




**T.C.**

**SELÇUK ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

# VERİ KÜMELERİNDEKİ EKSİK DEĞERLERİN YENİ YAKLAŞIMLAR KULLANILARAK HESAPLANMASI

**İbrahim Berkan AYDİLEK**

**DOKTORA TEZİ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Haziran 2013**
**KONYA**
**Her Hakkı Saklıdır**

# ÖZET

## DOKTORA TEZİ

## VERİ KÜMELERİNDEKİ EKSİK DEĞERLERİN YENİ YAKLAŞIMLAR KULLANILARAK HESAPLANMASI

**İbrahim Berkan AYDİLEK**

**Selçuk Üniversitesi Fen Bilimleri Enstitüsü**
**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Prof. Dr. Ahmet ARSLAN**

**2013, 91 Sayfa**

**Jüri**
**Prof. Dr. Ahmet ARSLAN**
**Prof. Dr. Novruz ALLAHVERDİ**
**Prof. Dr. Şirzat KAHRAMANLI**
**Doç. Dr. Harun UĞUZ**
**Doç. Dr. Erkan ÜLKER**

Veri kümeleri; veri madenciliği, makine öğrenmesi veya yapay zeka gibi disiplinlerin uygulanabilmesi için gereklidir. Veri kümelerindeki verinin kalitesi, doğru araştırma sonuçları elde edebilmek adına önemli bir konudur. Veri kümelerinde çeşitli nedenlerle veri kalitesini azaltan değeri olmayan nitelikler bulunabilmektedir. Değeri olmayan bu eksik değerler yapılmak istenen çalışmaya ait sonuçların güvenirliğini riske atabilmektedir. Bu nedenle veri kalitesini artırmaya yönelik yöntemler ile veri kümelerindeki eksik değer probleminin giderilmesi gerekmektedir. Bu tez çalışmasında eksik değer hesaplamasında kullanılan klasik yöntemlerden bahsedilerek alternatif gelişmiş yöntemler önerilmiştir. Daha önce konuyla ilgili yapılmış olan çalışmaların faydalarından bahsedilerek eksik değer hesaplamasının önemi vurgulanmıştır. Bulanık c-ortalamaları, destek vektör regresyonu ve genetik algoritmaların hibrit kullanımı ile geliştirilen bir yaklaşım ve ayrıca en yakın k-komşu ve yapay sinir ağlarının hibrit kullanımı sonucu geliştirilen bir diğer yaklaşım önerilmiştir. Bu yaklaşımlarda kullanılan temel algoritmalar olan bulanık c-ortalamaları ve en yakın k-komşu algoritmaları için en uygun parametre değerlerini bulan otomatik bir model önerilmiştir. Önerilen yaklaşımlar literatürde yaygın kullanılan veri kümeleri ile test edilmiş ve benzer diğer yaklaşımlar ile kıyaslanmıştır. Benzer yöntemlerin eksikliklerine karşı önerilen hibrit yaklaşımların literatüre kattığı yenilikler anlatılmıştır. Elde edilen araştırma sonuçlarında, önerilen hibrit yaklaşımların performanslarının benzer yöntemlere göre daha üstün ve tutarlı olduğunu görülmüştür.

**Anahtar Kelimeler:** Bulanık c-ortalamaları, Eksik değerler, En yakın k-komşu, Hibrit yaklaşımlar, Kayıp değerler, Kayıp veriler, Veri önişleme

# ABSTRACT

## Ph.D THESIS

## ESTIMATING MISSING VALUES IN DATASETS USING NOVEL HYBRID APPROACHES

**İbrahim Berkan AYDİLEK**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY**
**THE DEGREE OF DOCTOR OF PHILOSOPHY**
**IN COMPUTER ENGINEERING**

**Advisor: Prof. Dr. Ahmet ARSLAN**

**2013, 91 Pages**

**Jury**
**Prof. Dr. Ahmet ARSLAN**
**Prof. Dr. Novruz ALLAHVERDİ**
**Prof. Dr. Şirzat KAHRAMANLI**
**Assoc. Prof. Dr. Harun UĞUZ**
**Assoc. Prof. Dr. Erkan ÜLKER**

Data mining, machine learning or artificial intelligence algorithms need a dataset to produce and evaluate research results. Data quality is a significant issue to obtain accurate research results. Many datasets may contain one or more missing values in a row due to various reasons. Missing values reduce data quality and even may jeopardize research results. Therefore, before using missing values in data mining or machine learning methods, they should be handled and estimated without reduce the data quality. In this paper basic conventional and computational intelligence imputation techniques are mentioned. Advantages of closer literature researches bring out the importance of dealing with missing values in datasets. A novel hybrid approach using fuzzy c-means, support vector regression and genetic algorithms is proposed. Also another novel hybrid approach k-nearest neighbors, artificial neural networks is also proposed. Fuzzy c-means and k-nearest neighbors algorithms' parameters are automatically optimized. Approaches tested with different kinds of datasets, which are frequently used in literature and additionally proposed approaches are compared with other closer methods in literature. Disadvantages of closer methods are mentioned in order to assess the originality of the proposed approaches. Findings showed that new novel proposed hybrid approaches performances are more stable and better than the other closer methods.

**Keywords:** Data preprocessing, Fuzzy c-means, Hybrid approaches, Incomplete Values, Imputation, K-Nearest Neighbor, Missing Data, Missing Values

# ÖNSÖZ

Bu tez çalışmamda gösterdiği hedeflerle, fikirlerle ve örnek hoca kişiliğiyle çalışmalarıma katkı sağlayan danışmanım Sayın Prof. Dr. Ahmet ARSLAN hocama teşekkürlerimi sunarım.

Tez izleme komitesinde bulunan Sayın Prof. Dr. Novruz ALLAHVERDİ, Sayın Doç. Dr. Harun UĞUZ ayrıca Sayın Yrd. Doç. Dr. Halife KODAZ hocalarımın yaptıkları olumlu görüşleriyle çalışmalarımın iyileşmesine katkıda bulundukları için teşekkürler ederim.

Tüm hocalarım ve mesai arkadaşlarıma özellikle Barış KOÇER ve Ersin KAYA' ya desteklerinden dolayı teşekkür ederim.

Hayatım boyunca anlayış ve sabırlarından dolayı değerli Aile fertlerime ve manevi desteklerinden dolayı biricik Eşime sonsuz minnettarım.

Eksik değer hesaplama problemi dünyada uygulama alanlarında geniş ilgi görmeye devam etmektedir, ülkemizde bilgisayar bilimleri konusunda çalışma yapan araştırmacılara bu problemin giderilmesi hususunda ve konu ile ilgili gelecekte yapılacak olan araştırmalara bu tez çalışmasının yol gösterici faydalı bir Türkçe kaynak olmasını temenni ederim.

İbrahim Berkan AYDİLEK
KONYA 2013

# İÇİNDEKİLER

**ÖZET** ..... **iv**

**ABSTRACT** ..... **v**

**ÖNSÖZ** ..... **vi**

**İÇİNDEKİLER** ..... **vii**

**SİMGELER VE KISALTMALAR** ..... **ix**

**1. GİRİŞ** ..... **1**

**2. LİTERATÜR ARAŞTIRMASI** ..... **4**

**3. VERİ TABANLARINDA BİLGİ KEŞFİ VE VERİ MADENCİLİĞİ** ..... **9**

3.1. Veri, Enformasyon ve Bilgi Kavramları ..... 9
3.2. Veri Tabanı Sistemleri, Veri Kümesi ve Veri Ambarı ..... 10
3.3. Veri Tabanlarında Bilgi Keşfi Süreci ..... 11
3.4. Veri Madenciliği Teknikleri ..... 15
3.4.1. Sınıflama ve regresyon ..... 15
3.4.2. Kümeleme ..... 16
3.4.3. Birliktelik analizi ..... 17
3.5. Veri Madenciliğinde Karşılaşılan Problemler ..... 17
3.5.1. Veri tipi ..... 18
3.5.2. Veri boyutu ..... 18
3.5.3. Aykırı veri ..... 18
3.5.4. Gürültülü, tutarsız veri ..... 18
3.5.5. Eksik veri ..... 19

**4. KULLANILAN YARDIMCI METOTLAR** ..... **22**

4.1. Destek Vektör Regresyonu (DVR) ve Genetik Algoritmalar (GA) ile Eksik Değer Hesaplama ..... 22
4.1.1. Destek vektör regresyonu (DVR) ..... 22
4.1.2. Genetik algoritmalar (GA) ..... 26
4.2. Bulanık c-ortalamalar (BCO) ile Eksik Değer Hesaplama ..... 36
4.2.1. Bulanık c-ortalamalar uygulaması ..... 37
4.3 Yapay Sinir Ağları (YSA) ve Genetik Algoritmalar (GA) ile Eksik Değer Hesaplama ..... 38
4.3.1. Yapay sinir ağları (YSA) ..... 38
4.4. En Yakın k-Komşular (EYK) ile Eksik Değer Hesaplama ..... 45
4.2.1. En yakın k-komşular uygulaması ..... 47

**5. GELİŞTİRİLEN EKSİK DEĞER HESAPLAMA YAKLAŞIMLARI.....48**

5.1. Eksik Değerlerin Bulanık c-ortalamaları, Destek Vektör Regresyonu ve Genetik Algoritmalar (BcoDvrGa) Hibrit Kullanımı ile Hesaplanması.....48
5.2. Eksik Değerlerin En Yakın K-Komşu ve Yapay Sinir Ağları (EykYsa) Hibrit Kullanımı ile Hesaplanması.....50

**6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....52**

6.1. BcoDvrGa Ön Hazırlık.....52
6.2. Araştırma Sonuçları İçin Kıyaslama Ölçütleri.....53
6.3. BcoDvrGa Araştırma Sonuçları.....55
6.4. EykYsa Ön Hazırlık.....62
6.5. EykYsa Araştırma Sonuçları.....64
6.6. Eksik Değer Hesaplamasını Etkileyen Özellikler.....69

**7. SONUÇLAR VE ÖNERİLER.....79**

7.1. Sonuçlar.....79
7.2. Öneriler.....80

**KAYNAKLAR.....82**

**ÖZGEÇMİŞ.....91**

## SİMGELER VE KISALTMALAR

| | |
|---|---|
| Bco | : Bulanık c-ortalamalar |
| BcoDvrGa | : Bulanık c-ortalamaları, destek vektör regresyonu ve genetik algoritmalar yaklaşımı |
| BcoGa | : Bulanık c-ortalamaları ve genetik algoritmalar yaklaşımı |
| D | : Göreceli tahmin doğruluğu |
| Dvm | : Destek vektör makinaları |
| Dvr | : Destek vektör regresyonu |
| DvrGa | : Destek vektör regresyonu ve genetik algoritmalar yaklaşımı |
| Eyk | : En yakın k-komşular |
| EykYsa | : En yakın k-komşu, yapay sinir ağları yaklaşımı |
| Ga | : Genetik algoritmalar |
| Hkok | : Hata karelerinin ortalamasının karekökü |
| k | : En yakın k-komşular algoritması komşu adedi |
| Kayıt | : Veri kümesi nesnesi |
| KKT | : Karush-kuhn-tucker koşulları |
| Mod | : Bir dizideki en sık geçen değer |
| Nitelik | : Veri kümesi kayıtlarının bir özelliği |
| Ortanca | : Bir dizinin büyüklüğüne göre sırasındaki orta değer, medyan |
| t | : Çalışma süresi (saniye) |
| UCI | : California Üniversitesi Bilimsel Veri Tabanı |
| Ysa | : Yapay sinir ağları |
| YsaGa | : Yapay sinir ağları genetik algoritmalar yaklaşımı |
| W | : Wilcoxon sıra toplamı istatiksel güven testi |

# 1. GİRİŞ

Eksik değerler veri madenciliği, makine öğrenmesi ve diğer bilgi sistemlerinde istenmeyen bir durumdur (Nelwamondo ve ark., 2009). Son yıllarda eksik değerlerin ya da diğer adıyla kayıp verilerin hesaplanması ve tahmin edilmesi, araştırmacıların kaliteli veriye ulaşma isteğinden dolayı popüler bir konu haline gelmiştir.

İşlenmemiş ham veri işlendikten ve yorumlandıktan sonra bilgi elde edilmektedir. Verinin bilgiye dönüşme sürecine bilgi keşfi denmektedir. Veri kalitesi makine öğrenmesi, veri madenciliği ve bilgi keşfi için büyük öneme sahiptir (Patil, 2010). Veri temizleme veri önişleme aşamalarından biridir. Veri temizleme sürecinin amacı kaliteli veri üretmektir. Verilerden yola çıkılarak elde edilecek olan bilginin keşfi sürecinde var olan veriler eksik değerler içerebilmektedir. Bu eksik değerlerin giderilmesi bilgi keşfi sürecinde göz önünde bulundurulması gereken bir adımdır (Ramirez ve ark., 2011).

Çoğu bilgisayar bilimleri yöntemi, yapay sinir ağları, destek vektör makineleri ve karar ağaçları gibi kestirim yapan yaklaşımlar daha önce görülmüş veriyi girişte eğitim verisi olarak alarak çıkışta bir sınıflama yapmaktadır. Bu gibi kestirim modelleri girişte bir veya birden çok veride eksik değer olması durumunda ya çalışmaz ya da hatalı tahmin üretmektedir. Sonuç olarak eğer giriş veri nitelikleri tam değil ise bu durumda karar verme amacıyla kullanılamazlar (Marwala, 2009).

Veri kümelerindeki eksik değerlerle başa çıkabilmek için eksik veriyi göz ardı etmek, ilgili kaydı silmek, sıfır ile doldurmak, sık geçen ifade ile doldurmak veya ilgili kayıt ya da niteliğin satır, sütun ortalaması ile doldurulması gibi basit klasik yöntemler karmaşık hesaplama yöntemlerinin yerine kullanılmaktadır (Cheng ve ark., 2011; Meng ve Shi, 2012). Fakat bu gibi basit klasik yöntemler eksik gözlemleri yok sayarak verimi düşürmekte aynı zamanda var olan veriyi yanlılaştırarak sistematik anlamda kalitesizleştirmektedir (Little ve Rubin, 1987; Pelckmans ve ark., 2005; Nelwamondo ve Marwala, 2008).

Örneğin hava durumları kayıtlarının tutulduğu bir veri kümesinde yağmurlu havalarda ölçüm yapan metrekareye düşen yağmur miktarını ölçen ağırlık sensörü görevini yapmadığında tüm yağmurlu hava kayıtlarının silinmesi, hava durumu kayıtlarının tutulduğu bölgeye hiç yağmur yağmadığı gibi gösteren bir veri kümesi bırakabilmektedir. Yine buna benzer bir durum olarak bir ankette katılımcılardan siyasi

görüş, yaş, kilo veya maaş durumları ile ilgili bilgiler istendiğinde örnek olarak katılımcılar çok yüksek ya da çok düşük maaş aldığının bilinmesinin istemediği için alakalı soruya yanıt vermekten çekinebilmektedirler. Böyle bir veri kümesinde eksik cevap yerine ortalama maaş değeri ya da nitelikte sık geçen ifadeyi (düşük, orta, yüksek) koymak yapılmak istenen anket çalışması sonuçlarına olan güveni azaltabilmektedir. Kaliteli veri madenciliği ancak kaliteli veri ile yapılabilmektedir (Salama, 2010; Sim ve ark., 2011). Bu ve benzeri durumlarda eksik olan değerin eksik olmayan diğer nitelik ve kayıtlara bakılarak hesaplanması ve böylece verinin kalitesinin artırılması daha uygun görülmektedir (Han ve Kamber, 2001).

Biyolojik veya medikal deney sonuçları, kimyasal analiz sonuçları, meteorolojik, mikro dizi gen görüntüleme teknolojileri veya anket veri tabanları gibi verileri içeren veri kümeleri eksik değerler içermektedir (Mohamed ve Marwala, 2005; Abdella ve Marwala, 2005). Eksik değerlerin oluşma nedenleri arasında elektronik sensör bozukluğu, ölçüm hatalarından kaynaklanan nedenler, bilimsel bir deneyin sonuçlarının ortaya çıkmaması örneğin kimyasal bir tepkimenin gerçekleşmemesi veya deney sırasında sonuçları etkileyecek biyolojik bir hücrenin oksijen temin edememesi, yetersiz çözünürlükteki resim dosyaları, görüntüleme cihazlarının lenslerinde olması muhtemel toz parçacıkları veya çizikler, dijital sistemlerde veri transferinde meydana gelen problemler, verileri toplayan kişinin bazı verileri kayıt altına almaması veya bilgisayar operatörünün giriş hataları ve düzenlenen bir ankette katılımcıların bazı sorulara cevap vermek istememesi gibi nedenler gösterilmektedir (Abdella ve Marwala, 2005; Mohamed ve Marwala, 2005; Qiao ve ark., 2005; Nelwamondo ve ark., 2007; Mahmoud ve Emzir, 2012).

Eğer eksik değerler düzgün bir şekilde ele alınıp hesaplanmaz ise araştırma sonuçlarının geçerliliği azalabileceği gibi bazı durumlarda araştırma başarısızlığa bile uğrayabilmektedir (Hulse ve Khoshgoftaar, 2007; Genc ve ark., 2010; Di Nuovo, 2011; Bergmeir ve Benitez, 2012).

Eksik verileri tamamlamak için bazen anket sorularının tekrar katılımcılara sorulması veya eksik veriler için deneylerin tekrar edilmesi gibi yollar izlenebilmektedir. Fakat bu hem çok uzun zaman alabilmekte hem de çok maliyetli olabilmektedir (Sehgal ve ark., 2005). Ayrıca eksik olan değerlerin yine tespit edilebileceğinin garantisi yoktur. Yani ankete yine cevap verilmek istenmeyebilir veya deney sonuçları yine eksik değerler içerebilir. Özellikle mikro dizi gen çalışmalarında eksik değer oluşması normal olarak kabul görmektedir. Sistemlerde kullanılan

sensörlerin yedekli hale getirilmesi bu gibi sistemlerde eksik değer oluşmasını önlemesi için bir çözüm olarak görülebilir. Fakat günümüzde öngörülemeyen ve sayısı fazla giriş verisi kullanan sistemlerde hızlı çözümler üretilebilmesi için bilgi keşfi yöntemlerinin sistemlere gömülü hale getirilmesi gerekliliği doğmuştur. Örneğin uçak yolculuğu sırasında elektronik sensörlerden birinin bozulması veya bir sistemde problem sebebinin tam olarak ne olduğunun anlaşılmasının zaman alabileceği gibi durumlarda eksik veri hesaplama zorunluluğu oluşmaktadır (Marwala, 2009).

Tezin giriş bölümünde eksik değer nedir, eksik değerlerin hangi durumlarda ve hangi koşullarda oluşabileceği, eksik değerlerin hangi sorunlara neden olabileceği, eksik değerlerin giderilebilmesi için kullanılabilecek klasik yöntemler ve bu yöntemlerin eksik değer gidermede ne derece kullanışlı olduğundan bahsedilmiştir. İkinci bölümde ise daha önce konuyla ilgili yapılmış olan benzer çalışmalara değinilerek yapılmak istenen çalışmanın literatür içerisindeki yeri ve literatür çalışmalarında görülen eksikliklere getirilecek çözümlerden bahsedilmiştir. Üçüncü bölümde eksik veri probleminin bilgi keşfindeki yeri ve verilerden yola çıkılarak elde edilen bilgi kavramı için kullanılan temel yöntem ve kavramlardan bahsedilmiştir. Dördüncü bölümde önerilen yaklaşımlarla doğrudan veya dolaylı olarak ilgili yardımcı materyal ve yöntemler üzerinde durulmuştur. Beşinci bölümde tez çalışmasında önerilen bulanık c-ortalamaları, destek vektör regresyonu ve genetik algoritmalar (BcoDvrGa) ve en yakın k-komşu, yapay sinir ağları (EykYsa) hibrit yaklaşımları ve literatüre getirdiği yenilikler anlatılmıştır. Altıncı bölümde geliştirilen yaklaşımlar literatürdeki benzer metotlarla karşılaştırılmıştır. Karşılaştırılan algoritmalarının verimliliğinin ölçülebilmesi için hata karelerinin ortalamasının karekökü (HKOK), göreceli tahmin doğruluğu (D), Wilcoxon sıra toplamı istatiksel güven testi (W), saniye cinsinden çalışma zamanları (t) ve korelasyon katsayısı (r) performans ölçüm değerlerinin kıyaslaması yapılmıştır. Kıyaslama sonuçları grafik, şekil ve çizelgelerle gösterilmiş, bunlar üzerinde tartışmalar yapılmıştır. Son bölümde ise tezde yapılan çalışmalar ile ilgili elde edilen genel sonuçlara değinilmiş ve eksik değer hesaplamasıyla ilgili gelecekte yapılabilecek araştırmalara önerilerde bulunulmuştur.

## 2. LİTERATÜR ARAŞTIRMASI

Bu bölümde tezde eksik değer hesaplaması konusunda çalışılan en yakın k-komşu, yapay sinir ağları, bulanık c-ortalamaları, destek vektör regresyonu ve genetik algoritmalar metotları ile ilgili önceki yapılmış çalışmalar kısaca sunulmuştur.

Abdella ve ark. tarafından veri tabanındaki eksik değerler yapay sinir ağları ve genetik algoritmalar kullanarak tahmin edilmiştir (Abdella ve Marwala, 2005). Li ve ark. K-ortamalar kümeleme yöntemi ile aynı küme içindeki farklı nesnelerin birbirlerine olan uzaklığı ve kümenin merkezine olan uzaklıklarının hesaplamasını yapmıştır. Belirsizlik ve kesin olmama gibi durumlara karşı daha toleranslı esnek hesaplamalı kümeleme metodunu tam olmayan verileri düzeltmek için kullanmıştır (Li ve ark., 2004).

Liao ve ark. bulanık c-ortalamalar kümeleme algoritmasını kullanarak veri üzerinde kayan pencereler yöntemiyle eksik değer tamamlama yaparak verinin kalitesini artırmıştır (Liao ve ark., 2009).

Twala, karar ağaçları ile eksik veri hesaplaması yapmıştır. Eksik verileri karar ağaçları ile tamamlayabilmek için ilk başta ağaç modelini oluşturmuş. Eksik olmayan kayıtlar ile ağaç yapısı oluşturulmuş ve eksik olan değer sınıf niteliği olarak düşünülerek karar ağacı modeline göre tespit edilmiştir (Twala, 2009).

Garcia ve Hruschka, saf Bayes algoritması ile eksik değer hesaplaması yapmıştır. Veri kümesi kayıtlarının nitelik değerlerine bakarak eksik değer içeren niteliğin yerine gelebilecek en büyük olasılıklı nitelik verisi eksik değer olarak atanmıştır (Garcia ve Hruschka, 2005).

Pelckmans ve ark. eksik verileri yeniden onarmaktan ziyade alternatif bir yaklaşım önermiştir buna göre destek vektör makineleri kullanılarak eksik verinin sonuca ve tahmin edilen maliyet üzerine olan etkileri ile ilgilenmişler ve bu çalışmada eksik değerlerin kovaryansları için bir model oluşturularak en büyük olasılıklı değeri tercih eden bir kestirim metoduyla tahminler yapılmıştır. Bu çalışmanın avantajı sınıflama eğitim verisinde eksik değerlerin olmasına rağmen kuralların öğrenilmesi, dezavantajı ise eksik değerlerin tam olarak ne olduğundan çok sınıflama başarısının artırılmasına yönelik çalışma yapılmış olmasıdır (Pelckmans ve ark., 2005).

Lim ve ark. örüntü sınıflaması yapabilmek için eksik değer içeren eğitim ve test verileriyle çalışabilecek bulanık artmap ve bulanık c-ortalamalar kümelemesi ile birlikte

hibrit oluşturulmuş bir yapay sinir ağı önermiştir. Fakat bulanık artmap'ın dezavantajı eğitim veri sırasının, yani verilerin yerleştirilme sıra düzenin sonuç üzerindeki büyük etkisinin olması ve sonuçların buna bağlı olarak değişiklik göstermesine izin vermesidir. Bunun yanında bulanık artmap kullanımında ihtiyat parametresi olarak adlandırılan parametre değerinin seçimi biraz zor olabilmektedir. Bundan dolayı farklı parametre değerleri seçimi çok farklı sonuçların üretilmesine sebep olmaktadır (Lim ve ark., 2005).

Hathaway ve ark. eksik farklı verilerin kümelemesi üzerine bir yöntem önermiştir. Bu yöntemin avantajı bulanık c-ortalamalar metodunun eksik veriler için güvenilir bir kümeleme algoritması olarak kabul edilmiş olmasıdır (Hathaway ve Bezdek, 2002).

Feng ve ark. dik kod şeması kullanan destek vektör regresyonu metodu temelli DNA mikro dizi gen tanıma verisi için eksik değer hesaplaması yapılmasını önermiştir. Veri kümesinde bulunan çıkış veya sınıf nitelikleri giriş niteliği olarak, giriş niteliği ise çıkış sınıf niteliği olarak ayarlanmıştır. Yapılan çalışmada karşılaştırılan yöntemler en yakın k-komşu ve Bayes temel bileşen analiz tamamlama metotlarına göre destek vektör regresyonu yöntemi eksik değer tamamlama konusunda daha etkili olduğu gösterilmiştir (Feng ve ark., 2005). Destek vektör regresyonun bir diğer araştırmada ise önemli avantajından bahsedilirken daha az hesaplama süresinin olduğu öne sürülmüştür (Wang ve ark., 2006), fakat tek başına kullanılmayıp kümeleme algoritmalarıyla hibrit kullanılmaları aykırı veya gürültülü değerler için daha hassas ve iyi sonuçlar ürettiği görülmüştür.

Schneider, beklenti maksimizasyonu algoritması ile veri kümesi niteliklerinin ortalama ve kovaryans matrisini iteratif iyileştirmek suretiyle veri kümesinde bulunan eksik değerleri hesaplamıştır (Schneider, 2001).

Timm ve ark. eksik veri kümelerinin veri analizinde önemli ve büyük problemlere neden olduğu belirtmiştir. Çalışmalarında sınıf niteliği üzerinden istatiksel bir yöntem oluşturarak eksik veri noktalarını uygun küme noktalarına atayarak tespit etmiştir (Timm ve ark., 2004).

Farhangfar ve ark. daha önce önerilmiş eksik veri tamamlama tekniklerine kapsamlı bir inceleme yapmıştır. Düşük kaliteli tekli tamamlama tekniği ile elde edilen tamamlama isabet başarısının diğer bazı gelişmiş tamamlama tekniklerinin sonuçlarıyla mukayese edilebilecek başarıda olduğunu göstermiştir (Farhangfar ve ark., 2007).

Li ve ark. eksik nitelikleri sanki birer kesikli değer aralığı olarak varsayarak çalışmalarında en yakın k-komşu aralığından esinlenmiş özgün bir bulanık c-ortalamalar algoritması geliştirmiştir. Yöntemin dezavantajı, teorik olarak bulanık c-ortalamalar *küme merkezi sayısını* tespit edememesi olarak görülmüş ve ilerde yapılacak çalışmalarda bu problemin giderilmesi gerektiği önerilmiştir (Li ve ark., 2010).

Nuovo bulanık c-ortalamalar ile eksik veri tamamlama yöntemini silme yöntemiyle karşılaştırmıştır. Çalışmasında psikoloji araştırma ortamından elde edilen zihin bozukluğu yaşayan hasta verileri üzerinde çalışmalar yapılmıştır. Elde edilen sonuçlarda eksik değer hesaplama teknikleri ile yapılan düzeltmeler özellikle bulanık c-ortalamalar algoritmasının araştırma sonuçlarını daha güçlü hale getirdiği gösterilmiştir. Bulanık c-ortalamaların, regresyon eksik tamamlama ve beklenti maksimizasyonu eksik tamamlama yöntemlerine göre daha isabetli eksik veri tamamladığı ve başarılı olduğu gösterilmiştir (Di Nuovo, 2011). Fakat bu çalışmada görülen bulanık c-ortalamalar yönteminin eksikliği, uygulamalarda *ağırlık faktörü parametresi* olarak adlandırılan parametre değerinin sabit olarak *2* değerini almış olmasıdır. Aslında bu parametre değeri daha hassas ve iyi sonuçlar elde edilebilmesi için kullanılan veri kümesinin özellik veya tipine göre değişiklik göstermelidir.

Troyanskaya ve ark. DNA gen mikro dizilerinde bulunan eksik değerleri hesaplamak için ağırlıklı en yakın k-komşu yöntemini önermiştir. Fakat *k* komşu adedinin tam olarak en uygun miktarı hakkında tatmin edici bir araştırma sonucu ortaya konulamamıştır (Troyanskaya ve ark., 2001).

Oba ve ark. istatiksel olasılık prensiplerine göre eksik değeri tamamlayan Bayes temel bileşenler analizi ile hesaplama yapmıştır, bu yöntemde temel bileşenler analizi ile regresyon eğrisi oluşturularak Bayes olasılığı ile tahminler yapılmıştır (Oba ve ark., 2003).

Thompson ve ark. boyutlanabilir doğal oto kodlayıcı adını verdiği bir yapı geliştirmiş bu yapı ile eksik sensör verilerinin tamirini yapay sinir ağlarını temel alan bir yöntem kullanarak gerçekleştirmiştir (Thompson ve ark., 2003).

Qiao ve ark. eksik veri hesaplayıcı hibrit yöntemini parçacık sürü optimizasyonu ve sinir ağları ile gerçekleştirmiş (Qiao ve ark., 2005), bunu bir bitkinin hareket değişimlerini takip eden sensörden kaynaklanan eksik değer ölçümlerini gidermek amacıyla kullanmıştır.

Abdella ve ark. tarafından veri tabanındaki eksik değerler yapay sinir ağları ve genetik algoritmalar kullanarak tahmin edilmiştir (Abdella ve Marwala, 2005).

Mohamed ve ark. hem ajan temelli hem de yapay sinir ağları temelli veri tabanındaki eksik veriyi hesaplamak için bir yöntem önermiştir (Mohamed ve Marwala, 2005).

Nelwamondo ve ark. yapay sinir ağları ile beklenti maksimizasyonu tekniklerini karşılaştırmıştır (Nelwamondo ve ark., 2007). Mohamed ve ark. ayrıca yapay sinir ağlarını, temel bileşen analizini ve genetik algoritmaları birlikte kullanan eksik veri hesaplaması yapan bir yöntem önermiştir (Mohamed ve ark., 2007). Nelwamondo ve ark. daha önce yapılmamış özgün bir teknik olan dinamik programlama ile birlikte yapay sinir ağlarıyla eksik veri tahmini yapmıştır (Nelwamondo ve ark., 2009).

Blend ve ark. eksik veri tamamlama teknikleri ve etkileriyle ilgili kapsamlı bir karşılaştırma çalışması hazırlamıştır (Blend ve Marwala, 2008). Bu makalede kendini çağıran yapay sinir ağları ve sinirsel bulanık sistemler ile oluşturulan yöntemler sıcak-deste eksik tamamlama yöntemiyle kıyaslanmıştır.

Hlalele ve ark. eksik değer tamamlaması için temel bileşen analizi, sinirsel bulanık ve genetik algoritmalarla eksik değer tamamlama çalışması yapmıştır (Hlalele ve ark., 2009) önerdikleri yöntemde Güney Afrika'da HIV görülme sıklığı verilerini içeren veri kümesi kullanmış ve deney sonuçlarında isabet oranı yüksek şekilde eksik değer hesaplaması yapıldığı gösterilmiştir.

Patil, genetik algoritmaları temel alan çoklu eksik veri tamamlama teknikleri üzerine yoğunlaşmış ve çalışmalar yapmıştır (Patil, 2010) önerilen yöntemde karar ağaçlarının sınıflama başarısını uygunluk fonksiyonu olarak kabul eden genetik algoritmalarla veri kümelerindeki eksik veriler hesap edilmiştir.

Ramirez ve ark. eksik değerlerin tamamen rastlantısal olduğu durumlar için çok katmanlı sinir hücrelerini kullanarak eksik veri tamamlaması yapmıştır (Ramirez ve ark., 2011). Metodolojik çerçeve içerisinde yapay sinir ağları için geliştirilmiş otomatik olarak eksik değer tamamlaması yapan bir yöntem üzerine yoğunlaştılar. Çok katmanlı sinir hücreleri için değişik varyasyonlu mimariler ve geri yayılım algoritmaları ile testler yapılmıştır. Üç adet klasik eksik değer tamamlama yöntemi olan ortalama, mod ile tamamlama, regresyon ve sıcak-deste eksik değer tamamlama yöntemleriyle karşılaştırılmalar yapılmış ve önerilen yöntemle iyi sonuçlar elde edildiği gösterilmiştir.

