçilen ebeveynlerden iyi özellikler taşıyan yeni bireyler oluşturulur ve zaman ilerledikçe çözümün kalitesi de artmaktadır.

### 1.1.1. Genetik Algoritmanın Temel Kavramları

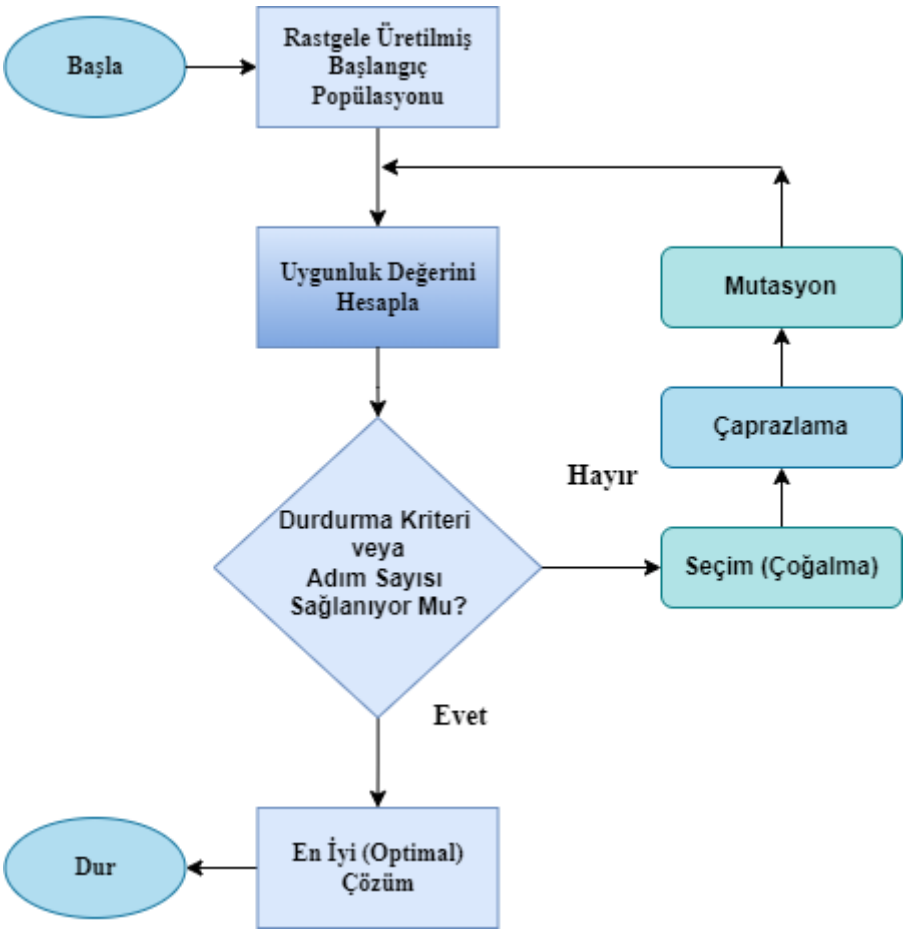
Gen: Karakterle ilgili en küçük anlamlı bilgiyi taşıyan genetik birimdir. Bir gen 0 ve 1 gibi bir bit veya A, B, C gibi çeşitli karakterlerle ifade edilen bir bilgi olabilir.

Kromozom (Birey, çözüm): Birden fazla genin bir araya gelerek oluşturduğu diziye denir. Bireyle ilgili kalıtsal bilgileri içermektedir. Her kromozom kendisine ait çözüme ait bilgileri temsil etmektedir.

Popülasyon: Kromozomlarla oluşturulan topluluğa denilmektedir. Popülasyonun büyüklüğü sabit veya kullanıcı tarafından değiştirilebilmektedir. Popülasyonun oluşturan topluluk çok küçük olduğunda, genetik algoritmada yerel bir optimuma takılma problemleri olabilmektedir. Popülasyonu oluşturan topluluğun çok büyük olması durumunda ise çözüm elde etme zamanı artarak çözüme ulaşmayı zorlaştırmaktadır [6].

### 1.1.2. Genetik Algoritmanın Aşamaları

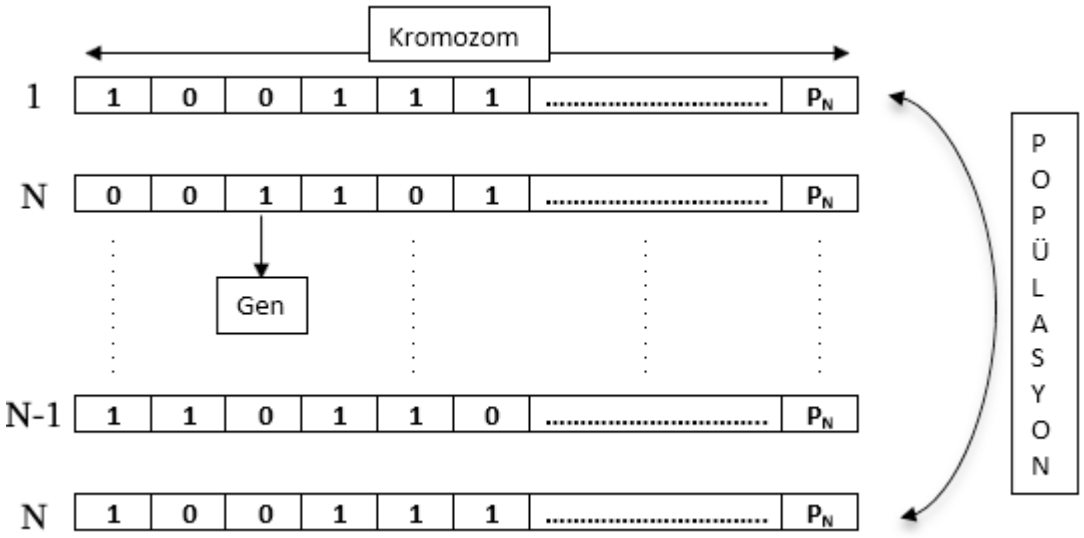
Genetik algoritma üreme, çaprazlama ve mutasyon gibi üç temel aşamadan oluşmaktadır [2]. Genetik algoritmalar ile uygun olmayan çözümler elenerek her nesilde daha iyi çözümler elde edilir. Bu sayede başlangıçta rastgele olarak alınan çözümlerden performans olarak düşük olanların yerleri daha iyi çözümlerle değiştirilmektedir [8]. Şekil 1.1'de GA'nın akış diyagramı gösterilmektedir.




Şekil 1.1 Genetik Algoritma Akış Diyagramı.

1.1.2.1. Başlangıç Popülasyonunun Oluşturulması.

Genetik algoritmanın birinci aşamasında çeşitli bireylerden oluşan bir başlangıç popülasyonu oluşturulur. Bu tezdeki popülasyondaki çözümler aynı boyutlara sahip bit dizileri ile gösterilmektedir. Popülasyonun büyüklüğü artıkça çözüme ulaşma zamanı ve maliyette artmaktadır [9]. Popülasyondaki her bir çözümün kodları da kromozom olarak adlandırılmaktadır. Genetik algoritma uygulamalarında genelde kromozom yapısı ikili bit dizisi ile oluşturulmaktadır. Kromozomlardaki genler oluşturulurken rastgele olarak 0 veya 1 değerine ayarlanmaktadır. $N$ adet kromozomdan oluşan bir popülasyon da kromozom gösterimi $[P1, P2, P3, \dots\dots, PN]$ile ifade edilir. Buradaki $P$ değerleri 0 veya 1 bit ile gösterilmektedir. $N$ adet kromozomdan oluşan popülasyon yapısı Şekil 1.2.'de gösterilmektedir.




Şekil 1.2 N Tane Kromozomdan Oluşan Popülasyonun Yapısı.

***1.1.2.1.1. Kodlama Yöntemleri***

Kodlama ile popülasyonu oluşturan kromozomlarla ifade edilen çözümlerin nasıl oluşturulacağı belirtilir. Problemlerin çeşitlerine göre farklı kodlama yöntemleri kullanılmaktadır [2]. Genetik algoritmalar da en çok kullanılan kodlama tekniği ikili kodlamadır. İkili kodlama ile her bir kromozom 0 ve 1'lerle oluşturulan bit dizisi ile temsil edilir. Değer kodlama ise kromozomlar reel sayılar ve karakterle ile oluşturulmaktadır. Permütasyon kodlama ise daha çok gezgin satıcı problemi gibi

sıralama problemlerinde kullanılmaktadır. Şekil 1.3'de farklı kodlama türleri görülmektedir.

**İkili Kodlama**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Kromozom A: | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| Kromozom B: | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |

**Değer Kodlama**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Kromozom A: | 4.15 | 3.10 | 4.44 | 5.35 | 6.74 | 4.12 | 3.75 | 5.12 | 2.35 |
| Kromozom B: | B | M | L | T | C | Z | A | A | M |

**Permütasyon Kodlama**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Kromozom A: | 7 | 8 | 6 | 3 | 2 | 4 | 5 | 1 | 9 |
| Kromozom B: | 6 | 8 | 3 | 2 | 5 | 7 | 9 | 1 | 4 |

Şekil 1.3 Kodlama Türleri.

1.1.2.2. Uygunluk Değerinin Hesaplanması

Başlangıç popülasyonu oluşturulduktan sonra her kromozomun uygunluk değeri hesaplanmalıdır. Uygunluk fonksiyonu ile her bireyin üstünlük değeri hesaplanır. GA'da en önemli aşamalardan biridir ve probleme özgü olarak uygunluk fonksiyonu değişmektedir. Çözümü aranan her problem için bir uygunluk değeri bulunmaktadır. Kromozomun kalite değeri uygunluk fonksiyonu ile hesaplanmaktadır [10]. Bireyler uygunluk değerlerine göre algoritmanın çalışma süreci boyunca bir sonraki kuşağa aktarılmaktadırlar [11]. Popülasyon istenen çözüme ulaşana kadar uygunluk fonksiyonu ile değerlendirilir.

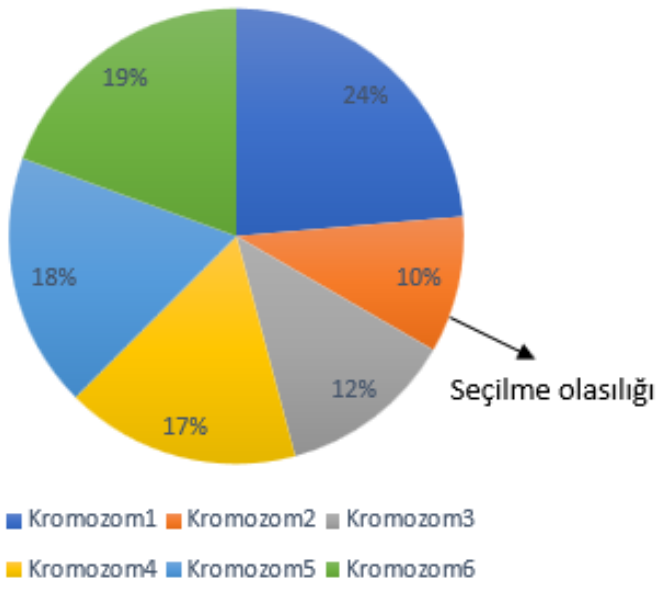
1.1.2.3. Seçim (Çoğalma-Yeniden Üretim) İşlemi

Başlangıç popülasyonu oluşturulduktan sonra kromozomların uygunluk değeri (iyi özellikleri) kullanılarak bir sonraki nesilde yer alacak bireylerin belirlenmesi için ebeveyn seçimleri yapılmaktadır. Popülasyonda yer alan her kromozom bir sonraki nesilde yer almak ister, uygunluk değeri yüksek olan kromozomların yeni popülasyona

aktarılma ihtimalleri daha yüksektir. Darwin'in evrim teorisine göre en iyi bireyler hayatta kalmaktadır. Uygunluk değeri yüksek olanların sürekli olarak yeni kuşağa aktarılması genetik çeşitliliğin azalmasına neden olabilir. Diğer yandan özellikleri yeterli olmayan bireyler ise evrim sürecinin yavaş ilerlemesine sebep olabilir [12]. Yeni kuşağa aktarılacak ebeveynler belirlenirken rulet tekerleği, sıralama ve turnuva seçimi en çok kullanılan seçim yöntemleridir.

#### *1.1.2.3.1. Rulet Tekerleği Seçim Yöntemi*

Popülasyonda her bireyin uygunluk değeri hesaplandıktan sonra toplanır. Her bireyin uygunluk değerinin, toplam uyguluk değerine oranı ile bireylerim seçilme olasılığı hesaplanır. Bireyin seçilme olasılığı rulet tekerleğinde bir dilim olarak temsil edilir. Sonraki adım da ise rulet tekerleği döndürülerek tekerleğin altındaki dilimde yer alan birey ebeveyn olarak seçilir [11]. Şekil 1.4'de rulet tekerleği seçilme olasılığı grafiğinde görüldüğü üzere uygunluk değeri yüksek olan birey rulet tekerleğinde yüzde olarak daha çok alanda temsil edileceğinden seçilme olasılığı artacaktır.




Şekil 1.4 Rulet Tekerleği Seçilme Olasılığı Grafiği.

Bireyin rulet tekerleği yönteminde seçilme olasılığı aşağıdaki Denklem 1.1'deki formüle göre hesaplanmaktadır.

$$P(i) = \frac{f(i)}{\Sigma f(i)} \quad \textbf{(1.1)}$$

$f(_i)$= Her bir birey için hesaplanan uygunluk değeridir.

$\sum f(_i)$= Popülasyonda yer alan tüm bireylerin uygunluk değerlerinin toplamıdır.

$P(_i)$ = Bireylerin seçilme olasılığı.

#### *1.1.2.3.2. Sıralama Seçim Yöntemi*

Sıralama seçim yönteminde popülasyonu oluşturan bireyler uygunluk değerlerine göre sıralanmaktadır. Bireylere uygunluk değerlerine göre sıraya konulur. En kötü uygunluk değerine sahip olan bireye 1 değeri verilir. En yüksek sıra(rank) değeri ($Sn$), en iyi uygunluğu sahip olan bireye verilir [13]. Sıralama yapıldıktan sonra göre olasılıklar Denklem 1.2'deki formüle göre hesaplanır ve rassal olarak yeni popülasyonda yer alacak bireyin seçimi yapılır.

$$P(i) = \frac{rank(i)}{Sn*(Sn-1)} \quad \textbf{(1.2)}$$

#### *1.1.2.3.3. Turnuva Seçim Yöntemi*

Popülasyon içerisinden rastgele n tane birey seçilir. Seçilen n tane birey uygunluk değerlerine göre karşılaştırılır. En iyi birey bir sonraki popülasyonu oluşturmak için ebeveyn olarak seçilmektedir. Problem maksimizasyon problemi ise uygunluk değeri yüksek olan birey seçilir. Problem minimizasyon problemi ise uygunluk değeri düşük olan birey seçilir [12]. Turnuva seçim yönteminde turnuvaya girecek bireylerin uygunluk değerleri düşükte olabilir. Turnuva seçimi ile bireylerin aralarından daha iyi olanı bir sonraki kuşağa aktarılmaktadır. Turnuva seçimini kazanan bireyler çaprazlama aşamasına katılırlar. Bu sayede popülasyondaki çeşitlilik sağlanabilmektedir.