Nelwamondo ve ark. çevrimiçi çalışan görüntüleme sistemleri için ara sıra elektronik sensör hatalarından kaynaklanan eksik verileri işleyebilmek için hibrit genetik algoritmalar ve hızlı benzetimli tavlama yaklaşımıyla eksik değer tahmini yapmıştır (Nelwamondo ve Marwala, 2008).

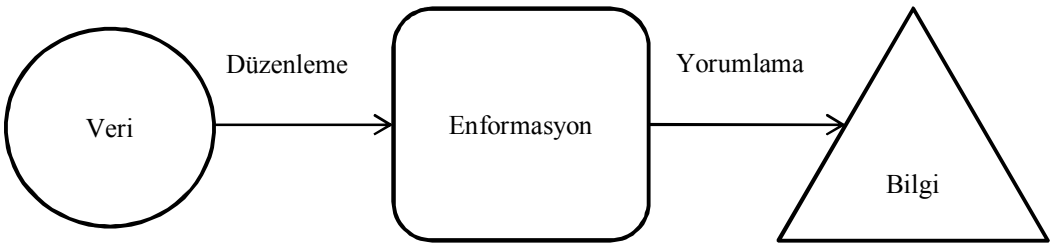
Literatürde bugüne dek belli bir algoritmanın belirgin bir üstünlüğü kanıtlanamamıştır. Bu yüzden eksik veri problemi üzerinde sıklıkla çalışılmaya devam edilen bir araştırma konusu olmaya devam etmektedir (Marwala, 2009, Aslan ve ark., 2011; Yozgatlıgil ve ark., 2013). Bu tez çalışmasında eksik veri hesaplaması konusunda yeni yapılan çalışmalar incelenmiş ve birden çok algoritmanın kullanıldığı hibrit çalışmalara önem verilmiştir. Literatür araştırmaları sonucunda bu tez çalışmasında geliştirilen yöntemlere benzer olan çalışmaların elde ettiği başarılı ve iyi sonuçların geliştirilen yöntemlerin eksik veri hesaplaması konusunda iyi sonuçlar üreteceğini onaylamıştır. Fakat yapılmış olan bu benzer çalışmalarda görülen eksiklik ve problemlerin giderilerek çok daha iyi hale getirilmesinin ise önerilen yaklaşımların gerekliliğini ve önerilen yaklaşımların doğru amaçları ortaya koyduğunu göstermiştir.

# 3. VERİ TABANLARINDA BİLGİ KEŞFİ VE VERİ MADENCİLİĞİ

## 3.1. Veri, Enformasyon ve Bilgi Kavramları

Veri yapılmak istenene bağlı olarak işlemlerin işlenmemiş bir biçimde kaydedilmesidir. Veri, açıklanmamış ve yorumlanmamış gözlemler, işlenmemiş gerçekler olarak tanımlanmaktadır (Guclu ve Sotirofski, 2006). Veri kavramı ham şekilde duran tek başına bir anlam ifade etmeyen çoğu zaman birden fazla olayın birbirine karışımından oluşan kayıt altına alınmış yığınlara verilen isimdir. Bilgisayar teknolojisi geliştikçe bu yığınlar dijital sistemlerde çok değişik dosya formatlarında saklanmaya başlanmıştır.

Enformasyon, işlenmiş, düzenlenmiş veri olarak tanımlanmıştır. İşleme veya düzenlenme başkaları tarafından yapılmakta ve enformasyon yalnızca ilgili kişi ya da kurum için bir anlam taşımaktadır (Barutçugil, 2002; Guclu ve Sotirofski, 2006). Değişik veri kaynaklarından bir amaç için elde edilmiş düzenlenmiş ve işlenmiş verilerdir. Enformasyon kavramı veriden bilgiye geçişte bir aşamadır ve çoğu kez veri ve enformasyon süreci birlikte adlandırılmasına rağmen birbirinden farklı ele alınmasında fayda vardır. Örneğin bir markete ait bütün veriler içindeki müşteri bilgilerini tutan yığın artık enformasyon adını almaktadır. Şekil 3.1'de verinin bilgiye dönüşmesi özetlenmiştir.




**Şekil 3.1.** Verinin bilgiye dönüştürülmesi

Bilgi kavramı yorumlanmış, işe yarar hale dönüştürülmüş enformasyon olarak tanımlanır ve insanlık gelişiminin temelinde büyük rolü vardır. Bilgi kavramı veri ve enformasyon kavramları ile yakından ilişkilidir. Bilginin ne olduğunun düşünülmesi milattan önce beşinci yüzyılda, felsefeci Sokrates'in bilginin sınırları tartışması ile başlamıştır (Malhotra, 1997; Guclu ve Sotirofski, 2006). Bilgi anlamlı, işe yarar,

yorumlanmış, değerlendirilmiş enformasyona verilen genel tanımdır. İnsana ait muhakeme gücü tarafından diğer insan ya da sistemlerin anlayabileceği yorumlar ya da değerlendirmeler olarak da bahsedilmektedir. Müşteri enformasyonunun kullanılmasıyla benzer satın alma eğiliminde bulunan müşteri profillerinin tespit edilerek gruplanması, daha sonra belirlenen gruplara farklı kampanyalar yapılması elde edilmiş bir bilginin sonucu olarak örneklenmektedir.

Hiçbir anlama gelmeyen *22451234* sayısı gürültü olarak adlandırılmaktadır. Eğer bu sayı değerinin öğrenci numarası veya bir banka hesap numarası olduğu biliniyor ise artık veri olarak adlandırılmaktadır. Bu veri *Hesap No: 22451234* şeklinde düzenlendiği takdirde enformasyona dönüşmektedir. Enformasyonlar arasındaki ilişkilerden faydalanarak üretilen nitelik kazanmış enformasyon topluluğuna bilgi adı verilmektedir. Örneğin kişinin hesap numarası *22451234* ise bu durumda ilgili hesap numarasıyla kişiye havale yapılması bilgi olarak elde edilmektedir (Allahverdi, 2002).

### 3.2. Veri Tabanı Sistemleri, Veri Kümesi ve Veri Ambarı

Veri tabanı, rastgele bir başlık altında birbirleri ile ilişkili verilerin sistemli olarak oluşturduğu yapılardır (Kaya ve Tekin, 2007; Özdemir ve ark., 2010). Veri oldukça geniş bir alana düz dosyalar halinde yayıldığı durumlarda bu verilerin kontrolü ve yönetimi oldukça zor bir hal almakta ve bu tip işlevsel verileri çok daha iyi kontrol altında tutan veri tabanı sistemlerine ihtiyaç duyulmaktadır (Narang, 2004; Özdemir ve ark., 2010). Bilgisayar sistemlerinde kaydedilen veri boyutu arttıkça beraberinde karışıklığı, güvensizliği ve düzensizliği peşinden getirmeye başlamıştır. Verileri belli bir düzende farklı formatlarda güvenli saklamaya yarayan sistemlere olan ihtiyaç artmaya başlayınca, bu eksikliği gidermek için veri tabanı sistemleri geliştirilmiştir. Bu sistemler sayesinde verilere hızlı, güvenli ve düzenli bir şekilde içinde barındırdığı sorgulama dilleri yardımıyla erişmek mümkün hale gelmiştir.

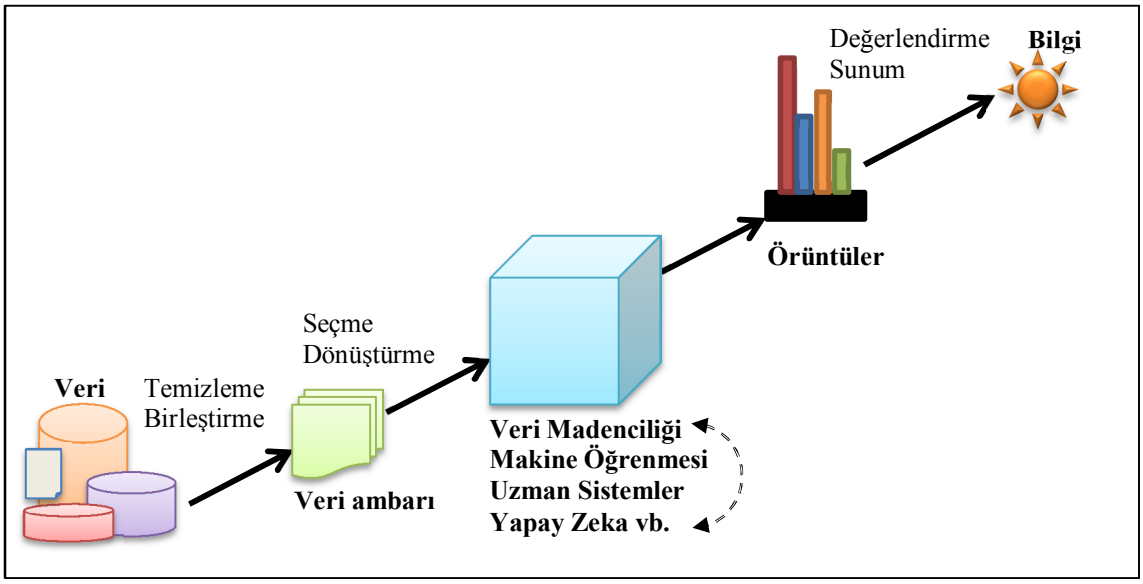
İşte bu sistemlerden elde edilen dijital verilere genel olarak veri tabanı, belirli bir amaç için özelleşmiş bir sorgu sonucunu yansıtıyorsa veri kümesi ya da veri seti içinde birden fazla veri veya veri formatı içerebilen farklı fiziksel ortamlardan toplanmış olabilen veri topluluğuna ise genel anlamda veri ambarı olarak adlandırmalar yapılmaktadır. Tüm bunların yanı sıra araştırmalarda ya da günlük hayatta tüm bu kavramların hepsine birden bilişim dilinde kısaca veri tanımı da yapılabilmektedir.

### 3.3. Veri Tabanlarında Bilgi Keşfi Süreci

Artık bilgisayarlar çok daha büyük miktardaki veriyi kayıt altına alabilmekte ve bu veriler üzerinde daha kısa sürelerde işlemler yapabilmektedir. Bunun yanında internet ağ cihazların çeşitlenmesi ve teknolojik olarak ilerlemesi ile bu verilere başka bilgisayarlardan hızlı erişebilmek mümkün olmaktadır (Alpaydın, 2000; Kayaalp, 2007). Günümüzde bilgisayar sistemlerinin ve bilgisayar ağ yapılarının gelişmesi ve buna bağlı olarak gelişen dijital işlemlerin tüm kayıtları ham veriler halinde kayıt edilmeye devam edilmektedir. Veriler gün geçtikçe katlanarak daha da artmakta, oluşan bu veri havuzundan anlamlı bilgiler çıkarılması ise zorlaşmaktadır.

Veri madenciliği, büyük hacimli verinin içinden geleceğin tahmin edilmesinde yardımcı olacak anlamlı ve faydalı, gizli bağlantı ve kuralların bilgisayar programları aracılığıyla aranması ve analizidir. Ayrıca veri madenciliği, çok büyük boyutlardaki verilerin içindeki ilişkileri inceleyerek aralarındaki bağıntıları bulmaya yardımcı olan veri analiz tekniğidir (Akpınar, 2000; Kayaalp, 2007).

Veri tabanlarında bilgi keşfi, verinin nasıl bir şekilde saklanması gerektiği ayrıca algoritmaların büyük veri kümelerine nasıl uygulanması gerektiği ile en son elde edilen sonuçların nasıl yorumlanması gerektiği gibi soruların cevaplarının arandığı aşamaları içermektedir (Altıntop, 2006). Verinin, bilgiye dönüşmesi için gereken işlem adımları yani bilgi keşfi süreci Şekil 3.2'deki gibi gösterilmektedir (Han ve Kamber, 2001).




**Şekil 3.2**. Veri tabanlarında bilgi keşfi süreci

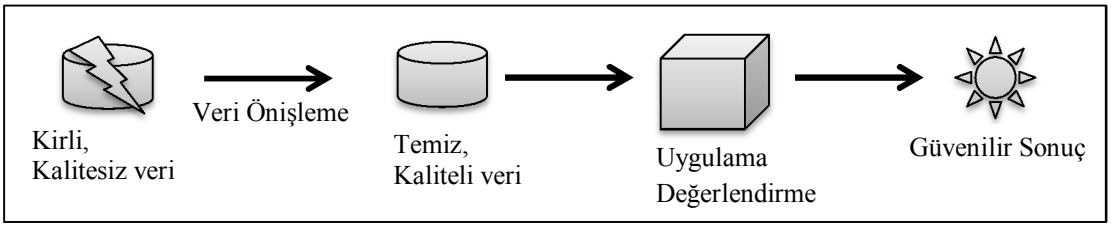
Veriden değerli bilgiler elde edilmesi süreci titiz ve dikkatli değerlendirilmesi gereken aşamalardan oluşmaktadır. Bilgi keşfinde izlenen aşamalar veri, veri önişleme, veri madenciliği, örüntü değerlendirme, bilgi ve bilgi sunumu şeklinde gerçekleşmektedir. Bu izlenen aşamaların her birine ayrı ayrı gereken önemin verilmesi topyekün bilgi keşfi sonuçları üzerinde olumlu etkiler oluşturacağı aşikardır.

Bilgi keşfi süreci adımları interaktif ve döngüseldir. Yani her adımda işlemler tekrar tekrar yapılabildiği gibi önceki ve sonraki aşamalara dönülebilmektedir. İstenen bir aşamadan tekrar geriye dönülüp önceki işlemler tekrar uygulanabilmektedir (Han ve Kamber, 2001; Maimon ve Rokach, 2010).

1. İlk adım olarak yapılmak istenen uygulama ait etki alanı oluşturulmakta, planlamalar yapılmakta ve son kullanıcı gözüyle çalışma amacı ifade edilmektedir.
2. İkinci aşama bilgi keşfi yapılacak veri kümesin oluşturulmaktadır. Araştırmada kullanılacak veri tabanı tablolarından ya da düz metin dosyalarından belirlenen niteliklerle birlikte bir alt veri kümesi oluşturulmaktadır.
3. Üçüncü aşamada veri üzerinde önişlemler yapılmaktadır. Bilgi keşfinde temiz ve kaliteli veri yapısı çok önemlidir (Aydın, 2007). *Veri temizleme*, veri içinde bulunan gürültülü, eksik, aykırı, tutarsız verilerle ilgilenme ve olası sorunlara karşı veriyi ele alarak gerekli düzeltmelerin yapılması yani verinin temizlenmesi sürecidir. *Veri birleştirme* işlemleri ise çok farklı veri kaynaklarından elde edilen değişik verilerin bir araya getirilmesi, düzenlenmesi veya mükerrer kayıtların silinmesi gibi görevleri içermektedir. Örneğin iki farklı veri tabanında aynı tür içeriği barındıran niteliklerin aynı isimle güncellenerek bir araya getirilmesi veya bir veri tabanında *metre* diğerinde *kilometre* olabilen birim ve benzeri değerlerin aynı formatlara çevrilmesi aşamasıdır. *Veri seçme*, analizi yapılacak çalışma için gerekli ve alakalı verilerin seçilmesi geri kalanlarının ise veri havuzundan çıkarılması işlemidir. Bu aşamada veri küpü, veri azaltma veya sıkıştırma teknikleriyle verinin gösterimi ya da boyutu düşürülerek daha az sayıya indirilmiş veri ile iyi araştırma sonuçları elde edilebilmesi amaçlanmaktadır. Temel bileşen analizi veya veri niteliklerini kesikli hale getirme yine bu aşamada

gerçekleştirilmektedir. *Veri dönüştürme* ise verinin algoritmaların istediği uygun formlara çevrilmesi, analiz yapacak yöntemin kabul edeceği ve sonuçlarını doğru, hızlı ve güvenli üretebileceği bir veri yapısına dönüştürülme sürecidir. Verinin normalizasyonu, yani *0* ile *1* arasında bir değere dönüştürülmesi yine bu aşamada gerçekleştirilmektedir (Han ve Kamber, 2001).

Veri madenciliğini için önemli süreçlerden biri olan veri hazırlaması aşaması olarak belirtilmiştir. Veri madenciliği uygulamalarında kaynakların yani zaman ve enerjinin %80'i verinin ön işlemden geçirilmesi ve temizlenmesi aşamaları için harcanmaktadır (Piramuthu, 2004). Veri kalitesi, veri madenciliğinde önemli bir konudur. Veri madenciliğinde sonuçlara olan güvenin artması için veri önişleme önemle yapılmalıdır. Aksi durumda yanlış girdi verileri araştırmaya hatalı çıktılar üretmektedir. Veri önişleme, çoğu zaman yarı otomatik olan ve büyük zaman isteyen bir veri madenciliği sürecidir. Veri boyutundaki artış ve buna bağlı olarak çok yüksek sayıda verilerin ön işlemden geçirilmesinin zorunluluğu, *otomatik veri önişleme* için etkin teknikleri önemli hale getirmiştir (Oğuzlar, 2003; Kayaalp, 2007). Şekil 3.3'de veri önişleme süreci gösterilmektedir. Veri önişleme bilgi keşfinde önemli ve uzun süren bir süreçtir. Böylesine zaman alan önemli bir süreç eğer doğru bir şekilde ele alınmazsa bilgi keşfi sürecinin sonuçları çok farklı sonuçlar üretebildiği gibi doğruluktan uzak raporlara, örüntülere dahi sebep olabilmektedir.

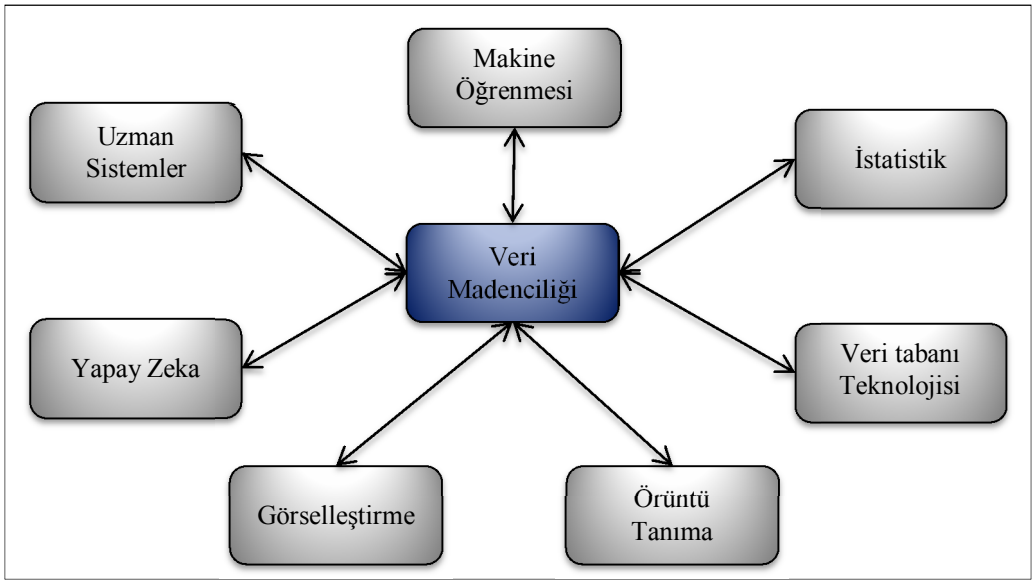



**Şekil 3.3**. Veri kalitesinde veri önişlemenin rolü

4. Dördüncü aşamada bilgi keşfi için kullanılması istenen veri madenciliği tekniği belirlenmektedir. Bu teknikler sınıflama, regresyon, kümeleme, birliktelik analizi olmak üzere gruplandırılmaktadır. Seçilecek olan algoritma *1.aşamada* yapılmak istenen amaca uygun olması

gözetilmektedir. Oluşturulan model ile gelecekte olası veri boyutu üzerindeki değişimler dikkate alınarak bu şekilde çalışabilecek öğrenme yapısına sahip algoritmalar yine bu adımda tercih edilmektedir.

5. Beşinci aşamada uygulama gerçekleştirilmektedir. Şekil 3.4'de veri madenciliği, makine öğrenmesi, uzman sistemler, istatistik, yapay zeka, görselleştirme, veri tabanı teknolojisi ve örüntü tanıma gibi alanlarının bir kesişim noktası olarak görülmekte ve bu yönde gelişmesini sürdürmektedir (Fayyad ve ark., 1994). *Bu yüzden eksik veri problemi bütün bu disiplinleri ilgilendiren aşılması gereken bir problem olarak görülmektedir.* Bilgi keşfinde veri madenciliği her ne kadar tek başına bir aşama olarak görülse de aslında veri madenciliği diğer disiplinler ile işbirliği yaparak bilgi kavramının üretilmesine yardımcı olmaktadır. *Bu nedenle günümüzde veri madenciliği kavramı genel isim olarak hem bilgi keşfinin hem de diğer disiplinlerin yerine kullanılabilmektedir.* Veri madenciliği, zeki yöntem kullanılarak veriden bilinmeyen, ilginç çıkarımlar yapılmasını sağlayan önemli temel bir süreçtir.




**Şekil 3.4.** Birçok disiplinin kesişim noktası olarak veri madenciliği

6. Altıncı aşama uygulama aşamasında kullanılan yaklaşımları optimize edilmesi aşamasıdır. Bu aşamada bilgi keşfinde kullanılan algoritmalar defalarca farklı parametre ve yapılar ile çalıştırılmaktadır. Elde edilen sonuçların yeterince tatmin edici seviyeye gelmesi sağlanmaktadır.

Örnek olarak karar ağacı için en küçük sınıflama yapabilecek ağaç yapısının oluşturulması bu aşamada yapılmaktadır.

7. Bu aşama yorumlama aşaması olarak kullanılmaktadır. Elde edilen örüntüler değerlendirilmekte, *1.aşamada* belirlenen amaçlar doğrultusunda çıkarımlar yapılmaktadır. Gerçekten faydalı ve ilginç örüntüler tespit edilmektedir. Bu çıkarımlar yorumlama olabileceği gibi kurallardan oluşan listeler şeklinde de olabilmektedir. Yapılan değerlendirmeler yazılı dokümanlar veya grafikler haline getirebilmektedir. Veri önişlemede yapılan iyileştirmelerin sonuçlar üzerindeki etkisi yine bu aşamada değerlendirilmekte ve istendiğinde tekrar *3.aşamaya* dönülerek veri yapısı üzerinde değişiklikler yapılabilmektedir. Örneğin bir nitelik eklenmesi ya da azaltılması yapılarak sonuç üzerindeki etkileri incelenebilmektedir.
8. Sekizinci son aşama ise keşfedilmiş bilgilerin kullanılma aşamasıdır. Elde edilen bilgiler başka bir sisteme transfer edilebilmekte veya daha sonra kullanmak üzere de saklanabilmektedir. Kazanılmış bilgilerin uygulanması sırasında ilgili sistem üzerindeki değişiklikler ölçülmektedir. Zaman içinde bilginin kalitesi sistem üzerinden elde edilen geri bildirimlerle değerlendirilmektedir. Bu son sürecin başarısı veri tabanlarında bilgi keşfi sürecinin başarısı olarak görülmektedir.

## 3.4. Veri Madenciliği Teknikleri

Bilgi Keşfinde kullanılan veri madenciliği teknikleri temelde sınıflama, regresyon, kümeleme veya birliktelik analizi yapan yöntemler olarak tanımlanmaktadır (Han ve Kamber, 2001). Sınıflama ve regresyon tahmine dayalı yöntemler olarak birliktelik analizi ve kümeleme ise tanımlamaya dayalı yöntemler olarak sınıflandırılmıştır (Dunham, 2003).

### 3.4.1. Sınıflama ve regresyon

Sınıflama bilinmeyen bir veri nesnesi sınıfının daha önce var olan kayıtlara bakılması sonucu bir karar verilip sınıfa ayrılmasıdır. Yapılan çalışmaya göre tahmin yapıldığı ya da zaman serilerinde kestirimde bulunulduğu da söylenmektedir. Veri

kümesinde bulunan her bir kayıt niteliklerden oluşmaktadır. Bu niteliklerin biri veya bazıları o kayıt verisini sınıflamaktadır. Örneğin veri kaydı bir hastalık ile nitelikler içerdiğinde, sınıf niteliği hastalık durumunun olup olmadığı evet veya hayır şeklinde sınıflayabilmektedir. Bu halde yeni bir hastaya ait hastalık belirtilerini gösteren nitelikler bilinir ise veri madenciliği sınıflama algoritması hastanın hasta olup olmadığını evet ya da hayır şeklinde karar verilebilmektedir.

Regresyon ise sınıflama gibi bilinen veriler eşliğinde matematiksel bir model oluşturularak bir kayıttaki bilinmeyen veri nitelik değerinin bulunmasını sağlamaktadır. Sınıflamadan farklı olarak bulunan sonuç değeri bir sınıf nitelik değeri yerine gerçek değerli matematiksel bir sayı olmaktadır. Literatürde veriye uyan matematiksel fonksiyonun bulunması veya eğri uydurma şeklinde de tanımlamaktadır. Sınıflama amaçlı kullanılan bazı yaklaşımlar aşağıdaki gibidir;

- Yapay sinir ağları
- Destek vektör makineleri
- En yakın k-komşu
- Karar ağaçları
- Saf Bayes teoremi
- Kaba kümeler
- Bulanık kümeler yaklaşımı
- Genetik algoritmalar
- Doğrusal ve çoklu regresyon

### 3.4.2. Kümeleme

Veri kümesinde bulunan kayıtları benzerliklerine göre alt gruplara ayırmayı hedefleyen bilgi keşfi konusuna kümeleme denmektedir. Kümelemede amaç oluşturulan alt grupları olabildiğince birbirinden farklı olacak şekilde oluşturmaktır. Örneğin internet üzerinden uzaktan eğitim ile ders alan öğrencilerin derslere olan ilgi düzeylerine göre aktif öğrenciler, pasif öğrenciler veya başarılı, başarısız öğrenciler olarak alt kümelere ayrılabilmektedir. Ayrılan bu gruplara yönelik farklı ödevler veya çalışmalar verilmekte kümeleme amacına yönelik bilgi keşfi sonucuna ulaşılmaktadır.

- K-ortamalar algoritması
- K-merkezler algoritması
- Pam algoritması
- Clara algoritması
- Clarans algoritması
- Dbscan yoğunluk bazlı kümeleme algoritması

Yukarıdaki yöntemler belli başlı kümeleme metotlarını oluşturmaktadır.

### 3.4.3. Birliktelik analizi

Birliktelik analizi, genel anlamda belli bir destek ve güven eşik değeri yardımıyla birlikte sık görülen işlemlerin, kayıtların veya verilerin tespit edilip analiz edilmesidir. Örneğin bir markette birlikte en çok satılan ürünler veya ürün grupları satış işlemleri içerisinden tespit edilebilmektedir. Böylece bu ürün grupları market içindeki raflarda birbirlerine daha yakın bir şekilde bir araya getirilmektedir. Birliktelik analizi için daha çok aşağıdaki yöntemler kullanılmaktadır;

- Ais algoritması
- Setm algoritması
- Apriori algoritması
- FP-Growth algoritması

## 3.5. Veri Madenciliğinde Karşılaşılan Problemler

Belli sayıda veri içeren bazı veri kümelerinde sorunsuz şekilde çalışan yöntemler, veri sayısının artması ya da azalmasına bağlı olarak farklı sonuçlara neden olabilmektedir. Gürültülü değer, eksik değer, aykırı değer içermeyen verilerle tutarlı ve iyi bir biçimde çalışan bilgi keşfi sistemi, veride meydana gelen bozukluklar karşısında ciddi derecede kötüleşebilmektedir. (Sever ve Oğuz, 2002; Han ve Kamber, 2001). Bilgi keşfi veya veri madenciliğinde kaliteli bilgiye ulaşmanın önünde verilerde olası bulunan aşılması ve kaçınılması gereken problemler şu şekilde özetlenmiştir.

### 3.5.1. Veri tipi

Nitelik değerlerinin farklı tiplerde örneğin sayı, metin, ses, resim, video vb. veri değerleri içermesi sonucu çalışmada kullanılan algoritma açısından farklı seviyelerde problemlere neden olabilmektedir. Algoritmalar geliştirilirken tüm veri tiplerini ele alabilen ve çıktı üretebilen yapıda tasarlanması ve gelecekte olası farklı veri formatları ile çalışacak yapıda olması istenmektedir.

### 3.5.2. Veri boyutu

Veri kümesinde bulanan kayıt adedinin çok fazla veya çok az olması veya nitelik adedinin çok ya da az olması algoritmalar için problem teşkil edebilmektedir. Ayrıca veri kümesindeki kayıt adedinin zaman içinde değişkenlik göstermesi yapılan çalışma sonuçları üzerinde farklılaşmış çıktılara neden olmaktadır. Veri boyutuna bağlı olarak algoritmalarda meydana gelen *Aşırı uyum* sorunu eğitilmiş sistemin var olan verilere göre iyi karar vermesi fakat yeni bilinmeyen kayıtlarda zafiyet göstermesi durumudur. *Az uyum* problemi ise sistemin var olan verilerle eğitiminin yetersiz kaldığı durumlarda ortaya çıkabilmektedir.

### 3.5.3. Aykırı veri

Verinin içindeki nitelik değerinin doğru olmasına rağmen diğer nitelik değerlerine göre aşırı farklılık gösteren değerler bulunmaktadır. Bu aşırılıklar çalışmada kullanılan algoritmanın belli bir yöne doğru meyilli, taraflı ve peşin hüküm veren hale gelmesine neden olabilmektedir. Bu durumda bu türdeki verilere aykırı veriler denilmektedir.

### 3.5.4. Gürültülü, tutarsız veri

Gürültülü veri, veri değerleri içerisinde bazı değerlerin gerçeği yansıtmaması durumunda karşılaşılan yanlış veri değerleridir ve sistemsel olarak bir niteliğe ait tüm değerlerde bulunabilmektedir. Bu tür verilerin veri kümesinin çalışma konusuna göre tespit edilmesinin zor olduğu durumlar olabilmektedir. Eğer gürültü, örneğin insan yaş nitelik değeri *1000* olarak kaydedilmiş gibi durumlarda tespit edildiği zaman tutarsız veri olarak da adlandırılmaktadır. Her gürültülü veri tutarsız değerler

göstermeyebilmekte fakat her tutarsız veri gürültü olarak adlandırılabilmektedir. Sonuç olarak bu türdeki veri değerleri veri kümelerinde istenmeyen yanlış değerli veriler olarak görülmektedir.

### 3.5.5. Eksik veri

Eksik veri, veri kümelerinde değeri olmayan, bilinmeyen değerler olarak adlandırılmaktadır. Eksik değerler bilgi keşfinde sorunlara neden olmaktadır. Bilgi keşfinde kullanılan yöntemlerin birçoğu girdi olarak eksiksiz verilerle çalışabilmek üzere geliştirilmiştir. Eksik değerler, nitelik değerinin veri kümesine kayıt edilememesi gibi durumunda oluşabileceği gibi aykırı, gürültülü, tutarsız değerlerin tespit edilip veri kümesinden silinmesi sonucunda da oluşabilmektedir.

Çizelge 3.1'de eksik değer içeren bir veri kümesinden bir kesit sunulmuştur. Çizelgede bulunan Y1, Y2, Y3, Y4, Y5 ve Y6 için kayıt, satır ifadeleri kullanılmaktadır. X1, X2, X3, X4 ve X5 için nitelik, sütun ifadeleri kullanılmaktadır. Y2, Y5 ve Y6 için eksik değer içermeyen kayıt ya da tam kayıt ifadeleri kullanılmaktadır. Y1, Y3 ve Y4 için ise eksik değer içeren kayıt ya da tam olmayan kayıt ifadeleri kullanılmaktadır.

**Çizelge 3.1.** Eksik veri içeren bir veri kümesi kesiti

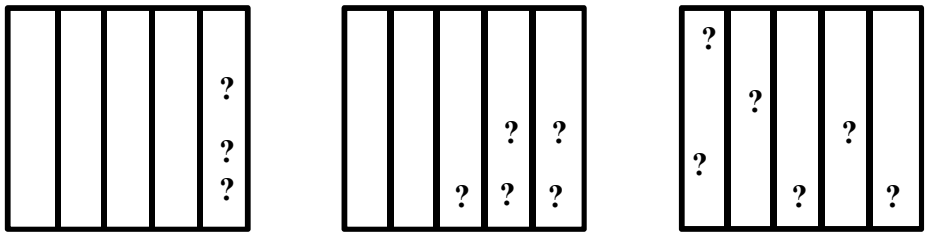
| | X1 | X2 | X3 | X4 | X5 |
|---|---|---|---|---|---|
| Y1 | 0,113524 | 0,084785 | **?** | 0,625473 | 0,06385 |
| Y2 | 0,112537 | 0,138211 | 0,15942 | 0,625473 | 0,068545 |
| Y3 | 0,110563 | **?** | 0,144928 | 0,624212 | 0,083568 |
| Y4 | 0,110563 | 0,170732 | 0,146998 | 0,623581 | **?** |
| Y5 | 0,108588 | 0,129501 | 0,144928 | 0,624212 | 0,076056 |
| Y6 | 0,108588 | 0,082462 | 0,112836 | 0,626103 | 0,015023 |

Eksik veriler veri kümelerinde üç değişik yapıda bulunmaktadırlar (Schafer ve Graham, 2002).

1. Tek değişkende bulunan eksik veri yapısı Şekil 3.5.a'daki gibi tek bir nitelikte meydana gelen eksik veri yapısıdır, elektronik sensör hataları bu tip eksik veri yapısına örnek olarak gösterilmektedir.
2. Monoton eksik veri yapısı Şekil 3.5.b ise bir değerin hesap edilememesi sonucu eksik değere bağlı diğer değişkenlerin de tespit edilememesinden oluşan yapılardır. Bir grup girişin bazı şartlar altında sonuç üretmemesinden

kaynaklanır. Medikal veri kümelerinde bu durumla sıklıkla karşılaşılmaktadır.

3. Rastgele eksiklik veri yapısı ise veri kümesinde rastgele pozisyonlarda eksik değerlerin olması ile oluşmaktadır. Şekil 3.5.c gösterildiği gibi eksik verilerin oluşmasında sistematik bir tutarlılık görülmediği durumlarda oluşabilmektedir.




Tek değişkende bulunan eksik veri yapısı (**a**)

Monoton eksik veri yapısı (**b**)

Rastgele eksik veri yapısı (**c**)

**Şekil 3.5**. Farklı türdeki eksik veri yapıları

Veri kümelerinde bulunan eksik değerleri giderebilmek için bazı basit klasik yöntemler aşağıdaki gibi verilmiştir;

- Eksik değeri *olduğu gibi* bırakma olarak adlandırılan hiçbir şey yapmama yöntemi bazı makine öğrenmesi veya veri madenciliği uygulaması yapan algoritmaların, örneğin C4.5 karar ağacı algoritması gibi olasılık temelli eksik değerleri kendi başına giderebilecek kabiliyetleri olması sonucunda ortaya çıkmıştır. Fakat bu gibi algoritmalar esas görev olarak sınıflama ya da kümelemeye yoğunlaştığı ve tekdüze bir eksik değer giderme yöntemi içerdiği için beklentileri karşılamamaktadır. Bu nedenlerden dolayı eksik değer hesaplamasının ayrı bir süreç olarak ele alınması daha sağlıklı bir yol olduğu görülmüştür.
- Eksik değer içeren kayıt satırı ya da nitelik sütununun silinerek veri kümesinden çıkarılması yani *silme* yöntemi.
- Eksik değer yerine matematiksel veya algoritmalar tarafından etkisiz eleman olarak kabul edilen *sıfır* sayısını koyma yöntemi
- Satırın ya da sütunun aritmetik *ortalamasını* koyma yöntemi

- Satır veya sütunda en *sık geçen* ifadenin eksik değer yerine koyma yöntemi

Bahsedilmiş bu basit klasik yöntemler veri kümesini yanlı ve peşin hüküm veren bir yapıya dönüştürmektedir. Bu nedenle eksik değerleri veri kümesinde var olan diğer verilerden yola çıkarak hesaplayan gelişmiş yöntemler daha popüler yaklaşımlardır. Bu şekildeki hesaplamalar sayesinde eksik değer ile veri kümesi arasındaki ilişki, veri kalitesi anlamında da korunmuş olmaktadır (Han ve kamber, 2001).

# 4. KULLANILAN YARDIMCI METOTLAR

## 4.1. Destek Vektör Regresyonu (DVR) ve Genetik Algoritmalar (GA) ile Eksik Değer Hesaplama

### 4.1.1. Destek vektör regresyonu (DVR)

Regresyon var olan veri kümesinden yola çıkarak bir anlamda veriye eğri uydurularak modellenmesidir. Destek vektör makinaları (DVM) Vapnik ve Lerner tarafından 1963 yılında oluşturulmuş istatiksel öğrenme teorisidir. Teori daha sonra 1996 yılında Scholkopf tarafından büyük veya sonsuz boyutlu bilinen eğitim verisi üzerinde bir veya birden çok hiper düzlem inşa ederek geliştirilmiştir. Böylece DVM başarılı şekilde bilinmeyen veriler için sınıflama yapma amacıyla kullanılmıştır (Smola ve Scholkopf, 2004; Huang ve Kao, 2009). Destek vektör makinalarının regresyon amaçlı değişik bir versiyonu olan destek vektör regresyonu (DVR) olarak adlandırılmış ve 1997 yılında Vapnik, Golowich ve Smola tarafından önerilmiştir (Vapnik ve ark., 1997). Bu önerilmiş model ile eğitim verisinden gerçek değerli bir fonksiyonun tahmin edilmektedir. Destek vektör regresyonu, destek vektör makinelerinin en yaygın biçimde kullanılma şeklini oluşturmaktadır (Basak ve ark., 2007).

Bir grup eğitim verisi $\{(x_1, y_1),....,(x_\ell, y_\ell)\}$ şeklinde verilsin, burada $x_i \subset R^n$ örneklerin giriş uzayını ve giriş verisinin her birine karşılık olarak yani *i*=1'den *l*'ye kadar bir çıkış değeri olarak $y_i \subset R$ kabul edilmektedir. Buradaki *l* eğitim veri adedidir. Regresyon konusunun temel amacı bilinmeyen sayısal değerleri tahmin eden fonksiyonu bulmaktır (Muller ve ark.,2000; Wu ve ark., 2004).

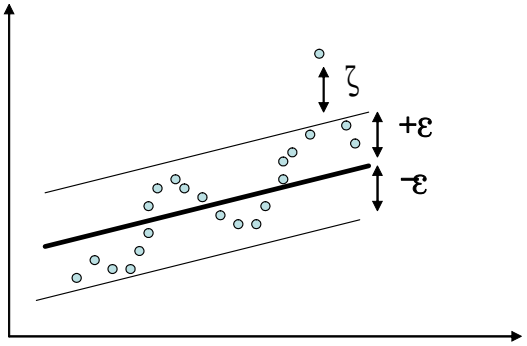
Genel destek vektör regresyonu tahmin fonksiyonu Denklem 4.1'deki gibidir.

$$f(x) = (w \cdot \Phi(x)) + b \qquad \text{Denklem 4.1}$$

Burada $w \subset R^n, b \subset R$ ve $\Phi$ ise $R^n$ boyutundan daha yüksek boyutlu uzaya doğrusal olmayan dönüşümü göstermektedir. *f(x) regresyon fonksiyonunu* bulabilmek için öncelikli amaç denklemde bilinmeyen $w$ katsayısını ve $b$ sabit değerini bulmaktır. Böylelikle regresyon sadece $x$ değerleri ile tespit edilebilir hale gelmektedir.

$$R_{reg}(f) = C\sum_{i=0}^{\ell}\Gamma(f(xi) - yi) + \frac{1}{2}\|w\|^2 \qquad \text{Denklem 4.2}$$

Regresyon fonksiyonu Şekil 4.1'deki gibi eğitim verilerini bir $\varepsilon$ yarıçaplı tüp içerisinde tahmin etmektedir ve Denklem 4.2 halini almaktadır. Denklemdeki $\Gamma(\cdot)$ maliyet fonksiyonudur ve hesaplanan ile gerçek değer arasındaki hatayı ölçmektedir. $C$ sabit bir değerdir ve toplam hataya uygulanan ceza miktarı olarak adlandırılmaktadır. $w$ vektörü ε yarıçaplı tüpten elde edilen veri noktaları Lagrange katsayıları yardımıyla ifade edilirse Denklem 4.3 halini almaktadır.




**Şekil 4.1.** Destek vektör regresyonu bir tüp şeklinde verileri içine alır (Wu ve ark., 2004)

$$w = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\Phi(x_i) \qquad \text{Denklem 4.3}$$

Denklem 4.3, Denklem 4.1'in içinde yerine konulursa, genel denklem şu şekilde elde edilmektedir.

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)(\Phi(x_i)\cdot\Phi(x)) + b$$

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)k(x_i, x) + b \qquad \text{Denklem 4.4}$$

Denklem 4.4'deki nokta çarpımı, çekirdek fonksiyonu olarak bilinen $k(x_i, x)$ ile değiştirilmektedir. Çekirdek fonksiyonları dönüşüme uğramış $\Phi$ düşük uzay boyutlu giriş verisi üzerindeki nokta çarpımını yüksek boyutlu özellik uzayında gerçekleştirmeyi sağlamaktadır. Çekirdek fonksiyonları özellik uzayının iç katmanına karşılık gelen

Mercer teoremi koşullarını sağlamalıdır. Radyal tabanlı çekirdek fonksiyonu Denklem 4.5'de verilmiştir ve en yaygın regresyon çekirdek fonksiyonu olarak kullanılmaktadır.

$$k(x_i, x) = \exp\left\{-\gamma \left| x - x_i \right|^2\right\} \qquad \text{Denklem 4.5}$$

Diğer bazı yaygın kullanılan çekirdek fonksiyonları ile Çizelge 4.1'de gösterilmiştir.

**Çizelge 4.1.** Yaygın kullanılan çekirdek fonksiyonları

| Çekirdekler | Fonksiyonlar |
|---|---|
| Doğrusal | $x \cdot y$ |
| Polinom | $\left[\left(x * x_i\right) + 1\right]^d$ |
| Sigmoid | $\tanh(\gamma x^T * x_i + 1)$ |
| RTF | $\exp\left\{-\gamma \left| x - x_i \right|^2\right\}$ |

Denklem 4.6'da $\varepsilon$-hassas olmayan kayıp fonksiyonu Denklem 4.1'de kullanılan maliyet fonksiyonu olarak kullanılmaktadır.

$$\Gamma(f(x) - y) = \begin{cases} \left| f(x) - y \right| - \varepsilon, & \left| f(x) - y \right| \geq \varepsilon \\ 0 & diger \end{cases} \qquad \text{Denklem 4.6}$$

Kuadratik optimizasyon problemi çözümü olan regresyon fonksiyonu (Denklem 4.2), kısıt fonksiyonları Denklem 4.7 ve $\varepsilon$-hassas olmayan kayıp fonksiyonu (Denklem 4.6) ile en küçük hale getirilerek aşağıdaki formu almaktadır.

$$\frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) - \sum_{i=1}^{\ell}\alpha_i^*(y_i - \varepsilon) - \alpha_i(y_i + \varepsilon)$$

Kısıt fonksiyonu Denklem 4.7 olarak verilmektedir (Yang, 2003).

$$\sum_{i=1}^{\ell} \alpha_i - \alpha_i^* = 0, \quad \alpha_i, \alpha_i^* \in [0, C] \qquad \text{Denklem 4.7}$$

Lagrange katsayıları ($\alpha_i, \alpha_i^*$), kuadratik optimizasyon probleminin çözümünü $y_i$ değerini tahmin ederek bulmaktadır. Denklem 4.7'deki sıfırdan farklı Lagrange katsayıları regresyon çizgisindeki tahminlerde kullanılmaktadır ve destek vektörleri olarak bilinmektedir. $\varepsilon$-tüpünün içindeki tüm noktalarda Lagrange katsayıları sıfır olduğundan regresyon fonksiyonuna katkıda bulunmazlar. Sadece $|f(x) - y| \geq \varepsilon$ olduğu durumda Şekil 4.1'deki gibi, Lagrange katsayıları sıfırdan farklı değer almakta ve destek vektörleri olarak kullanılmaktadır.

Denklem 4.2'de verilen *C* sabit katsayısı hesaplama hatalarındaki ceza miktarını belirlemektedir. Büyük değerli *C* katsayısı hataya karşı daha büyük ceza uygulamaktadır. Bu nedenle regresyon fonksiyonu hatayı en düşük hale getirmek için düşük genelleme yapmaktadır. Düşük *C* katsayısı ise hatalara daha düşük cezalar uygulamaktadır. Böylece hata sınırı en düşük hale gelirken daha yüksek genelleme kabiliyeti sağlanmaktadır. *C* sonsuz değere giderse, destek vektör regresyonu herhangi bir hataya izin vermemekte ve sonuç karmaşık bir yapıda olmaktadır. *C* sıfıra doğru giderken ise sonuç çok sayıda hataya karşı toleranslı olmakta ve yapısı daha az karmaşık olmaktadır.

Buraya kadar Denklem 4.1'deki $w$ değeri Lagrange katsayıları ile bulunmuştur. $b$ sabit değişkeni ise Karush-Kuhn-Tucker (KKT) koşulları ile hesaplanır. Lagrange katsayıları ve sabitler sıfıra eşitlenmektedir (Yang, 2003).

$$\alpha_i(\varepsilon + \zeta_i - y_i + (w, x_i) + b) = 0$$
$$\alpha_i^*(\varepsilon + \zeta_i^* + y_i - (w, x_i) - b) = 0 \qquad \text{Denklem 4.8}$$

ve

$$(C - \alpha_i)\zeta_i = 0$$
$$(C - \alpha_i^*)\zeta_i^* = 0 \qquad \text{Denklem 4.9}$$

Burada $\zeta_i$ ve $\zeta_i^*$ gevşeklik katsayısıdır ve $\varepsilon$-tüpünün dışındaki hataları ölçmek için kullanılmaktadır. $\alpha_i, \alpha_i^* = 0$ ve $\zeta_i^* = 0$, $\alpha_i^* \in (0, C), b$ olduğunda $b$ Denklem 4.10'daki gibi hesaplanmaktadır. Böylece *f(x)* regresyon fonksiyonu bulunmuş olmaktadır (Muller ve ark.,2000; Yang, 2003; Wu ve ark., 2004).

$$b = y_i - (w, x_i) - \varepsilon \quad for\ \alpha_i \in (0, C)$$
$$b = y_i - (w, x_i) + \varepsilon \quad for\ \alpha_i{}^* \in (0, C)$$

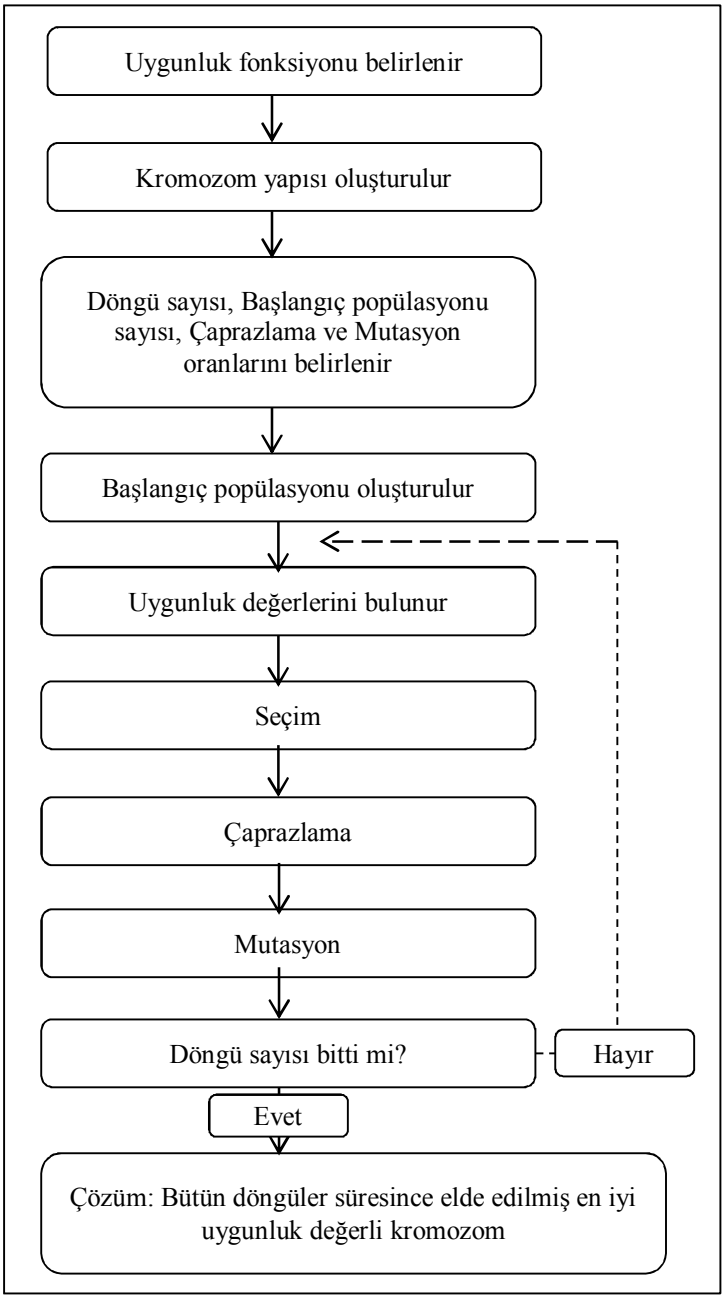
Denklem 4.10

### 4.1.2. Genetik algoritmalar (GA)

1975 yılında, John Holland birey havuzundan en uygun bireyin hayatta kalma kanunundan yararlanan doğal seçim yöntemi olan genetik algoritmaları (GA) tanıtmıştır. GA'ın temel prensibi aile neslinden daha iyi türler seçmek ve rastgele genleri karşılıklı değiştirerek daha iyi nesil üretmektir (Chang, 2009). GA'ın amacı doğal seçim prensiplerini taklit eden rehber eşliğinde çözüm uzayında daha iyi çözümü bulmaktır (Li ve ark., 2009). Birkaç nesil sonra uygun olmayan genler elenerek daha uygun genler üretilmeye başlanmaktadır. GA çözüm uzayını gezmek ve en iyi çözümü bulmak konusunda dengeleme yapmaktadır (Braik ve ark., 2008). Fakat yine de bazı durumlarda çözüm uzayı tam gezilmeden en iyi çözüm bulunduğu durumlar oluşabilmektedir. Genetik algoritmalar genelden özele veya basitten karmaşık olana doğru giden ve geniş alanda kullanılan etkili bir arama tekniğidir (Marwala ve Chakraverty, 2006). Bu doğal yöntem optimizasyon problemleri için kullanılmaktadır. Bu yüzden son yıllarda GA eksik değer hesaplama problemlerinde kullanılmıştır (Hengpraphrom ve ark., 2010, Yang ve Xu, 2011).

#### 4.1.2.1. Genetik algoritmalar süreci

Genetik algoritmalar bir popülasyon havuzundan yeni popülasyon havuzu oluşturma sürecidir (Şekil 4.2). Bu süreç seçim, çaprazlama ve mutasyon temel aşamaları şeklinde devam etmektedir (Marwala ve Chakraverty, 2006). En başta uygunluk fonksiyonu belirlenmekte ve kromozom yapısı oluşturulmaktadır. Daha sonra döngü sayısı, başlangıç popülasyon sayısı, çaprazlama ve mutasyon oranları belirlenmektedir. Başlangıç popülasyon sayısına göre başlangıç popülasyon havuzu oluşturulmakta ve popülasyon havuzundaki bireylerin uygunluk fonksiyonu değerleri hesaplanmaktadır. En sonda ise seçim işlemi gerçekleştirilmekte çaprazlama ve mutasyon işlemleri gerçekleştirilmektedir. Böylece genetik algoritmaların bir döngü süreci tamamlanmış olmaktadır. Genetik algoritmaların sonlanma kriteri olan döngü sayısı kadar algoritma bu şekilde devam etmekte ve sonlanmaktadır.



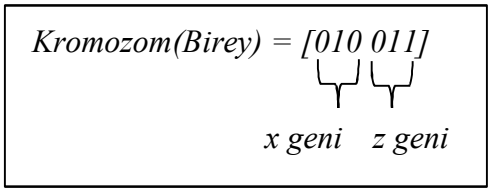

**Şekil 4.2.** Genetik algoritmalar çalışmasının akış diyagramı

- **Başlangıç popülasyon havuzunun oluşturulması**

Genetik algoritmalarda kromozom ya da birey olarak adlandırılan yapılar çözümü istenen problemdeki Denklem 4.11'deki gibi değişkenlerin birleşimden oluşmaktadır. Her bir değişkene gen adı verilmektedir. Örneğin Denklem 4.12'de verilen *y uygunluk fonksiyonunun* maksimum olduğu değerin bulunması istendiğinde *x* ve *z* değişkenlerinin alabileceği değerler aralığı göz önünde bulundurularak, Şekil 4.3'deki gibi ikili sayı sistemi ile oluşturulan bir kromozom yapısı oluşturulmaktadır.

$$Kromozom = [Değişken1\ Değişken2 \dots \dots . Değişken N] \qquad \text{Denklem 4.11}$$

$$y = ax + bz, \qquad 1 < x, z < 10 \qquad \text{Denklem 4.12}$$




**Şekil 4.3.** Genetik algoritmalarda kromozom yapısı

Genetik algoritmaların çalışabilmesi için başlangıç popülasyon sayısının başta belirlenmesi gerekmektedir. Başlangıç popülasyon sayısı çözüm havuzunda kaç adet kromozom yani bireyin olacağını belirleyen bir sayıdır. Bu sayı çok küçük olursa lokal minimum problemi olabilmektedir. Çok büyük olması durumunda ise çözüme ulaşılması çok fazla zaman alabilmektedir (Emel ve Taşkın, 2002). Başlangıç popülasyon havuzu çözüm olabilecek bireyleri yani kromozomları değişken sınırları içinde rastgele bir şekilde üretilmesi sonucu oluşturulmaktadır. Rastgele bu üretim çözüme erişilmesinde bazen gecikmelere neden olabileceği gibi aynı zamanda tüm çözüm uzayını tarama imkanı da sunmaktadır.

- **Uygunluk fonksiyonu**

Genetik algoritmaların, optimize yapılması istenen problemin çözümüne ulaşılıp ulaşılmadığını derecelendiren fonksiyona *uygunluk fonksiyonu* denmektedir. Genetik algoritmalar bir problemin çözümü için en düşük ya da en büyük uygunluk değerini bulmayı amaçlamaktadır. Karşılaşılan problemi en iyi bir şekilde ifade eden uygunluk fonksiyonunun belirlenmesi, yapılacak genetik algoritmalar çalışması bakımından oldukça önemli bir konudur. Her bir genetik algoritmalar döngüsü sonucunda uygunluk fonksiyonu tekrar tekrar hesaplanmakta böylece o anda bulunan bireylerin problemin çözümü için ne derece uygun olduğunun tespiti yapılmaktadır. Uygunluk fonksiyonunu en küçük değere ulaştıran çözümün bulunması istendiğinde en büyük çözüm amaçlı uygunluk fonksiyonu *-1* sayısı ile çarpılarak kullanılmaktadır.

- **Seçim**

Birey havuzu oluşturulduktan sonra problemin çözümü için uygunluk değeri yardımıyla her bir bireye bir skor verilmektedir. Buna göre daha uygun olan bireyler havuzda kalarak güçlü olmayan bireyler havuzun dışında bırakılmaktadır. Seçim için rulet tekerleği, turnuva seçimi, sıra seçimi ve seçkinlik yöntemlerinden biri tercih edilmektedir.

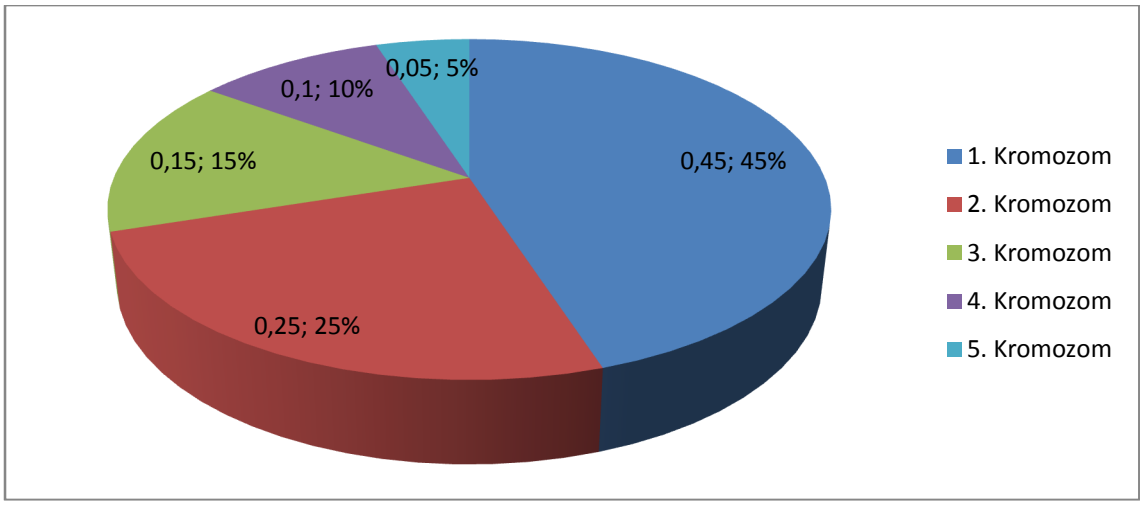
$$Rulet\ tekerleği\ seçim\ değeri = \frac{Kromozom\ uygunluk}{Toplam\ uygunluk} \qquad \text{Denklem 4.13}$$

Rulet tekerleği seçimi değerlerinin oluşturulmasında Denklem 4.13 kullanılarak her bireye ait seçilme oranları belirlenmektedir. Kümülatif toplam haline getirilen seçim değerleri Çizelge 4.2 ve Şekil 4.4'de gösterilmiştir.

**Çizelge 4.2.** Genetik algoritmaların seçim aşamasında rulet ve sıra değerleri

| Kromozom | Uygunluk Değeri | Rulet Tekerleği Seçimi | Rulet Kümülatif | Sıra Seçimi | Sıra Kümülatif |
|---|---|---|---|---|---|
| 1.Kromozom | 45 | 0.45 | 0.45 | 0.33 | 0.33 |
| 2.Kromozom | 25 | 0.25 | 0.70 | 0.26 | 0.59 |
| 3.Kromozom | 15 | 0.15 | 0.85 | 0.20 | 0.80 |
| 4.Kromozom | 10 | 0.10 | 0.95 | 0.13 | 0.93 |
| 5.Kromozom | 5 | 0.05 | 1.00 | 0.06 | 1.00 |

Burada başlangıç popülasyon sayısı kadar rastgele *0* ile *1* arasında üretilen değerler yardımıyla kümülatif değerlere göre yeni nesil bireyler oluşturulmaktadır. Rastgele üretilen sayı ilk kümülatif değerden itibaren sırasıyla karşılaştırılmaktadır. Eğer rastgele sayı kümülatif değerden küçük ise karşılık gelen kromozom seçilmektedir. Eğer havuzda uygunluk değeri diğerlerine göre daha yüksek birey ya da bireyler varsa rulet tekerleğinde çok fazla yer işgal etmektedir. Böylece bu uygun bireyler birkaç döngü sonra havuzda baskın hale gelmektedir. Sıra seçiminde ise bireyler uygunluk fonksiyonu değerlerine göre büyükten küçüğe sıralanmakta ve sıralarına uygun kümülatif değerlere göre seçim işlemi gerçekleştirilmektedir. Kümülatif değerler sıralamaya göre verildiği için rulet tekerleğine göre her seferinde daha fazla çeşitlilik olmakta bu yüzden çözüme geç ulaşılmaktadır.




**Şekil 4.4.** Genetik algoritmalarda rulet tekerleği uygunluk değerleri ve yüzdesi gösterimi

Sıra seçim değerleri Denklem 4.14'e göre hesaplanmaktadır. Denklemdeki $N$ en büyük sıra değerini göstermektedir. Turnuva seçimi ise karşılıklı iki bireyden uygunluk fonksiyonu daha yüksek olan bireyin havuzda kalması diğerinin ise çıkarılmasıyla yapılmaktadır. Seçkinlik yöntemi belli orandaki en iyi birey ya da bireylerin popülasyon havuzuna direk olarak seçilmesiyle gerçekleştirilmektedir.

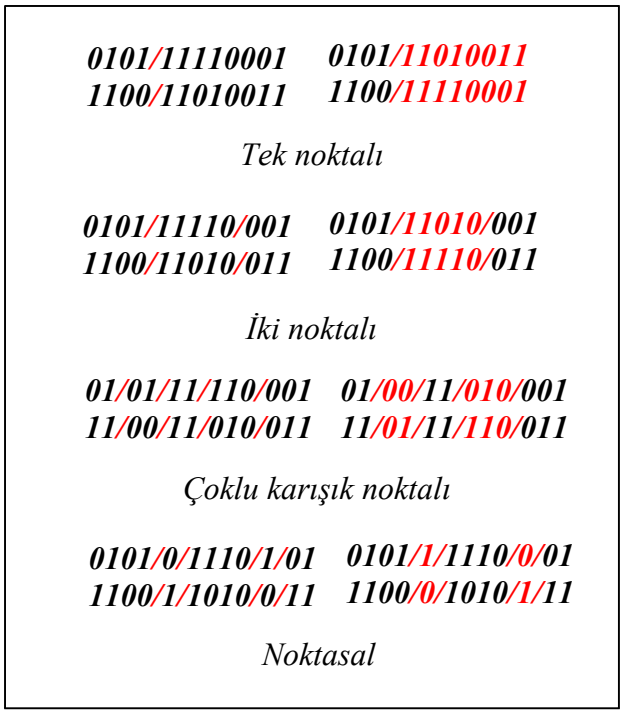
$$Sıra\ seçim\ değeri = \frac{((N+1) - Kromozom\ sıra)}{(N \times (N+1))/2} \qquad \text{Denklem 4.14}$$

Genetik algoritmalar her uygunluk değeri hesaplamasından sonra çalışma süresince gelmiş geçmiş en iyi olan bireyi saklamaktadır. Her döngü sonucunda yeni nesilde daha iyi bir birey elde edilmiş ise bu birey en iyinin yerini alarak genetik algoritmalar sonlanana kadar saklanmaktadır. Bu saklanan en iyi birey genetik algoritmaların elde ettiği optimum çözüm olarak adlandırılmaktadır.

- **Genetik operatörler**

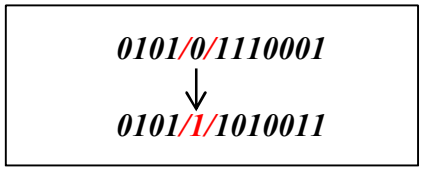
Genetik algoritmaların başarısı var olan geçerli popülasyondan seçilmiş üyelerin karşılıklı olarak bilgilerinin değiştirilmesi ve mutasyona maruz kalmalarına bağlı olarak değişmektedir. Birey bilgilerinin karşılıklı değişmesini sağlayan çaprazlama operatörü ve bireyin mutasyon sonucu değişmesini sağlayan operatör ise mutasyon operatörü olarak adlandırılmaktadır. Bu operatörler belirli yüzdelik ya da bindelik oranlarda bireyler veya bitler üzerinde uygulanmaktadır.

Çaprazlama operatörü anne baba olarak adlandırılan bit dizelerinden bilgileri almakta ve iki yeni birey üretmektedir. Belirli bir bit sırasından itibaren anne ve baba bireylerinin bit dize değerleri karşılıklı olarak yer değiştirilmektedir. Şekil 4.5'da gösterildiği gibi çaprazlama tek noktalı, iki noktalı, çoklu karışık noktalı ve noktasal olarak yapılmaktadır. Tek noktadan yapılan çaprazlama ile ebeveyn bireyleri belirlenmiş bir bit sırasından itibaren karşılıklı olarak diğer bitlerle yer değiştirmektedir. İki noktalı çaprazlama ise iki nokta olarak belirlenmiş başlangıç ve bitiş yer değişim sırası boyunca karşılıklı olarak bitler yer değiştirmektedir. Çoklu karışık noktalı çaprazlama bir veya birden çok bit değerleri karşılıklı olarak yer değiştirmektedir. Noktasal çaprazlama ise tek bir bit değeri diğer ebeveynin aynı pozisyonundaki bit değeri ile yer değiştirmekte yeni iki adet birey oluşmasını sağlamaktadır.




**Şekil 4.5.** Çaprazlama yöntemleri

Şekil 4.6'de mutasyon operatörü gösterilmektedir ve bir birey üzerinde meydana gelen değişiklikleri ifade etmektedir. Mutasyon işlemi çaprazlama operatörüne göre çok daha küçük bir değişime sebep olmakla birlikte güçlü olanın ayakta kalma ve yeni nesil oluşturma prensibi için bir aşama olarak görülmektedir (Mitchell, 1997; Han ve Kamber, 2001).




**Şekil 4.6**. Mutasyon operatörü

#### 4.1.2.2. Genetik algoritmaların bir uygulaması

Matematikte türevi alınabilen bir fonksiyonun en büyük ya da en küçük yapan değeri fonksiyonun türevi alınarak bulunmaktadır. Fakat türevi alınamayan fonksiyonların çözümü için genetik algoritmalar kullanışlı bir hale gelmektedir. Örneğin $y = 24x - 3x^2 + 5,\ 0 < x < 16$ denklemini en büyük yapan $x$ değerini bulmak için fonksiyonun matematiksel türevi $\frac{dy}{dx} \rightarrow 24 - 6x = 0$ olmaktadır. Bu eşitlikten $x = 4$ olarak $y$ fonksiyonunu en büyük yapan değer olarak bulunmuştur.