#### *1.1.2.3.4. Elitizm Yöntemi*

Uygunluk değerinin en yüksek olduğu bireyin bir sonraki popülasyonda yer alması sağlanarak genetik algoritmanın performansı artırılmaktadır. Çaprazlama ve mutasyon aşamaları uygulandıktan sonra popülasyonda yer alan en iyi bireyler yok edilebilir [14]. Uygunluk değeri yüksek olan birey veya bireyler yeni oluşturulacak

popülasyona dahil edilerek durdurma kriterine ulaşıncaya kadar popülasyonun iyileştirilmesine katkı sağlanır.

#### 1.1.2.4. Standart Çaprazlama Operatörleri

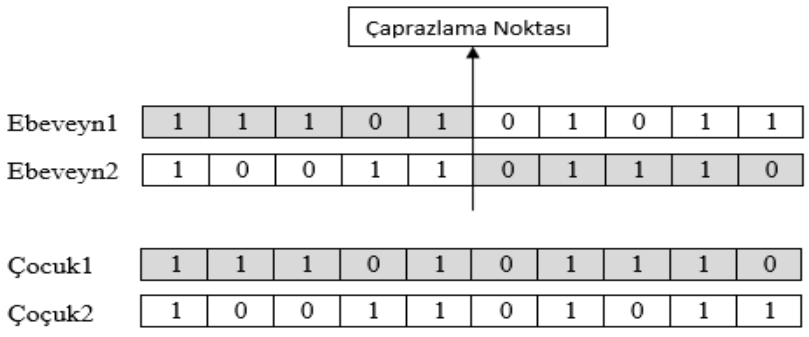
İki veya daha fazla ebeveynin kromozomlarının yerleri değiştirilerek uygunluk değeri yüksek olan yeni bir kromozom elde etme işlemidir [7]. Çaprazlama işlemleri üç adımda gerçekleşmektedir [9].

- Adım 1: Çaprazlama işlemi için ebeveynler seçilir.
- Adım 2: Seçilen kromozomlar içerisinde rastgele çapraz noktalar belirlenir.
- Adım 3: Çapraz noktaları takiben genler yer değiştirilerek yeni kromozomlar elde edilir.

Çaprazlama işleminin yapılıp yapılamayacağı çaprazlama oranına $(Pc)$ bağlı olarak değişmektedir. Çaprazlama oranı genellikle [0.6, 1] aralığında bir değerle temsil edilmektedir, 0 ve 1 arasından rassal olarak seçilen değer $Pc$ oranından küçük olduğunda ebeveynler arasında çaprazlama işlemi gerçekleştirilir. Rassal değer $Pc$ oranından büyük olduğunda ise çaprazlama işlemi gerçekleştirilmeyecektir.

##### *1.1.2.4.1. Tek Noktalı Çaprazlama*

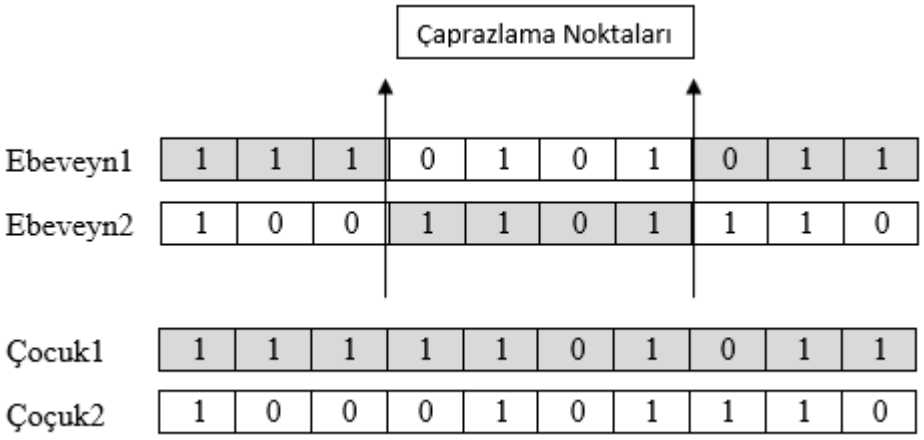
Çaprazlama işlemi olarak öncelikle ebeveyn çifti seçimi yapılır. Rastgele bir çaprazlama noktası belirlendikten sonra bu noktadan önceki gen bilgileri ebeveyn 1'den alınır, belirlenen noktadan sonraki kısım ise ebeveyn 2'den alınarak bir sonraki kuşağa aktırılacak birey oluşturulur [15]. Şekil 1.5'de basit bir tek noktalı çaprazlama örneği görülmektedir.




Şekil 1.5 Tek Noktalı Çaprazlama ile İki Ebeveynden Oluşturulan İki Çocuk.

#### *1.1.2.4.2. Çok Nokta Çaprazlama*

Seçilen ebeveyn çiftinden çocukları üretirken, çaprazlama noktası birden fazla seçilir. İki ebeveyn için çaprazlama ve genleri ayırma noktaları rasgele olarak seçilmektedir [16]. Şekil 1.6'da birden fazla çaprazlama noktası seçimi sonucunda meydana gelen yavrular görülmektedir.




Şekil 1.6 Çok Nokta Çaprazlama ile Oluşturulan Çocuklar.

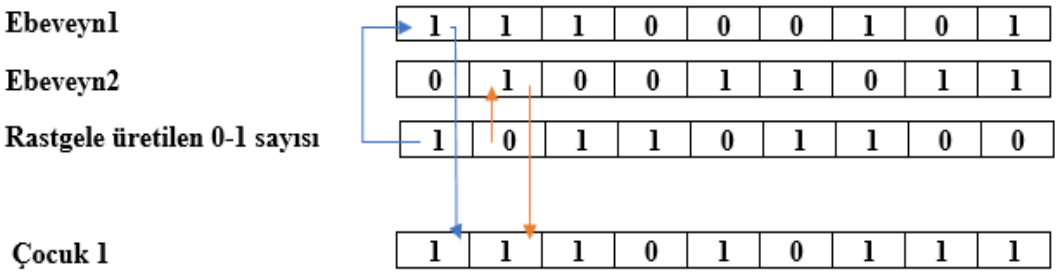
### 1.1.2.5. Çoklu-Ebeveyn Çaprazlama Operatörleri

Canlılar doğada nesillerini devam ettirmek için ürerler. Üreme ise canlılarda aseksüel (bir ebeveyn) ve biseksüel (iki ebeveyn) olmak üzere iki çeşittir [17]. GA'da nüfus çeşitliliğini artırmak için çaprazlama işleminde iki veya ikiden daha fazla sayıda ebeveyn kullanarak yavrular üretilebilir. Bu çalışmada kullanılacak çaprazlama operatörleri; Tek biçimli (Bir örnek) çaprazlama, oluşum temelli çaprazlama, uygunluk temelli çaprazlama ve diyagonal çaprazlama olarak belirlenmiştir.

#### *1.1.2.5.1. Tek Biçimli (Bir Örnek) Çaprazlama*

Tek biçimli çaprazlamada (İng.: Uniform Crossover) genler ebeveynlerden rastgele alınacak şekilde kopyalanarak çocuk bireyler oluşturulmaktadır [18]. Tek biçimli çaprazlama işleminde iki tane ebeveyn seçilir. Ebeveynlerden eşit olarak seçilecek genlerden iki tane çocuk meydana getirilir. Çocuktaki her gen rastgele olarak ebeveynlerden seçilir. Rastgele 0 ve 1 arasında bir sayı üretilir. Rastgele sayı değeri 1 ise çocuğun geni ebeveyn 1'den alınacak, 0 ise çocuğun geni ebeveyn 2'den alınacaktır [16]. Bu işlem kromozomun uzunluğu boyunca tekrar edilecektir. Şekil

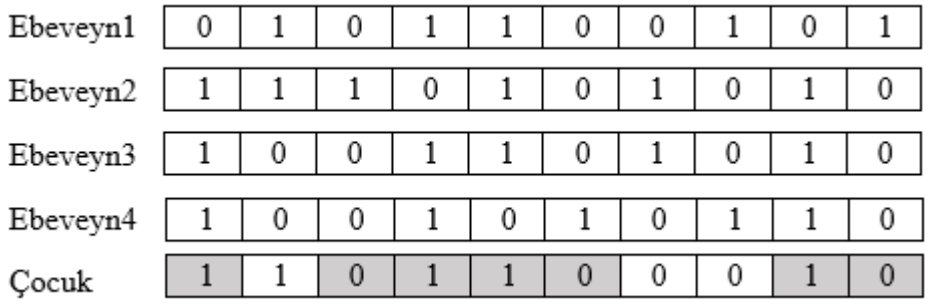
1.7’de tek biçimli çaprazlama ile ebeveynlerden seçilen genlerle çocuk oluşumu görülmektedir. Çoklu ebeveyn uygulamasında oluşturulacak yeni bireyin, her bir geni ebeveynlerden rastgele olarak seçilerek genetik çeşitlilik sağlanmaktadır.




Şekil 1.7 Tek Biçimli Çaprazlama.

#### *1.1.2.5.2. Oluşum Tabanlı Çaprazlama*

Oluşum tabanlı çaprazlama (İng: Occurrence Based Crossover) ile uygunluk değerlerine göre seçilen ebeveynlerden seçilecek genlerin belirli bir konumda en çok ortaya çıkan değerinin seçilebilecek en iyi değer olduğu belirtmektedir. Şekil 1.8’de ebeveynlerin birinci genlerine bakıldığında 1’lerin sayısının 0’lardan daha fazla olduğu görülmektedir, bu durumda çoğunluk 1 olduğu için ebeveynlerden elde edilecek bireyin birinci geninin değeri 1 olacaktır. Bireyin bütün genlerinin değeri belirlenene kadar bu işlem tekrar edecektir. Şekil 1.8’de görüldüğü üzere, seçilen konumda çoğunluk bir değer yoksa yani 0 ve 1’lerin sayısı eşitse seçilecek gen, ya ebeveyn1’den alınır ya da ebeveynler arasından rastgele seçilir [18].



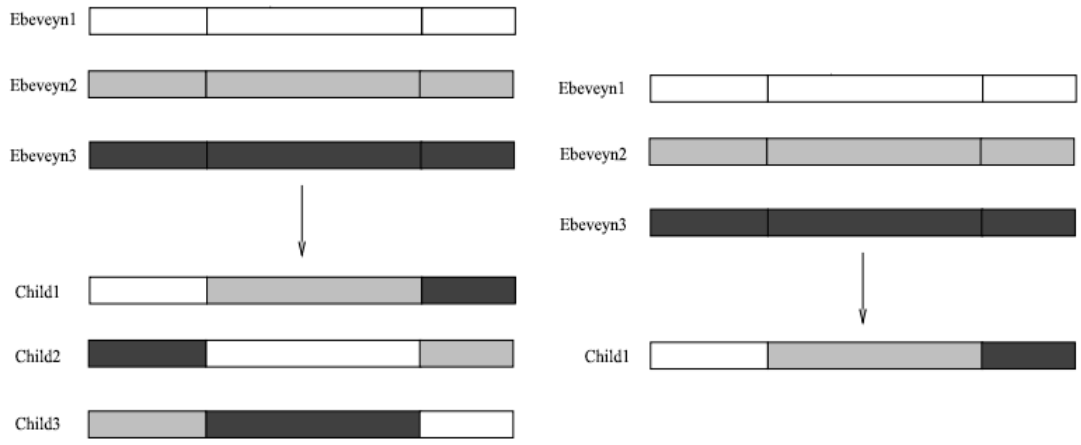

Şekil 1.8 Oluşum Tabanlı Çaprazlama Gen Seçimi.

#### *1.1.2.5.3. Uygunluk Temelli Çaprazlama*

Uygunluk temelli çaprazlama (İng: Fitness Based Crossover) da turnuva aşaması sonucunda belirlenen ebeveynlerden hangi kromozomun alınacağına karar verilirken rulet tekerlek seçimi kullanılmaktadır. Rulet tekerlek seçimi yöntemi ile popülasyondaki tüm bireylerin uygunluk değerleri toplanır ve her bireyin seçilme olasılığı, uygunluk değerinin bu toplam değere oranı kadardır [18]. Rulet tekerleğinde seçilen ebeveynden ilgili gen alınarak birey oluşturulmaktadır. Bu işlem bireyin tüm genleri seçilene kadar devam etmektedir.

#### *1.1.2.5.4. Diyagonal Çaprazlama*

Diyagonal çaprazlama (İng: Diagonal Crossover) yöntemi ise çok ebeveynli üreme için bir veya birden fazla çaprazlama noktalarını kullanarak n tane ebeveyn için n tane çocuk birey oluşturur. Şekil 1.9'da çaprazlama noktaları olarak n-1 tane nokta seçilir. N-1 tane çaprazlama noktası ebeveyni n tane parçaya ayırır. Oluşturulan her çocuk ebeveynlerin farklı parçalarını içerdiğinden popülasyonda çeşitliğe sebep olmaktadır [19].



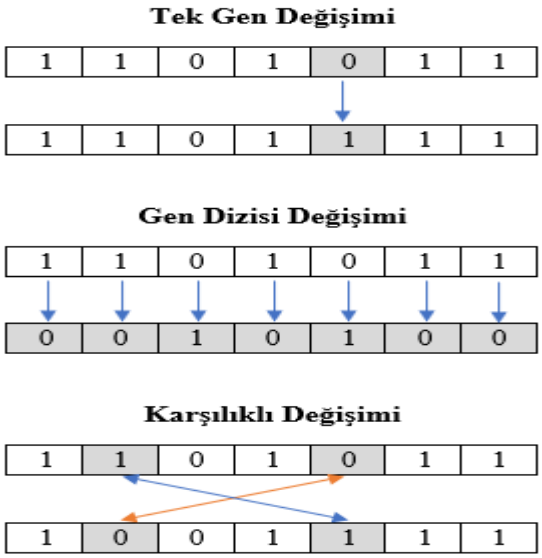

Şekil 1.9 Diyagonal Çaprazlama ile Üç Ebeveyn ve Üç Çocuk (Solda) ve Üç Ebeveyn ve Bir Çocuk Sağda [19].

### 1.1.2.6. Mutasyon

Çaprazlama aşamasından sonra yeni oluşturulan yavruların genleri atalarına benzeyebilmektedir. Bu durum çeşitliliğin azalmasına neden olmaktadır. Bireyler birbirlerine benzedikleri için genetik algoritma problemin çözümünde yerel en iyiye takılarak çözümün kısıtlanmasına sebep olmaktadır. Mutasyon işlemi ile üretilen

yavruların genlerinde rastgele değişimler yapılarak popülasyon içinde çeşitlilik sağlanmaktadır. Problemin tanımlandığı aşamada [0, 1] aralığında mutasyon oranı $(Pm)$ belirlenir. Bireydeki her gen için rastgele 0 ve 1 arasında bir mutasyon olasılık değeri verilir. Pm değerinden büyük olasılığa sahip olan bireyler mutasyon işlemine tutulmazlar. $Pm$ değerinden daha küçük olasılığa sahip olan bireyler ise mutasyon işlemi ile değişime uğrarlar. İkili kodlama yöntemi ile oluşturulan kromozomlarda bulunan değeri 0 olan genin yeni değeri 1, değeri 1 olan genin yeni değeri ise 0 olarak değiştirilerek mutasyon işlemi gerçekleştirilir [6]. Bu sayede popülasyon içerisinde genetik çeşitlilik sağlanır. Mutasyon işlemi sonucunda kromozomun genlerinin değeri 0 ise en az bir genin değerinin 1 olması sağlanır. Çaprazlama yöntemlerinde olduğu gibi, problemin çeşidine göre mutasyon yöntemleri değişmektedir [20]. Şekil 1.10'da mutasyon yöntemleri ile ilgili örneklerin açıklaması gösterilmiştir.