Aynı fonksiyon genetik algoritmalar ile çözülmek istendiğinde ilk başta uygunluk fonksiyonu tespiti yapılmalıdır. Uygunluk fonksiyonu olarak $y$ fonksiyonunun kendisini kullanılmıştır. Fakat gerçekte bu uygunluk fonksiyonu üzerinde çalışılan probleme uygun belirlenmelidir. Örneğin en uygun banka kredi maliyet hesabı ya da nesnelerin bir bölgeye en uygun şekilde yerleştirilmeleri için kullanılacak uygunluk fonksiyonu yapılacak çalışmanın amacına hizmet edecek şekilde oluşturulmalıdır. Uygunluk fonksiyonu seçiminden sonra başlangıç popülasyon sayısı belirlenmektedir. Çözümü istenen *y fonksiyonu* için *5* adet kromozom tercih edilmiştir. Kromozomdaki *x* değişkeni *1* ile *15* arasında tam sayı değerleri almaktadır. Bu nedenle bir kromozom ikili sayı düzeninde *4* bit uzunluğunda bir sayı ile temsil edilmiştir. Genetik operatörler çaprazlama oranı bu örnek için *0.6* ve mutasyon oranı *0.01* olarak alınmıştır.

Çizelge 4.3'deki gibi başlangıç popülasyonu kromozom değerleri için rastgele *1* ile *15* arasında *5* adet sayı üretilmesiyle oluşturulurmuştur. Daha sonra Çizelge 4.4'deki gibi bu kromozomlara ait uygunluk fonksiyonu değerleri hesaplanmıştır.

Örneğin *1.kromozom* uygunluk değeri $x = 2\ için\ y = (24 \times 2) - (3 \times 2 \times 2) + 5 = 41$ olarak hesaplanmıştır. Diğer değerler de bu şekilde hesaplanarak Çizelge 4.4'de verilmiştir.

**Çizelge 4.3.** Başlangıç popülasyonu

| Kromozom | Onluk Sayı Değerleri | İkili Sayı Değerleri |
|---|---|---|
| 1.Kromozom | 2 | 0010 |
| 2.Kromozom | 11 | 1011 |
| 3.Kromozom | 7 | 0111 |
| 4.Kromozom | 14 | 1110 |
| 5.Kromozom | 5 | 0101 |

Çizelge 4.4'deki gibi uygunluk değerlerinin negatif olması durumunda rulet tekerleği seçilme oranları negatif olamayacağından dolayı bu değerleri pozitife çevirmek için bütün uygunluk değerleri çok büyük bir sabit sayı ile örneğin *1000* ile toplanmıştır.

**Çizelge 4.4.** Uygunluk fonksiyonu değerleri

| Kromozom | Onluk Sayı Değerleri | İkili Sayı Değerleri | Uygunluk Değerleri |
|---|---|---|---|
| 1.Kromozom | 2 | 0010 | 41 |
| 2.Kromozom | 11 | 1011 | -94 |
| 3.Kromozom | 7 | 0111 | 26 |
| 4.Kromozom | 14 | 1110 | -247 |
| 5.Kromozom | 5 | 0101 | 50 |

Ayrıca normalizasyon ile *[0, 1]* aralığına dönüştürülerek kullanılabilir. Bu şekilde elde edilmiş rulet tekerleği kümülatif değerleri Çizelge 4.5'da verilmiştir.

**Çizelge 4.5.** Rulet tekerleği kümülatif değerleri

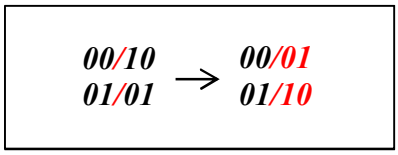
| Kromozom | Sayı Değerleri | İkili Sayı Sistemi | Düzeltilmiş Uygunluk Değerleri | Rulet Tekerleği Seçimi | Rulet Tekerleği Kümülatif |
|---|---|---|---|---|---|
| 1.Kromozom | 2 | 0010 | 1041 | 0.2180 | 0.2180 |
| 2.Kromozom | 11 | 1011 | 906 | 0.1897 | 0.4077 |
| 3.Kromozom | 7 | 0111 | 1026 | 0.2148 | 0.6225 |
| 4.Kromozom | 14 | 1110 | 753 | 0.1577 | 0.7802 |
| 5.Kromozom | 5 | 0101 | 1050 | 0.2198 | 1.00 |
| | | | Toplam: 4776 | Toplam: 1.00 | |

Rulet tekerleği kümülatif değerleri elde edildikten sonra seçim aşamasına geçilmiştir. Seçim aşamasında popülasyon sayısı kadar yani *5* adet rastgele *0* ile *1.00* değerleri arasında ondalık sayı üretilmiştir ve ilk bireyden başlayarak kümülatif değerler ile üretilen sayı karşılaştırılmıştır. Eğer rastgele sayı karşılık gelen bireyin kümülatif değerinden küçük ise o birey seçim havuzuna Çizelge 4.6'daki gibi seçilmiştir. Bu örnekte görüldüğü gibi kümülatif değerleri büyük olan *1., 3. ve 5.* bireylerin seçilme oranları daha yüksek olmaktadır.

**Çizelge 4.6.** Yeni birey havuzu

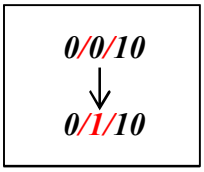
| Kromozom | Onluk Sayı Değerleri | İkili Sayı Değerleri |
|---|---|---|
| 1.Kromozom | 2 | 0010 |
| 2.Kromozom | 11 | 1011 |
| 3.Kromozom | 7 | 0111 |
| 4.Kromozom | 5 | 0101 |
| 5.Kromozom | 5 | 0101 |

Oluşturulan birey havuzunda çaprazlama yapmak için ilk bireyden başlanarak rastgele *0* ile *1.00* arasında ondalık sayı üretilmiştir. Rastgele sayı en başta verilen *0.6* çaprazlama oranından küçük ise o birey rastgele seçilen bir birey ile yine rastgele seçilen bir noktadan itibaren Şekil 4.7'daki gibi yer çaprazlanmıştır. Bu süreç yeni birey havuzundaki tüm bireyler için tekrarlanmıştır.




**Şekil 4.7.** Çaprazlama işlemi

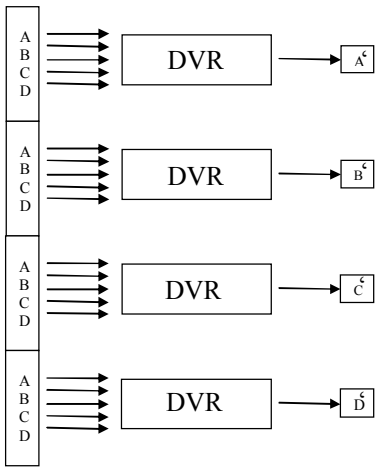
Mutasyon için ilk bireyden başlanarak rastgele *0* ile *1.00* arasında ondalık sayı üretilmiştir. Rastgele sayı belirtilmiş *0.01* mutasyon oranından küçük ise o bireyin herhangi bir bit değeri değiştirilmiştir (Şekil 4.8). Çaprazlama veya mutasyon $x$ değişkenin başlangıç aralık sınırları dışında olması durumunda birey bazında tekrarlanmıştır.




**Şekil 4.8.** Mutasyon işlemi

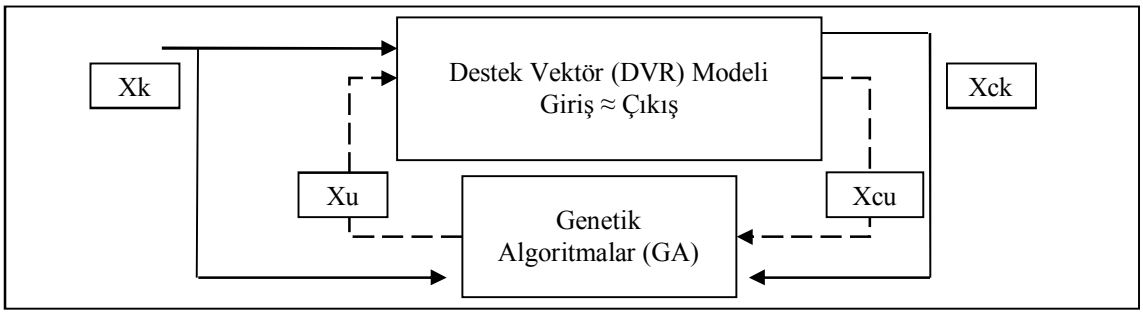
Genetik algoritmalar mutasyon işleminden sonra tekrar döngünün en başına dönmektedir. Daha sonra birey havuzunda bulunan yeni bireylerin uygunluk değerleri yeniden hesaplanarak sonlanana kadar devam ettirilmiştir. Sonlanma kriteri en başta belirlenen döngü sayısıdır. Zaten belirli bir döngü sayısı sonrası bireylerin en uygun çözüm olan $x = 4$ değerine yakınsadığı görülmüştür.

Anlatılanlara göre Destek vektör regresyonu (DVR) ve genetik algoritmalar (GA) ile eksik değer hesaplamasının yapılabilmesi için önce veri kümesinden eksik değer içermeyen tam kayıtlar seçilmektedir. Bu tam kayıtların Şekil 4.9'de görüldüğü gibi her seferinde bir tanesi çıkış diğer tüm nitelikler giriş olacak şekilde kullanılarak veri kümesindeki nitelik sayısı kadar destek vektör regresyonu yapısı oluşturulmaktadır. Böylece tüm regresyon yapıları eğitilerek genel regresyon modeli oluşturulmaktadır. Bu sayede veri kümesinin her bir tam kaydının çıkışta yaklaşık olarak benzer bir şekilde geri elde edilmesinin sağlandığı bir model oluşturulmuştur (Feng ve ark., 2005; Marwala ve Chakraverty, 2006; Nelwamondo ve ark., 2007).




**Şekil 4.9.** Destek Vektör Regresyonu (DVR) modeli

Şekil 4.10'de bahsedilmiş bu destek vektör regresyon modeli ve genetik algoritmalar optimizasyonu kullanılarak eksik değeri tamamlamak üzere oluşturulmuş yapı görülmektedir (Marwala, 2009).




**Şekil 4.10.** Destek Vektör Regresyonu (DVR) ve Genetik Algoritmalar (GA) ile eksik değer hesaplama

Buna göre destek vektör regresyon modelinin girişine (Denklem 4.15) tam olmayan bir kayıt verilmektedir. Bu kayıtta *Xk* bilinen nitelikler, *Xu* ise eksik nitelik değerleri olarak kabul edilmektedir. Destek vektör çıkışı *f fonksiyonu* (Denklem 4.16) olarak adlandırılmakta ve değerleri başta bilinen *Xck* ve sonradan hesaplanmış *Xcu* vektörlerinden oluşmaktadır. Eğitilmiş DVR modelinin girişinin çıkışına yaklaşık olarak eşit olması istenmekte ve aradaki *fark hata* (Denklem 4.17) olarak adlandırılmaktadır. Genetik algoritmaların amacı *giriş* ile *çıkış* arasındaki bu *hatayı* negatif olmayan uygunluk fonksiyonu (Denklem 4.18) yardımı ile *minimum hata* yapan eksik değerleri hesaplamaktır (Marwala ve Chakraverty, 2006; Nelwamondo ve ark., 2007).

$$DVR\ giriş = \binom{Xk}{Xu} \quad \text{Denklem 4.15}$$

$$DVR\ çıkış = \binom{Xck}{Xcu} = f\binom{Xk}{Xu} \quad \text{Denklem 4.16}$$

$$hata = \binom{Xk}{Xu} - f\binom{Xk}{Xu} \quad \text{Denklem 4.17}$$

$$GA\ uygunluk\ fonksiyonu\ = (\binom{Xk}{Xu} - f\binom{Xk}{Xu})^2 \quad \text{Denklem 4.18}$$

### 4.2. Bulanık c-ortalamalar (BCO) ile Eksik Değer Hesaplama

Kümeleme yapmanın genel amacı veri kümesinde var olan verilmiş bir takım nesneleri, nesnelerin benzerliğine göre alt gruplara bölme ve bu alt kümeler arasındaki benzerliği en aza indirmektir. Bulanık c-ortalamalar (BCO) yöntemi var olan bir nesneyi bir veya birden çok kümeye üye yapmaktadır. Bu yöntem 1973 yılında Dunn tarafından ortaya atılmış (Dunn, 1973; Das ve Sil, 2010), 1981 yılında Bezdek (Wang, 1983) tarafından iyileştirilmiş ve sık sık örüntü tanıma alanında kullanılmaya devam edilmektedir. Yöntemin başlıca hedefi amaç fonksiyonunu (Denklem 4.19) minimum hale getirmeye çalışmaktır.

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} {u_{ij}}^m \left(x_i - c_j\right)^2 \quad \text{Denklem 4.19}$$

$$2 \le c \le N \quad \text{Denklem 4.20}$$

$$1 \le m \le \infty \quad \text{Denklem 4.21}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\left\| x_i - c_j \right\|}{\left\| x_{i-} c_k \right\|} \right)^{\frac{2}{m-1}}} \qquad \text{Denklem 4.22}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \qquad \text{Denklem 4.23}$$

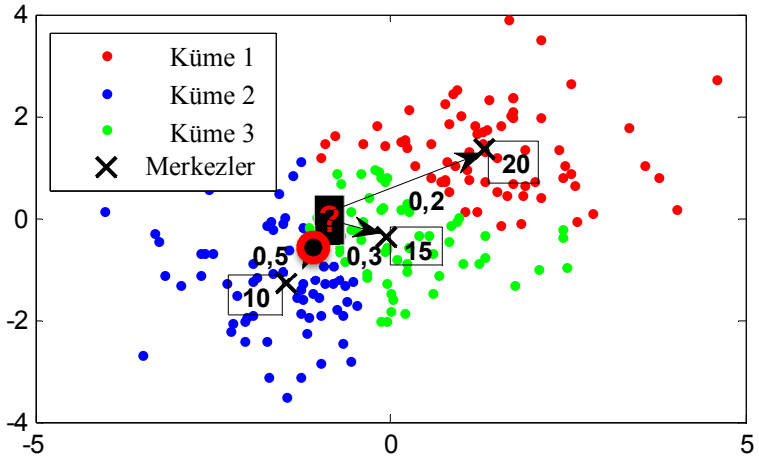
Denklem 4.20'deki *c* parametresi *küme merkez sayısını* göstermektedir ve *2* ile veri kümesi kayıt adedi (*N*) arasında bir tam sayı değeri almaktadır. Denklem 4.21'de ise *m* parametresi *ağırlık faktörü* olarak adlandırılmakta ve *1* ile ∞ arasında ondalık sayı değeri almaktadır. Ağırlık faktörü parametresi kümeleme sürecinde bulanıklık miktarını kontrol etmektedir (Wang, 1983). Teorik olarak *c* ve *m* parametresinin optimum seçim değeri olmamaktadır (Li ve ark., 2004; Ghosh ve Mujumdar, 2006). Bu parametreler veri kümesinin karakteristik özelliğine ve veri kümesinde bulunan niteliklerin birbirleriyle olan ilişki düzeylerine bağlı olarak değişmektedir. Bu tez çalışmasında önerilen bulanık c-ortalamalar ile eksik veri hesaplaması yapılırken o anda kullanılan veri kümesi yapısına en uygun, optimum *c* ve *m* parametre değerlerinin tespit edilmesi sağlanmıştır.

Bulanık kümelemede her veri nesnesi bir üyelik fonksiyonu (Denklem 4.22) değerine sahiptir. Bu değer nesnenin hangi küme merkezine (Denklem 4.23) ne derece ait olduğunu tanımlamaktadır. Üyelik fonksiyon değerlerini ve küme merkezlerini güncelleme sürecinde sadece veri kümesinde eksik değer içermeyen tam kayıtlar hesaba alınmaktadır. Bu süreçte temel klasik k-ortalamalar kümeleme yöntemine nazaran her bir veri nesnesi tek bir kümeye ait olmamaktadır. Veri nesnesi bütün küme merkezlerine üyelik fonksiyon değerleri nispetince üye olmaktadır. Tam olmayan veri kaydının eksik değeri bu üyelik fonksiyon değerleri ve küme merkezi olarak kabul edilen değerler yardımıyla hesaplanmaktadır. Deneysel çalışmalar bulanık eksik değer tamamlama algoritmasının temel klasik kümeleme algoritmasına göre daha iyi performans sonuçları ortaya koyduğunu göstermiştir (Li ve ark., 2004).

### 4.2.1. Bulanık c-ortalamalar uygulaması

Şekil 4.11'da eksik bir değerin bulanık c-ortalamalar ile hesaplanması bir örnek üzerinden açıklanmıştır. Örneğin *soru işareti* (*?),* veri kümesindeki bir eksik değer olarak kabul edilmektedir.

Eksik değer içermeyen veri kümesinin tam elemanları *3* küme merkezine ayrılmıştır. Bulanık c-ortalamalar *m*, *ağırlık faktörü* parametre değeri *2* olarak tercih edilmiştir. Bu durumda *eksik değerin (?),* küme merkezlerine olan üyelik fonksiyonu değerleri sırasıyla $U_{?-1}$*=0.2,* $U_{?-2}$*=0.5 ve* $U_{?-3}$*=0.3* olarak hesaplanmıştır.




**Şekil 4.11.** Bulanık c-ortalamalar (Bco) ile eksik değer hesaplaması

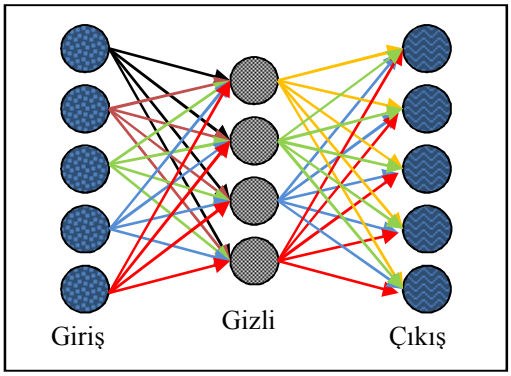
Küme merkez değerleri Denklem 4.23 ile sırasıyla $c_1$*=20,* $c_2$*=10 ve* $c_3$*=15* olarak hesaplanmıştır. Böylece eksik olan *soru işareti (?)* değeri $c_1$**U*$_{?-1}$*+*$c_2$**U*$_{?-2}$*+*$c_3$**U*$_{?-3}$*'*den yani *0.2*20+0.5*10+0.3*15* işleminin hesabından sonra *13.5* olarak bulanık c-ortalamalar eksik değer hesaplama yöntemi ile bulunmuştur.

## 4.3 Yapay Sinir Ağları (YSA) ve Genetik Algoritmalar (GA) ile Eksik Değer Hesaplama

### 4.3.1. Yapay sinir ağları (YSA)

Yapay sinir ağları (YSA) insan sinir sistemi gibi davranır ve insan sinir sistemi gibi öğrenme işlemlerini gerçekleştirir. YSA giriş, gizli ve çıkış katmanlarından oluşmaktadır. Sinir hücreleri bir araya gelerek bir katmanı oluşturmakta ve her bir sinir hücresi sonraki katmandaki sinir hücrelerine ağırlık değerleri denilen parametre değerleri ile bağlanmaktadır. Ağırlık değerlerinin optimizasyonundan sonra, eğitilmiş bir yapay sinir ağı önceden öğrenmesi için verilen bilgi kategorisinde bir uzman olarak kabul edilmektedir. Bu uzman olan sistem yeni bilinmeyen durumlar için tahminlerde bulunması amacıyla kullanılmaktadır. Yapay sinir ağları örüntü tanıma, sinyal işleme, zaman serileri tanıma, doğrusal olmayan kontrol, problem tanımlama ve benzeri

alanlarda geniş bir şekilde kullanılmaktadır (Gao ve ark., 2009). Yapay sinir ağları, sınıflama değerlerinin bilindiği bir eğitim veri kümesiyle yapılan öğrenme yöntemidir. Bu yöntem değişik sınıflama görevlerinde fazlaca başarı hikayesi olan pratik bir yaklaşım olarak kendisini kanıtlamıştır (Chen ve ark., 2009). YSA, giriş katmanındaki sinir hücresi sayısı, gizli katman sayısı ve çıkış katmanındaki sinir hücresi sayısındaki değişiklikle bağlı olarak ağ yapısı itibariyle farklılıklar göstermektedir. Şekil 4.12'de bir yapay sinir ağ yapısı gösterilmiştir.




**Şekil 4.12.** Yapay Sinir Ağları (YSA) yapısı

Yapay sinir ağlarında öğrenme işlevi geri yayılım öğrenme algoritması ile yapılmaktadır. Bu öğrenme algoritması ağ çıkışının gerçekte olması gereken sınıf değeri ile bulunan değer arasındaki toplam farkı azaltmaya çalışmaktadır. Bu sayede katmanları birbirine bağlayan ağırlık parametre değerleri güncellenmektedir. En uygun yapay sinir ağ yapısının tam olarak oluşturulmasının kolay olmaması ve öğrenilen kuralların insanlar tarafından kolay yorumlanamaması ayrıca ağ yapısının eğitiminin uzun süre alabilmesi nedenleriyle yapay sinir ağları bazen eleştirilmektedir. Fakat gürültülü veriler üzerinde yüksek toleransa sahip, iyi bir şekilde çalışması ve birçok uygulamada başarılı sınıflama yapabilmesi yapay sinir ağlarının kullanılmasını avantajlı bir hale getirmektedir.

#### 4.3.1.1. Geri yayılım algoritması

Geri yayılım algoritması ileri beslemeli çok katmanlı yapay sinir ağlarında öğrenmeyi gerçekleştirmektedir. İleri beslemeli yapay sinir ağı ağırlık değerleri vasıtasıyla giriş katmanını orta katman olarak da adlandırılan gizli katman veya gizli katmanlara oradan da en son çıkış katmanına aktaran ağ modelidir. Geri besleme ise adından da anlaşılabileceği üzere çıkış katmanındaki gerçek değer ile tahmin edilen

değer arasındaki hata oranını temel alarak ağ yapısı üzerinde geriye doğru düzeltmeler yapılmasını sağlamaktadır. Geri yayılım algoritmasının çalışma adımları şu şekilde ifade edilmektedir (Han ve Kamber, 2001).

Ağırlık ve bias değerlerine başlangıç değerlerinin verilmesi aşamasında yapay sinir ağında bulunan tüm ağırlık ve bias değerlerine rastgele *-1.00* ile *+1.00* veya *-0.5* ile *+0.5* arasında küçük değerli ondalık sayılar verilmektedir. Bundan sonra giriş değerleri vasıtasıyla gizli katmandaki nöron değerleri hesaplanmaktadır. Gizli nöron değeri hesaplaması ilgili nörona gelen tüm giriş nöron değerleri ile ağırlık değerlerinin çarpımlarının toplamı şeklinde Denklem 4.24 ile hesaplanmaktadır. Burada $I_j$ gizli nöron değerini, $w_{ij}$ ise o nörona $O_i$ girişinden gelen ağırlık değerini temsil etmektedir. Denklemde görülen $\theta$, bias olarak adlandırılarak gizli nöron değerinin sıfır olmasını engellemek ve Denklem 4.25'deki aktivasyon fonksiyonun sonuç üretebilmesi amacıyla eşik değeri olarak kullanılmaktadır. Aktivasyon fonksiyonu büyük ölçekli verileri *0* ile *1* arasındaki bir değer aralığına dönüştürmekte ve öğrenme işlemini kolaylaştırmaktadır.

$$I_j = \sum (w_{ij} \times O_i) + \theta \qquad \text{Denklem 4.24}$$

$$O_j = \frac{1}{(1 + e^{-I_j})} \qquad \text{Denklem 4.25}$$

Bu şekilde çıkış katmanı nöron değerleri hesaplandıktan sonra geri yayılım yapılarak hata değerleri hesaplanmaktadır. *Hata* eğitim verisindeki gerçek çıkış değeri ile hesaplanmış değer arasındaki fark temel alınarak Denklem 4.26 ile hesaplanmaktadır.

$$Hata_j = O_j(1 - O_j)(H_j - O_j) \qquad \text{Denklem 4.26}$$

Buradaki $O_j$ hesaplanmış çıkış değeri, $H_j$ ise gerçek, hedef çıkış değerini temsil etmektedir. Bu şekilde bütün çıkış nöronları ve çıkış nöronlarındaki *hata* değerleri hesaplanmaktadır.

$$Hata_j = O_j(1 - O_j) \sum_k (Hata_k \times w_{jk}) \qquad \text{Denklem 4.27}$$

Bundan sonra gizli katmandaki her bir nöron için *hata* değerleri hesaplaması yapılmaktadır. Bunun için Denklem 4.27 kullanılır. $w_{jk}$, gizli katmanda bulunan nörondan çıkış katmana giden ağırlık değerini, $Hata_k$ ise o ağırlık değerinin ulaştığı çıkış katmanındaki nöron değerinin hatasıdır. En son aşamada ise hata oranlarına karşılık *ağırlık* ve *bias* değerleri güncellenmektedir. Yeni ağırlık değerleri Denklem 4.28'deki gibi yeni bias değerleri ise Denklem 4.29'daki gibi hesaplanmaktadır.

$$w_{ij} = w_{ij} + (\alpha \times Hata_j \times O_i) \quad \text{Denklem 4.28}$$

$$\theta_j = \theta_j + (\alpha \times Hata_j) \quad \text{Denklem 4.29}$$

$\alpha$ öğrenme katsayısı olarak bilinmektedir. *0* ile *1* arasında ondalık sayı değeri almakta ve bu katsayı karar verme uzayında yerel minimuma sıkışıp kalınmasını önlemektedir. Eğer bu katsayı düşük tutulursa öğrenme işlemi yavaş adımlarla ilerlemekte, yüksek tutulur ise çözümün etrafında sarkaç gibi yakınlaşıp uzaklaşan dalgalanmalara neden olmaktadır.

Geri yayılım algoritması eğitim veri kümesindeki her bir kayıttan sonra *ağırlık* ve *bias* değerlerinde güncelleme yapabildiği gibi algoritmanın çalışma hızı açısından tüm eğitim veri kümesi kayıtlarının en sonunda kümülatif toplam güncelleme de yapabilmektedir. Fakat daha iyi sınıflama başarısı için her kayıt sonrasında güncelleme tercih edilmektedir. Geri yayılım algoritması belli bir döngü sayısı veya ağırlık değerlerindeki değişim belli bir eşik seviyesini geçmeyinceye kadar devam etmekte ve sonlanmaktadır. Farklı tipteki aktivasyon fonksiyonu, öğrenme katsayısı veya hata fonksiyonu seçimiyle değişik şekillerde geri yayılım algoritması uygulanmaktadır.

#### 4.3.1.2. Yapay sinir ağlarının bir uygulaması

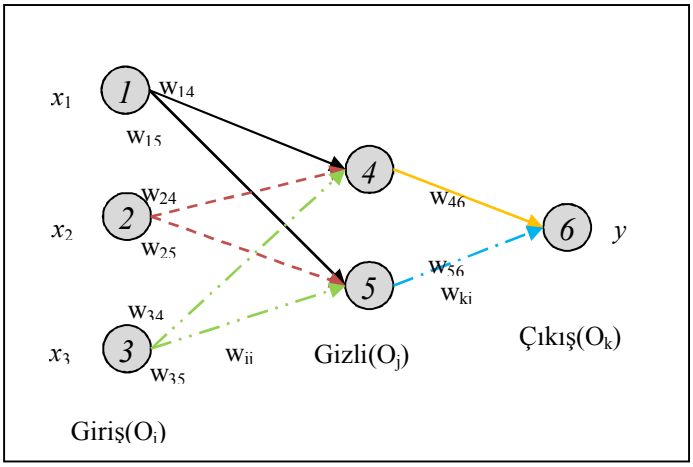
Çizelge 4.7'de verilen veri kümesinin yapay sinir ağları ile öğrenilmesi istenmiştir. Veri kümesinde $x_1$, $x_2$, $x_3$ olmak üzere *3* giriş ve sınıflama yapan *y* çıkış nitelik değeri bulunmaktadır (Han ve Kamber, 2001).

Yapay sinir ağının yapısı veri kümesinin yapısına uygun olarak Şekil 4.13'daki gibi oluşturulmuştur. *1, 2, 3* giriş, *4, 5* gizli, *6* ise çıkış katmanında bulunan nöronları temsil etmektedir.

**Çizelge 4.7.** Yapay sinir ağları eğitim örneği veri kümesi

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |

*1, 2, 3* giriş nöronlarına verilen veri kümesinin *1, 0, 1* eğitim örneğine karşılık *6* numaralı çıkış katmanında bulunan nöronda veri kümesinin *y* nitelik değeri olarak *1* elde edilmek istenmiştir.




**Şekil 4.13.** Yapay sinir ağlarının bir örneği

Başlangıç yapay sinir ağının değerleri Çizelge 4.8'da gösterilmiştir. Başlangıç *ağırlık* değerleri ve *bias* değerleri küçük sayılarla rastgele oluşturulmuştur. Öğrenme katsayısı olarak *0.9* alınmıştır.

**Çizelge 4.8.** Başlangıç giriş, ağırlık ve bias değerleri

| | | | |
|---|---|---|---|
| $x_1$ | 1 | $w_{34}$ | -0.4 |
| $x_2$ | 0 | $w_{35}$ | 0.3 |
| $x_3$ | 1 | $w_{46}$ | -0.2 |
| $w_{14}$ | 0.3 | $w_{56}$ | -0.1 |
| $w_{15}$ | -0.2 | $\theta_4$ | -0.5 |
| $w_{24}$ | 0.5 | $\theta_5$ | 0.3 |
| $w_{25}$ | 0.2 | $\theta_6$ | 0.2 |

Bu değerlere karşılık *4, 5* gizli katman nöronlarının ve aktivasyon fonksiyonu çıkış değerleri Çizelge 4.9'daki gibi hesaplanmıştır. Bu değerler tespit edildikten sonra çıkış katmanında bulunan *6* numaralı nöron, *4, 5* nöron değerleri yardımıyla elde edilmiştir.

**Çizelge 4.9.** *4, 5, 6* numaralı nöron değerlerinin hesaplanması

| Nöron | Giriş | Aktivasyon Çıkış |
|---|---|---|
| ***4*** | 0.3+0-0.4-0.3 = -0.4 | $\frac{1}{(1+e^{0.4})} = 0.401$ |
| ***5*** | -0.2+0+0.3+0.3 = 0.4 | $\frac{1}{(1+e^{-0.4})} = 0.599$ |
| ***6*** | (-0.2)(0.401)-(0.1)(0.599)+0.2 = 0.06 | $\frac{1}{(1+e^{-0.06})} = 0.515$ |

Böylece yapay sinir ağının ileri besleme aşaması yapılmış olmaktadır. Bundan sonra geriye doğru hataları yayma aşaması olan geri yayılım algoritması uygulanmaktadır. Gerçekte Çizelge 4.7'de ki veri kümesine göre yapay sinir ağının *1, 2, 3* numaralı nöron girişlerine *1, 0, 1* değerleri verilince *6* numaralı çıkış katmanındaki nörondan *1* değeri elde edilmeliydi fakat bu değer Çizelge 4.9'da görüldüğü gibi *0.515* olarak hesaplanmıştır. Buna göre hatalar Çizelge 4.10'deki gibi hesaplanmıştır.

**Çizelge 4.10.** *6, 5, 4* numaralı nöron hata değerlerinin hesaplanması

| Nöron | Hata |
|---|---|
| ***6*** | (0.515)(1-0.515)(1-0.515) = 0.1211 |
| ***5*** | (0.599)(1-0.599)(0.1211)(-0.1) = -0.0029 |
| ***4*** | (0.401)(1-0.401)(0.1211)(-0.2) = -0.0058 |

En son aşamada ise hata değerlerine karşılık *ağırlık* ve *bias* değerleri güncellenmiştir. Buna göre yeni *ağırlık* ve *bias* değerleri Çizelge 4.11'de verilmiştir.

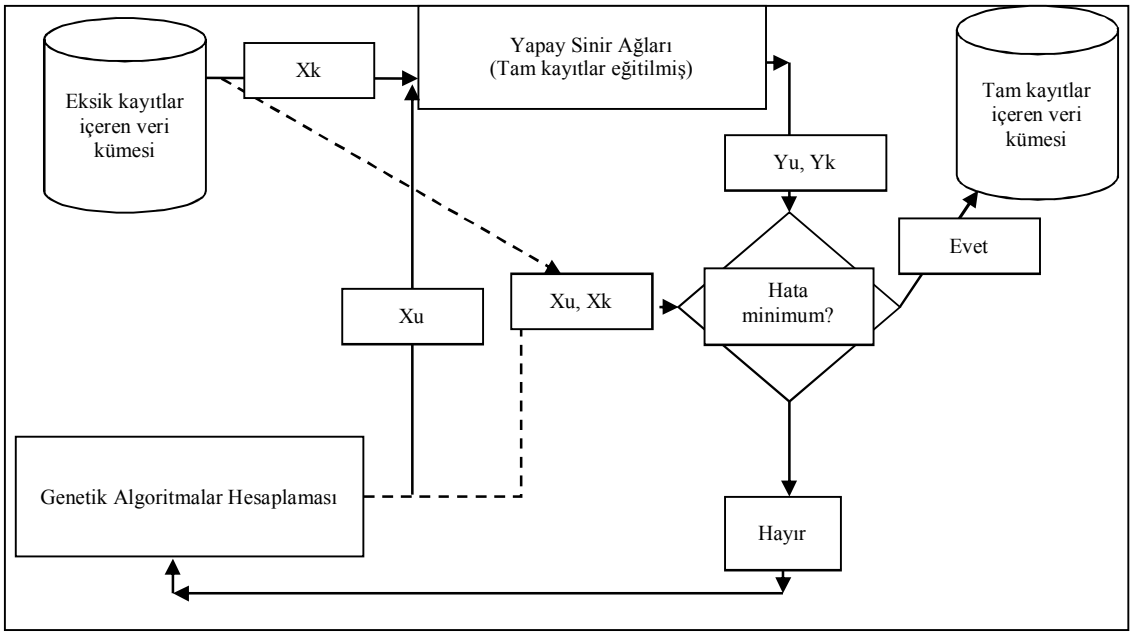
**Çizelge 4.11.** Yeni ağırlık ve bias değerleri

| | |
|---|---|
| $w_{14}$ | 0.3+(0.9)(-0.0058)(1) = 0.2948 |
| $w_{15}$ | -0.2+(0.9)(-0.0029)(1) = -0.2026 |
| $w_{24}$ | 0.5+(0.9)(-0.0058)(0) = 0.5 |
| $w_{25}$ | 0.2+(0.9)(-0.0029)(0) = 0.2 |
| $w_{34}$ | -0.4+(0.9)(-0.0058)(1) = -0.4052 |
| $w_{35}$ | 0.3+(0.9)(-0.0029)(1) = 0.2974 |
| $w_{46}$ | -0.2+(0.9)(0.1211)(0.401) = -0.1563 |
| $w_{56}$ | -0.3+(0.9)(0.1211)(0.599) = -0.2347 |
| $\theta_4$ | -0.3+(0.9)(-0.0058) = -0.3052 |
| $\theta_5$ | 0.3+(0.9)(-0.0029) = 0.2974 |
| $\theta_6$ | 0.2+(0.9)(0.1211) = 0.3090 |

Çizelge 4.11'deki güncellemeler yapıldıktan sonra veri kümesinde bulunan bir sonraki eğitim veri kaydı yapay sinir ağı girişine verilmiş ve yeniden *ağırlık* ve *bias* değerleri güncellenmiştir. Bu şekilde yapay sinir ağı eğitim sonlanma kriterine kadar devam edilerek öğrenme işlemi gerçekleştirilmiştir.

Abdella ve Marwala, Şekil 4.14'daki modeli kullanarak eksik verileri hesaplamıştır. Yapay sinir ağ yapısı veri kümesinin özel olarak modellemesinde yani giriş katman değeri çıkış katman değeriyle neredeyse aynı olacak şekilde oluşturulmuştur (Thompson ve ark., 2003; Betechuoh ve ark., 2006).

Giriş ile çıkış arasındaki fark genetik algoritmaların uygunluk fonksiyonu olarak eksik veri hesaplamasında kullanılmaktadır (Mohamed ve Marwala, 2005; Qiao ve ark., 2005; Abdella ve Marwala, 2005; Nelwamondo ve ark., 2007; Nelwamondo ve Marwala, 2007; Blend ve Marwala, 2008; Nelwamondo ve ark., 2009; Hlalele ve ark., 2009).




**Şekil 4.14.** Yapay Sinir Ağları (YSA) ve Genetik Algoritmalar (GA) ile eksik değer hesaplama

Yapay sinir ağ yapısı *giriş(X)*, *çıkış(Y)* ve *ağırlık(w)* değerlerinden oluşmaktadır. Matematiksel olarak yapay sinir ağı ifade edilirse Denklem 4.30 şeklinde yazılmaktadır.

$$Y = f(X, w) \qquad \text{Denklem 4.30}$$

Eğer ağ yapısı girişte verilen vektör değerlerini çıkışta tahmin edecek üzere eğitilirse *giriş(X)* ile *çıkış(Y)* çıkış neredeyse birbirine eşit olmaktadır (Denklem 4.31).

$$X \approx Y \qquad \text{Denklem 4.31}$$

Gerçekte giriş ile çıkış vektörü her zaman mükemmel bir şekilde birbirine eşit olamazlar. Bu yüzden giriş ve çıkış vektörünün farkını ifade eden bir *hata (e)* fonksiyonu Denklem 4.32'deki gibi gösterilmiştir.

$$e = X - Y \qquad \text{Denklem 4.32}$$

Burada $Y$ yerine Denklem 4.30 eşitliği yazılırsa hata fonksiyonu Denklem 4.33 halini almaktadır.

$$e = X - f(X, w) \qquad \text{Denklem 4.33}$$

Hatanın sıfır olmayan minimum değeri tercih edildiğinden dolayı, hata fonksiyonu karesi alınarak ifade edilmektedir (Denklem 4.34).

$$e = (X - f(X, w))^2 \qquad \text{Denklem 4.34}$$

Eksik değer, giriş vektöründe ($X$) bazı değerlerin olmadığı durumlarda oluşmaktadır. Giriş ($X$) vektöründe bilinen değerler için ($Xk$), bilinmeyen değerler ise ($Xu$) olarak ayrı ayrı yazılabilmektedir.

$$e = (\binom{Xk}{Xu} - f(\binom{Xk}{Xu}, w))^2 \qquad \text{Denklem 4.35}$$

Böylece genetik algoritmaların (GA) uygunluk fonksiyonu (Denklem 4.35) elde edilmiştir. GA uygunluk fonksiyonunu minimize edecek değerler eksik değerler olarak kabul edilmekte ve böylece eksik değerlerin hesaplaması yapılmış olmaktadır.

### 4.4. En Yakın k-Komşular (EYK) ile Eksik Değer Hesaplama

Eksik değerler benzerlik ölçütü vasıtasıyla tespit edilen en benzer kayıtların eksik değerlere karşılık gelen nitelik değerlerinin bir araya getirilmesiyle hesaplanmaktadır. Benzer kayıtlarda en sık geçen ifade, aritmetik ortalama (Batista ve Monard, 2003) veya ağırlıklı aritmetik ortalama (Troyanskaya ve ark., 2001) alınarak eksik değer hesaplaması yapılmaktadır. Bu hesabın yapılabilmesi için benzerlik ölçütü

ve en yakın *k* komşu adedinin belirlenmesi gerekmektedir (Troyanskaya ve ark., 2001; Bras ve Menezes, 2007). Öklid uzaklık ölçütü (Denklem 4.36) yardımıyla en yakın *k* adet eksik değer içermeyen benzer komşu kayıtlar *($x_k$)* ile eksik değer içeren kayıt *($y_i$)* arasındaki benzerlik hesaplanmaktadır.

$$d_{ik} = \sqrt{\sum_{j=1}^{n} (x_{kj} - y_{ij})^2} \qquad \text{Denklem 4.36}$$

Denkleme göre *j* ifadesi eksik değer içeren kaydın tam olan sütunlarını ifade etmektedir. Yani benzerlik ölçütü, eksik olan kaydın sadece tam nitelikleri ile o niteliklere karşılık gelen diğer komşuların nitelikler değerleri kullanılarak hesaplanmaktadır. Benzerlik ölçütü hesaplandıktan sonra *k* adet komşunun ağırlıklı olarak benzerlik oranları Denklem 4.37'deki gibi bulunur.

$$w_{ik} = \frac{\frac{1}{d_{ik}}}{\sum_{k=1}^{K} \frac{1}{d_{ik}}} \qquad \text{Denklem 4.37}$$

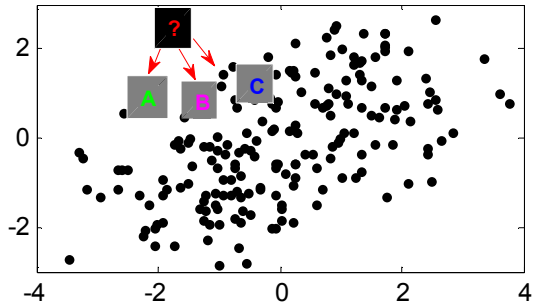
En yakın *k* adet komşuya olan uzaklık değerleriyle hesaplanan ağırlıklı benzerlik oranlarının toplamı *1* olmaktadır. Ağırlık değerleri ($w_{ik}$) bulunduktan sonra eksik olan kaydın eksik nitelik değeri, o niteliğe karşılık gelen en yakın *(k)* komşu kayıtlarındaki niteliklerden yararlanılarak Denklem 4.38'deki gibi hesaplanmaktadır.

$$y_{ij} = \sum_{k=1}^{K} w_{ik}\, x_{kj} \qquad \text{Denklem 4.38}$$

Denklemdeki $y_{ij}$, *i* numaralı kaydın *j* sütunundaki eksik değeri, $w_{ik}$, *i* kaydının *k.* komşu kayıtla olan ağırlıklı uzaklık ölçütü ve $x_{kj}$, *k.* komşunun *j* sütununda bulunan değerini ifade etmektedir (Bras ve Menezes, 2007).

### 4.2.1. En yakın k-komşular uygulaması

Teorik anlatımın pratik kullanımı gösteren basit bir örnek Şekil 4.15 üzerinden izah edilmiştir.




**Şekil 4.15.** Eksik değerlerin en yakın k-komşu hesaplanması örneği

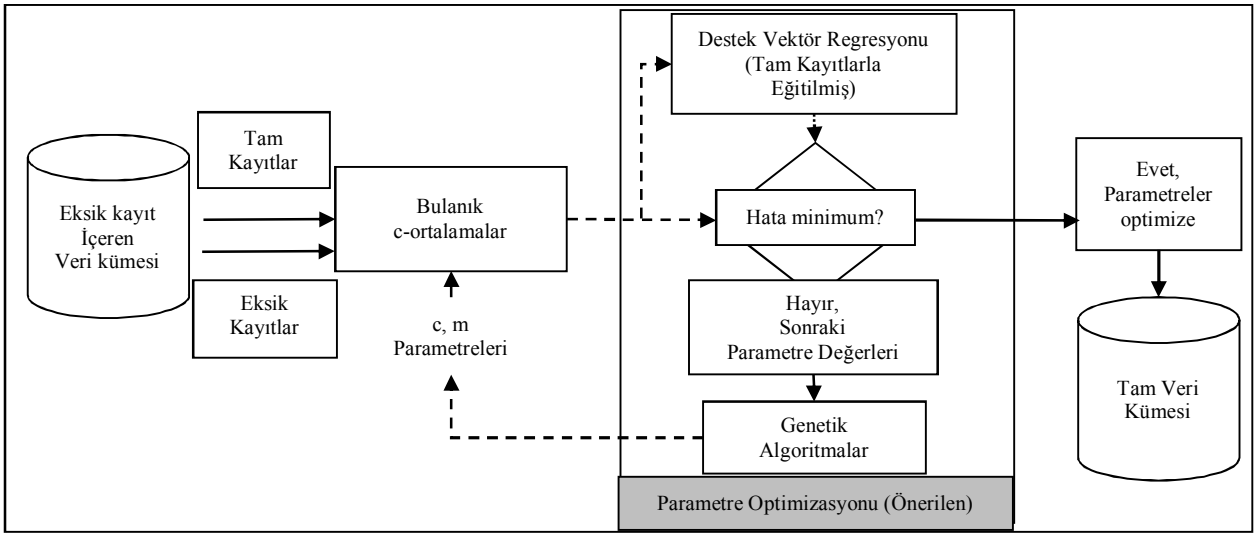
*A, B, C ve ?* veri kümesinin birer kaydı olarak düşünüldüğünde, *soru işareti (?)* eksik veri içeren bir kayıt olarak gösterilmiştir. Öklid uzaklık ölçütüne (Denklem 4.36) göre *soru işareti (?)* kaydına en yakın üç kayıt *A, B, C* oldukları varsayılmıştır. Uzaklık ölçütü hesaplaması sırasıyla $d_{?-A}$*=13.43,* $d_{?-B}$*=9.43,* $d_{?-C}$*=10.13 birim* olarak hesaplanmıştır. Buna göre ağırlıklı uzaklık ölçütleri (Denklem 4.37) ise sırasıyla $w_{?-A}$*=0.2668,* $w_{?-B}$*=0.3797,* $w_{?-C}$*=0.3535* birim olarak hesaplanmıştır. *Soru işareti (?)* kaydındaki eksik niteliğe karşılık gelen *A, B, C* kayıtlarındaki tam nitelik(*X*) değerleri sırasıyla $X_{Ai}$*=3,* $X_{Bi}$*=2,* $X_{Ci}$*=4* olarak kabul edilmiştir. Bu sayısal değerler yardımıyla *soru işareti (?)*'da bulunan eksik nitelik değeri Denklem 4.38'e göre *3*0.2668+2*0.3797+4*0.3535* hesabının sonucu olarak *2.9738* bulunmuştur. Bu hesaplama sonucu en yakın k-komşu sayısının *k=3* alındığı durum için elde edilmiştir.

En yüksek başarı isabetiyle eksik değerin hesaplanabilmesi için komşu sayısının en uygun şekilde belirlenmesi gerekmektedir. *Komşu (k)* sayısı farklı alındığı durumlar için hesaplama sonucu tamamen farklılık göstermektedir. *(k)* komşu sayısı veri kümesi tipi, veri kümesi kayıt sayısı, nitelikler arasındaki matematiksel ilişki, nitelik adedi ve eksiklik oranına bağlı olarak değişebilmektedir. Bu yüzden veri kümesinin özel durumuna uygun komşu sayısı bulma gerekliliği ortaya çıkmaktadır. Bu tez çalışmasında en uygun *(k)* komşu sayısı probleminin çözümü için hibrit bir yaklaşım önerilmiş ve başarılı sonuçları tutarlı olan bu eksik değer hesaplama yöntemi üzerinde araştırmalar yapılmıştır.

# 5. GELİŞTİRİLEN EKSİK DEĞER HESAPLAMA YAKLAŞIMLARI

## 5.1. Eksik Değerlerin Bulanık c-ortalamaları, Destek Vektör Regresyonu ve Genetik Algoritmalar (BcoDvrGa) Hibrit Kullanımı ile Hesaplanması

Tipik olarak eksik veri içeren bir veri kümesi, tam kayıtlar ve eksik kayıtlar olarak iki bölüme ayrılmaktadır. Eksik kayıt bir veya birden çok nitelik değeri olmayan kayıtlara, tam kayıt ise tüm nitelik değerleri var olan veri kümesi satırına ya da nesnesine denmektedir. Şekil 5.1'de önerilen yöntem gösterilmektedir (Aydilek ve Arslan, 2013). Eksik değerler bulanık c-ortalamalar algoritması vasıtasıyla hesaplanabilmektedir. Şekilde *c, küme merkezi adedi* ve *m ağırlık faktörü* parametresini göstermektedir.




**Şekil 5.1.** Önerilen bulanık c-ortalamalar, destek vektör regresyonu ve genetik algoritmalar (BcoDvrGa) eksik değer hesaplama yöntemi

Literatür araştırması bölümünde incelenen bulanık c-ortalamalar ile yapılmış çalışmalarının eksikliğinin giderilmesi için ve ayrıca Başlık 4.2'de bulanık c-ortalamalar (BCO) ile en iyi eksik değer hesaplama başarısının elde edilebilmesi için vurgulanmış olan *c küme merkezi adedinin* ve *m ağırlık faktörü* parametre değerinin veri kümesi yapısına en uygun şekilde belirlenmesi gerekmektedir. Geliştirilen BcoDvrGa yaklaşımı ile bu parametrelerin optimizasyonu yapılmaktadır. Bilinmeyen, gizli veri kümesi yapısı destek vektör regresyonu modeli ile öğrenilmektedir. Genetik algoritmalar, destek vektör regresyonu ile işbirliği yaparak giriş ile çıkış arasındaki

hesaplama *hatasını* en aza indirmeye çalışmaktadır. Hata fonksiyonu değeri Denklem 5.1 ile hesaplanmaktadır.

$$hata = (X - Y)^2 \qquad \text{Denklem 5.1}$$

Burada $X$ destek vektör regresyon çıkışını, $Y$ ise bulanık c-ortalamalar tahmin çıkışını ifade etmektedir. Eksik değerler tahmin edilmeden önce destek vektör regresyonu (DVR), veri kümesindeki tam olan kayıtlarla, giriş değerleri neredeyse çıkış değerlerine benzer olacak şekilde eğitilmiş olması gerekmektedir. Bu sayede genetik algoritmalar (GA) $X$ giriş ile $Y$ çıkış arasındaki en düşük farkı bulmayı amaçlayan uygunluk fonksiyonu yardımıyla en uygun *c, m* parametre değerlerini tespit etmektedir.

Önerilen hibrit yöntem olan bulanık c-ortalamalar (BCO) destek vektör regresyonu (DVR) ve genetik algoritmalar (GA) ile eksik değer hesaplaması aşağıdaki aşamalardan oluşmaktadır.

1. Destek vektör regresyon modeli veri kümesinin tam kayıtlarıyla giriş çıkışa yaklaşık eşit olacak şekilde eğitilir, giriş($X$) ≈ çıkış($Y$).
2. Eksik olan kayıtlar bulanık c-ortalamalar (BCO) ve destek vektör regresyonu (DVR) ile ayrı ayrı hesaplanır.
3. Genetik algoritmalar ile BCO ve DVR arasındaki farkı minimize eden en uygun *c* ve *m* parametre değerleri tespit edilir.
4. Optimize edilmiş parametre değerleri BCO ile kullanılarak eksik değerler hesaplanır.

Bir diğer önerilen yaklaşımda ise sadece bulanık c-ortalamalar (BCO) ve genetik algoritmalar (GA) kullanılarak eksik değer hesaplaması yapılmaktadır. Bu yöntemde tam olan veri kümesinden bazı değerler yapay olarak silinmektedir. BCO eksik değer hesaplaması ile silinmiş olan aslında bilinen gerçek değerlere en yakın hesaplama yapılan optimum *c* ve *m* parametre değerleri genetik algoritmalar yardımıyla tespit edilmektedir. Daha sonra eksik olan değerlerin hesabı bu optimum parametre değerleri vasıtasıyla BCO eksik değer hesaplaması tarafından yapılmaktadır. BcoGa eksik değer hesaplama yaklaşımı aşağıdaki aşamalardan oluşmaktadır.

1. Tam olan veri kümesinden bazı kayıtlar yapay olarak eksik hale getirilir.
2. Yapay olarak eksik hale getirilmiş değerler BCO tarafından hesaplanır. Eksiltilmiş değerler ile hesaplanan değerler arasındaki farkı minimize eden genetik algoritmalar optimum *c* ve *m* parametre değerlerini bulur.
3. Optimum parametre değerleri ile BCO tarafından veri kümesindeki gerçekte eksik değerler hesaplanır.

## 5.2. Eksik Değerlerin En Yakın K-Komşu ve Yapay Sinir Ağları (EykYsa) Hibrit Kullanımı ile Hesaplanması

Eksik değerler *(k)* en benzer kayıt adedi olmak üzere ağırlıklı en yakın k-komşu yöntemiyle hesaplanabilmektedir. Literatür araştırması bölümünde incelenen en yakın k-komşu ile yapılmış çalışmalarının eksikliğinin giderilebilmesi için ve ayrıca Başlık 4.4'de en yakın k-komşu (EYK) ile en başarılı hesaplama doğruluğunu elde edebilmek için vurgulanmış veri kümesine en uygun *(k)* komşu adedi kullanılması gerekliliği ifade edilmiştir. Çıkış($Y$) katmanında, giriş($X$) katmanında girilen değerlere en yakın çıkış($Y$) üreten yapay sinir ağları modelinin fark fonksiyonu Denklem 5.2'deki gibi kabul edilmektedir. Denklemdeki $X$ yapay sinir ağları girişini, $Y$ ise yapay sinir ağları çıkışını ifade etmektedir.

$$fark = X - Y \qquad \text{Denklem 5.2}$$

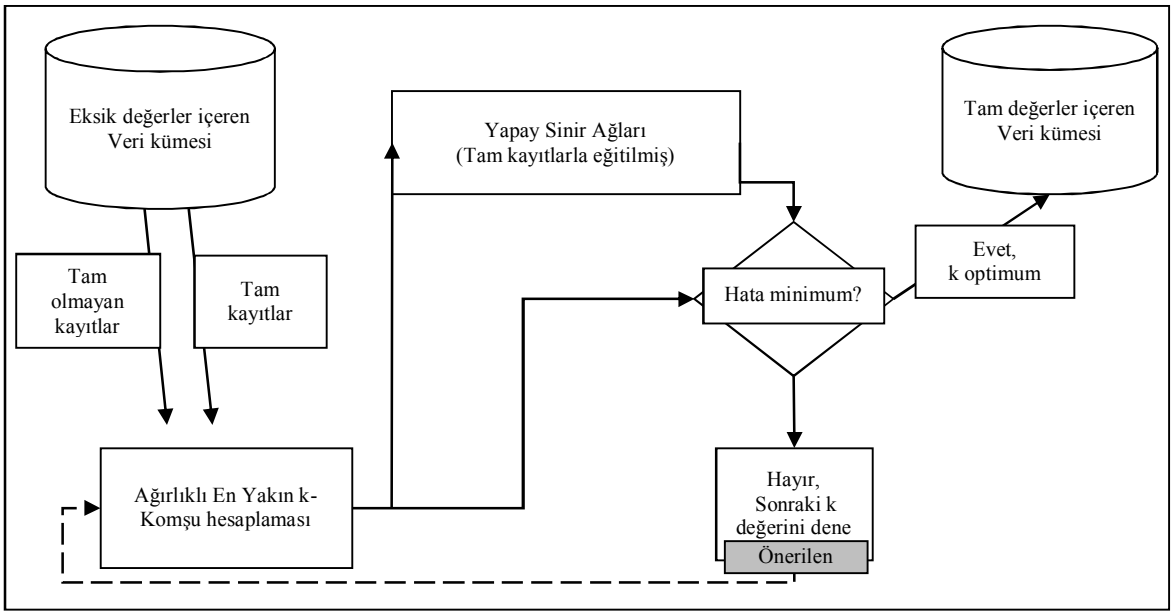
$$amaç\ fonksiyonu = (X - Y)^2 \qquad \text{Denklem 5.3}$$

Fark fonksiyonunun Denklem 5.2'nin karesi aynı zamanda en uygun *(k)* komşu adedini bulmak için kullanabilecek negatif olmayan minimum fark (Denklem 5.3) fonksiyonu olarak kullanılmaktadır. Eksik veri kaydı bir veya daha fazla nitelik değeri olmayan, tam veri kaydı ise bütün nitelikleri var olan veri kümesi kayıtlarına denmektedir. Eksik veri içeren tipik bir veri kümesi tam kayıtlar ve eksik kayıtlar olmak üzere iki alt kümeye ayrılmaktadır.

Şekil 5.2'de önerilen yöntem gösterilmektedir. En yakın k-komşu (EYK) ve yapay sinir ağları (YSA) ile eksik değer hesaplaması aşağıdaki aşamalardan oluşmaktadır (Aydilek ve Arslan, 2012).

Eksik değer hesaplaması yapılabilmesi için veri kümesindeki kayıtların ve niteliklerin veri kümesindeki diğer kayıt ve niteliklerle olan bilinmeyen matematiksel

ilişkilerinin çözümlenmesi, anlaşılması gerekmektedir. Önerilen yöntemde veri kümesindeki gizli ilişkileri çözümleyebilmek için yapay sinir ağları modeli kullanılmaktadır. Böylece veri kümesi yapısı bir model olarak analiz edilmiş ve eksik değerler ağırlıklı en yakın k-komşu algoritmasıyla tahmin edilmiştir.




**Şekil 5.2.** Önerilen yöntem, Eksik değerlerin hibrit en yakın k-komşu, yapay sinir ağları ile hesaplanması

1. Yapay sinir ağı modeli tam olan kayıtlarla giriş çıkışa yaklaşık eşit olacak şekilde eğitilir, giriş ($X$) ≈ çıkış ($Y$).
2. Eksik değer içeren kayıtlar k=1'den k=tam komşu kayıt sayısına kadar ağırlıklı EYK ile hesaplanır.
   a. Eksik değeri hesaplanmış kayıtlar giriş($X$) olarak yapay sinir ağları modeline verilerek çıkış($Y$) değerleri elde edilir.
   b. En düşük farka $(X - Y)^2$ (Denklem 5.3) yani hataya karşılık gelen ($k$) komşu değeri tespit edilir.
3. Eksik değerler tespit edilmiş olan en uygun ($k$) ile ağırlıklı en yakın k-komşu yaklaşımı ile hesaplanır.

# 6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

## 6.1. BcoDvrGa Ön Hazırlık

Deneysel araştırma sonuçlarının elde edilmesinde literatürde sıklıkla kullanılan ve makine öğrenmesi veri kümelerini bünyesinde içeren UCI (California Üniversitesi Bilimsel Veri Tabanı) veri havuzundan (Blake ve ark., 1998) Çizelge 6.1'deki altı adet veri kümesi kullanılmıştır.

**Çizelge 6.1.** Kullanılan veri kümeleri

| Veri kümesi | Kayıt Sayısı | Nitelik Sayısı |
|---|---|---|
| Glass | 214 | 11 |
| Haberman | 306 | 4 |
| Iris | 150 | 5 |
| Musk1 | 476 | 167 |
| Wine | 178 | 14 |
| Yeast | 1489 | 9 |

Testlerin çok sayıda ve farklı kayıt sayısı, nitelik sayısına sahip veri kümeleriyle yapılmasının nedeni genel olarak önerilen yaklaşımın nasıl çalıştığını ve araştırma sonuçlarının ne derece tutarlı bir şekilde ortaya çıktığının görülmesi amaçlanmıştır. Aynı zamanda yaklaşımların performanslarının *6* farklı şart altında ölçülebilmesi için veri kümeleri üzerinde *%1, %5, %10, %15, %20 ve %25* oranlarında yapay olarak eksik değerler oluşturulmuştur.

Veri kümelerinin nitelikleri farklı boyut aralıklarında olduklarından ve tüm algoritmaların eşit şartlar altında test edilmesi istendiğinden dolayı tüm veri kümelerinin Min-max normalizasyonu Denklem 6.1 ile yapılmıştır. Bu sayede veri kümesindeki tüm niteliklerin değerleri küçük bir aralığa indirilerek *0* ile *1* arasında ondalık değerler alması sağlanmıştır.

$$Norm_x = \frac{x - Min_x}{Max_x - Min_x} \qquad \text{Denklem 6.1}$$

Normalizasyon özellikle sınıflama algoritmaları için çok kullanışlıdır, eğitim veri kümesinde bulunan her bir nitelik değişkenine uygulanan normalizasyon öğrenme sürecinin hızlanmasına yardımcı olmaktadır. Uzaklık ölçütü kullanan algoritmalarda ise

normalizasyon büyük aralıklı niteliklerin küçük aralıklı nitelikler üzerinde baskınlık oluşturmasını önlemektedir (Han ve Kamber, 2001). Bu özelliklerinden dolayı önerilen hibrit yaklaşımda veri kümeleri üzerinde normalizasyon uygulanmış ve destek vektör regresyonu eğitim süresi azaltılmıştır. Ayrıca kıyaslanacak tüm algoritma ve yöntemlerin eşit koşullarda bulunan veri kümeleri ile çalışması garanti altına alınmıştır.

Önerilen yaklaşım Matlab 2009b'nin 7.9 versiyonu kullanılarak kodlanmış ve 'least squares support vector (LS-SVM) toolbox' (Suykens ve ark., 2002) ile regresyon yapısı 'radial basis kernel' kullanılarak oluşturulmuştur. Genetik algoritmalar Matlab 'genetic algorithm toolbox' ile popülasyon sayısı *20*, nesil sayısı *40*, çaprazlama %60, mutasyon oranı %3 kullanılarak gerçekleştirilmiştir. Bu parametre tercihleri literatür araştırması sonucu elde edilen bilgiler sonucunda diğer kıyaslanan benzer algoritmalarla kullanılan parametre değerleri oldukları için tercih edilmiştir.

### 6.2. Araştırma Sonuçları İçin Kıyaslama Ölçütleri

Eksik veri hesaplama algoritmalarının verimliliğinin ölçülebilmesi için hata karelerinin ortalamasının karekökü (HKOK), göreceli tahmin doğruluğu (D), Wilcoxon sıra toplamı istatiksel güven testi (W) ve saniye cinsinden çalışma zamanları (t) performans ölçüm değerlerinin kıyaslaması yapılmıştır. Hata karelerinin ortalamasının karekökü (HKOK) Denklem 6.2 ile gösterilmiştir. HKOK, gerçek değerler ile hesaplanmış eksik değerler arasındaki farkı yani hatayı ölçer ve tahmin isabet niceliğini belirtmektedir (Abdella ve Marwala, 2005; Marwala, 2009).

$$HKOK = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x}_i)^2}{n}} \qquad \text{Denklem 6.2}$$

Burada $x_i$ gerçek değer, $\overline{x}_i$ hesaplanmış değer ve $n$ sayısı ise eksik değer adedini göstermektedir. Göreceli tahmin doğruluğu (D) belirli bir tolerans aralığında yüzde kaç doğru tahmin yapıldığı ölçmektedir (Abdella ve Marwala, 2005; Marwala, 2009). Tolerans aralığı (Nelwamondo ve ark., 2007) tarafından *%10* olarak kullanılmıştır. *%10* göreceli tahmin doğruluğu toleransı, hesaplanmış eksik değeri *±%10* gerçek değer aralığı içerisinde doğru bir hesaplama olarak kabul etmektedir. Örneğin gerçek değer *100* olduğunda, algoritma tarafından hesap edilen eksik değer eğer *90* ile *110* arasında bulunursa bu durumda doğru bir tahmin yapıldığı kabul edilmektedir.

$$D = \frac{n_T}{n} \times 100 \qquad \text{Denklem 6.3}$$

Denklem 6.3'de görülen ölçümde $n$ toplam tahmin sayısını $n_T$ ise belirli bir toleransta yapılan doğru tahmin sayısını göstermektedir.

Wilcoxon çift taraflı sıra toplamı hipotez testi (W) eksik değer hesaplama yöntemlerinin istatiksel olarak güvenirlik ve geçerliliğini sınamak için kullanılmıştır. Gerçek değer vektörü ile hesaplanmış eksik değer vektörünün dağılımlarının istatiksel olarak birbirine benzeyip benzemediğini test etmek için kullanılmaktadır. Testin istatiksel güven seviyesi *0.05* olarak kullanılmıştır (Hajek ve Sidak, 1967). Bu testin kullanılmasının nedeni karşılaştırılacak olan veri vektörlerinin varyanslarının eşit olma zorunluluğunun olmamasıdır. Özellikle eksik değer yerine sıfır koyma yöntemi gibi yüksek hatalı hesaplamalar yapabilen yöntemlerde varyans değer değişiminden kaynaklanan etkileri yok saymaktadır. Boş hipotez $H_0 Y = Y_{est}$, *P-değeri>0.05* olduğunda eşit dağılım olduğu kabul edilmektedir. *Y*, gerçek değer $Y_{est}$, hesaplanmış değer vektörünü ifade etmektedir. *P-değeri* Denklem 6.4 ile elde edilmektedir. Buna göre *Y* ve $Y_{est}$ küçükten büyüğe sıralanmakta ve sıra numarası verilmektedir. *R*, gerçek değer vektörünün sıra numaraları toplamıdır. *Pr, z* rastgele değişkenine karşılık gelen standart normal dağılım tablosu olasılık değeridir. $n_1, n_2$ ise sırasıyla *Y* ve $Y_{est}$ vektörlerinde bulunan eleman sayısını belirtmektedir (Sehgal ve ark., 2008).