- Tek gen değişimi: Pm olasılığına göre rastgele seçilen bir gendeki bit ters çevrilir.
- Gen dizisi değişimi: Pm olasılığına göre kromozomdaki tüm genler bit bit ters çevrilir.
- Karşılıklı değişim: Pm olasılığına göre kromozomda rastgele seçilen iki genin yerleri değiştirilir.




Şekil 1.10 Mutasyon Yöntemleri Örnek Uygulamaları.

## 1.2. ÖZNİTELİK SEÇİMİ

Öznitelik seçimi, genetik algoritmada verilerin analizinde ve işlenmesinde önemli aşamalardan biridir. Veri kümelerinde eksik ve hatalı veriler olabilmektedir. Veri kümesi içerisinde yer alan alakasız ve ilgisiz veriler normalizasyon yapılmadan kullanılırsa analizlerde yanıltıcı sonuçlar elde edilebilmektedir. Öznitelik seçimi (Diğer adıyla özellik seçimi) ile problemin çözümü ile ilgisi olmayan değişkenleri azaltarak en faydalı verilerin seçilmesi ile birlikte hesaplama maliyetinin azaltılmasıyla çözümün doğrulunun artırılması hedeflenmektedir. Öznitelik seçimi ile veri kümesi içerisindeki değerlendirme kriterlerine uygun olan n adet veri içerisinde yer alan özelliklerden, en iyi k adet özelliği seçerek alt küme oluşturmaktır [21].

### 1.2.1. Öznitelik Seçimi Aşamaları

Öznitelik seçimiyle problemin çözümünden kullanılacak veri kümesi içerisinden daha küçük boyutlarda bir özellikler alt kümesi oluşturulmaktadır. Alt küme içerisinde yer alan özellikler, probleme özgü değerlendirme kriterlerine göre belirlenen aşamalardan geçirilerek, hangi özelliğin seçilip seçilmeyeceğine karar verilerek seçilen özellikler belirlenen alt kümeye dahil edilmektedir [22]. Öznitelik seçimi aşamaları ile ilgisiz ve gerekli olmayan verilerden arındırılmış olarak seçilen alt kümeler ile veri kümesi içerisindeki gürültüler azaltılır ve veri kalitesi artırılarak kullanılacak olan algoritmanın başarı oranı arttırılır.

### 1.2.2. Öznitelik Seçim Yöntemleri

Öznitelik seçiminde kullanılacak olan yöntemler, filtreleme yöntemleri sarmal yöntemler ve gömülü yöntemler olarak 3 kategoriye ayrılabilir.

#### 1.2.2.1. Filtreleme Yöntemleri

Filtreleme yöntemlerinde, bir öğrenme algoritması kullanılmadan önce istatiksel olarak belirlenen ölçülere bağlı olarak özellik seçim işlemi yapılmaktadır [22]. Veri kümesi içerinde yer alan özellikler, sınıflandırmanın doğru yapılabilmesi için değerlendirme kriterlerine uygun olarak belirlenen bireysel skorlar hesaplanarak, en yüksek skor değerlerine sahip olan özelliklerin seçilmesiyle, en iyi özelliklerden oluşan alt küme oluşturulur.

#### 1.2.2.2. Sarmal Yöntemler

Sarmal Yöntemler ile özellik seçimi sınıflandırma algoritmasına bağlı olarak öğrenme algoritmaları ile en iyi tahmin yöntemiyle gerçekleştirilmektedir. Öznitelik seçimi işlemi, veri kümesi içerindeki öznitelikler en iyi tahminlime yapılarak alt kümede istenen sayıda özniteliğe ulaşılan kadar devam edilir [23]. İki çeşit sarmal alt küme oluşturma yaklaşımı vardır. İleri yönlü arama: Arama işlemine boş bir öznitelik kümesi ile başlanır her aşamada en iyi olan özellik alt kümeye eklenir, karar verilen durdurma kriteri sağlanıncaya kadar alt kümeye öznitelik ekleme işlemine devam edilir. Geri yönlü arama; En iyi öznitelik alt kümesi bulunana kadar arama işlemine bütün öznitelikleri içeren alt kümeyle başlanır ve belirlenen durdurma kriteri sağlanana kadar her aşamada en kötü öznitelik alt kümeden çıkartılmaktadır. Sarmal yöntemlerle en iyi özelliklerin seçiminde filtreleme yöntemlerine göre daha başarılı sonuçlar alınmaktadır fakat büyük veri içeren kümelerde hesaplama maliyeti daha yüksek olmaktadır [22].

#### 1.2.2.3. Gömülü Yöntemler

Gömülü yöntemlerde öznitelik seçimi için alt küme belirlemede sınıflandırma algoritması ile özellik seçimi algoritması birlikte çalışmaktadır. Sınıflandırma ve öznitelik seçimi eş zamanlı yapıldığı için hesaplama maliyeti filtreleme yöntemlerine göre dezavantajlıdır. Gömülü yöntemler, sarmal yöntemlerde alt kümeleri yeniden sınıflandırmak için harcanan hesaplama zamanını azaltmak istemektedir[24]. Karar Ağaçları ve Destek Vektör Makinaları hem sınıflandırma hem de özellik seçim işleminde kullanılmaktadır.

## 1.3. DENETİMLİ ÖĞRENME

Makine öğrenmesi uygulamalarında giriş ve çıkış verilerinde ilişki bulunan yöntemlerden birisi de denetimli öğrenme yöntemidir. Denetimli öğrenme yönteminde kullanılan veri kümesi içerisinde yer alan verilerin etiketli veri olması gerekmektedir. Örneğin öğrenci bilgilerinin yer aldığı bir veri kümesinde, öğrencilerin yaşları, cinsiyetleri ve başarı oranları gibi bilgilerin bulunduğu veri kümeleri, etiketli veri kümeleridir. Tasarlanan sistemi eğitmek için kullanılan etiketli verilerin çoğunluğu sistemi eğitirken, geriye kalan veriler sistemi test etmek için kullanılmaktadır. Sistemi

eğitmek için kullanılan verilerle, eğitim işlemi sonucunda ortaya çıkan değerlerin gerçekten istenilen değerler olup olmadığını denetleyen veya tahmin eden modele denetimli öğrenme modeli denilmektedir [25].

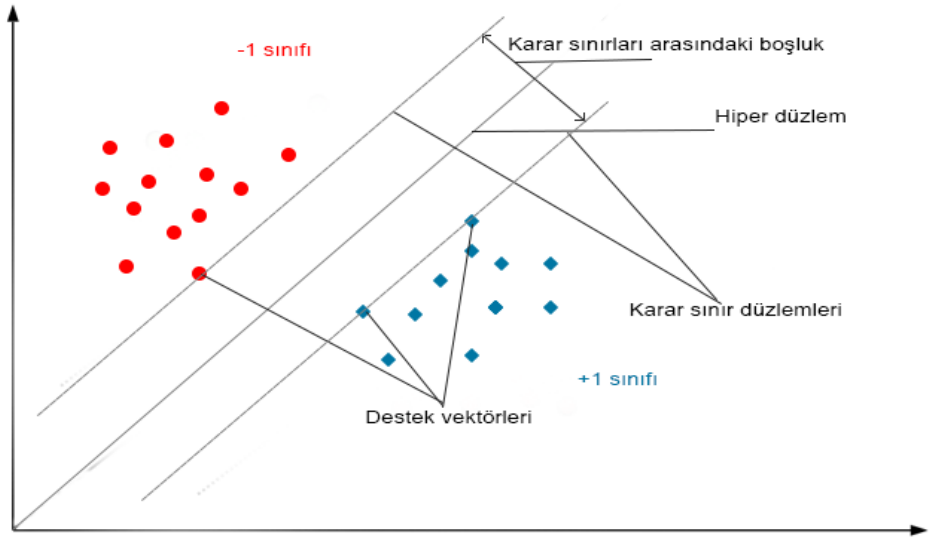
Sınıflandırma algoritmalarıyla veri kümesi içerisinde yer alan sınıfı belli olmayan test verilerini, önceden belirlenen etiketlere uygun olarak belirlenen çeşitli sınıflara veriyi ayırma işlemidir. Sınıflandırma uygulamalarında denetimli öğrenme modelinden yararlanılmaktadır. Literatürde en çok kullanılan denetimli sınıflandırma yöntemlerinden bazıları; Karar Ağacı, Naive Bayes, K-En Yakın Komşu (İng.: K-Nearest Neighbors, KNN) ve Destek Vektör Makinalarıdır. Bu tezde Destek Vektör Makinaları (DVM) verileri, çeşitli vektörler yardımıyla sınıflara ayırmak için kullanılan denetimli sınıflandırma yöntemlerinden birisi olarak tercih edilmektedir.

### 1.3.1. Destek Vektör Makinaları

DVM'ler sınıflandırma veya regresyon problemlerinin çözümünde kullanılan denetimli bir makine öğrenme algoritmasıdır. DVM, algoritması Vapnik tarafından geliştirilmiştir [26]. Birden çok etiketten oluşan veri kümelerinde sınıflandırma DVM ile vektörler yardımıyla gerçekleşmektedir. DVM algoritması kullanılarak doğrusal veya doğrusal olmayan sınıflandırma işlemleri yapılabilmektedir. DVM'ler ile ilk başlarda ikili veri kümeleri doğrusal vektörlerin kullanılmasıyla sınıflara ayrılırken, öte yandan çekirdek fonksiyonlarının kullanılmaya başlanmasıyla birlikte doğrusal olmayan birçok karmaşık ve iç içe geçen sınıfın , sınıflandırma işlemleri de yapılmaya başlanmıştır [25].

Şekil 1.11'de görüldüğü üzere DVM iki sınıflı sınıflandırmada, eğitim verileri kullanılarak destek vektörlerinin üzerinde yer aldığı karar sınırlıları arasında ki en iyi boşluk (İng.: Margin) bulunarak hiper düzlem çizilir. Sınıfları birbirinden ayırmak için her iki sınıfa ait karar sınırlarına eşit mesafede olan hiper düzlem çizgisi çizilmelidir [27]. Destek vektörleri içerisinde bulunduğu sınıfın karar sınır çizgilerini belirler. Destek noktalarında yer alan değerler değişirse karar sınırları da değişecektir. Karar sınırları arasında yer alan boşluk mümkün olan en iyi değer olmalıdır. Bu sayede sınıflandırma için gelen test verileri ve yeni gelen veriler en uygun sınıfa

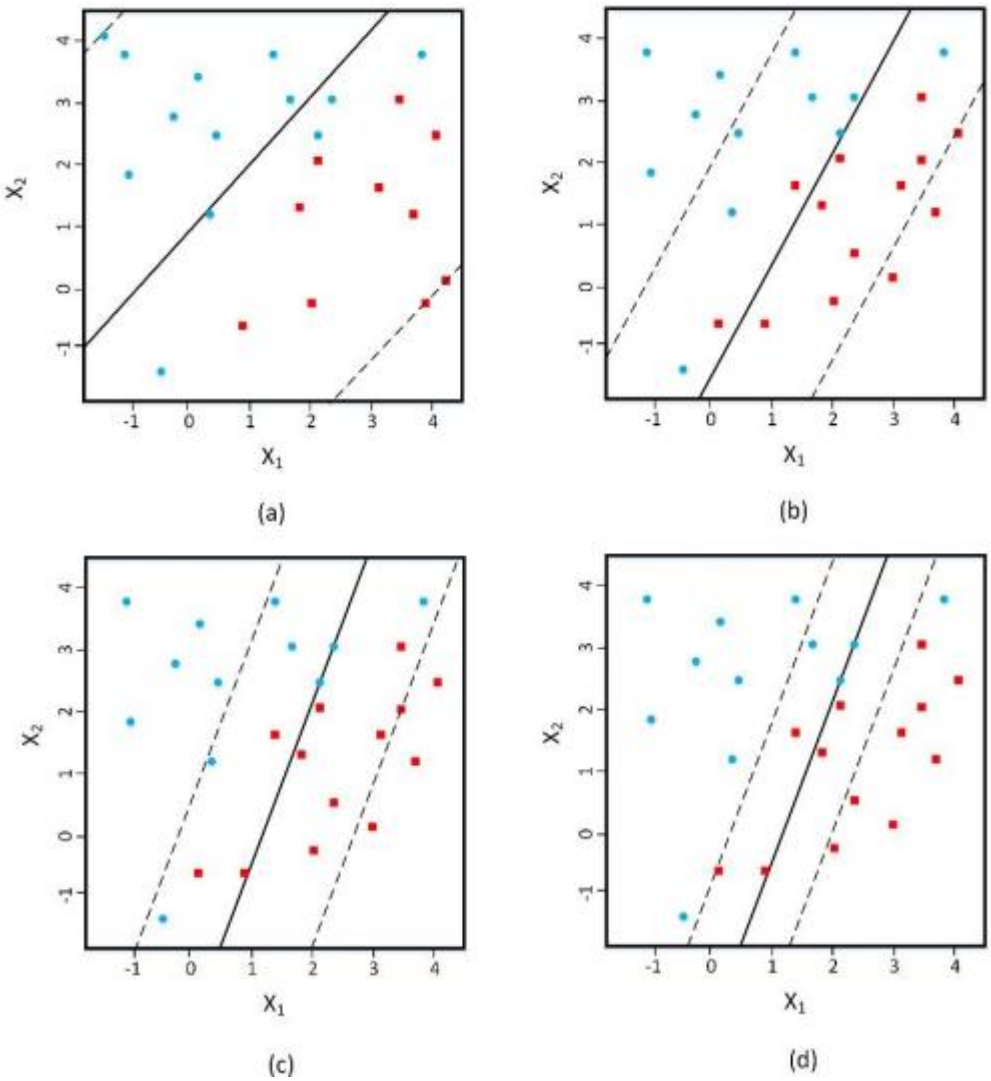
yerleştirilecektir. Veri kümesine yeni değerler eklenirse çizilen karar sınırları ve hiper düzlem doğrularının konumu değişecektir.




Şekil 1.11 İki Sınıflı Doğrusal Olarak Ayrılabilen Destek Vektör Makinası.

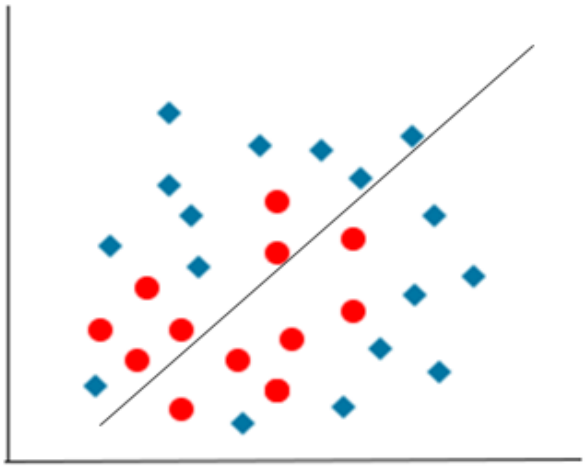
Tasarlanan modelde karar sınırları arasında ki alana gelen aykırı değerler olabilmektedir. Doğrusal sınıflandırmada bu durumda sistemin varyansı yükselmektedir. Varyans değeri tahmin edilen verilerin, eğitim verileri etrafında yerleştikleri alanların dağılımı hakkında bilgi vermektedir [25]. Varyans değerinin yüksek olması durumunda tasarlanan model öğrenmek yerine ezberlemeye (Aşırı öğrenme, İng.: Overfitting) çalışacaktır.

Eğitim örneklerini doğru bir şekilde sınıflandırmak için karar sınırları arasında yer alan boşluğun kontrol edilerek hata oranını azaltmak ve esneklik sağlayabilmek için $C$ parametresi kullanılmaktadır. $C$ parametresi ile karar sınırları arasındaki boşluk artırılıp azaltılmaktadır. $C$ parametresinin değeri arttıkça karar sınırları arasındaki boşluk azaltılmaktadır. Şekil 1.12(d)'da $C$ değerinin büyük seçilmesi durumunda esneklik azalacaktır ve sınıflandırmada katı bir tutum sergilenecektir. $C$ değerinin küçük belirlenmesi durumunda Şekil 1.12(a)'da görüldüğü üzere esneme payı büyük olacaktır. Tasarlanan modelde aşırı öğrenme gerçekleşirse $C$ parametresinin azaltılması gerekmektedir.




Şekil 1.12 C Parametresi Değerine Göre Esneme Payları [25].

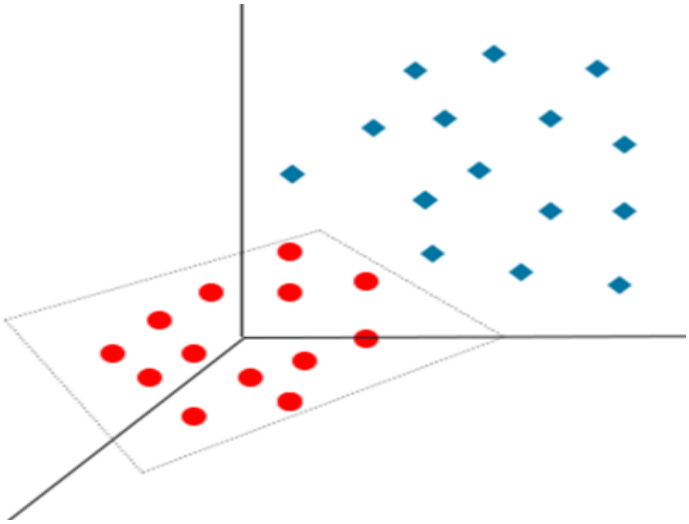
DVM algoritması ile doğrusal olan sınıflandırma amaç verileri ayıracak doğrular içerisinden karar sınırları arasındaki mesafeyi maksimum olacak şekilde tespit ederek mümkün olan en iyi ayırma hiper düzlem doğrusunu bulmaktır. DVM ile Şekil 1.13'de görüldüğü gibi bir doğru çizilerek ayrıştırılamayan veri setlerini başka yöntemlerle ayırmak gerekmektedir. Birbiri içerisine giren sınıfları ayırmak için üçüncü bir boyut eklenerek sınıflar birbirlerinden ayrılmaktadır.



Şekil 1.13 Doğrusal Olarak Ayrılamayan Destek Vektör Makinası.

Veri kümesindeki veriler doğrusal düzlemlerle birbirinden ayrılamayan veriler içeriyorsa çekirdek kavramı kullanılmaktadır. Çekirdek, kendisine verilen giriş verilerini alarak sınıfları birbirinden ayıran boyutu ekleme işlemi gerçekleştirir. Çekirdek kullanarak bir boyutta ayıramadığımız verileri farklı boyutlara taşıyarak çözümler getirilebilmektedir. Hiper düzlem doğrusu ile ayrılamayan veri setleri Şekil 1.14'de görüldüğü üzere sınıfların boyutunun arttırılması ile birbirlerinden ayrılmaktadırlar. Çekirdek fonksiyonu ile DVM'ler farklı birçok uygulama tarafından sınıflandırma probleminin çözümünde tercih edilmektedirler. Literatürde kullanılan çekirdek fonksiyonlarının bazıları şunlardır;

- Doğrusal,
- Polinom,
- Radyal Tabanlı



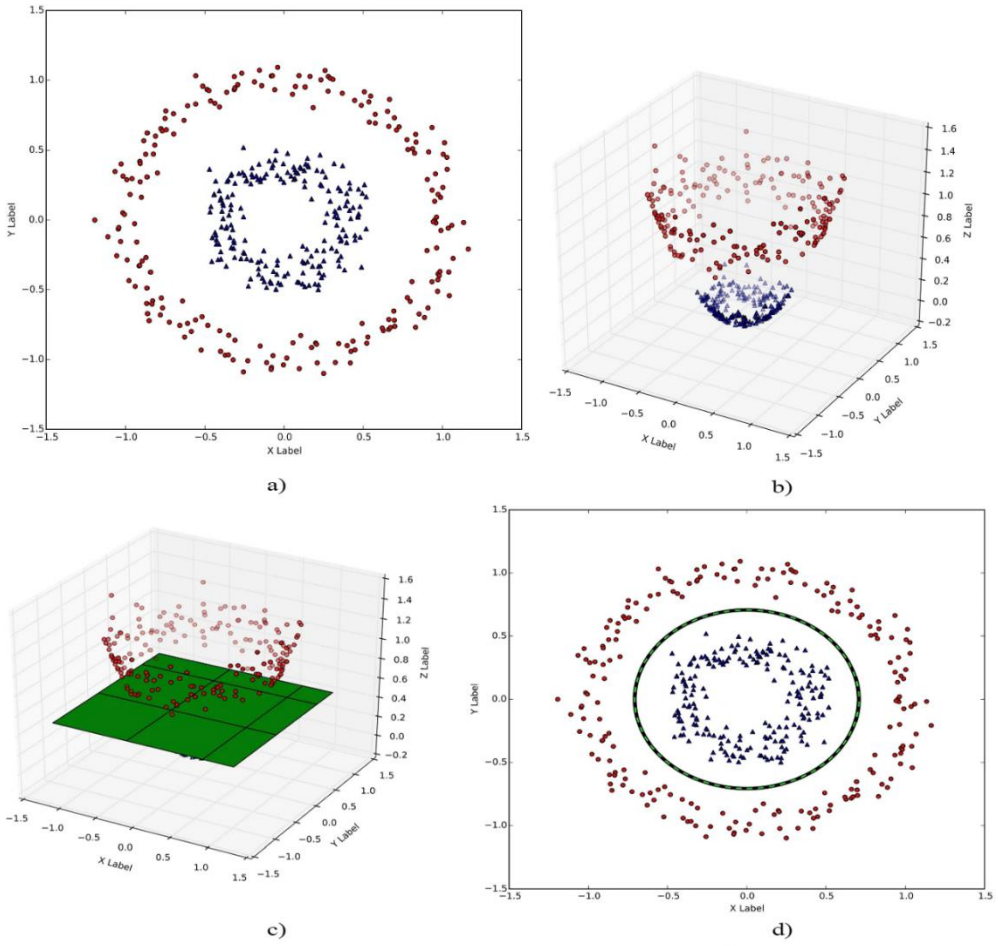
Şekil 1.14 Boyut Artırma ile Sınıfları Birbirinden Ayırma.

#### 1.3.1.1. Doğrusal Çekirdek Fonksiyonu

Doğrusal olarak sınıflandırma yapılacak modellerde, tercih edilen çekirdek türüdür. İki farklı sınıfı düzlemlerle birbirinden ayırmak için mümkün olan en iyi karar sınırlarını bularak, sınıfları birbirinden ayırmaktadır. Bir başka deyişle verileri birbirinden ayırmak için çizilen birçok doğru arasından en iyi doğruyu bularak sınıflandırma işlemi gerçekleştirilmektedir.

#### 1.3.1.2. Polinom Çekirdek Fonksiyonu

İki boyutlu bir düzlemde sınıflandırmanın bir doğru ile yapılamadığı durumlarda sınıflandırma işlemini polinom çekirdek fonksiyonu ile gerçekleştiririz. Polinom çekirdek fonksiyonu ile iki boyutta sınıflandırma yapamadığımız için üçüncü boyut eklenerek verileri sınıfları ayırmak için düzlem kullanılmaktadır. Şekil 1.15(a)'da veriler iki boyutlu uzayda gözükmektedir, Şekil 1.15(b)'de veriler üç boyutlu uzayda gözükmektedir, Şekil 1.15(c)' üç boyutlu uzayda karar sınırı için oluşturulan düzlem gözükmektedir ve Şekil 1.15(d) ise karar sınırının uzayda izdüşümü gösterilmektedir.




Şekil 1.15 Polinom Çekirdek Fonksiyonu ile Boyut Artırma [28].

1.3.1.1. Gauss Radyal Tabanlı Çekirdek Fonksiyonu

Veriler doğrusal olarak iki boyutlu düzlemde sınıflandırılmadığı zaman Gauss radyal tabanlı çekirdek fonksiyonu kullanılarak sınıflandırma işlemi daha yüksek boyutlu uzayda gerçekleştirilir [29]. Gauss radyal tabanlı çekirdek fonksiyonunda gamma parametresi kullanılarak sınıflandırma düzleminde yer alan noktaların belirli noktalara ne kadar benzediği normal dağılım ile bulunmaktadır. Ayarlanabilir gamma değeri ile normal dağılım arasında ters orantı bulunmaktadır. Gamma değeri yüksek olursa çekirdeğin genişliği büyük olur ve veriler uzayda düz bir hiper düzleme yerleşirler. Bu sebeple tasarlanan model doğrusal davranmaya başlar ve hatalı sınıflandırmalar yapılabilir. Gamma değerinin küçük olması karar sınırlarını keskinleştirir, destek vektörlerini, eğitim ve test hatalarının sayısı artırır [28]. Gamma değeri çok küçük olursa çekirdeğin genişliği azalır ve model küçük bir veri kümesine odaklanılır. Bu nedenle modelde aşırı uyum sorunları ve yüksek varyans oluşur.

## 1.4. LİTERATÜR ÇALIŞMALARI

Literatürde öznitelik seçimi için genetik algoritmanın kullanılmasıyla ilgili birçok farklı çalışma bulunmaktadır. Farklı uygulama alanlarında, farklı veri kümelerinden öznitelik seçimi için genetik algoritmalarla çalışmalar yapılmıştır. Aşağıda yapılan bazı çalışmalar hakkında bilgi verilmiştir.

2014 yılında yapılan [30] çalışmada ilgili sınıflandırıcının performansını arttırmak için GA tabanlı özellik seçimi kullanılmıştır. Bu çalışmada Flavia görüntü veri kümesinden yüz adet özellik elde edilmiştir. Çalışmada bu özellikleri elde etmek için k-NN tabanlı sınıflandırma hata oranı ile yeni bir uygunluk fonksiyonu kullanan GA tabanlı öznitelik seçici geliştirilmiştir. Çalışma elde edilen sonuçlar WEKA yazılımında çeşitli öznitelik seçicilerle karşılaştırılmıştır. Çalışmada sınıflandırıcı doğruluğu açısından WEKA özellik seçicilerden daha iyi sonuçlar elde edilmiştir.

2016 yılında yapılan çalışmada [31] çok sayıda ebeveynin GA performansına etkisini incelemek için ikili kodlanmış çok-ebeveynli GA önerilmiş ve geleneksel iki ebeveynli GA ile karşılaştırma yapılmıştır. Çalışmada çaprazlama operatörü olarak, çok sayıda ebeveynden yeni yavrular elde etmek için diyagonal çaprazlama kullanılmıştır. Yapılan çalışma sonucunda çok sayıda ebeveynin önemli bir artış olmaksızın daha hızlı yakınsadığını göstermektedir.

2019 yılında yapılan çalışmada [32] çocuklarda ve ergenlerde depresyon teşhisine yardımcı olmak için sınıflandırıcıların performansını artırmak ve en alakalı öznitelikleri seçmek için genetik algoritmaları kullanılmıştır. Bu çalışmada elde edilen sonuçlarda en çok kişinin kendi görünüşüyle nasıl hissettiğini ifade ettiğini gösteren özellik kullanılmıştır. GA ile seçilen 55 öznitelikten oluşan veri kümesinin kullanılması, sınıflandırıcıların performansını artırarak, depresyon tanısının teşhis edilmesine katkı sağlamıştır.

2019 yılında yapılan bir diğer çalışmada [33] ağ güvenliği ve izinsiz giriş tespiti alanındaki öznitelik seçimi problemini ele almaktadır. Makine öğrenmesiyle ağ güvenliği ve saldırı tespit sistemleri geliştirilmektedir. Makine öğrenme yöntemleri ağ trafiği izlenerek saldırgan tespit edilmektedir. Bu çalışma yüksek veri boyutunun performans üzerindeki olumsuz etkilerini kaldırmak ve benzersiz verilerin birçoğunu

minimum sayıda koruyan bir yöntem tasarlamak için GA tabanlı özellik seçimi için yeni bir uygunluk fonksiyonu geliştirmektedir. Ağ veri kümeleri üzerinde test edilen yöntem maksimum %99,80 doğruluk elde ederek, doğrulukların arttığını göstermiştir.