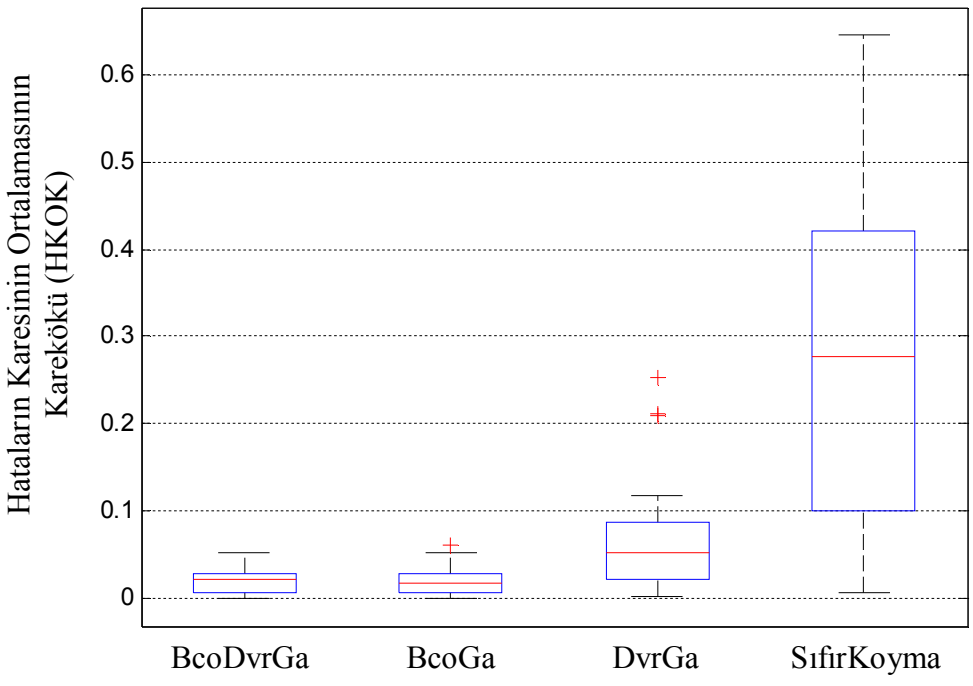
$$P - değeri = 2\big(1 - Pr(|z|)\big), \quad z = \frac{R - (\frac{n_1(n_1 + n_2 + 1)}{2})}{\sqrt{n_1 n_2 (n_1 + n_2 + 1)/12}} \qquad \text{Denklem 6.4}$$

İki yöntemin sonuçlarının karşılaştırılması sonucu yüksek *P-değeri*, *düşük P-değerine* göre daha isabetli hesaplama yapıldığı anlamına gelmektedir (Sidak ve ark., 1999; Sehgal ve ark., 2008). Bir diğer performans kıyaslama kriteri olarak çalışma süresi (t) saniye cinsinden ifade edilmiştir. Bu amaçla yapılan hesaplamaların hepsi aynı standart bilgisayar üzerinde yapılmış ve çalışma zamanları hakkında ölçümler tutulmuştur. Çalışma zamanları ölçümleri Intel Core 2 Quad işlemcisi, 4.00 GB hafıza ve Microsoft Windows XP SP2 işletim sistemi özelliklerine sahip bilgisayar ile yapılmıştır.

### 6.3. BcoDvrGa Araştırma Sonuçları

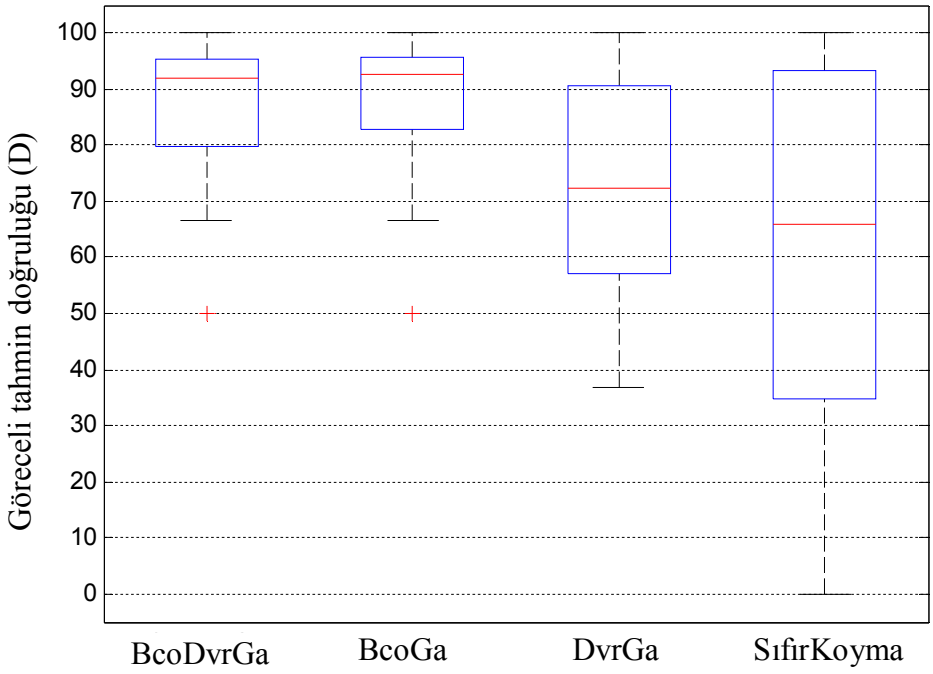
Şekil 6.1-6.4 arasındaki grafiklerde *%1, %5, %10, %15, %20* ve *%25* oranlarında eksik veri içeren Glass, Haberman, Iris, Musk1, Wine ve Yeast veri kümelerinin araştırma sonuçları kutu grafikleri şeklinde gösterilmiştir. Sonuçlar hibrit bulanık c-ortalamalar yaklaşımının eksik değer hesaplaması konusunda başarılı sonuç değerlerine sahip olduğunu onaylamıştır. Kutu grafiklerde dikdörtgen kutu alanı toplam sonuçların *%50*'sini kapsamaktadır. Kutunun içinde bulunan çizgi ortanca değeri, kutunun kenarları *%25* ve *%75* sonuç değerlerinin başlangıç ve bitiş noktalarını, kutu dışına uzayan kesikli çizgiler ise geri kalan sonuçları göstermektedir. Ayrıca bireysel olarak gösterilen artı işaretleri ise aykırı değerler olarak kabul edilen sonuç değerlerini ifade etmektedir.

Şekil 6.1, BcoDvrGa, BcoGa, DvrGa, SıfırKoyma isimli *4* yöntemin *%1* ile *%25* eksik arasında eksik veri içeren *6* veri kümesi çalışma sonuçlarının karşılaştırmasını göstermektedir. Her bir kutu her bir yöntemle elde edilen *36* adet hatalar karesinin ortalamasının karekökü (HKOK) sonuç değerini içermektedir. HKOK, genel olarak yöntemin hesaplama hatasını ölçer, düşük değerli hata değeri daha iyi performans değeri ifade etmektedir. Ortanca HKOK değerleri sırasıyla *0.0215, 0.0178, 0.0511 ve 0.2765* olarak bulunmuştur.



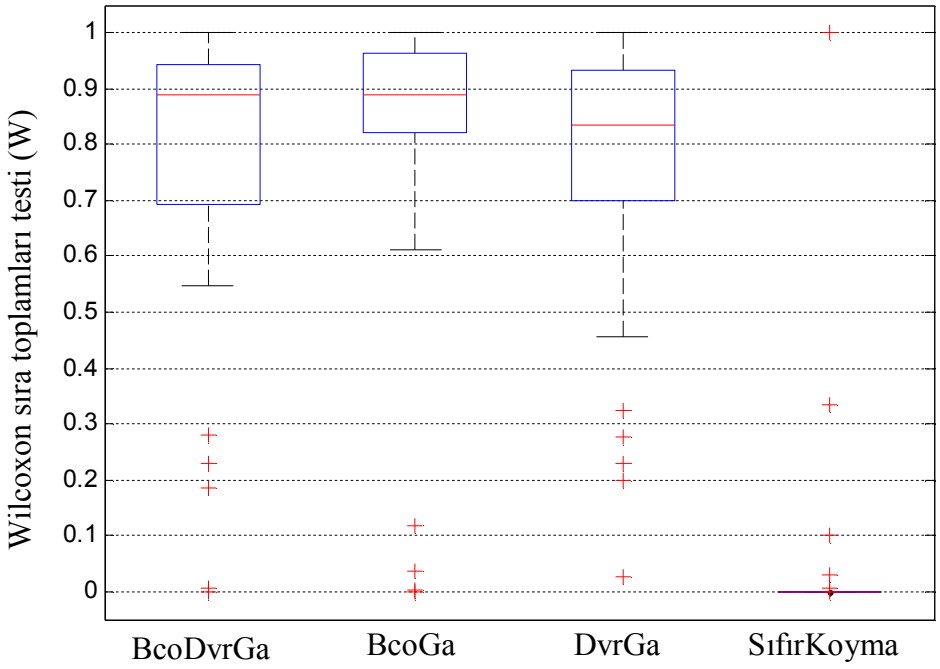

**Şekil 6.1.** %1-25 arasında eksik veri içeren tüm veri kümelerinin HKOK hata sonuçları

Şekil 6.2'de hesaplama doğruluk oranları gösterilmiştir. Bu oranlar gerçek eksik değer ile hesaplanan eksik değer arasındaki farkın belirli bir tolerans aralığı dahilinde olması sonucunda yüzde olarak bulunmuştur. Grafik sonuçlarında kullanılan *%1* ile *%25* arasındaki eksik veri içeren *6* veri kümesi için kullanılan tolerans değeri *%10* olarak belirlenmiştir. Yüksek değerli doğruluk ilgili yaklaşımın daha iyi eksik veri hesaplandığını göstermektedir. Göreceli tahmin doğruluğu (D) ortanca değerleri sırasıyla *92.03, 92.65, 72.17 ve 65.95* olarak bulunmuştur.




**Şekil 6.2.** %1-25 arasında eksik veri içeren tüm veri kümelerinin göreceli tahmin doğruluğu sonuçları

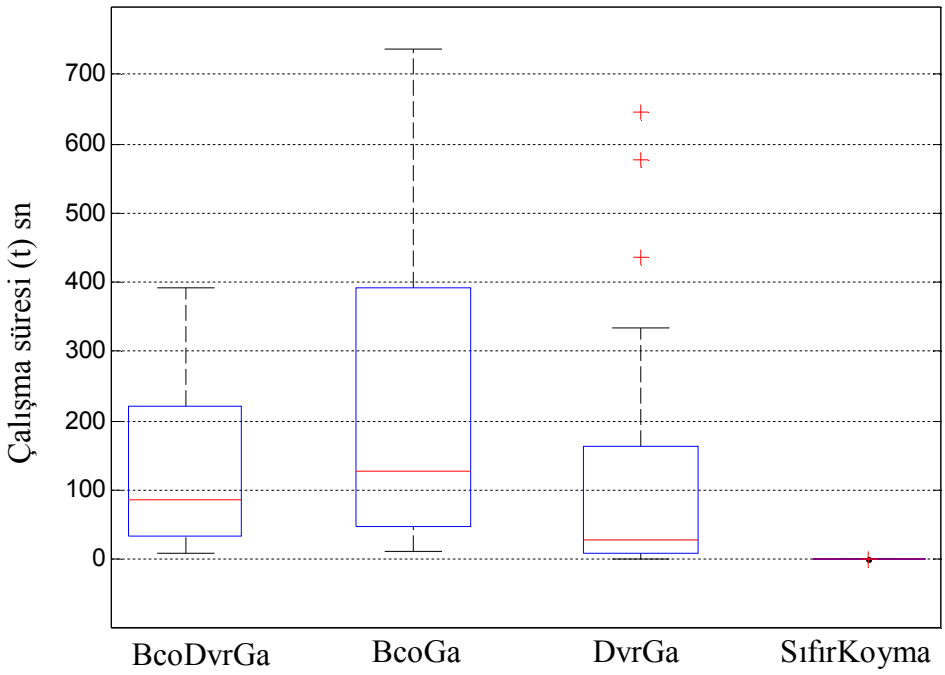
Şekil 6.3, Wilcoxon sıra toplamları istatiksel güven testi sonuçlarını göstermektedir. Yüksek değerli Wilcoxon P-değerine (Denklem 6.4) sahip yöntemin diğer yönteme göre üstün olduğu anlamına gelmektedir. *4* yöntemin *%1* ile *%25* arasında eksik veri içeren *6* veri kümesi üzerindeki ortanca değerleri sırasıyla *0.8877, 0.8908, 0.8352 ve ≈0* olarak bulunmuştur.




**Şekil 6.3.** %1-25 arasında eksik veri içeren tüm veri kümelerinin Wilcoxon testi sonuçları

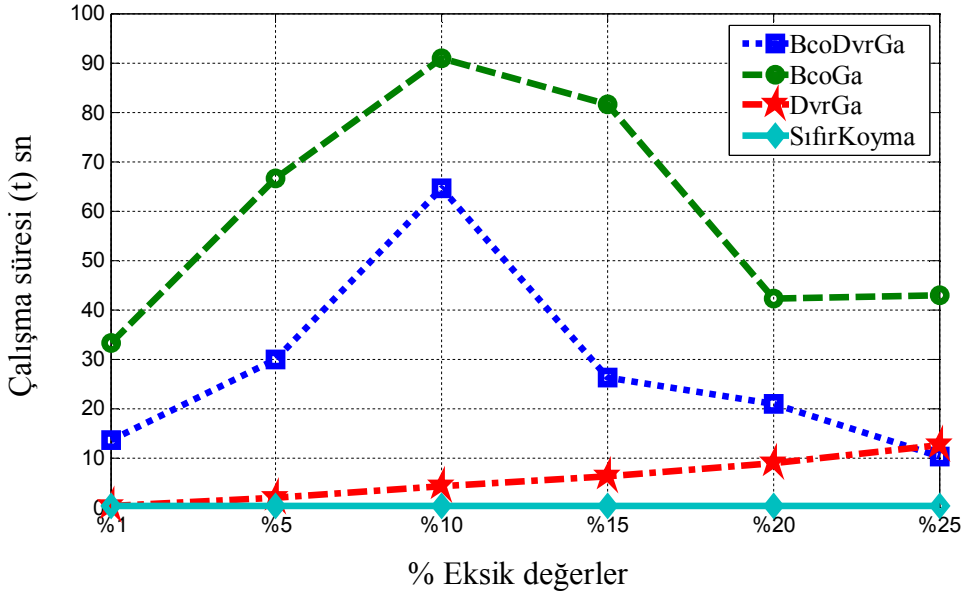
Şekil 6.4, aynı standart bilgisayarda *4* yöntemin çalışma sürelerini saniye cinsinden göstermektedir. Çalışma zamanları önerilen hibrit yöntemin bulanık c-ortalamalar algoritmasının amaç fonksiyonun (Denklem 4.19) diğerlerine nazaran biraz daha karmaşık yapıda olduğu göstermektedir. Bu nedenle çalışma süresi açısından önerilen yöntemin diğerlerine göre biraz daha uzun sonuçlar doğurduğu görülmüştür. Çalışma süresi ortanca değerleri sırasıyla *85.28, 128.14, 27.03 ve 0.001* saniye olarak bulunmuştur.

Şekil 6.1-6.4 arasındaki grafikler elde edilen ana sonuçların oluşmasının sebepleri hakkında da bilgiler içermektedir. Bulanık c-ortalamalar veri kümelerinde bulunan kayıtların sınıf değerlerinin değişmesini anlayan ve aynı zamanda belirsiz veriler (örneğin aykırı, gürültülü veriler) için esnek ve güçlü bir yöntemdir (Shahi ve ark., 2009). Bu faydalarından dolayı bulanık c-ortalamalar yaklaşımı eksik değer hesaplama performansını arttırmış ve önerilen yöntemin üstünlüğünü sağlamıştır.



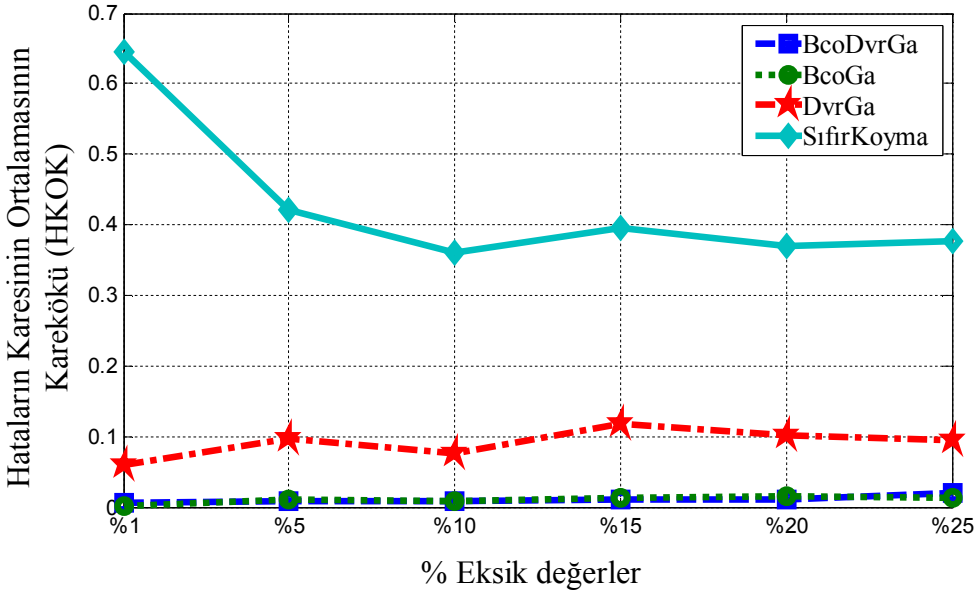

**Şekil 6.4.** %1-25 arasında eksik veri içeren tüm veri kümelerinin çalışma süresi (sn.) sonuçları

Şekil 6.5-6.8 arasındaki grafikler *%1, %5, %10, %15, %20 ve %25* oranlarında eksik değer içeren *Iris veri kümesi* için araştırma sonuçlarını göstermektedir. Şekil 6.5'de önerilen hibrit bulanık c-ortalamalar yönteminin daha uzun çalışma süresi olduğu görülmektedir. Yöntem diğer yöntemlere nazaran bulanık c-ortalamalar kümeleme ile birlikte destek vektör regresyonu eğitim süresini içerdiğinden dolayı daha uzun süre çalışmaktadır. Eğitim için kullanılan tam veri kümesi kümeleme ve regresyon yapısı için daha uygun yapıda olursa buna bağlı olarak çalışma süresi aynı oranda azalmaktadır. *%20* ve *%25* eksik veri içeren *Iris veri kümesi* için bu durum grafikteki sonuçlara yansıdığı görülmektedir. Bir optimizasyon yöntemi olan genetik algoritmaların, veri kümesi içindeki ilişkileri çözümleyebilen destek vektör regresyonu ile birlikte kullanılması sonucu oluşturulan önerilen yöntem BcoDvrGa, *Iris veri kümesi* için BcoGa'ya göre daha kısa süre çalışmaktadır. Bunun nedeni *Iris veri kümesinin* düşük sayıda kayıt, nitelik içermesine bağlı olarak regresyon yapısının hızlı oluşması ve böylece genetik algoritmaların daha kısa sürede çözüme ulaşması neden olmuştur.



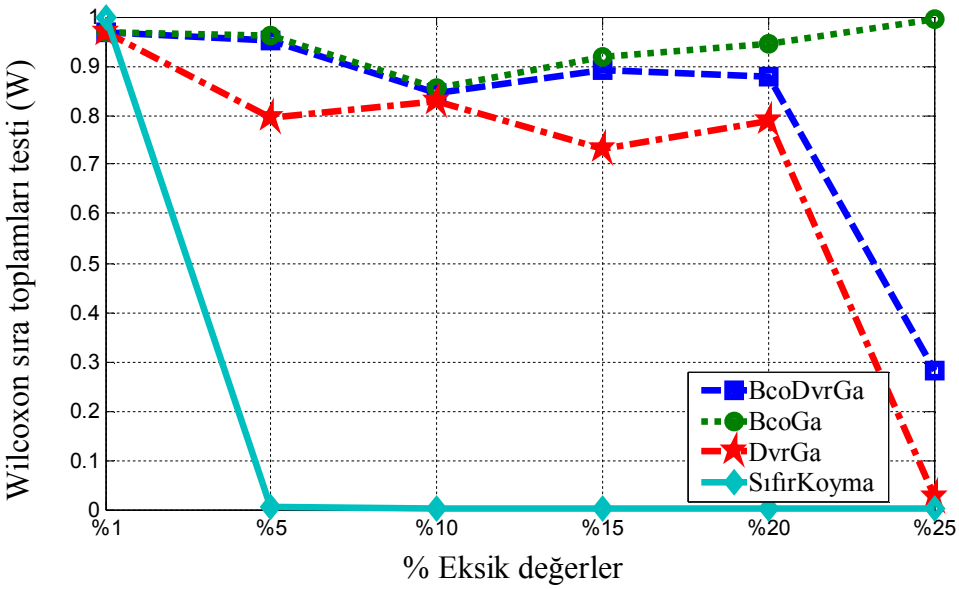

**Şekil 6.5.** %1-25 arasında eksik veri içeren Iris veri kümesi için çalışma süresi (sn.) sonuçları

Şekil 6.6'da hibrit bulanık c-ortamalar yaklaşımının DvrGa ve SıfırKoyma yöntemlerine göre Iris veri kümesi için daha düşük hata değerlerine sahip olduğu görülmektedir.



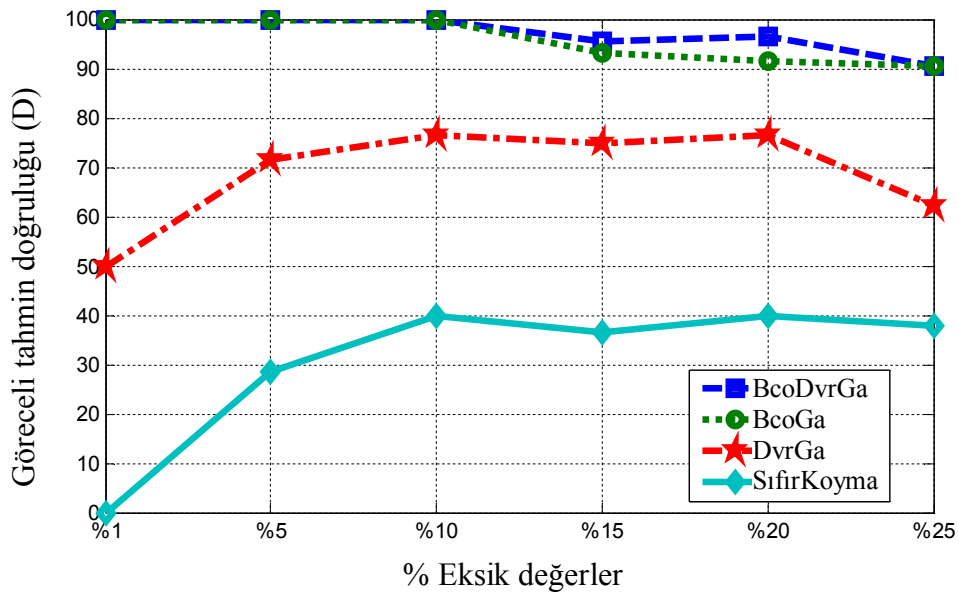

**Şekil 6.6.** %1-25 arasında eksik veri içeren Iris veri kümesi için HKOK hata sonuçları

Şekil 6.7 hibrit bulanık c-ortamalar yaklaşımının Iris veri kümesi üzerinde diğer yöntemlere göre istatiksel olarak daha güvenilir sonuçlar edildiğini kanıtlamaktadır.




**Şekil 6.7.** %1-25 arasında eksik veri içeren Iris veri kümesi için Wilcoxon testi sonuçları

Şekil 6.8'de yapay olarak silinmiş yani gerçekte bilinen eksik değerler ile sonradan hesaplanmış eksik değerler arasındaki *%10* tolerans farkla doğru tahmin yapılıp yapılmadığı gösterilmektedir. *%1* ve *%10* eksik değer oranlarında Iris veri kümesi için önerilen yöntemin daha isabetli hesaplamalar yaptığı ortaya çıkmaktadır.




**Şekil 6.8.** %1-25 arasında eksik veri içeren Iris veri kümesi için göreceli tahmin doğruluğu (D) sonuçları

Önerilen yaklaşımın özgün olduğu özellikler, eksiklikleri ve diğer kıyaslanan yöntemlere göre avantajları şu şekilde ifade edilebilir. Temel klasik bulanık c-ortalamalar eksik değer hesaplama modeli sabit, önceden belirlenmiş *merkez küme*

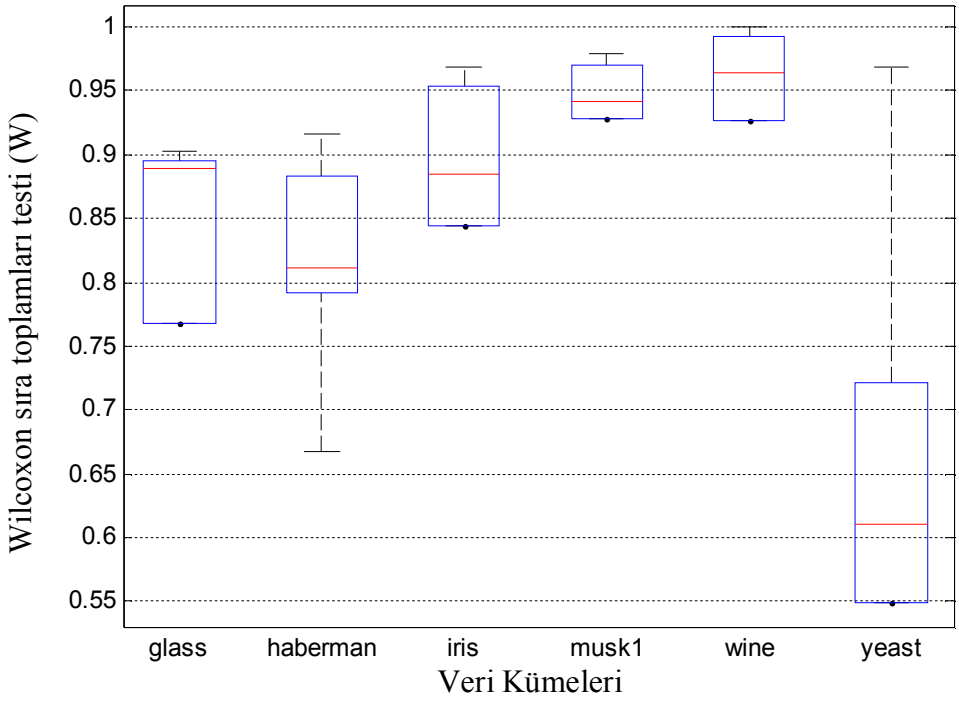
*adedi (c)* ve *ağırlık faktörü* (*m)* parametre değerleri kullanmaktadır. Bunun yanında bu sabit ve baskın parametre değerleri veri kümesi tipi ayırt etmeksizin tüm veri kümeleri üzerinde ve aynı zamanda farklı yüzde oranlardaki eksik olma durumunda dahi uygulanmaktadır. Bu durum özellikle hassas eksik değer hesaplaması gerektiren durumlarda olumsuz sonuçlar doğurmaktadır. Oysa bu parametre değerleri veri tipinin zaman serisi, tam sayı, ondalık sayı olup olmamasına göre veya veri kümesinin kayıt veya nitelik sayısına, en önemlisi veri kümesi niteliklerinin birbirleriyle olan ilişkilerine bağlı olarak değişmelidir.

Kıyaslanan benzer DvrGa yöntemi bazı aykırı değerli veriler üzerinde yanlış hesaplamalar yapmakta ve genetik algoritmalar ise yerel minimum sonuçlara takılı kalabilmektedir yani tüm çözüm uzayı aranmadan en iyi çözümün bulunduğu varsayılmaktadır. Genetik algoritmaların eksik değer hesaplamadaki problemi yüksek derecede uygun fakat optimum olmayan bireylerin popülasyon üzerinde baskın hale gelmesinden kaynaklanmaktadır. Bir kez popülasyon belli bir noktaya yakınsadığında daha iyi çözüm için arama kabiliyeti kısıtlanmakta ayrıca çaprazlama ve mutasyon oranlarının uygun tutulmadığı durumlarda çözümün bulunması yavaş ve uzun sürebilmektedir (Goldberg, 1989). Diğer taraftan önerilen yaklaşım BcoDvrGa, destek vektör regresyonu ile veri nitelikleri arasındaki matematiksel ilişkiler tespit edilerek veri için en uygun parametre seçimi yapılmaktadır. Böylece araştırma sonuçları en başta tercih edilen parametre seçimlerine bağlı kalmamaktadır. Ayrıca bulanık c-ortalamalar yöntemi aykırı ve gürültülü veriler için tutarlı veri hesaplaması sağlamaktadır. Bulanık hesaplama gürültü azaltma ve iyileştirme için başarılı bir şekilde kullanılmakta ve gürültü altında bile kümeleme başarısını arttırdığı bilinmektedir (Gil ve ark., 1999; Ahmed ve ark., 2002; Xiaohong ve ark., 2011).

Önerilen yöntemin bazı eksiklikleri vardır. Bunlardan biri destek vektör regresyonu eğitimi önemli bir konudur, eğitim yapısında kullanılacak regresyon çekirdek tipi (Suykens ve ark., 2002) ve eğitim sonlandırma kriteri eksik veri hesaplamasından önce belirtilmesi gerekmektedir. Önerilen yaklaşımla ilgili gelecekte eksik değer hesaplama performansını artırabilecek iyileştirme olarak eğitim verisi üzerinde özellik seçimi ve/veya boyut azaltma teknikleri (Fodor, 2002) uygulanabilir olmasıdır. Örneğin temel bileşenler analizi gibi analizler destek vektör regresyonu eğitim süresini azaltmak için kullanılması uygun görülmektedir.

Şekil 6.9'da görüldüğü üzere kümeleme işlemi ve destek vektör regresyonu eğitim aşamalarına daha uygun yapıda veriler için önerilen yaklaşım daha başarılı

sonuçlar üretmektedir. Yaklaşımla ilgili gelecekte yapılabilecek bir başka öneri ise eksik değer hesaplaması yapılmadan önce veri kümesinin bulanık kümeleme süreci için ne derece tutarlı ve uygun olduğunu belirleyen çeşitli bulanık kümeleme geçerlilik analizleri (Xie ve Beni, 1991) uygulanması ve buna bağlı sonuçlar üzerinde geliştirmeler yapılması tavsiye edilmektedir.




**Şekil 6.9.** Tüm veri kümeleri için Wilcoxon sıra toplamları testi (W) sonuçları

## 6.4. EykYsa Ön Hazırlık

Çizelge 6.2'de araştırma sonuçlarının elde edildiği veri kümeleri görülmektedir. Araştırma sonuçları elde edilmesinde kullanılan çizelgedeki veri kümelerinin tercih edilmesinin nedeni literatürde kıyaslama yapılan diğer benzer çalışmalar ile birlikte sıklıkla tercih edilmelerinden kaynaklanmaktadır. İlk veri kümesi Fransa'da bulunan 120-MW elektrik üreten santralin normal çalışma koşullarındaki verileridir. Bu veri kümesi *5* girişten oluşur ve her 1228.8 saniyedeki kaydedilen toplam *200* kayıttan oluşmaktadır (Mohamed ve Marwala, 2005; Qiao ve ark., 2005; Nelwamondo ve ark., 2007; Marwala, 2009). İkinci veri kümesi *827* kayıt içeren Ogawa, saccharomyces cerevisiae isimli mayanın polifosfat metabolizması ve fosfat birikimi ile ilgili kayıtları *8* nitelikten faydalanarak kayıt altına almıştır (Ogawa ve ark., 2000). Üçüncü veri kümesi maya hücresindeki yaşam döngüsü ve DNA mikro dizi zaman serisi verisinin

birleşiminden oluşmakta ve ismini Cdc28 bazlı senkronizasyondan almaktadır (Spellman ve ark., 1998), dördüncü veri kümesi ise endüstriyel bobin projesi testlerini temsil etmektedir her *0.1* saniyede kaydedilmiş *2500* kayıt ve *5* girişten oluşmaktadır (Marwala, 2009).

**Çizelge 6.2.** EykYsa araştırma sonuçlarının elde edildiği veri kümeleri

| Veri Kümesi | Kayıt | Nitelik | Eksik değerli Kayıt sayısı | Eksik olmayan Kayıt Sayısı |
|---|---|---|---|---|
| Powerplant | 200 | 5 | 20 | 180 |
| Ogawa | 827 | 8 | 82 | 745 |
| Cdc28 | 1370 | 17 | 137 | 1233 |
| Winding | 2500 | 5 | 250 | 2250 |

Performans ölçümü ve önerilen yöntemin literatürdeki diğer benzer yöntemle eşit şartlar altında kıyaslanabilmesi için veri kümelerinin bazı değerleri yapay olarak silinmiştir. Veri kümelerinin bütün niteliklerine, yapay sinir ağının aktivasyon fonksiyonunun aktif aralığına girmesi için Min–max normalizasyonu (Denklem 6.1) uygulanmıştır.