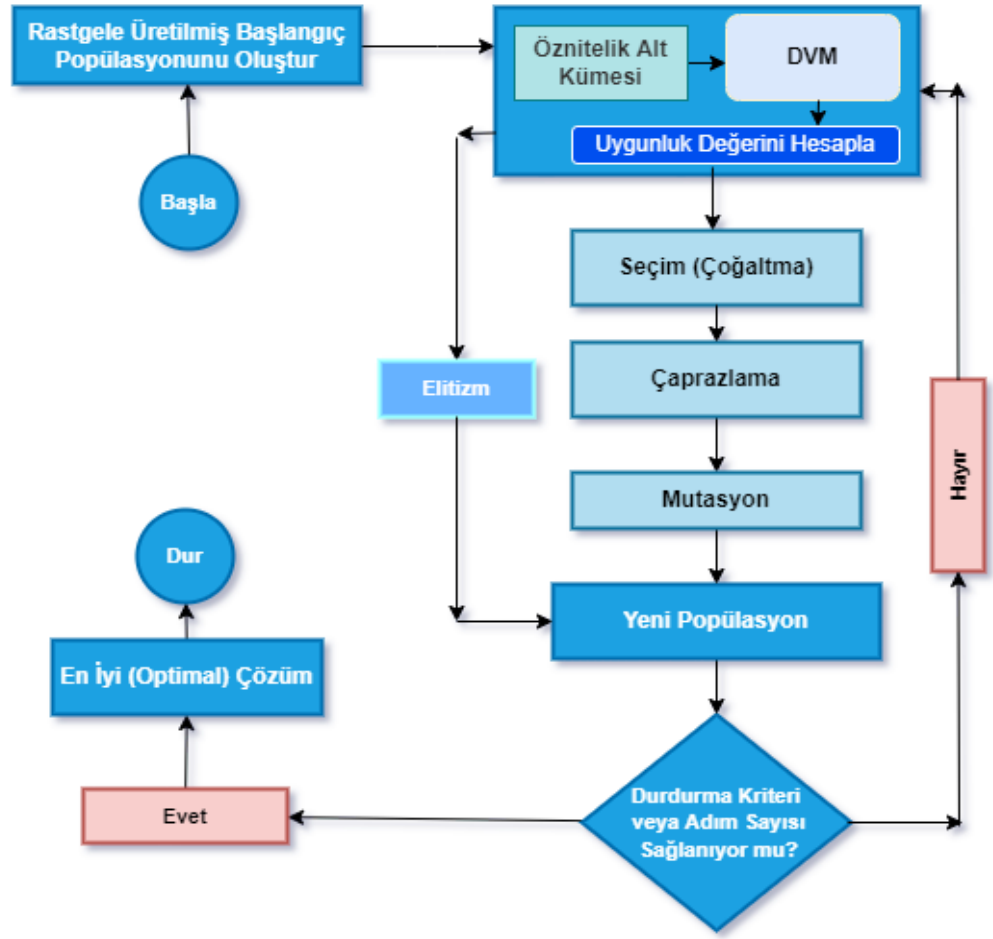
2021 yılında yapılan çalışmada [34] hastaların, akciğer kanseri evresini erken teşhis etmek için veri kümesinin ilgisiz özelliklerini azaltmak ve sınıflandırma hızını artırmak için gerekli öznitelik seçimi için k-NN sınıflandırma tekniğini kullanan GA uygulanmıştır. Yapılan çalışmada sınıflandırma doğruluğunun arttığı gözlenmiştir.

2021 yılında yapılan başka bir çalışmada makine öğrenmesi kullanılarak tıp alanında karaciğer kanser verilerinin sınıflandırması yapılmıştır. Sınıflandırma doğruluğunu artırmak için, makine öğrenmesinin yaygın olarak kullanılan Rastgele Orman ve XGBoost (Extreme Gradient Boosting) algoritmasıyla birlikte özellik seçimi için GA uygulanmıştır. Çalışmada XGBoost'un Rastgele Orman algoritmasına göre daha yüksek doğruluk puanı elde ettiği görülmüştür [35].

# İKİNCİ BÖLÜM

## 2. ÖZNİTELİK SEÇİMİ İÇİN GENETİK ALGORİTMA

Bu tezde öznitelik seçimi içim çoklu-ebeveyn çaprazlama operatörlerinin GA içerisinde kullanılması ve performanslarının karşılaştırılması hedeflenmektedir. Çoklu-ebeveyn çaprazlama operatörleri iki veya daha fazla sayıdaki ebeveynden yavrular üretmek için tasarlanmıştır. Yapılan bu tezde kullanılan GA'nın öznitelik seçimi için akış diyagramı Şekil 2.1'de ve GA'nın temel akışını gösteren sözde kodu Şekil 2.2'de verilmiştir.




Şekil 2. 1 Öznitelik Seçimi İçin Genetik Algoritma Akış Diyagramı

Başlangıçta belirlenen birey sayısı kadar geliştirilecek popülasyon rastgele oluşturulur ve bireylerin öznitelik sınıflandırma hata oranı ve öznitelik sayısını dikkate alan Denklem 2.1'deki formül ile uygunluk değerleri hesaplanır. GA'da minimizasyon problemi için uygunluk değerinin en düşük olan birey küresel olarak en iyi parçacık olarak tanımlanmaktadır. Her yinelemede yeni ve geliştirilecek popülasyonda bir önceki popülasyonda ki en iyi uygunluk değerine sahip bireyin yer alması için elitizm yöntemi kullanılır. Elitizm uygulandıktan sonra popülasyonu oluşturacak bireylerin seçim işleminde elitizm birey sayısı kadar az seçim yapılmalıdır. Örneğin popülasyon boyutu 50 olarak belirlenen bir problemde elitizm uygulanacak birey sayısını 1 olarak belirlersek, geriye kalan 49 bireyi çaprazlama operatörleri ile belirleriz.

Bir sonraki aşamada başlangıçta belirlenen çaprazlama operatörü ile bireyler arasında gen alışverişinin yapılması sağlanarak genetik çeşitlilik hedeflenmektedir. Çaprazlama aşamasında yer alacak ebeveynlerin belirlenmesinde turnuva seçim yöntemi kullanılmaktadır. Seçim aşamasında hangi bireylerin ebeveyn olarak çaprazlama işleminde yer alacağı belirlenmektedir. Belirlenen turnuva seçimi yöntemiyle bir önceki popülasyondan rassal olarak belirlenen iki birey arasında ki uygunluk değerine göre yapılan yarışmadan kazanan birey seçilmektedir. Seçim işlemi istenen sayıda ebeveyn seçilene kadar tekrarlanmaktadır. Ebeveynler belirlendikten sonra seçilen çaprazlama operatörüne göre yeni birey oluşturulur ve popülasyona dahil edilir.

Seçim ve çaprazlama işlemlerine rağmen bireylerin genleri, ebeveynlerine benzeyebilmektedir. Mutasyon işlemi ile bireylerin genlerinde rastgele değişimler yaparak bir önceki popülasyondan farklı bireyler oluşturularak genetik çeşitliliğin sağlanması amaçlanmaktadır. Mutasyon işleminde oluşturulan bireyin bir önceki bireyden daha iyi olacağı anlaşılmamalıdır. İkili kodlamanın kullanıldığı problemlerde mutasyon işlemi bit değişimi (0>1 veya 1<0) ile sağlanmaktadır. Mutasyon işlemi gerçekleştirildikten sonra popülasyonda ki bireylerin uygunluk değerleri hesaplanmaktadır. Her yinelemenin sonucunda en iyi çözüm güncellenir .Yukarıda ki aşamalar belirlenen durdurma kriterine ulaşılana kadar tekrar edilir.Son olarak küresel en iyi çözüme ulaşılır.

**1: Girdi** Parametreleri ayarla.
**2:** Başlangıç popülasyonu oluştur.
**3:** Başlangıç popülasyondaki bireylerin uygunluk değerleri hesaplanır.
**4: while** (iterasyon <MakİterasyonSayısı) **do**
**5:** Yeni popülasyonu oluştur.
**6:** Elitizm varsa bir önceki popülasyonda en iyi uygunluk değerine sahip bireyi yeni popülasyona aktar.
**7:** İkili turnuva yöntemini kullanarak ebeveynleri seç.
**8:** Çoklu-ebeveyn çaprazlama türüne göre yeni birey oluştur.
**9:** Bit değişim mutasyonunu uygula.
**10:** Yeni bireyi popülasyona dahil et.
**11:** Yeni bireyin uygunluk değerini hesapla.
**12: end while**
**13: Çıktı** Küresel en iyi uygunluk değerine sahip birey

Şekil 2. 2 Genetik Algoritma Sözde Kodu.

Genetik algoritmalar , öznitelik çözümlerini kodlar ve uygunluk değerlerini hesaplar. Genetik algoritma bu tezde öznitelik seçiminde kromozomları ikili bit dizisi ile kodlamaktadır. İkili kodlama da bitin değerinin 1 olması özniteliğin seçildiğini , bitin değerinin 0 olması ise özniteliğin seçilmediğini göstermektedir. Şekil 2.3'de öznitelik sayısı 10 olan bir çözüm dizisinde X çözümü ={1, 0, 1, 1 , 0, 1, 0 , 1 , 1 , 0} ifade edilsin. X çözümde 6 özellik 1., 3., 4., 6., 8., ve 9. seçilmiştir [36].

| **Öznitelikler** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **X Çözümü** | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| **Seçilen Öznitelikler** | **1** | | **3** | **4** | | **6** | | **8** | **9** | |

Şekil 2. 3 Öznitelik Seçimi Örnek Çözüm.

Öznitelik seçiminde veri kümesi içerisinde yer alan özelliklerin alt kümeleri oluşturulmaktadır. Öznitelik sayısı n olan bir kümede $2^n$ farklı özellik alt kümesi bulunacaktır. En iyi alt kümenin seçiminde bütün kombinasyonların çözümünün aranması zaman ve bellek yönetimi açısından maliyetlidir. Bu çalışmada [36] önerilen çözümlerin uygunluk değerinin matematiksel olarak hesaplanmasında hem sınıflandırma hata oranını hem de özellik sayısını dikkat alan fonksiyon uygulanmıştır. Uygunluk fonksiyonu matematiksel olarak şu şekilde ifade edilebilir:

$$f(i)=\alpha ER(K)+(1-\alpha)\left(\frac{s}{d}\right) \quad \textbf{(2.1)}$$

Uygunluk fonksiyonu denkleminde yer alan parametrelerin tanımları aşağıdaki gibidir:

$f(i)$: Her birey için hesaplanan uygunluk fonksiyonu değeridir.

$ER(K)$: K seçim kararına göre seçilen öznitelik sınıflandırma hata oranıdır.

$d$: Veri kümesindeki öznitelik sayısıdır.

$s$: Seçilen özniteliklerin sayısıdır.

$\alpha$: Parametresi sınıflandırma hata oranının etkisini kontrol eden parametredir.

$\alpha$ parametresinin değeri sınıflandırma performansı önemli olduğu için [36] çalışmada önerilen 0,99 olarak belirlenmiştir.

Yapılacak olan testlerde öznitelik seçiminde çoklu-ebeveyn çaprazlama operatörlerinin genetik algoritma içerisinde kullanılmasının performansları incelenecektir.

# ÜÇÜNCÜ BÖLÜM

# 3. DENEYSEL ÇALIŞMA

Bu bölümde genetik algoritma içerisinde çoklu-ebeveyn çaprazlama operatörlerinin veri kümeleri ve farklı ebeveyn sayıları ile öznitelik seçimindeki performans incelemesinde kullanılan programlama dili, veri kümeleri, performans metriği, parametre atamaları hakkında bilgi verilecektir.

## 3.1. PROGRAMLAMA DİLİ

Bu tez çalışmasında sonuçların elde edileceği uygulama Python programla dili kullanarak geliştirilmiştir. Python programlama dilinin tercih edilmesinin sebebi yapay zekâ ve makine öğrenmesi alanında oldukça gelişmiş kütüphanelere sahip olmasıdır. Bu çalışmada Python'ın Numpy, Pandas, Matplotlib ve makine öğrenmesi modellerini oluşturmak için Scikit-Learn kütüphanelerinden yararlanılmıştır.

## 3.2. ÇALIŞMADA KULLANILAN VERİ KÜMELERİ

Çalışmada kullanılan üç veri kümesi, UCI makine öğrenimi veri kümesinden elde edilmiştir. Kullanılan veri kümeleri; Wine veri kümesi, Wisconsin Diagnostic Breast Cancer veri kümesi (WDBC), Musk (Musk Versiyon 1) veri kümesidir. Veri kümeleri örnek sayısı, öznitelik sayısı ve sınıf sayısı açısından değerlendirilmiştir.

Wine veri kümesi; 178 örnek, 13 öznitelikten ve 3 sınıftan oluşmaktadır. Veri kümesi içerisinde İtalya'da aynı bölgede üç üretici tarafından yetiştirilen şarapların kimyasal analizleri sonucunda kaliteleri hakkında bilgi verilmektedir. Wine veri kümesinde şarapların kalitelerine göre belirlenen 3 sınıfta, sınıf başına düşen örnek sayısı 59, 71 ve 48 olarak belirlenmiştir.

WDBC veri kümesi; 569 örnek sayısı, 30 öznitelikten ve 2 sınıftan oluşmaktadır. WDBC veri seti göğüs kanseri hastalığının teşhisi için 1995 yılında Wisconsin Üniversitesi hastanesinde Dr. William H.Wolberg tarafından

oluşturulmuştur. Veri seti içerisinde 30 gerçek değerli öz nitelik ve ID ve 1 adette teşhis için kullanılan özellik olmak üzere toplam 32 öz nitelikten oluşmaktadır. Veri seti içerisinde 10 adet öz nitelik tümörlü hücre doğrudan gözlemlenerek gerçek değerlerden oluşmaktadır. Bu özelliklerden elde edilen ortalama, standart hata, en küçük ve en büyük (en büyük üç değerin ortalaması) değerleri kullanılarak gerçekleştirilen hesaplamalar sonucunda, 20 tane öz nitelik elde edilmiştir. Eksik özellik değeri yoktur. WDBC veri kümesinde sınıflarda yer alan örnekler; 357 örnek iyi huylu, 212 örnek ise kötü huylu olarak ayrılmıştır.

Musk veri kümesi; 476 örnek sayısı, 166 öznitelikten ve 2 sınıftan oluşmaktadır. Molekülleri tanımlayan 166 öz nitelik, molekülün tam şekline ve konformasyonuna (Moleküllerin uzayda doğal olarak sahip oldukları üç boyutlu yapı) bağlıdır. Molekülleri oluşturan bağlar değişe bildiği için bir molekül birçok şekil alabilir. Musk veri setinde 47'si musk (pozitif) olarak işaretlenen ve geri kalanı işaretlenmemiş 92 molekül bulunmaktadır. Amaç yeni moleküllerin musk mı yoksa musk olmayan mı olacağını öğrenmektir. Musk veri kümesinde musk olan ve olmayan olarak ayrılan sınıflarda, sınıf 1 için 207 örnek bulunurken sınıf 2 için ise 269 örnek bulunmaktadır. Tablo 3.1'de veri setleri listesi ve kısa açıklamaları verilmiştir.

Tablo 3. 1 Veri Kümeleri Listesi ve Kısa Açıklamaları

| **Veri Kümesi Numarası** | **Veri Kümesi Adı** | **Örnek Sayısı** | **Öznitelik Sayısı** | **Sınıf Sayısı** |
|---|---|---|---|---|
| **1** | Wine | 178 | 13 | 3 |
| **2** | WDBC | 569 | 30 | 2 |
| **3** | Musk | 476 | 166 | 2 |

## 3.3. PARAMETRE ATAMASI

Denklem (2.1)'de verilen uygunluk fonksiyonunda ki sınıflandırma hata oranı hesaplanırken; Sınıflandırma için DVM algoritması tercih edilmiştir. Modelimizde doğrusal olarak sınıflandırma yapılacağı için çekirdek fonksiyonu olarak doğrusal (İng.: Linear) seçilmiştir. Sınıflandırmada hata oranını azaltmak ve esneklik sağlayabilmek için $C$ parametresi kullanılmaktadır. LinearSVC sınıflandırıcısının C

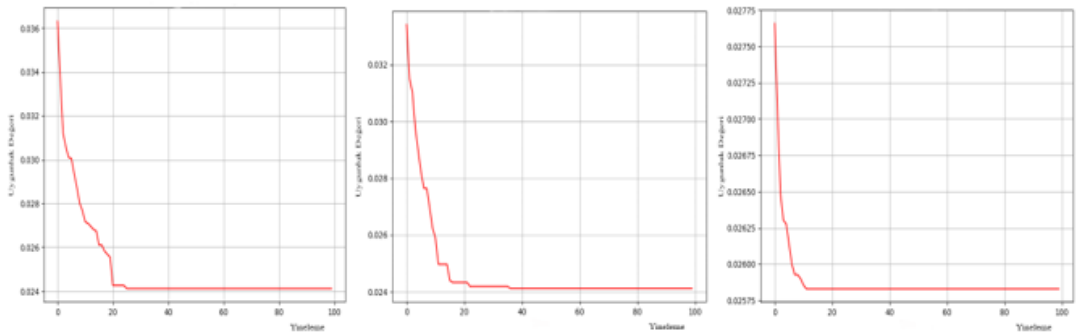
parametresinin optimizasyonu yapılırken, ayrıntılı arama olarak da bilinen ızgara araması (İng.: Grid Search) yöntemi kullanılmış ve en başarılı sonucu veren C parametresinin 1 olduğuna karar verilmiştir.

Makine öğrenmesinde veriler eğitim ve test kümeleri olarak bölünmektedir. Eğitim veri kümesiyle tasarlanan model eğitilirken, test verileriyle modelin performansının gerçekten istenen sonuçları verip vermediği tespit edilmektedir. Veri kümesindeki değerlerin, eğitim ve test verisi olarak ayrılması için bir yüzde değeri belirlenmektedir. Modelde veri kümeleri %80 eğitim için ve %20 test için ayrılmıştır. Modelin performansı değerlendirilirken eğitim ve test verilerinin her uygulamada aynı olmaması nedeniyle birbirinden farklı doğruluk (İng.: Accuracy) değerleri elde edilmektedir. Çalışmada veri kümesi eğitim ve test verisi olarak ayrılırken rastgele durum (İng.: Random State) değeri belirlenerek, verilerin her seferinde o değere göre bölünmesi sağlanmaktadır. Uygulamada [0, 5, 12] olmak üzere 3 farklı rastgele durum değeri belirlenmiştir. Örneğin rastgele durum değerini 5 olarak belirlediğimizde eğitim ve test verilerindeki bölünmede her zaman aynı çıktı alınmaktadır.

Çalışmada aynı algoritma kullanılırken üç farklı rastgele durum ile belirlenen test kümelerinde elde edilen ortalama doğruluk değerlerinin birbirinden farklı olduğu görülmüştür. Bu sorunun çözümü için $K$ katlamalı çapraz doğrulama (İng.: K Fold Cross Validation) yöntemi kullanılmaktadır. Çalışmada üç farklı veri kümesine ait veriler çapraz doğrulama yöntemi ile eğitilip ardından test edilmiştir. $K$ kez tekrarlanan yöntemde her defasında $k$ alt kümelerinden biri, test seti olarak kullanılırken geriye kalan $k$-1 alt küme bir arada gruplanarak eğitim seti olarak kullanılır. Gerçekleştirilen çalışmada $k$ alt küme sayısı 10 olarak belirlenmiştir. $K$ katlamalı çapraz doğrulama ile her veri bir kez test kümesine ve $k$-1 kez eğitim setine girmektedir. On parçaya bölünen verileri her bir parçası, bir kez test kümesi olacak şekilde sistem eğitilir ve test kümesiyle de test edilir. Son olarak 10 adet deneme sonucunda doğruluk değerlerinin ortalaması elde edilmiştir.

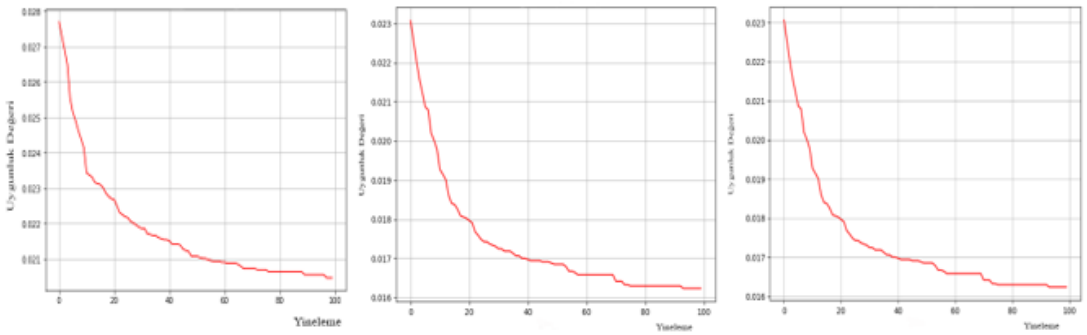
Şekil 3.1'de Wine veri kümesiyle sırasıyla 50, 100 ve 200 olarak belirlenen popülasyon büyüklüğü değerleri için ve yineleme sayısı 100 olduğunda elde edilen örnek yakınsama grafikleri gözükmektedir. Şekil 3. 2'de WDBC veri kümesiyle sırasıyla 50, 100 ve 200 olarak belirlenen farklı popülasyon büyüklüğü değerleri için

ve yineleme sayısı 100 olduğunda örnek yakınsama grafikleri gözükmektedir. Şekil 3. 3'de Musk veri kümesiyle sırasıyla 50, 100 ve 200 olarak belirlenen popülasyon büyüklüğü değerleri için ve yineleme sayısı 250 olduğunda örnek yakınsama grafikleri görülmektedir.



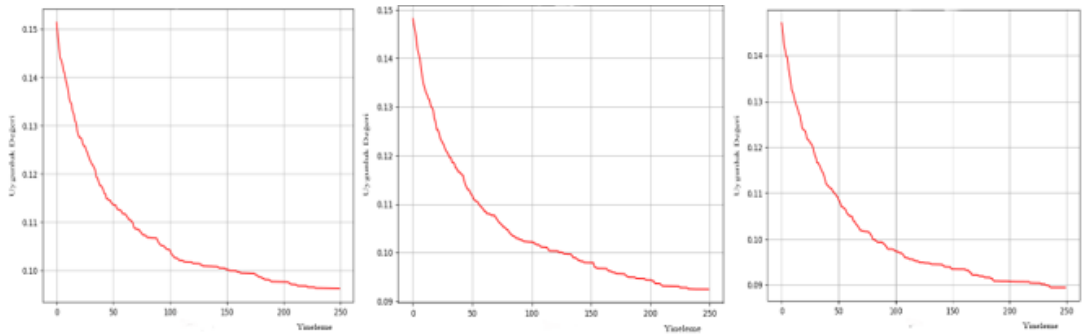
a) Popülasyon Boyutu: 50 b) Popülasyon Boyutu: 100 c) Popülasyon Boyutu: 200

Şekil 3. 1 Wine Veri Kümesiyle Farklı Popülasyon Değerleri İçin Oluşturulan Yakınsama Grafiği.



a) Popülasyon Boyutu: 50 b) Popülasyon Boyutu: 100 c) Popülasyon Boyutu: 200

Şekil 3. 2 WDBC Veri Kümesiyle Farklı Popülasyon Değerleri İçin Oluşturulan Yakınsama Grafiği.



a) Popülasyon Boyutu: 50 b) Popülasyon Boyutu: 100 c) Popülasyon Boyutu: 200

Şekil 3. 3 Musk Veri Kümesiyle Farklı Popülasyon Değerleri İçin Oluşturulan Yakınsama Grafiği.

Veri kümeleriyle farklı popülasyon değerleri sonucunda elde edilen yakınsama grafikleri doğrultusunda yapılacak olan testlerde popülasyon sayısı 50, yineleme sayıları ise Wine veri kümesi için 100, WDBC veri kümesi için 250, Musk veri kümesi için 500 olarak belirlenmiştir. Kullanılan yaklaşımda Tablo 3.2'de belirlenen parametrelerin kullanılmasına karar verilmiştir.

Tablo 3. 2 Uygulamada Kullanılacak Parametre Değerleri

| **Parametre** | **Değeri** |
|---|---|
| Çaprazlama Oranı, $Pc$ | 1.0 |
| Mutasyon Oranı, $Pm$ | 1/d |
| Seçim Yöntemi | Turnuva |
| Sınıflandırma | DVM Doğrusal |
| Ebeveyn Sayıları | 2,3,5,10 |
| Popülasyon Boyutu | 50 |
| Çaprazlama Operatörleri | DC, FBC, UC, OBC |
| Yineleme Sayısı( WINE) | 100 |
| Yineleme Sayısı( WDBC) | 250 |
| Yineleme Sayısı( MUSK) | 500 |
| Çalıştırma Sayısı | 11 |

## 3.4. PERFORMANS METRİĞİ

Çalışmada sınıflandırma başarımının ölçülmesinde kullanılan en popüler ve basit yöntem, doğruluk metriği kullanılarak yapılan değerlendirmedir. Doğruluk metriği Tablo 3.3'de verilen sınıflandırma hata matrisinden elde edilmiştir. Doğruluk, sınıflandırma başarısının performansının ölçüsüdür. Doğruluk ne kadar yüksek ise sınıflandırma o kadar başarılıdır[37]. Doğruluğun değeri 0 (En kötü) ile 1 (En iyi) arasında değişmektedir.

Tablo 3. 3 Sınıflandırmada Kullanılan Hata Matrisi

| **Tahmin Edilen Sınıf** | | |
|---|---|---|
| **Gerçekleşen Sınıf** | DP | YN |
| | YP | DN |

Doğruluk Denklem 4.1'de verilen formüle göre hesaplanmıştır, doğru tanımlanmış (DP+YP) örnek sayısının, toplam örnek sayısına (DP+DN+YP+YN) oranıdır.

$$Doğruluk = \frac{DP+DN}{DP+DN+YP+YN} \quad \textbf{(4.1)}$$

Denklem 4.1'deki parametrelerin tanımları şunlardır;

$DP$: Pozitif olarak tahmin edilen pozitif örneklerdir.

$DN$: Negatif olarak tahmin edilen negatif örneklerdir.

$YP$: Pozitif olarak tahmin edilen negatif örneklerdir.

$YN$: Negatif olarak tahmin edilen pozitif örneklerdir.

# DÖRDÜNCÜ BÖLÜM

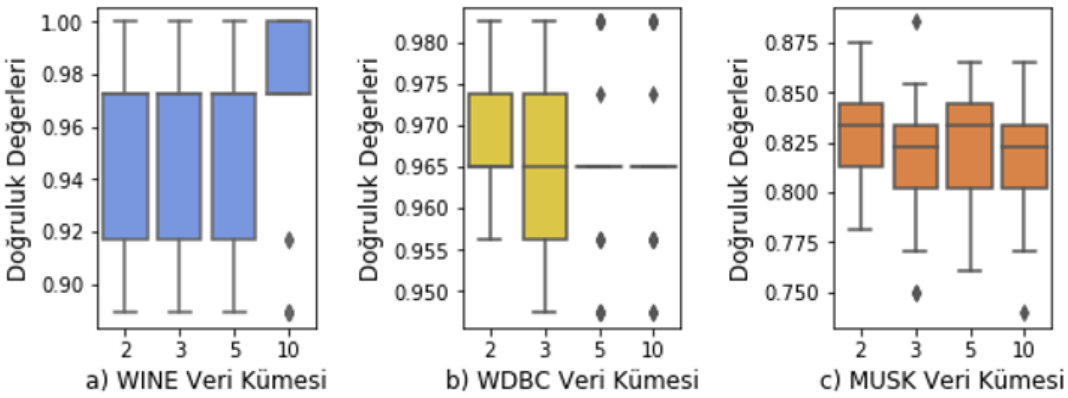
## 4. BULGULAR VE TARTIŞMA

Genetik algoritma ile dört farklı çaprazlama operatörü ile farklı ebeveyn sayıları için üç veri kümesi ile [0, 5, 12] rastgele durum ile testler yapılmıştır. Her bir veri kümesi, dört farklı çaprazlama operatörü ve farklı ebeveyn sayıları ile on bir defa çalıştırılmıştır. Çalıştırma sayının on bir olmasının sebebi, elde edilen sonuçların tutarlı olması ve performansların istatiksel analizini kolaylaştırmaktır. Çalışmalar sonucunda her bir çaprazlama operatörü için elde edilen ortalama doğruluk değerleri Tablo 4.1'de verilmiştir. Tablo 4.1'de en iyi sonuçlar kalın harflerle vurgulanmıştır.

Tablo 4. 1 Farklı Çaprazlama Operatörü ve Farklı Ebeveyn Sayıları İçin Üç Veri Kümesi Ortalama Doğruluk Değerleri

| Çaprazlama Operatörü | Ebeveyn Sayısı | WINE | WDBC | MUSK |
|---|---|---|---|---|
| **DC** | 2 | 0.9486 | **0.9673** | **0.8273** |
| | 3 | **0.9587** | 0.9659 | 0.8147 |
| | 5 | 0.9486 | 0.9654 | 0.8222 |
| | 10 | 0.9511 | 0.9662 | 0.8184 |
| **FBC** | 2 | 0.9537 | 0.9657 | **0.8203** |
| | 3 | **0.9562** | **0.9667** | 0.8181 |
| | 5 | 0.9537 | 0.9659 | 0.8153 |
| | 10 | 0.9511 | 0.9665 | 0.8134 |
| **UC** | 2 | **0.9638** | 0.9675 | 0.8238 |
| | 3 | 0.9486 | 0.9659 | 0.8194 |
| | 5 | 0.9537 | **0.9683** | **0.8295** |
| | 10 | 0.9486 | 0.9646 | 0.8147 |
| **OBC** | 2 | **0.9638** | 0.9630 | 0.8194 |
| | 3 | 0.9562 | 0.9688 | 0.8327 |
| | 5 | 0.9612 | **0.9702** | **0.8393** |
| | 10 | 0.9654 | 0.9699 | 0.8200 |

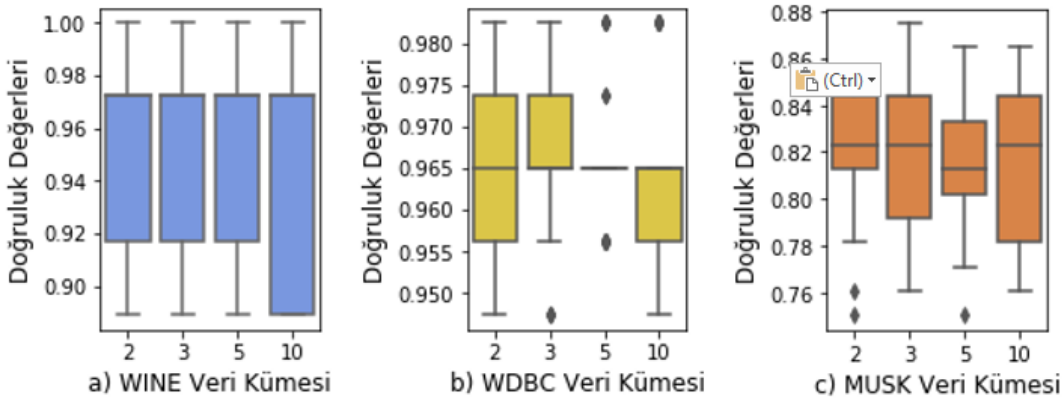
Kutu grafikleriyle sayısal verilerin ve değişkenliklerin dağılımı, çeyrekleri ve ortalamaları gösterilmektedir. Kutu grafikleriyle verilerin beş özelliği; minimum, ilk (%25) çeyrek, medyan (Ortanca değer), üçüncü (%75) çeyrek ve maksimum değerlerinin dağılımı görselleştirilmektedir. Veri analizinde görsel kutuların ortasında yer alan çizgi medyan değerini, diğer çizgiler ise verinin çeyreklerini göstermektedir. Kutu grafiklerinde başlangıç ve bitiş kesim noktaları dışında kalan değerler, aykırı değerler olarak kabul edilmektedir. Grafiklerde kutunun boyunun uzunluğu ne kadar fazla olursa verilerin o kadar dağılmış olduğu, kutunun boyunun küçük olması ise verinin o kadar az dağılmış olduğu görülmektedir. Şekil 4.1, Şekil 4.2, Şekil 4.3 ve Şekil 4.4'de çoklu ebeveyn çaprazlama operatörleri ile farklı ebeveyn sayılarına göre veri kümelerinin doğruluk dağılım grafikleri verilmiştir.