Deneysel araştırma sonuçlarının elde edilmesi için Matlab 2009b yazılımının 7.9 versiyonu kullanılarak algoritmalar geliştirilmiştir. Yapay sinir ağları için Matlab'da bulunan 'neural network toolbox' kullanılmıştır, *1* adet gizli katmanına sahip olan ve aynı zamanda giriş, çıkış ve gizli katmanında bulunan nöron sinir hücresi sayıları aynı olan bir yapay sinir ağı modeli oluşturulmuştur. Ağırlık değerlerini güncelleyen ölçeklenmiş birleşik gradyan (trainscg), ağ eğitim yöntemi olarak kullanılmıştır. Gizli katmanda transfer fonksiyonu olarak hiperbolik tanjant sigmoid (tansig), çıkış katmanında ise lineer transfer (purelin) fonksiyonu kullanılmıştır. Veri kümesi %60 eğitim, %20 doğrulama ve geri kalan %20 veri ise test verisi olarak ayrılmış ve ağ performans sonlanma kriteri *0.0001* olacak şekilde eğitilmiştir. Genetik algoritmalar için ise Matlab'da bulunan 'genetic algorithm toolbox' kullanılmış, *20* popülasyona ve *40* nesile sahip yapı, %60 çaprazlama ve %3 mutasyon oranları ile çalıştırılmıştır. Kullanılan bu parametre değerleri literatürde kıyaslanan benzer yöntemlerde sıklıkla kullanıldığı için tercih edilmiştir.

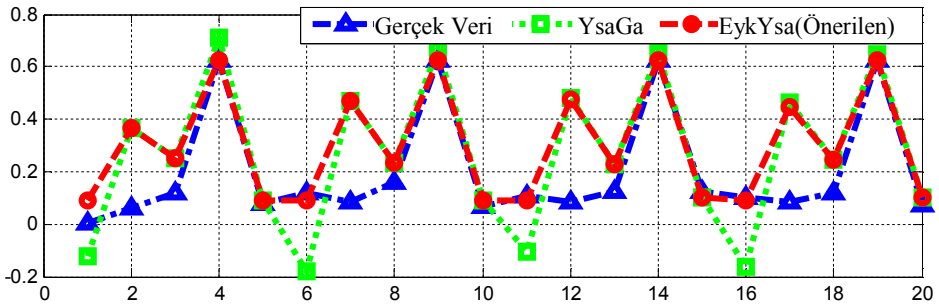
Kıyaslanan yöntemin önerilen yöntemle performansının karşılaştırabilmesi için hata karelerinin ortalamasının karekökü (HKOK) (Denklem 6.2) (Troyanskaya ve ark., 2001; Abdella ve Marwala, 2005; Bras ve Menezes, 2007; Mohamed ve ark., 2007; Marwala, 2009), göreceli tahmin doğruluğu (D) (Denklem 6.3) (Abdella ve Marwala,

2005; Mohamed ve ark., 2007; Marwala, 2009) ve korelasyon katsayısı (r) ölçümleri kullanılmıştır. Korelasyon katsayısı (r) gerçek eksik değerler vektörü ile daha sonra hesaplanmış değerler vektörü arasındaki doğrusal benzerliği ölçmede kullanılmaktadır (Abdella ve Marwala, 2005; Bras ve Menezes, 2007; Mohamed ve ark., 2007; Marwala, 2009). Korelasyon katsayısı değeri *-1* ile *+1* arasında ondalık değer almakta, mutlak değerli korelasyon katsayısı iki vektör arasındaki ilişkinin gücünü göstermekte iken katsayının önünde bulunan işaret ise ilişkinin yönünü göstermektedir. Bu nedenle *1* sayısına yakın değerler güçlü eksik değer hesaplama yeteneğini göstermektedir. Korelasyon katsayısı Denklem 6.5'deki gibi hesaplanmaktadır. Denklemdeki $x_i$ gerçek değerleri, $mx_i$ gerçek değerlerin aritmetik ortalamasını, $\overline{x}_i$ hesaplanmış eksik değerleri, $m\overline{x}_i$ ise hesaplanmış eksik değerlerin aritmetik ortalamasını göstermektedir.

$$r = \frac{\sum_{i=1}^{n}(x_i - mx_i)(\overline{x}_i - m\overline{x}_i)}{[\sum_{i=1}^{n}(x_i - mx_i)^2 \sum_{i=1}^{n}(\overline{x}_i - m\overline{x}_i)^2]^{1/2}} \qquad \text{Denklem 6.5}$$

## 6.5. EykYsa Araştırma Sonuçları

Şekil 6.10'da Powerplant veri kümesinde yapay olarak eksiltilmiş *20* adet eksik değerin önerilen en yakın k-komşu, yapay sinir ağları (EykYsa) yaklaşımı ve yapay sinir ağları, genetik algoritmalar (YsaGa) yaklaşımı tarafından hesaplanarak bulunmuş olan değerlerinin birebir karşılaştırılması gösterilmiştir.




**Şekil 6.10.** Powerplant veri kümesindeki gerçek ve hesaplanan değerlerin kıyaslaması

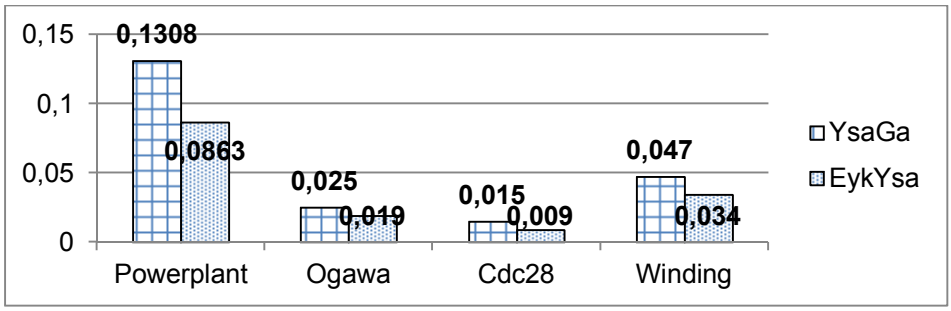
Çizelge 6.3'de grafiksel olarak gösterilen bu değerler çizelge halinde sunulmuştur buna göre önerilen yaklaşımın kıyaslanan diğer yöntem YsaGa'ya göre gerçek değerlere daha yakın hesaplamalar yaptığı görülmüştür. Sonuçların oluşmasına neden olan etkenlere bakıldığında sıra numarası *1, 4, 6, 9, 11, 14, 16, 19* olan eksik

değerlerde daha düşük hesaplama başarısı elde edilmiştir. Bunun nedeni bu eksik değerlerin gerçekte aykırı, tutarsız, gürültülü olabildiğinden veya yapay sinir ağı eğitiminin yetersiz kalabildiğinden kaynaklanmıştır. Diğer bir neden ise optimizasyon için kullanılan genetik algoritmaların tüm çözüm uzayını aramadan en iyi çözümü bulduğunu varsaymasından kaynaklanmıştır. Genetik algoritmalar yüksek derecede uygunluk fonksiyonuna uyan fakat en uygun çözüm olmayan genlerin popülasyon havuzunda baskın olmasına izin verebilen yerel minimum problemine maruz kalabilmektedir (Goldberg, 1989). Öte yandan önerilen en yakın k-komşu algoritması ve yapay sinir ağları yöntemi, en benzer *k* adet komşu sayısını kullanarak, tahmin için ağırlıklı bir ortalama değer üretip gürültü azaltıcı bir etki göstermiştir. Şekil 6.10'da görülebileceği üzere sıra numarası *1, 4, 6, 9, 11, 14, 16 ve 19* olan eksik değerlerde kullanılan ağırlıklı ortalama değeri tam olarak belli olmayan aykırı, tutarsız, gürültülü verilerde ve aynı zamanda yapay sinir ağlarının eğitiminden kaynaklanan yetersiz tahminlerde daha yumuşak bir hesaplama kabiliyeti sunmuştur.

**Çizelge 6.3.** Gerçek değer ile hesap edilen değerler

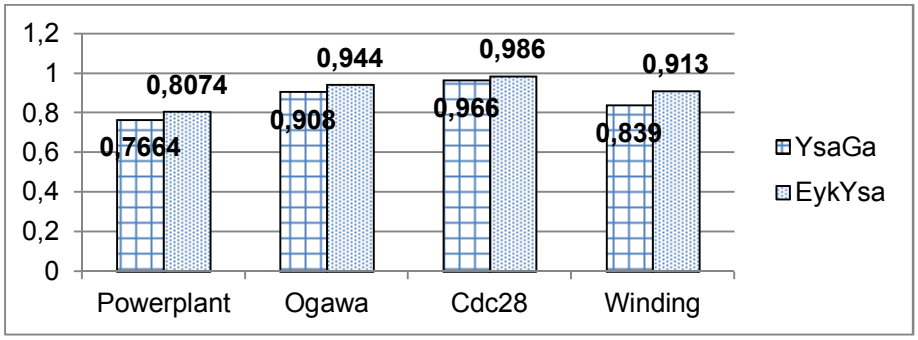
| Sıra | Gerçek | YsaGa | EykYsa (Önerilen) | Sıra | Gerçek | YsaGa | EykYsa (Önerilen) |
|---|---|---|---|---|---|---|---|
| 1 | 0.006 | -0.120 | 0.092 | 11 | 0.109 | -0.108 | 0.093 |
| 2 | 0.062 | 0.368 | 0.367 | 12 | 0.082 | 0.482 | 0.475 |
| 3 | 0.121 | 0.250 | 0.250 | 13 | 0.123 | 0.229 | 0.229 |
| 4 | 0.627 | 0.712 | 0.626 | 14 | 0.624 | 0.651 | 0.625 |
| 5 | 0.077 | 0.090 | 0.090 | 15 | 0.124 | 0.099 | 0.099 |
| 6 | 0.118 | -0.178 | 0.093 | 16 | 0.103 | -0.160 | 0.093 |
| 7 | 0.085 | 0.471 | 0.471 | 17 | 0.083 | 0.464 | 0.446 |
| 8 | 0.159 | 0.234 | 0.233 | 18 | 0.121 | 0.244 | 0.244 |
| 9 | 0.624 | 0.666 | 0.625 | 19 | 0.622 | 0.645 | 0.626 |
| 10 | 0.070 | 0.092 | 0.092 | 20 | 0.072 | 0.100 | 0.100 |

Şekil 6.11'da *4* veri kümesi için hata karelerinin ortalamasının karekökü (HKOK) karşılaştırması gösterilmiştir. HKOK eksik değer hesaplama hatasını ölçmektedir. Daha düşük bir hata değeri daha iyi bir performans ölçümü anlamına gelmektedir.



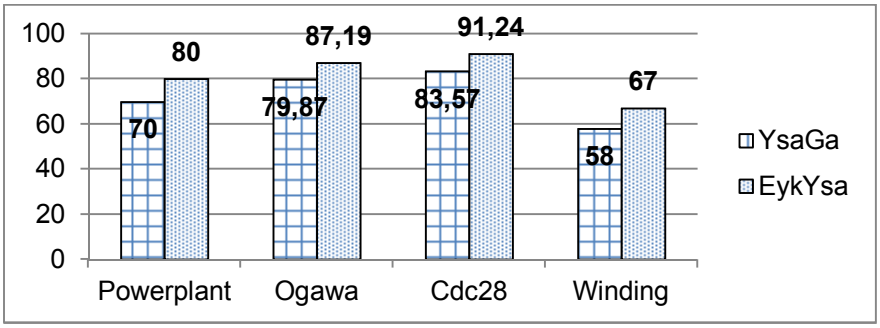

**Şekil 6.11.** Dört veri kümesindeki hata karelerinin ortalamasının karekökü kıyaslaması

Şekil 6.12'de *2* yöntemin *4* veri kümesi üzerindeki hesaplama sonuçlarının gerçek değerlerle olan korelasyon benzerliği katsayısı (r) gösterilmektedir. Korelasyon katsayısı gerçek ile hesap edilmiş değerler vektörünün birbirine olan benzerliğini ölçmektedir. Yüksek değerli korelasyon daha iyi performans ölçütü olarak kabul edilmektedir.




**Şekil 6.12**. İki yöntemin dört veri kümesi üzerindeki korelasyon katsayı kıyaslaması

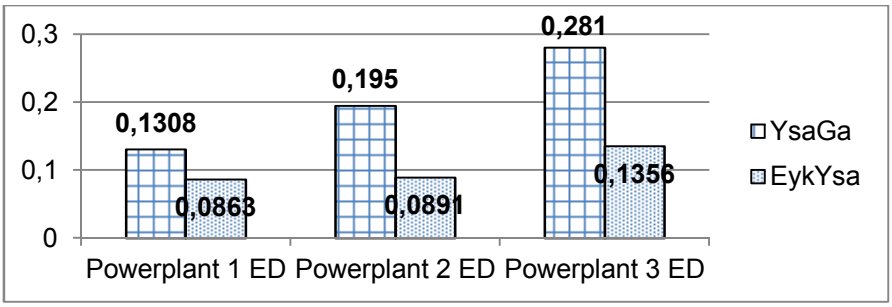
Şekil 6.13'de göreceli tahmin doğruluğu (D) yüzdeleri gösterilmiştir. Gerçek değere *%10* tolerans aralığı içerisinde yapılan başarılı hesaplama adetlerinin tüm eksik değerler adedi içerisindeki yüzde oranı olarak bulunan göreceli tahmin doğruluğunun yüksek değerli olması daha iyi sonuçlar elde edildiğini göstermektedir.




**Şekil 6.13**. İki yöntemin dört veri kümesi üzerindeki göreceli tahmin doğruluğu yüzdesi kıyaslaması

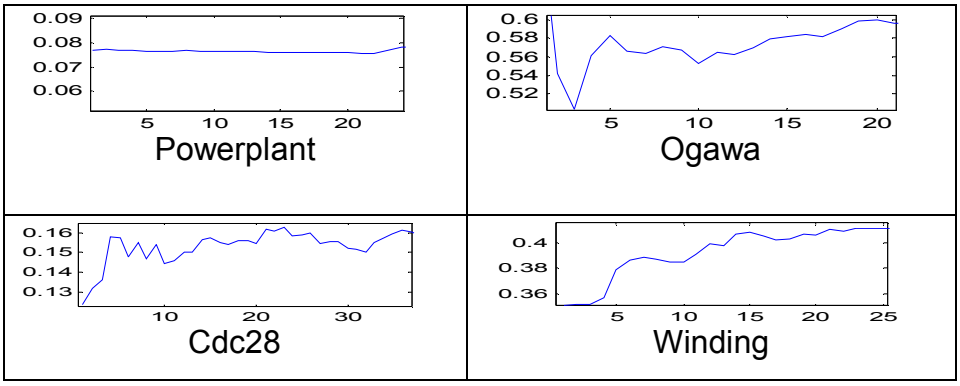
Sonuçların elde edildiği Şekil 6.11, Şekil 6.12 ve Şekil 6.13'de görüldüğü üzere Ogawa, Cdc28 mikro dizi veri kümelerinin endüstriyel zaman serisi veri kümeleri olan Powerplant ve Winding'e göre daha iyi sonuçlar elde edilmiştir. Bu çalışmada açıkça mikro dizi veri kümelerinin yapay sinir ağları eğitim süreci için daha uygun olduğu görülmüştür. En yakın k-komşu çalışma sonuçları veri kümesinde ilgisiz veya gereksiz nitelikler olup olmaması konusunda çok hassastır. Çünkü var olan bütün nitelikler benzerlik ölçütünün hesap edilmesinde kullanılmakta bu yüzden veri kümesindeki nitelikler eksik değer hesaplanması başarısını etkilemektedir. Hesaplama başarısı uygun özellik seçimi veya ağırlıklı nitelik kullanımı ile çok daha iyi hale getirilmektedir (Cunningham ve Delany, 2007). Genetik algoritmalar ve yapay sinir ağları güçlü, çözüm araması yapabilen algoritmalardır, öğrenebilmekte ve yüksek derece doğrusal olmayan veri modelleriyle başa çıkabilmektedirler. Bunu başarmak için amaç fonksiyonu dışında diğer ekstra karmaşık fonksiyonlara ihtiyaç duymamaktadırlar. Fakat bu esnek yapıları aynı zamanda sorunlara da neden olabilmektedir. Bu nedenle genetik algoritmalar ve yapay sinir ağ yapıları özenli oluşturulmalı, kodlanmalı ve kullanılacak veri kümelerine uygun hale getirilip, özelleştirilmelidir (Busetti, 2001). En yakın k-komşu algoritmasının avantajı veri kümesindeki çoklu sınıf yapısına uygun olması ve sınıf kararını az sayıdaki benzer komşu kayıtlara bakarak verebilmesidir. Veri kümesi nitelikleri birbirinden bağımsız değişik alt veri kümelerinden gelmiş olsa bile yeterli seviyede sınıflama başarısı elde edilebilmektedir. Sadece en yakın k-komşu algoritmasıyla çalışan bazı gürültü azaltıcı teknikler bulunmaktadır. Bu nedenle en yakın k-komşu yöntemi eksik değer hesaplama başarısını artırmada etkili bir rol oynayabilmektedir (Cunningham ve Delany, 2007).

Şekil 6.14'de Powerplant veri kümesindeki bazı kayıtların birden fazla eksik değer içermesi durumunda önerilen yaklaşımın çalışma sonuçlarını göstermektedir. Sonuçlar bir kayıtta *1, 2* veya *3* değerin aynı anda eksik olması durumunda önerilen en yakın k-komşu, yapay sinir ağları modelinin kıyaslanan yapay sinir ağları genetik algoritmalar yaklaşımına göre daha düşük hata karelerinin ortalamasının karekökü değerine sahip olduğunu göstermektedir. Yani bir kayıtta birden çok eksik değer olması durumunda bile önerilen yaklaşım EykYsa'nın başarılı sonuçlar ürettiği görülmüştür.




**Şekil 6.14**. Powerplant veri kümesinin bir kaydından birden fazla eksik değer olma durumundaki hata sonuçları

Ayrıca önerilen yaklaşımın özgün, yenilik getiren özelliklerine Şekil 6.15 üzerinden değinilmiştir. Görüldüğü üzere değişen komşu *(k)* adedine göre tüm veri kümelerinde hesaplanan eksik değerlerin hata oranlarda farklılıklar olmaktadır. Klasik temel en yakın k-komşu eksik değer hesaplama algoritması her bir veri kümesi için önceden sabit belirlenmiş bir komşu *(k)* adedi ile eksik değer tahmini yapmaktadır. Fakat komşu adedi veri kümesinin özel durumuna göre yani kayıt sayısı, nitelik sayısı, nitelik tipleri, zaman seri olup olmama gibi durumlarına göre farklı ve özel seçilmesi gerekmektedir.




**Şekil 6.15**. Klasik en yakın k-komşu algoritmasının k adedine (*x ekseni*) karşılık hata oranı (*y ekseni*) değişimleri

Yapay sinir ağları bazı aykırı değerler üzerinde hatalı sonuçlar üretmekte ve genetik algoritmalar ise yerel minimum problemi ile tüm çözüm havuzunu aramadan en iyiyi bulduğunu düşünebilmektedir. En yakın k-komşu algoritması ağırlıklı en benzer kayıtlarından yola çıkarak bu olumsuzları ortadan kaldırmakta ve veri kümesine uygun *(k)* komşu adedi belirleyerek daha tutarlı hesaplamalar yapılmasına olanak tanımaktadır. Bunun yanı sıra önerilen yaklaşımın bazı eksiklikleri bulunmaktadır. Yapay sinir ağları yapısının modelinin oluşturulması oldukça zor olabilmektedir. Yapay sinir ağ tipi ve

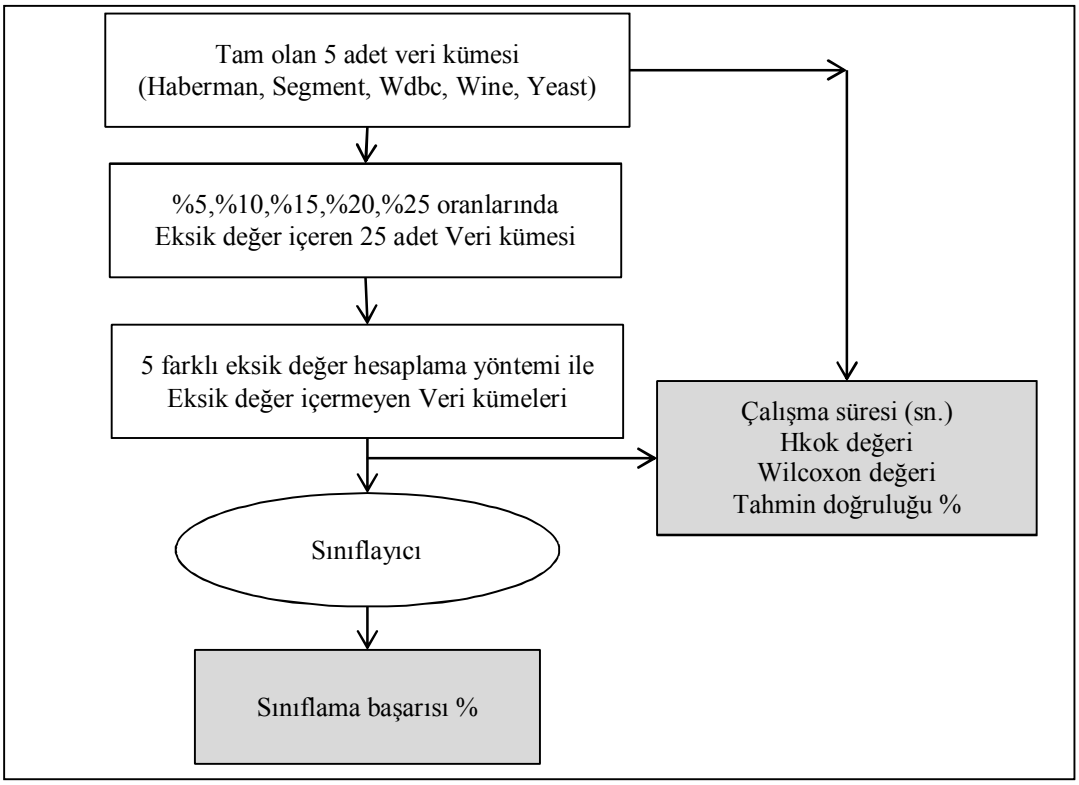
hangi performans sonlanma kriterine göre eğitilmesi gerektiği eksik değer hesaplaması yapılmadan önce ayrıntılı ve detaylı bir şekilde belirlenmesi gerekmektedir. Hesaplama süresinin klasik eksik değer tamamlama yöntemlerine göre uzun sürmesi önerilen yaklaşımın bir diğer eksikliği olarak kabul edilmektedir. Gelecekte yapılacak çalışmalara öneri olarak sonuçların daha hassas ve başarılı elde edilmesinin istendiği veri kümelerinde her bir eksik değere karşılık farklı bir komşu *(k)* adedinin belirlenmesi olarak düşünülmektedir. Bu durum çalışma süresini daha fazla uzatmasına rağmen performans kriteri üzerinde başarılı, olumlu sonuçlar doğuracağı öngörülmektedir. Ayrıca gelecekte yapılacak çalışma süresi performans iyileştirmesi olarak temel bileşenler analizi gibi özellik seçim yöntemlerinin yapay sinir ağları eğitiminden önce veri kümesi niteliklerine uygulanmasıdır. Bir diğer iyileştirme olarak ise yapay sinir ağı yapısını dinamik olarak güncelleyen, değiştiren algoritmaların önerilen yaklaşım ile birlikte kullanılmasıyla elde edilecek çalışmalar olarak öngörülmektedir.

### 6.6. Eksik Değer Hesaplamasını Etkileyen Özellikler

Yöntemlerin genel olarak değişik özelliklerdeki veri kümeleri üzerinde nasıl çalıştığı ve veri kümeleri özelliklerinin sonuçlara nasıl etki ettiğini araştırmaya yönelik tezde bahsedilen *5* farklı yöntem ile *5* değişik karakteristik özellik gösteren veri kümesi üzerinde Şekil 6.16'da gösterildiği gibi araştırmalar yapılmıştır.

Şekil 6.16'da görüldüğü üzere yaygın olarak kullanılan *5* farklı özelliğe sahip veri kümeleri araştırma için tercih edilmiştir. Haberman veri kümesi *306* kayıt ve *4* nitelikten oluşmaktadır. Bu veri seti meme kanseri nedeniyle ameliyat geçirmiş hastaların sağ kalıp kalmaması üzerine Chicago Billings Hastanesi Üniversitesi'nde 1958 ve 1970 yılları arasında yürütülen çalışma sonuçlarını göstermektedir. Hastanın beş yıl içinde hayatta olup olmadığını kayıt altında tutmaktadır. Segment veri kümesi *2310* kayıt ve *20* nitelik içermektedir. *7* açık hava resmini 3x3 pixel boyutlu olarak sınıflayarak her bölgenin yüzey türünü tespit etmekte kullanılmaktadır. Wdbc veri kümesi *569* kayıt ve *31* nitelikten oluşmakta ve meme kanserinde tespit edilen tümörün iyi ya da kötü huylu olup olmadığını sınıflamaktadır. Wine veri kümesi ise *178* kayıt ve *14* nitelikten meydana gelmekte ve İtalya'da aynı bölgede yetişen üç farklı şarap türünü on dört farklı niteliği ile sınıflamaya çalışmaktadır. Yeast ise *1484* kayıt ve *9* nitelikten oluşmakta ve maya hücreleri hakkında bilgiler içererek hücrenin hangi bölgede olduğunu tespit etmeyi amaçlamaktadır.

Bu veri kümelerinin her biri yapay olarak *%5, %10, %15, %20 ve %25* oranlarında eksik değer içeren veri kümelerine dönüştürülerek toplam *25* adet eksik değer içeren veri kümesi elde edilmiştir. Eksik değer içeren her bir veri kümesi tezde bahsedilmiş *5* farklı eksik değer hesaplaması yapan yöntem olan Bulanık c-ortalamaları, destek vektör regresyonu ve genetik algoritmalar (BcoDvrGa), Bulanık c-ortalamaları ve genetik algoritmalar (BcoGa), Destek vektör regresyonu ve genetik algoritmalar (DvrGa), Yapay sinir ağları ve genetik algoritmalar (YsaGa), En yakın k-komşu ve yapay sinir ağları (EykYsa) yaklaşımları ile hesaplanarak sonuçlar üzerinde değerlendirmeler yapılmıştır.




**Şekil 6.16.** Beş veri kümesinin beş farklı yöntemle eksik değer hesaplamasının kıyaslanması

Çizelge 6.4'de Haberman veri kümesi için hata karelerinin ortalamasının karekökü değerleri çizelge olarak görülmektedir. Grafiksel çizimi ise Şekil 6.17'deki gibi verilmiştir. Çizelgede kalın olarak gösterilen sayılar yöntem bazında en düşük hata elde edilmiş eksik yüzde oranını, eğik olarak gösterilen sayılar ise eksiklik yüzde oranına göre en düşük hata değeri elde edilen yöntemi göstermektedir.

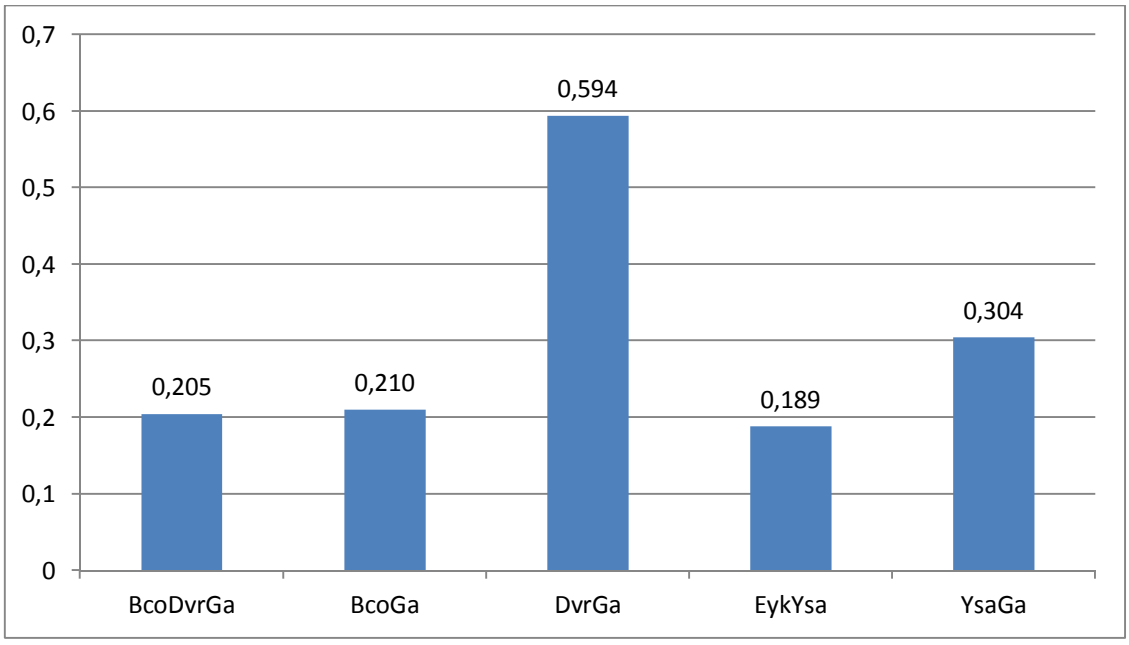
**Çizelge 6.4.** Haberman veri kümesi hata karelerinin ortalamasının karekökü (HKOK) sonuçları

| Yöntem/ Yüzde | %5 | %10 | %15 | %20 | %25 | En Düşük Yöntem için |
|---|---|---|---|---|---|---|
| BcoDvrGa | **0.0330345** | *0.0377793* | 0.0492397 | 0.0424476 | 0.0420740 | **0.0330345** |
| BcoGa | **0.0368129** | 0.0382693 | 0.0499382 | 0.0433462 | 0.0416842 | **0.0368129** |
| DvrGa | 0.2104778 | 0.1398456 | 0.0945349 | **0.0706278** | 0.0781833 | **0.0706278** |
| EykYsa | *0.0330215* | 0.0386291 | *0.0393646* | *0.0395236* | *0.0380155* | **0.0330215** |
| YsaGa | 0.0601967 | 0.0583911 | 0.0877459 | 0.0514424 | **0.0466423** | **0.0466423** |
| En Düşük Yüzde | *0.0330215* | *0.0377793* | *0.0393646* | *0.0395236* | *0.0380155* | |

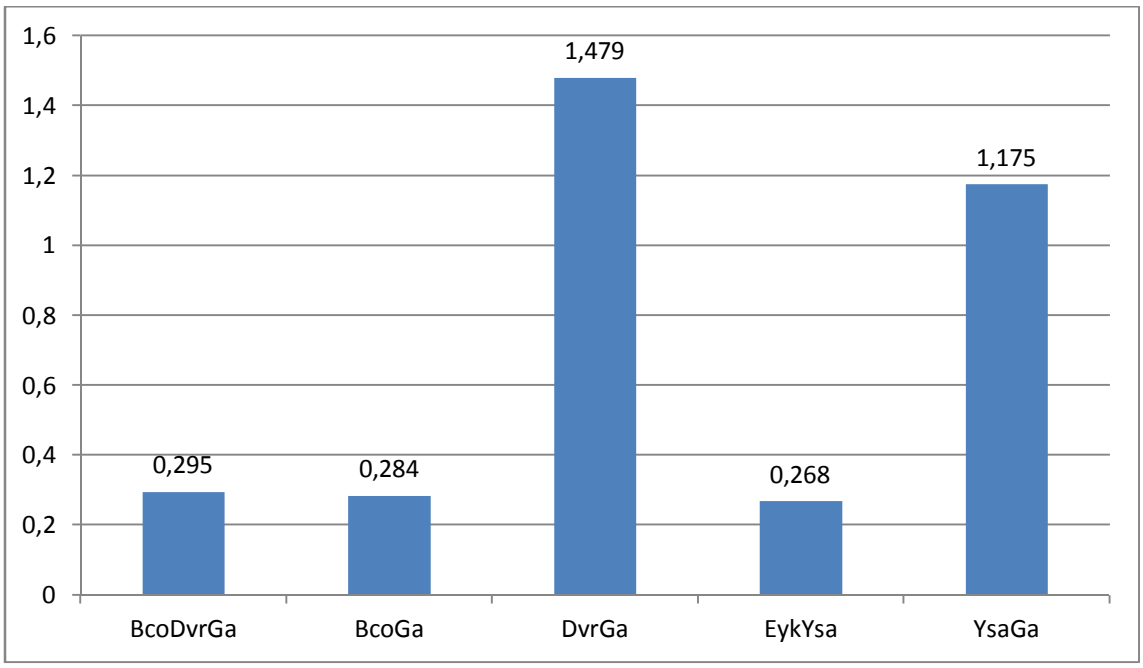
Buna göre *%5* eksik olma oranında *3* yöntem BcoDvrGa, BcoGa ve EykYsa yöntemlerinin en az hata oranına sahip olduğu görülmüştür. *%5* oranında hem yöntem hem de eksiklik yüzde oran olarak ikisine birden en iyi performansın EykYsa yöntemiyle elde edilmiştir. Ayrıca *%10* ve *%15* eksik olma durumunda hiçbir algoritma diğerleri üstün başarı elde edememiştir. Buna ilaveten EykYsa yönteminin *%5, %15, %20 ve %25* oranlarında hepsinde birden başarılı olduğu bunun aksine BcoGa, DvrGa ve YsaGa algoritmalarının yüzde oranların hiçbirinde diğer algoritmalardan daha iyi performans üretemediğini gözlemlenmiştir.