Şekil 4. 1 Diyagonal Çaprazlama (DC) Operatörü, Veri Kümeleri ve Ebeveyn Sayıları Doğruluk Değerleri Kutu Grafiği.

Şekil 4.1(a)'da DC operatörü ve Wine veri kümesi ile yapılan testlerde ebeveyn sayısının 2, 3, 5 olduğu durumlarda verilerin dağılımının benzer olduğu görülmektedir. Ebeveyn sayısının 10 olarak durumda verilerin dağılımının daha az olduğu ve dağılımda aykırı değerler olduğu görülmektedir. Tablo 4.1'deki verilere de baktığımızda verilerin birbirine yakın olduğu gözükmekte olup en yüksek ortalama doğruluk değeri ebeveyn sayısının 3 olduğu durumda elde edildiği belirlenmiştir. Şekil 4.1(b)'de ise WDBC veri kümesi ile yapılan testlerde ebeveyn sayısına göre elde edilen verilerin ortalama doğruluk değerlerinin birbirine yakın olduğu görülmektedir. Ebeveyn sayısının 5 ve 10 olarak belirlendiği durumlarda dağılımda aykırı değerler bulunmaktadır. Tablo 4.1'deki ortalama doğruluk değerlerinin birbirine yakın olduğu

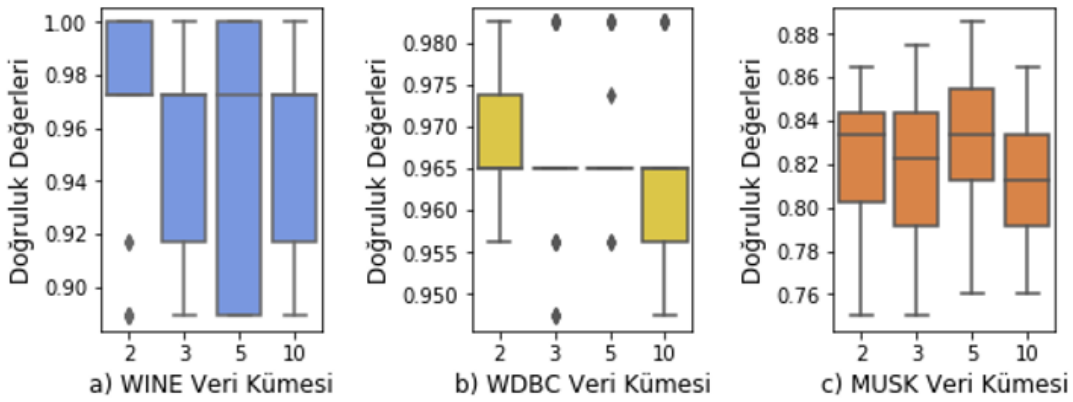
gözlenmektedir. DC operatörü ile WDBC veri kümesi ile yapılan testlerde en yüksek ortalama doğruluk değeri ebeveyn sayısının 2 olduğu durumda elde edilmiştir. Şekil 4.1(c)'de Musk veri kümesi ile yapılan testlerin ortalama doğruluk dağılımı verilmiştir. Dağılım incelendiğinde ortalama doğruluk değerlerinin birbirine yakın olduğu görülmektedir. Tablo 4.1'de DC operatörü ve Musk veri kümesi ile yapılan testlerdeki verilere bakıldığında en yüksek ortalama doğruluk değeri ebeveyn sayısının 2 olduğu durumda elde edilmiştir. DC operatörü ve veri kümeleriyle birlikte farklı ebeveyn sayıları ile [0, 5, 12] farklı rastgele durum ve toplam 33 farklı çalıştırma sonucunda elde edilen doğruluk değerleri incelendiğinde ortalama doğruluk değerlerinin en yüksek olduğu ebeveyn sayıları; Wine veri kümesi için 3, WDBC veri kümesi 2 ve Musk veri kümesi için ebeveyn sayısının 2 olduğu durumlarda elde edilmiştir.




Şekil 4. 2 Uygunluk Temelli Çaprazlama (FBC) Operatörü, Veri Kümeleri ve Ebeveyn Sayıları Doğruluk Değerleri Kutu Grafiği.

Şekil 4.2(a)'de FBC operatörü ve Wine veri kümesi ile yapılan testlerde ebeveyn sayısının 2, 3, 5 olduğu durumlarda verilerin dağılımının birbirine benzediği görülmektedir. Ebeveyn Sayısının 10 olarak durumda ise verilerin dağılımının daha fazla olduğu görülmektedir. Tablo 4.1'deki Wine veri kümesi ile elde edilene ortalama doğruluk değerlerine göre en yüksek ortalama doğruluk değeri ebeveyn sayısının 3 olduğu durumda elde edilmiştir. Şekil 4.1(b)'de ise WDBC veri kümesi ile yapılan testlerde ebeveyn sayısının belirlenmesinde kullanılan 33 farklı çalışma sonucu ile oluşturulan dağılımların birbirinden farklı olduğu görülmektedir. Ebeveyn sayısının 5 ve 10 olarak belirlendiği durumlarda dağılımda aykırı değerler bulunmaktadır. Tablo

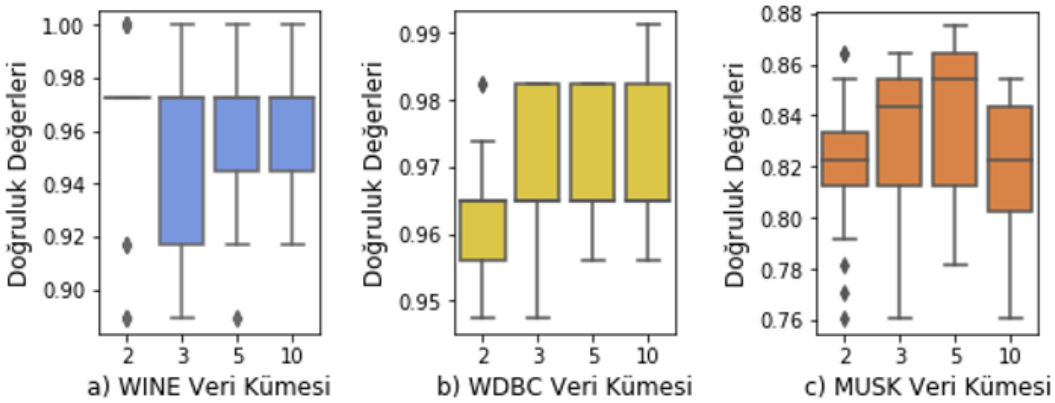
4.1’deki ortalama doğruluk değerlerinin birbirine yakın olduğu gözlenmektedir. FBC operatörü ile WDBC veri kümesi ile yapılan testlerde en yüksek ortalama doğruluk değeri ebeveyn sayısının 3 olduğu durumda elde edilmiştir. Şekil 4.1(c)’de Musk veri kümesi ile yapılan testlerde elde edilen verilerle oluşturulan dağılımı grafiği verilmiştir. Dağılım incelendiğinde doğruluk değerlerinin birbirine yakın olduğu görülmektedir. Ebeveyn sayısının 2 ve 3 olduğu kutu grafiklerinde aykırı değerlerde gözlenmektedir. Tablo 4.1’de FBC operatörü ve Musk veri kümesi ile yapılan testlerdeki ortalama doğruluk değerleri incelendiğinde, en yüksek ortalama doğruluk değeri ebeveyn sayısının 2 olduğu durumda elde edilmiştir. FBC operatörü ile farklı ebeveyn sayıları ile [0, 5, 12] farklı rastgele durum ve toplam 33 farklı çalıştırma sonucunda elde edilen doğruluk değerleri incelendiğinde ortalama doğruluk değerlerinin en yüksek olduğu ebeveyn sayıları; Wine veri kümesi için 3, WDBC veri kümesi 3 ve Musk veri kümesi için ebeveyn sayısının 2 olduğu durumlarda elde edilmiştir.




Şekil 4. 3 Tek Biçimli Temelli Çaprazlama (UC) Operatörü, Veri Kümeleri ve Ebeveyn Sayıları Doğruluk Değerleri Kutu Grafiği.

Şekil 4.3(a)’de UC operatörü ve Wine veri kümesi ile yapılan testlerde farklı ebeveyn sayıları ile yapılan testler sonucunda elde edilen doğruluk değerleriyle oluşturulan kutu grafiklerinde doğruluk değerlerinin dağılımlarının farklı olduğu görülmektedir. Ebeveyn sayınının 3 ve 10 olduğu durumlarda benzer dağılım görülmekte olup, ebeveyn sayının 5 olduğu durumda ise kutu grafiğinin daha uzun olduğu bu sebeple doğruluk değerleri dağılımının daha fazla gerçekleştiği görülmektedir. Ebeveyn sayısının 2 olduğun da ise dağılımın daha sınırlı olduğu ve

dağılımda aykırı değerlerin olduğu görülmektedir. Tablo 4.1'deki UC operatörü ve Wine veri kümesi ile elde edilene ortalama doğruluk değerlerine göre en yüksek ortalama doğruluk değeri ebeveyn sayısının 2 olduğu durumda elde edilmiştir. Şekil 4.3(b)'de ise WDBC veri kümesi ile yapılan testlerde elde edilen doğruluk değerleri ile oluşturulan kutu grafiklerinde sonuçların dağılımlarının birbirinden farklı olduğu görülmektedir. Ebeveyn sayısının 3, 5 ve 10 olduğu durumlarda dağılımda aykırı değerler bulunmaktadır. Tablo 4.1'deki UC operatörü ve WDBC veri kümesi ile yapılan testler sonucunda ortalama doğruluk değerlerinin birbirine yakın olduğu gözlenmektedir ve en yüksek ortalama doğruluk değerinin ebeveyn sayısının 5 olduğu durumda elde edildiği belirlenmiştir. Şekil 4.3(c)'de Musk veri kümesi ile yapılan testlerde elde edilen verilerle oluşturulan dağılımı grafiği verilmiştir. Dağılımdaki kutu grafikleri incelendiğinde ortalama doğruluk değerinin en yüksek ebeveyn sayısının 5 olduğu durumda elde edildiği görülmektedir. UC operatörü ile farklı ebeveyn sayıları ile [0, 5, 12] farklı rastgele durum ve toplam 33 farklı çalıştırma sonucunda elde edilen doğruluk değerleri incelendiğinde Tablo 4.1'de verilen ortalama doğruluk değerlerinin en yüksek olduğu ebeveyn sayıları; Wine veri kümesi için 2, WDBC veri kümesi 5 ve Musk veri kümesi için ebeveyn sayısının 5 olarak belirlendiği çalışmalarda elde edilmiştir




Şekil 4. 4 Oluşum Tabanlı Çaprazlama (OBC) Operatörü, Veri Kümeleri ve Ebeveyn Sayıları Doğruluk Değerleri Kutu Grafiği.

Şekil 4.4(a)'de OBC operatörü ve Wine veri kümesi ile yapılan testlerde farklı ebeveyn sayıları ile yapılan testler sonucunda elde edilen doğruluk değerleriyle oluşturulan kutu grafiklerinde verilerin dağılımları görülmektedir. Ebeveyn sayınının

2 olduğu durumda doğruluk değerlerinin dağılımı ortanca değerde yoğunlaşmaktadır ve dağılımda aykırı değerler de bulunmaktadır. Ebeveyn sayının 3 olduğu durumda ise kutu grafiğinin daha uzun olduğu bu sebeple doğruluk değerleri dağılımının daha fazla gerçekleştiği görülmektedir. Ebeveyn sayısının 5 ve 10 olduğu kutu grafiklerinde ise dağılımlar birbirine benzemektedir. Tablo 4.1'deki operatörü ve Wine veri kümesi ile elde edilene ortalama doğruluk değerlerine göre en yüksek ortalama doğruluk değeri ebeveyn sayısının 2 olduğu durumda elde edilmiştir. Şekil 4.4(b)'de ise WDBC veri kümesi ile yapılan testlerde elde edilen doğruluk değerleri ile oluşturulan kutu grafiklerinde sonuçların dağılımlarının ebeveyn sayılarının 3, 5, ve 10 olduğu durumlarda, kutu grafiklerinin birbirilerine benzedikleri görülmektedir. Ebeveyn sayısının 2 olduğu durumda ise dağılımın daha sınırlı bir alanda yoğunlaştığı görülmektedir. Tablo 4.1'deki OBC operatörü ve WDBC veri kümesi ile yapılan testler sonucunda ortalama doğruluk değerlerinin birbirine yakın olduğu gözlenmektedir ve en yüksek ortalama doğruluk değerinin ebeveyn sayısının 5 olduğu durumda elde edildiği belirlenmiştir. Şekil 4.4(c)'de Musk veri kümesi ile yapılan testlerde elde edilen verilerle oluşturulan dağılımı grafiği verilmiştir. Dağılımdaki kutu grafikleri incelendiğinde ortalama doğruluk değerinin en yüksek ebeveyn sayısının 5 olduğu durumda elde edildiği görülmektedir. OBC operatörü ile farklı ebeveyn sayıları ile [0, 5, 12] farklı rastgele durum ve toplam 33 farklı çalıştırma sonucunda elde edilen doğruluk değerleri analiz edildiğinde Tablo 4.1'de verilen ortalama doğruluk değerlerinin en yüksek olduğu ebeveyn sayıları; Wine veri kümesi için 2, WDBC veri kümesi 5 ve Musk veri kümesi için ebeveyn sayısının 5 olarak belirlendiği testlerde elde edilmiştir

Tablo 4.1'de farklı çoklu-ebeveyn çaprazlama operatörlerinin üç farklı durum için elde edilen ortalama doğruluk değenleri on bir çalışma sonucunda elde edildi. On bir çalışma sonucunda elde edilen tüm test sonuçları istatiksel olarak karşılaştırıldı. Yapılan karşılaştırmalar sonucunda her bir çoklu-ebeveyn çaprazlama operatörü için performanslar karşılaştırılarak en iyi ebeveyn sayısına, karar vermede kullanıldı. Tablo 4.2'de verilen çoklu-ebeveyn çaprazlama operatörleri için üç farklı veri kümesinin farklı ebeveyn sayıları ile yapılan üç farklı rastgele durum karşılaştırmaları için kazanma, beraberlik ve yenilgi sayıları toplam 36 olmaktadır. İstatiksel olarak yapılan

karşılaştırmalar %95 güven aralığında Anova Tukey HSD testi ile yapılmıştır. Anova Tukey testi grup ortalamaları arasında istatiksel olarak anlamlı bir fark olup olmadığına karar vermektedir. Anova Tukey testi üç ya da daha fazla grup ortalamasının eşit olup olmadığını test eder.

Tablo 4. 2 Ebeveyn Sayısı Seçimi Anova Tukey Testi Karşılaştırma Sonuçları

| **Çaprazlama Operatörü** | **Ebeveyn Sayısı** | **Kazanma** | **Beraberlik** | **Yenilgi** |
|---|---|---|---|---|
| **DC** | 2 | 0 | 9 | 0 |
| | 3 | 0 | 9 | 0 |
| | 5 | 0 | 9 | 0 |
| | 10 | 0 | 9 | 0 |
| **FBC** | 2 | 0 | 9 | 0 |
| | 3 | 0 | 9 | 0 |
| | 5 | 0 | 9 | 0 |
| | 10 | 0 | 9 | 0 |
| **UC** | 2 | 0 | 9 | 0 |
| | 3 | 0 | 9 | 0 |
| | 5 | 0 | 9 | 0 |
| | 10 | 0 | 9 | 0 |
| **OBC** | 2 | 0 | 6 | 3 |
| | 3 | 0 | 9 | 0 |
| | 5 | **3** | **6** | 0 |
| | 10 | **1** | **7** | 1 |

Tablo 4.2'deki istatiksel sonuçlara göre OBC çaprazlama operatörü için ebeveyn sayısının 5 olmasına karar verilmiştir. DC, FBC ve UC çaprazlama operatörü için sonuçlara bakıldığında istatiksel olarak fark görülmemektedir. DC, FBC ve UC çaprazlama operatörleri için ebeveyn sayısına karar verilirken Tablo 4.1'de verilen ortalama doğruluk değerlerine göre seçim yapılmıştır. DC için ebeveyn sayısı 2, FBC için ebeveyn sayısı 3 ve UC için ebeveyn sayısının 5 olmasına göre karar verilmiştir.

Çoklu-ebeveyn çaprazlama operatörleri için belirlenen ebeveyn sayıları ile her veri kümesi, [0, 5, 12] rastgele durum için iki kere çalıştırılmıştır. Elde edilen doğruluk sonuçları ve öznitelik sayılarının, ortalama doğruluk değerleri ve ortalama öznitelik sayıları Tablo 4.3'de verilmiştir.

Tablo 4. 3 Farklı Çaprazlama Operatörü ve Farklı Üç Veri Kümesi İçin Ortalama Doğruluk Değerleri ve Ortalama Öznitelik Sayısı Değerleri

| **Veri Kümesi** | **WINE** | | **WDBC** | | **MUSK** | |
|---|---|---|---|---|---|---|
| **Çaprazlama Operatörü** | **Ortalama Doğruluk** | **Öznitelik Sayısı** | **Ortalama Doğruluk** | **Öznitelik Sayısı** | **Ortalama Doğruluk** | **Öznitelik Sayısı** |
| **DC** | **0.9537** | 5.66 | 0.9634 | 12.33 | 0.8194 | 62.66 |
| **FBC** | **0.9537** | **5.33** | **0.9678** | 12.50 | 0.8263 | **55.12** |
| **UC** | 0.9361 | 6.33 | 0.9663 | **11.16** | 0.8281 | 63.00 |
| **OBC** | **0.9537** | 7.16 | 0.9663 | 18.16 | **0.8368** | **155.00** |

GA ile öznitelik seçimi çoklu-ebeveyn çaprazlama operatörleri ile veri kümelerinde alakasız vere gerekli olmayan değişkenleri azaltarak en faydalı verilerin seçilmesi için istatiksel testler yapılmıştır. Tablo 4.1'deki istatiksel sonuçlar doğrultusunda çoklu-ebeveyn çaprazlama operatörleri için ebeveyn sayıları belirlenmiş ve elde edilen ebeveyn sayıları ile tekrar testler yapılmıştı. Yapılan testeler sonucunda Tablo 4.3'deki ortalama doğruluk ve öznitelik sayıları elde edilmiştir. Tablo 4.1'deki ortalama doğruluk değerleri sonuçları ve Tablo 4.2'deki Anova Tukey testi karşılaştırma sonuçları incelendiğinde ebeveyn sayısının ikiden fazla olduğu durumlarda OBC operatörünün ortalama doğruluk değerlerinde iyi sonuçlar verdiği gözlenmektedir. Test sonuçlarına bakıldığında ebeveyn sayısının yüksek olmasının önemli fark olmaksızın daha iyi performans sağlamıştır.

Öznitelik seçiminde Tablo 4.3'deki veriler analiz edildiğinde DC, FBC, UC operatörleri ile veri kümeleri ile öznitelik seçimi yapılırken başarılı sonuçlar elde edilmiştir, OBC operatörü yapılan testler sonucunda ortalama doğruluk değerlerinde başarılı sonuçlar elde edilmiş fakat Musk veri kümesinden 166 öznitelik arasından ortalama 155 öznitelik seçilmiştir. Şekil 1.8'de verilen OBC operatörü ile bireyin gen seçimi yapılırken, ebeveynlerden seçilecek genlerin belirli bir konumda en çok ortaya çıkan değerinin seçilebilecek en iyi değer olduğu belirtmektedir. OBC operatörünün karakteristik özelliği kromozomda yer alan genlerin değerlerini 1'e götürdüğü için

öznitelik sayısı seçiminde daha yüksek değerler vermektedir. Popülasyon sayısını artırmak ve mutasyon oranını daha yüksek tutarak OBC ile öznitelik seçiminde daha faydalı sonuçlar elde edilebilir. OBC operatöründe ebeveyn sayısının 10 olduğu durumda iyi sonuç elde edilememesinin nedeni popülasyon boyutunun 50 olması sebebiyle ebeveynlerden üretilen yavruların atalarına daha çok benzemesi nedeniyle çeşitliliğin azalmasından kaynaklanmaktadır. Popülasyon boyutunu artırılmasıyla birlikte popülasyona çeşitlilik sağlanarak daha iyi sonuçlar elde edilebilir.

# SONUÇ

Bu tez çalışmasında öznitelik seçimi problemi için çoklu-ebeveyn çaprazlama operatörlerini genetik algoritma içerisinde kullanılarak performanslarının karşılaştırılması incelenmiştir. Öznitelik seçimi orijinal veri kümesini temsil edebilecek en iyi alt kümenin seçimi olarak tanımlanmaktadır. Genetik algoritma ise çeşitli çözümler arasında en iyi çözümü bulmayı amaçlayan popülasyon tabanlı algoritmadır. Öznitelik seçimi için genetik algoritma kullanılarak, dört farklı çaprazlama operatörü, farklı ebeveyn sayılarıyla üç farklı veri kümesi ile test edilmiş ve elde edilen sonuçlar One-way ANOVA ve Tukey HSD testleri %95 güven seviyesinde istatiksel olarak karşılaştırılmıştır. Yapılan çalışmada çoklu-ebeveyn çaprazlama operatörlerinin farklı ebeveyn sayıları üzerinde etkisi incelenmiş ve ortalama doğruluk değerlerine göre ebeveyn sayılarına karar verilmiştir. Çoklu-ebeveyn çaprazlama operatörleri için ebeveyn sayısı belirlendikten sonra yapılan testler sonucunda öznitelik sayısı belirlenmiştir. Öznitelik sayısı sonuçları incelendiğinde; Çoklu-ebeveyn çaprazlama operatörlerinin birbirine yakın sonuçlar verdiği, OBC operatöründe ise ebeveyn sayısının 5 olduğu durumda elde edilen ortalama doğruluk değerlerinde iyi performans gösterdiği belirlenmiş fakat öznitelik sayısında yüksek değerler verdiği gözlenmiştir. OBC operatörünün ebeveyn sayısı 10 olduğundan elde edilen ortalama doğruluk değerinin, ebeveyn sayısı 5 iken elde edilen ortalama doğruluk değerinden düşük olmasının sebebi popülasyonun büyüklüğünün 50 olması sebebiyle ebeveynlerden elde edilen bireylerin atalarına daha çok benzeyerek çeşitliliğin azalmasından kaynaklanmaktadır. OBC operatöründe öznitelik sayısında daha iyi sonuçlar elde etmek için popülasyonu boyutu artırılabilir ve mutasyon oranını daha yüksek tutmak faydalı olabilir. Çok amaçlı problemler için geliştirilen GA'lar özel uygunluk işlevlerini kullanarak çözüm çeşitliliğine katkı sağlayabilirler.

Yapılan çalışma sonucunda gelecekte çoklu-ebeveyn çaprazlama operatörleriyle farklı veri kümeleri, popülasyon sayıları ve farklı mutasyon oranlarıyla

öznitelik seçimi yapılabilir. Çeşitli sınıflandırma algoritmalarının çoklu-ebeveyn çaprazlama operatörleri üzerindeki etkisi incelenerek sınıflandırma algoritmaları karşılaştırılabilir.

# KAYNAKÇA


[1] J. H. Holland, “Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence,” p. pp.207-211., 1992.

[2] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning.,” *Reading*, 1989.

[3] S. Gu, R. Cheng, and Y. Jin, “Feature selection for high-dimensional classification using a competitive swarm optimizer,” *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018, doi: 10.1007/s00500-016-2385-6.

[4] S. S. Hong, W. Lee, and M. M. Han, “The feature selection method based on genetic algorithm for efficient of text clustering and text classification,” *Int. J. Adv. Soft Comput. its Appl.*, vol. 7, no. 1, pp. 22–40, 2015.

[5] J. H. Holland, “Adaptation in Natural and Artificial Systems,” *Adapt. Nat. Artif. Syst.*, vol. 100, p. 33, 2019, doi: 10.7551/mitpress/1090.001.0001.

[6] Ç. Taşkın and G. Emel, “Genetik Algoritmalar ve Uygulama Alanları,” *Uludağ Üniversitesi İİBF Derg.*, vol. 21, no. February 2002, pp. 129–152, 2002.

[7] Necdet Özçakar, “Genetik Algoritmalar,” *İ.Ü. İşletme Fakültesi Derg.*, vol. 27, no. 1, pp. 69–82, 1998.

[8] Ö. M. Aydın, “Hedef Programlama Olarak Modellenmiş Regresyon Problemine Genetik Algoritma Yaklaşımı,” Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1998.

[9] S. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. 2008. doi: 10.1007/978-3-540-73190-0.

[10] J. McCall, “Genetic algorithms for modelling and optimisation,” *J. Comput. Appl. Math.*, vol. 184, no. 1, pp. 205–222, 2005, doi: 10.1016/j.cam.2004.07.034.

[11] N. Moin, O. Sin, and M. bin Omar, “Hybrid Genetic Algorithm with Multiparents Crossover for Job Shop Scheduling Problems,” *Math. Probl. Eng.*, vol. 2015, pp. 1–12, Jan. 2015, doi: 10.1155/2015/210680.

[12] N. Saini, “Review of Selection Methods in Genetic Algorithms,” *Int. J. Eng. Comput. Sci.*, vol. 6, no. 12, pp. 23261–23263, 2017, doi: 10.18535/ijecs/v6i12.04.

[13] K. Jebari, “Selection Methods for Genetic Algorithms,” *Int. J. Emerg. Sci.*, vol. 3, pp. 333–344, Dec. 2013.

[14] P. Sharma, A. Wadhwa, and M. Komal, “Analysis of Selection Schemes for

Solving an Optimization Problem in Genetic Algorithm," *Int. J. Comput. Appl.*, vol. 93, pp. 1–3, May 2014, doi: 10.5120/16256-5714.

[15] O. Engin and A. Fığlalı, "Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu," 2002.

[16] U. A.J. and S. P.D., "Crossover Operators in Genetic Algorithms: a Review," *ICTACT J. Soft Comput.*, vol. 06, no. 01, pp. 1083–1092, 2015, doi: 10.21917/ijsc.2015.0150.

[17] A. E. Eiben, "Multiparent Recombination in Evolutionary Computing." Springer, pp. 175-192 BT-Advances in Evolutionary Computing, 2002.

[18] A. E. Eiben, P. Raué, and Z. Ruttkay, "Genetic algorithms with multi-parent recombination," 1994.

[19] A. E. Eiben and C. H. M. Van Kemenade, "Diagonal crossover in genetic algorithms for numerical optimization," *Control Cybern.*, vol. 26, no. 3, pp. 442–465, 1997.

[20] U. Bodenhofer and Q. Ai, "Genetic Algorithms : Theory and Applications Genetic Algorithms : Theory and Applications," no. May, 2014.

[21] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res. - JMLR*, vol. 3, Mar. 2003.

[22] H. Budak, "Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım," 2018. doi: 10.19113/sdufbed.01653.

[23] V. Kannan, "Feature Selection using Genetic Algorithms," pp. 1–21, 2018, doi: https://doi.org/10.31979/etd.6mq4-cp5p.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[25] S. Metlek and K. Kayaalp, "Makine Öğrenmesinde, Teoriden Örnek Matlab Uygulamalarına Kadar Destek Vektör Makinaları," Jan. 2021.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[27] S. Ayhan and Ş. Erdoğmuş, "Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi," vol. 9, no. 1, pp. 175–198, 2014.

[28] A. Tharwat, "Parameter investigation of support vector machine classifier with kernel functions," *Knowl. Inf. Syst.*, vol. 61, no. 3, pp. 1269–1302, 2019, doi: 10.1007/s10115-019-01335-4.

[29] N. Atasoy and D. Tabak, "Destek Vektör Makineleri Kullanarak Yüz Tanıma Uygulaması Geliştirilmesi," *E-Journal New World Sci. Acad.*, vol. 13, pp. 119–127, Apr. 2018, doi: 10.12739/NWSA.2018.13.2.1A0406.

[30] O. H. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A genetic algorithm-based feature selection," 2014.

[31] S. P. T. P. Phyu and G. Srijuntongsiri, “Effect of the number of parents on the performance of multi-parent genetic algorithm,” in *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, 2016, pp. 1–6.

[32] R. Santana *et al.*, “Genetic algorithms for feature selection in the children and adolescents depression context,” *Proc. - 18th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019*, pp. 1470–1475, 2019, doi: 10.1109/ICMLA.2019.00241.

[33] E. C. Matel, A. M. Sison, and R. P. Medina, “Optimization of Network Intrusion Detection System Using Genetic Algorithm with Improved Feature Selection Technique,” *2019 IEEE 11th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2019*, 2019, doi: 10.1109/HNICEM48295.2019.9073439.

[34] N. Maleki, Y. Zeinali, and S. T. A. Niaki, “A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection,” *Expert Syst. Appl.*, vol. 164, no. September 2020, p. 113981, 2021, doi: 10.1016/j.eswa.2020.113981.

[35] V. S. Desdhanty and Z. Rustam, “Liver Cancer Classification Using Random Forest and Extreme Gradient Boosting (XGBoost) with Genetic Algorithm as Feature Selection,” pp. 716–719, 2022, doi: 10.1109/dasa53625.2021.9682311.

[36] J. Too, A. R. Abdullah, and N. M. Saad, “Binary competitive swarm optimizer approaches for feature selection,” *Computation*, vol. 7, no. 2, Jun. 2019, doi: 10.3390/COMPUTATION7020031.

[37] Z. Halim *et al.*, “An effective genetic algorithm-based feature selection method for intrusion detection systems,” *Comput. Secur.*, vol. 110, p. 102448, 2021, doi: 10.1016/j.cose.2021.102448.