Haberman veri kümesi için toplam hata sonuçları Şekil 6.17'de gösterilmiştir. Buna göre yöntem bazında yüzde değerleri toplandığında Haberman veri kümesi için en iyi yöntemin EykYsa olduğu söylenebilir.

Şekil 6.18'de ve Çizelge 6.4'de Haberman veri kümesi için elde edilen hata karelerinin ortalamasının karekökü değerleri diğer *4* veri kümesi içinde elde edilmiş ve tümü yöntem bazında toplanarak yöntemlerin genel HKOK sonuçları topluca gösterilmiştir. Buna göre her bir çubuk grafik *5* veri kümesinin *5* farklı yüzde oranlardaki eksik değer sonuçlarını yani toplamda *25* adet HKOK değerinin toplanması ile elde edilmiştir. Grafiğe göre toplamda DvrGa ve YsaGa yüksek derecede hata değerine sahipken BcoDvrGa, BcoGa, EykYsa gibi yaklaşımlar daha iyi performansa sahip olmuştur.

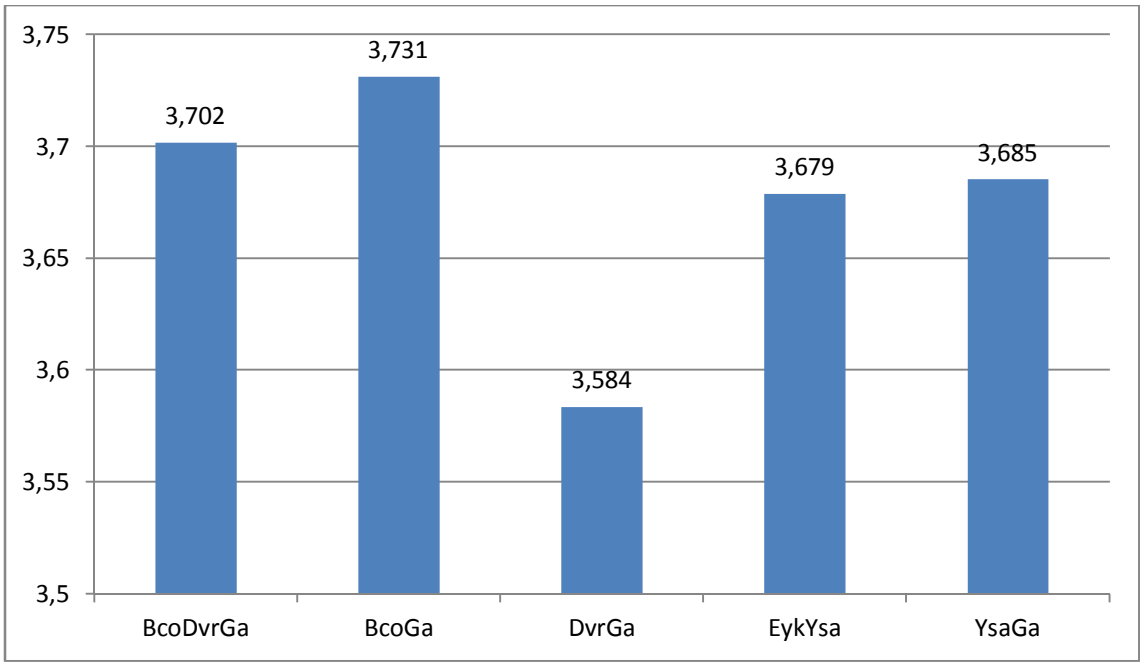



**Şekil 6.17.** Haberman veri kümesi yöntem bazında HKOK sonuçları grafiksel gösterimi




**Şekil 6.18.** Tüm yöntemlerin HKOK sonuçları toplamının grafiksel gösterimi

Şekil 6.19'da Haberman veri kümesi için toplam sınıflama başarısı grafiksel olarak gösterilmiştir.




**Şekil 6.19**. Haberman veri kümesi yöntem bazında toplam sınıflama başarısı

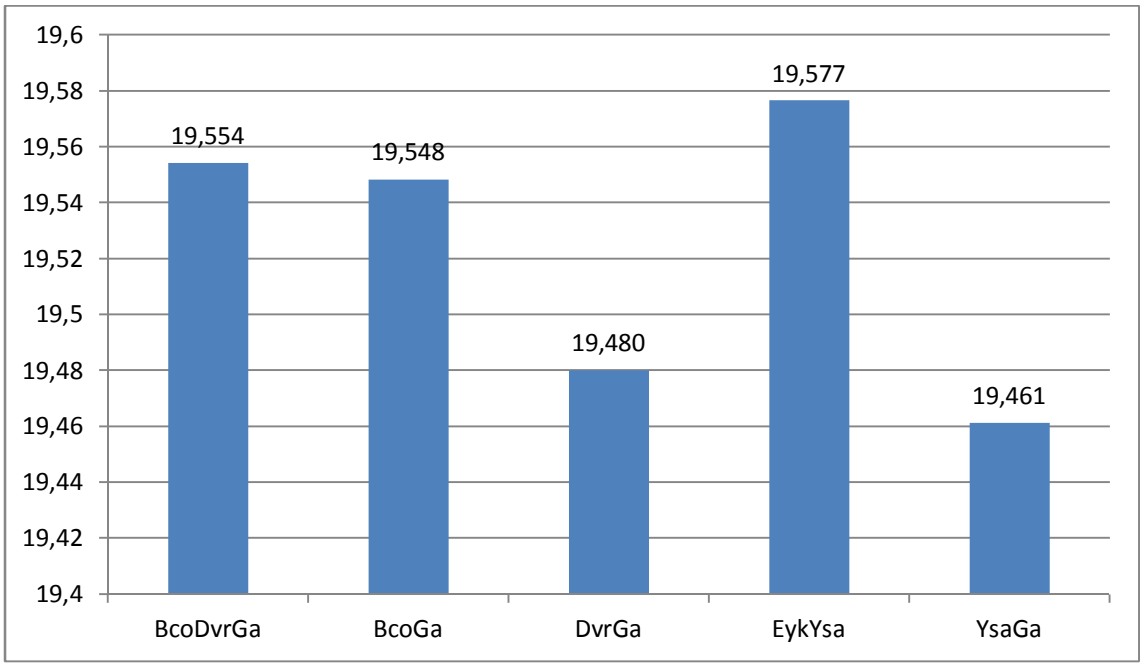
Ayrıca Çizelge 6.5'de sınıflama başarıları çizelge olarak gösterilmiştir. Bunun için yapay olarak eksik değer oluşturulan Haberman veri kümesinin *5* farklı yöntem ile eksik değerleri hesaplanmış ve sonra *10* kez en yakın *3*-komşu algoritmasına göre sınıflaması yapılmıştır. On katlı çapraz doğrulama sınıflama başarısı ölçütüne göre sınıflama başarısı sonuçları elde edilmiş ve değerlendirilmiştir. On-kat çapraz doğrulamada veri kümesi rastgele on eşit parçaya bölünerek her seferinde *1* parça test, geri kalan *9* parça eğitim verisi olarak kullanılmakta ve sınıflama yapılmaktadır. Bu işlem tüm parçalar test parçası olarak kullanılıncaya kadar devam ettirilerek bitirilmektedir. Elde edilen toplam doğru sınıflama oranı, on-kat çapraz doğrulama sonucu olarak kabul edilmektedir (Breiman ve ark., 1984, Kohavi, 1995).

**Çizelge 6.5.** Haberman veri kümesi sınıflama başarısı

| Yöntem/ Yüzde | %5 | %10 | %15 | %20 | %25 | En Yüksek Yöntem için |
|---|---|---|---|---|---|---|
| BcoDvrGa | 0.7377049 | 0.7311475 | 0.7213114 | **0.7606557** | 0.7508196 | **0.7606557** |
| BcoGa | *0.7442622* | 0.7213114 | *0.7442622* | *0.7606557* | *0.7606557* | **0.7606557** |
| DvrGa | 0.6950819 | 0.7278688 | 0.6950819 | 0.7213114 | **0.7442622** | **0.7442622** |
| EykYsa | **0.7442622** | 0.7245901 | 0.7311475 | 0.7409836 | 0.7377049 | **0.7442622** |
| YsaGa | 0.7344262 | *0.7475409* | 0.7180327 | 0.7344262 | **0.7508196** | **0.7508196** |
| En Yüksek Yüzde | *0.7442622* | *0.7475409* | *0.7442622* | *0.7606557* | *0.7606557* | |

Orijinal Haberman veri kümesinin sınıflama başarısı *0.7179*'dır. Eksik değerlerin hesaplaması sonucu elde edilmiş yeni veri kümelerinin genel başarısı artmış ve başarı ortalaması *0.7352* olarak bulunmuştur. BcoGa yöntemi *4* adet yüzde eksik değer *%5, %15, %20 ve %25* oranlarında sınıflama başarısı yüksek en iyi yöntem olmuştur. BcoDvrGa, DvrGa ve EykYsa yöntemlerinin diğer yöntemlere göre en yüksek sınıflama değeri elde edemediği gözlemlenmiştir. *%20* ve *%25* eksiklik oranlarında diğer oranlara göre daha fazla yöntemin en yüksek başarıya ulaştığı görülmüştür. Bunun nedeni yapay olarak silinmiş olan eksik değerlerin yerine veri kümesi yapısına daha uygun değerlerin hesaplanarak sınıflama başarısına olumlu etki yapmış olmasıdır.

Şekil 6.20'de tüm *5* adet veri kümesinin *5* eksik değer hesaplama yöntemine göre en yakın *3*-komşu algoritması sınıflama başarılarının toplamları gösterilmektedir. Buna göre BcoDvrGa, BcoGa ve EykYsa metotlarının toplam sınıflama başarısını artırmada daha iyi olduğu görülmüştür.



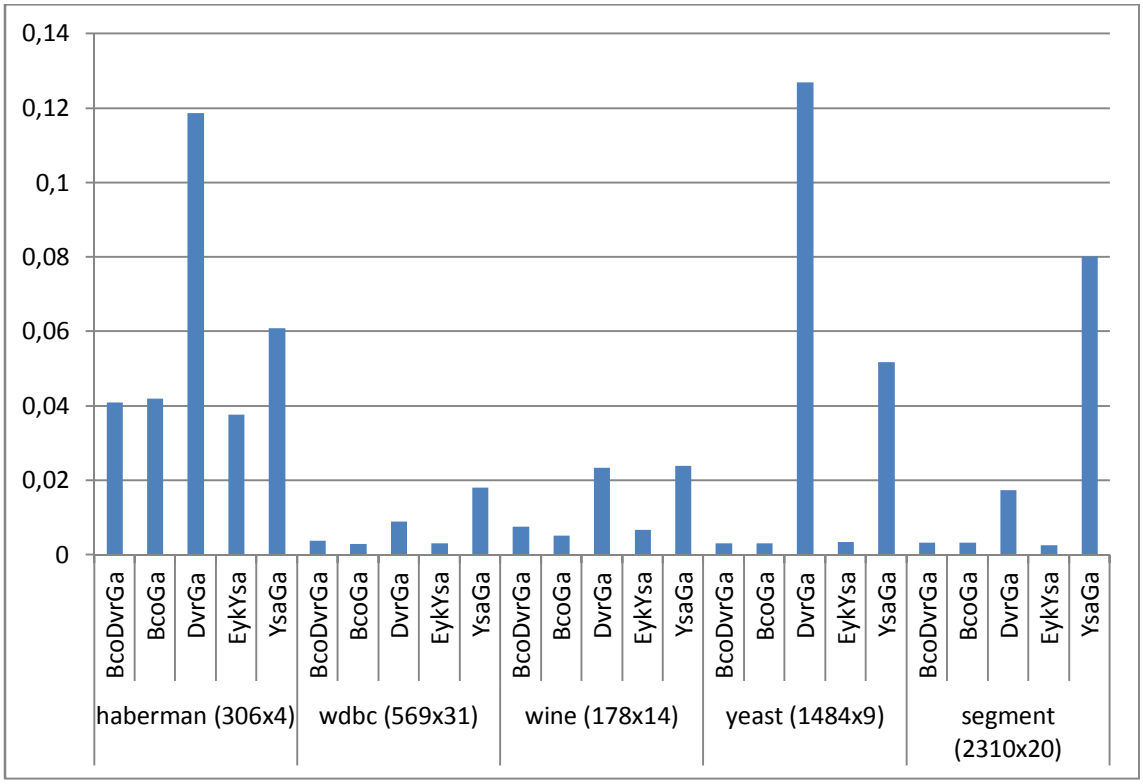

**Şekil 6.20**. Tüm veri kümelerinin sınıflama başarısı toplamı

Eksik değerleri hesaplayan algoritmaların performanslarının ölçülebilmesi için genellikle ortalama toplam hata kullanılmaktadır. Bunun için çalışmalardan önce genellikle yapay olarak eksik değer oluşturulmaktadır. Fakat yapay olarak eksik değer oluşturulmadan eksik değerlerin ne kadar başarılı hesaplandığının ölçümü sınıflama

başarı yüzdesi veya kümeleme geçerlilik indeksi kullanılarak yapılabilmektedir (Sun ve ark., 2009; Celton ve ark., 2010; Nandhini ve Sakthivel, 2012).

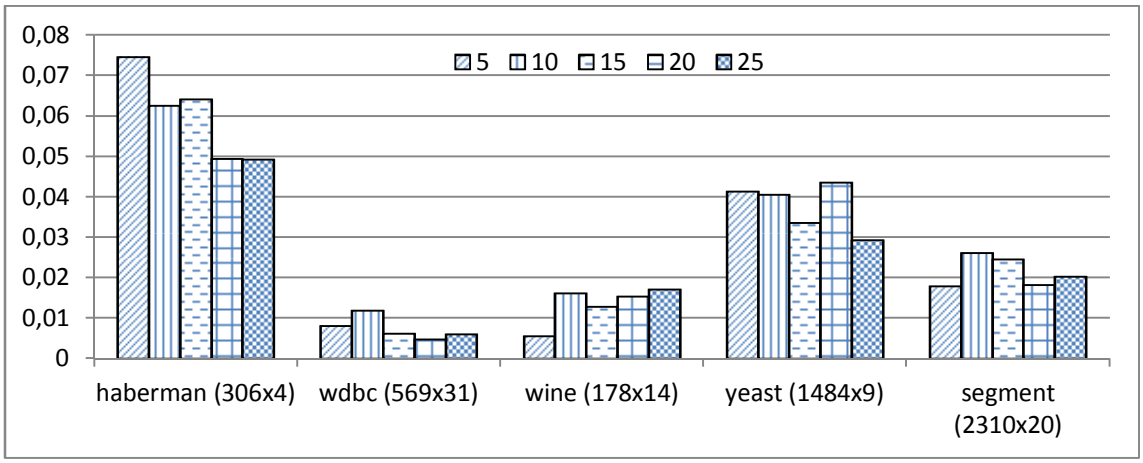
Özellikle sınıflama veya kümeleme başarısının yüksek olmasının istendiği çalışmalarda veri kümelerinde bulunan gürültülü veya aykırı değerler uygun algoritmalarla tespit edilip, silinebilmektedir. Bu şekilde silinmiş yani eksik değerler sınıflama başarısı veya kümeleme geçerlilik indeksini artıracak şekilde yeniden hesaplanarak yapılan çalışmanın sınıflama veya kümeleme başarısı artırılabilmektedir.

Veri kümesinde bulunan kayıt sayısı, nitelik sayısı veya eksik değer adedinin elde edilen sonuçlar üzerindeki etkilerini incelemek amacıyla veri kümesi bazında sonuçlar incelenmiştir. Şekil 6.21'de *5* yöntemin *5* farklı oranlardaki eksik değerli veri kümesi üzerindeki ortalama HKOK sonuçları gösterilmektedir. Her bir çubuk grafik ilgili veri kümesi ile ilgili elde edilmiş *%5, %10, %15, %20 ve %25* HKOK hata değerinin ortalamasını göstermektedir. Haberman veri kümesi *306* kayıt ve *4* adet nitelik içermektedir. Haberman veri kümesiyle elde edilen ortalama hata miktarı diğer *569* kayıt ve *31* niteliğe sahip Wdbc, *178* kayıt ve *14* niteliğe sahip Wine, 1484 ve 9 niteliğe sahip Yeast ile *2310* kayıt ve *20* niteliğe sahip Segment veri kümelerine göre daha fazla çıkmıştır.



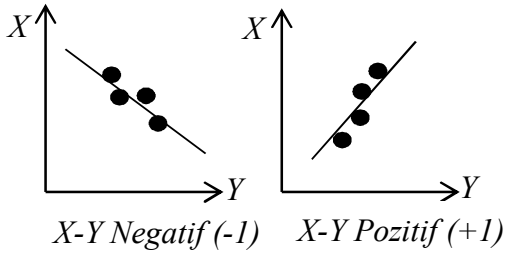

**Şekil 6.21**. Veri kümeleri bazında ortalama HKOK sonuçları

Şekil 6.22'de ise *5* yöntemin *5* farklı oranlardaki eksik değerli veri kümesi ile elde edilen ortalama HKOK değerleri yüzde eksik oran bazında gösterilmiştir. Eksik yüzde oranı artıkça Haberman veri kümesinde elde edilen HKOK hata değerinin azaldığı görülmüştür.




**Şekil 6.22**. Veri kümelerinin eksik oran bazında ortalama HKOK sonuçları

Bu çalışmada kullanılan veri kümesi ve kıyaslama ölçütleri arasındaki doğrusal ilişki ölçüsü olan korelasyon katsayı değerleri Çizelge 6.6 ile korelasyon matrisi şeklinde gösterilmiştir. Elde edilen toplu sonuçlar yardımıyla veri kümesi özelliğinin sonuçlar üzerindeki etkileri açıklanmıştır. Korelasyon matrisinde bulunan katsayı *-1* ile *+1* arasında bir ondalık değer almaktadır. Artı değerli bir korelasyon katsayısı değeri *pozitif korelasyon* olarak adlandırılmaktadır. Yani iki değişken doğru orantılı olarak biri artarken ya da azalırken birlikte hareket ettikleri anlamına gelmektedir. *Negatif korelasyon* katsayı değeri ise iki değişken arasında ters orantı olduğu anlamına gelmektedir. Yani bir değişken artarken diğeri azalmakta veya tam tersi biri azalırken diğerinin artmakta olduğu anlamına gelmektedir (Şekil 6.23).




**Şekil 6.23.** Doğrusal negatif ve pozitif korelasyon gösterimi

Mutlak değerli korelasyon katsayı değeri ise iki değişken arasındaki ilişkinin gücünü yansıtmaktadır. Korelasyon katsayı değerinin *-1* veya *+1*'e yaklaştıkça iki değişken arasındaki pozitif veya negatif ilişki gücünün de arttığı, uzaklaştıkça ise ilişki gücünün zayıfladığı söylenmektedir.

**Çizelge 6.6.** Performans ölçütleri ve veri kümesi özellikleri arasındaki korelasyon matrisi

| | Eksik oranı% | Çalışma süresi (sn.) | Hkok değeri | Wilcoxon değeri | Tahmin doğruluğu % | Sınıflama başarısı% | Kayıt Sayısı | Nitelik Sayısı | Eksik Sayısı |
|---|---|---|---|---|---|---|---|---|---|
| Eksik oranı% | 1 | | | | | | | | |
| Çalışma süresi (sn.) | 0.116 | 1 | | | | | | | |
| Hkok değeri | -0.058 | -0.163 | 1 | | | | | | |
| Wilcoxon değeri | -0.027 | -0.051 | -0.505 | 1 | | | | | |
| Tahmin doğruluğu % | 0.060 | 0.219 | -0.858 | 0.320 | 1 | | | | |
| Sınıflama başarısı% | 0.012 | 0.219 | -0.267 | 0.016 | 0.266 | 1 | | | |
| Kayıt Sayısı | 0 | 0.473 | -0.034 | -0.400 | 0.290 | 0.201 | 1 | | |
| Nitelik Sayısı | 0 | 0.109 | -0.434 | 0.146 | 0.545 | 0.744 | 0.139 | 1 | |
| Eksik Sayısı | 0.454 | 0.486 | -0.049 | -0.390 | 0.250 | 0.160 | 0.805 | 0.112 | 1 |

Çizelgeye göre eksik değer hesaplama başarısı ölçümünde kullanılan kıyaslama ölçütleri birbirleri ve veri kümesinin kayıt, nitelik ve eksik adedi ile olan ilişkileri korelasyon katsayısı ile tespit edilmiştir. Çizelgedeki güçlü korelasyon ilişkilerinden bazıları şu şekilde yorumlanmıştır. Çalışma süresi (sn.), kayıt sayısı ve eksik sayısı ile doğrusal pozitif ilişki içermektedir. HKOK hesaplama hatası Wilcoxon değeri, tahmin doğruluğu, sınıflama başarısı ve nitelik adedi ile negatif bir ilişkisi olduğu görülmektedir. Yani nitelik sayısının artmasıyla HKOK değerinin düştüğü aralarındaki "*-0.434"* korelasyon katsayısı yardımıyla söylenmektedir. Yine Wilcoxon değerinin kayıt adedi ve eksik adedi ile negatif korelasyon ilişkisi olduğu görülmektedir. Kayıt adedi veya eksik adedi azaldığında Wilcoxon değeri yükselmektedir. Tahmin doğruluğu

yüzdesinin ise veri kümesinde bulunan nitelik sayısıyla birlikte arttığı veya azaldığı görülmektedir. Tüm bu çıkarımların yanında veri kümesinin kaliteli ve birbiriyle tutarlı kayıtlar içermesinin elde edilen başarılı sonuçlara katkıda bulunduğu düşünülmektedir (Sun ve ark., 2009).

# 7. SONUÇLAR VE ÖNERİLER

## 7.1. Sonuçlar

Bu tez çalışmasında veri kümesinde bulanan eksik değerleri hesaplamak üzere özgün hibrit çalışmalar yapılmış ve araştırmalar sonucunda diğer benzer yöntemlere göre önerilen yaklaşımların daha iyi ve tutarlı sonuçlar elde ettiği görülmüştür. Güvenilir bir makine öğrenmesi tekniği olan destek vektör regresyonu ve başarılı bir yapay zeka tekniği olan genetik algoritmaların desteği alınarak bulanık c-ortalamalar kümeleme metoduyla eksik değerler hesaplanmış ve sonuçta bütün kayıtları tam nitelik değerleri içeren veri kümesi oluşturulmuştur. Bunun sağlanabilmesi için önce eksik değer içermeyen tam kayıtlar birbirlerine olan benzerliklerine dayanılarak bulanık prensip ilkelerine göre gruplara ayrılmıştır. Her bir veri kümesi kaydı birden fazla küme merkezine üye hale getirilerek daha hassas eksik değer hesaplaması yapılması sağlanmıştır. Birbirinden farklı karakteristik özelliklere sahip literatürde yaygın kullanılan *6* adet veri kümesi deneysel araştırma sonuçlarının elde edilmesi için kullanılmıştır. Bulanık c-ortalamalar kümelemenin amaç fonksiyonu için gerekli olan küme merkezi adedi ve ağırlık faktörü parametre değeri o anda işlem gören veri kümesine uygun olarak optimize edilen otomatikleşmiş bir çalışma yapısıyla oluşturulmuştur. Önerilen hibrit BcoDvrGa'ın destek vektör regresyonu, genetik algoritmalar ve sıfır ile tamamlama yöntemleriyle karşılaştırıldığında daha iyi hesaplama başarısı elde eden bir yaklaşım olduğu görülmüştür. Deneysel araştırma sonuçları önerilen hibrit yaklaşımın kümelemeye uygun veri kümelerindeki eksik değer probleminin üstesinden gelmede tek başına yeterli bir yaklaşım olduğunu kanıtlamıştır. Bu konuda ilerde yapılacak çalışmalardan önce kümeleme geçerlilik analizleri ile veri kümelerinin kümeleme analizleri yapılarak veri kümesi hakkında kümelemeye uygunluğu ile ilgili ayrıntılı bilgiler edinilebileceği öngörülmüştür.

En yakın k-komşu algoritması, yapay sinir ağları ile işbirliği yaparak ağırlıklı ortalama değer ile eksik değer hesaplaması yapmıştır. En yakın k-komşu algoritmasının yapay sinir ağları ile birlikte hibrit kullanılması sonucu en yakın k- komşu eksik değer hesaplama algoritması önemli avantajlara sahip bir yöntem halini almıştır. Literatürde kıyaslanan benzer yöntemle birlikte yaygın kullanılan farklı özelliklere sahip *4* adet veri kümesi ile deneysel araştırmalar yapılmıştır. Karşılaştırılan yapay sinir ağları ve genetik

algoritmalar eksik değer hesaplama yöntemi ve önerilen en yakın k-komşu, yapay sinir ağları yaklaşımın her ikisinde de daha iyi sonuçların eksik olmayan tam verilerle eğitilmiş yapay sinir ağının eğitim performansına bağlı olduğu görülmüştür. Bu yüzden daha kaliteli, tutarlı eğitim verisine sahip veri kümelerinin eksik değerlerinin en yakın k-komşu algoritmasıyla daha hassas ve doğru başarıyla hesaplanabildiği ortaya çıkmıştır. Fakat genetik algoritmaların çözümü bulamadığı ya da gerçekte bulunması gereken eksik değer veri kümesinde hesaplanmasının zor olduğu aykırı, gürültülü veri olması durumunda önerilen yöntem en yakın benzer $k$ adet komşunun ağırlıklı ortalaması almıştır. Alınan ağırlıklı ortalama eksik değer tahminini daha yumuşak, kabul edilebilir bir değerde olmasına neden olmuştur. Ayrıca önerilen EykYsa yaklaşımın veri kümesinin tek bir kaydında birden çok eksik değer olma durumunda bile başarılı sonuçlar ürettiği görülmüştür. Veri kümesi kayıtlarında birden çok eksik değer olması durumunda bile araştırmacıların en yakın k-komşu yaklaşımını tercih etmeleri yapılan çalışmalara daha fazla avantaj sağlayacağı görülmüştür.

## 7.2. Öneriler

Elde edilen araştırma sonuçları ışığında, yapay sinir ağları ve destek vektör regresyonu algoritmalarının veri kümesini başarılı bir şekilde modelleyen giriş değerlerine karşılık anlamlı çıkış değerleri üreten yöntemler oldukları görülmüştür. Bu yöntemler veri kümesine uygun parametreler ile eğitilmesi ve eksik değer hesaplaması yapılan veri kümesinin niteliklerinin birbiriyle daha ilişkili ya da tutarlı olmasının sonuçlar üzerinde etkili olduğu görülmüştür. Bu ilişkileri tespit eden ve veri kümesi nitelikleri arasındaki korelasyonu ölçen algoritmalarla gereksiz, fazla niteliklerin veri kümesinden çıkarılması sağlanabildiği gibi nitelikler arasında ağırlıklı seçimler yapılarak veri kümesini daha iyi temsil eden niteliklerin daha baskın hale getirilmesi sağlanabilmektedir. Ayrıca yapay sinir ağı yapısını dinamik olarak güncelleyen algoritmalarla eğitim süresi ve performans değeri veri kümesi tipine uygun olarak çalışma anında iyileştirilebilmektedir.

Genel olarak eksik değer hesaplaması yapabilen algoritma veya yöntemlerin kıyaslanması sonucunda eksik değer hesaplamasında hangi yöntemin diğerine göre hangi şartlarda üstün geldiğinin açıklaması yapılabilmiştir. Fakat genel olarak tam manada tek bir yöntemin diğerlerine göre daha üstün olduğunun veya en iyisi olduğunun kanıtı yapılamamıştır. Bu nedenle bu tez çalışması konusu alanında halen

üzerinde araştırma ve çalışmalar yapılan popüler bir konu olmaya devam etmektedir. Veri kümesinin yapısını modelleyen nitelikler ve kayıtlar arasındaki ilişkileri çözmede etkili makine öğrenmesi yöntemlerinin çok değişik farklı şekillerde kullanımları sonucunda gelecekte daha etkili bilinmeyen veya eksik değer hesaplamasının mümkün hale geleceği öngörülmektedir.

## KAYNAKLAR

Abdella, M., Marwala, T., The use of genetic algorithms and neural networks to approximate missing data in database, Comput Inform, 24, 577-589, 2005.

Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T., A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data, Ieee T Med Imaging, 21, 193-199, 2002.

Akpınar, H., Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, İstanbul Üniversitesi İşletme Fakültesi Dergisi, Cilt 29, Sayı 1/Nisan, s. 1–22, 2000.

Allahverdi N., Uzman Sistemler, Bir Yapay Zeka Uygulaması, Atlas Yayın-Dağıtım, İstanbul, Mart 2002

Alpaydın, E., 2000. Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2k-notlar.doc, Erişim Tarihi: 12.06.2013

Altıntop, Ü., Internet Tabanlı Öğretimde Veri Madenciliği Tekniklerinin Uygulanması, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi, Yüksek Lisans Tezi, 2006.

Aslan, S., Yozgatlıgil, C., İyigün, C., Batmaz, İ., Tatlı, H., Meteorolojik Zaman Serilerinde Kayıp Veri Tahmin Yöntemlerinin Başarımlarının Korelasyon Boyutu Analiziyle Karşılaştırılması, İstatistik Araştırma Dergisi, 8 (2), 55-67, 2011.

Aydın S., Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Doktora Tezi, 2007.

Aydilek, I.B., Arslan, A., A Novel Hybrid Approach to Estimating Missing Values in Databases Using K-Nearest Neighbors and Neural Networks, Int J Innov Comput I, 8, 4705-4717, 2012.

Aydilek, I.B., Arslan, A., A hybrid method for imputation of missing values using optimized Fuzzy c-means with Support Vector Regression and a Genetic Algorithm, Information Sciences, Volume 233, Pages 25–35, 1 June 2013.

Barutçugil, İ., Bilgi Yönetimi. İstanbul: Kariyer Yayıncılık, 2002.

Basak, D., Pal, S., Patranabis, D.C., Support Vector Regression, Neural Information Processing – Letters and Reviews Vol. 11, No. 10, October 2007.

Batista, G.E.A.P.A., Monard, M.C., An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence 17, 519-533, 2003.

Bergmeir, C., Benitez, J.M., On the use of cross-validation for time series predictor evaluation, Inform Sciences, 191, 192-213, 2012.

Betechuoh, B.L., Marwala, T., Tettey, T., Autoencoder networks for HIVclassification, CurrentScience, Vol 91, No 11, pp1467-1473, December 2006.

Blake, C.L., Merz, C.J., UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html last accessed 02 may 2013, Irvine, CA, U. of California, Department of Information and Computer Science, 1998.

Blend, D., Marwala, T., Comparison of Data Imputation Techniques and their Impact, arXiv, scientificcommons, 2008.

Braik, M., Sheta, A., Arieqat, A., A Comparison between GAs and PSO in Training ANNto Model the TE Chemical Process Reactor, AISB 2008 Convention Communication, Interaction and Social Intelligence, 2008.

Bras, L.P., Menezes, J.C., Improving cluster-based missing value estimation of DNA microarray data, Elsevier Sciencedirect, Biomolecular Engineering 24, 273–282, 2007.

Breiman, L., Friedman, JH., Olshen, RA & Stone, CJ., Classification and regression trees. Monterey,CA: Wadsworth & Boks / Cole Advanced Boks & Software, 1984

Busetti, F., Genetic Algorithms overview, 2001.

Celton, M., Malpertuy, A., Lelandais, G., De Brevern, A.G, Comparative analysis of missing value imputation methods to improve clustering and interpretation of microarray experiments, BMC Genomics, 11-15, 2010.

Chang, J.F., A Performance Comparison Between Genetic Algorithms And Particle Swarm Optimization Applied In Constructing Equity Portfolios, International Journal of Innovative Computing, Information and Control ICIC International 2009 ISSN 1349-4198 Volume 5, Number 12(B), December 2009.

Chen, Z., Yang, B., Chen, Y., Wang, L., Wang, H., Abraham, A., Grosan, C., Improving Neural Network Classification Using Further Division Of Recognition Space, International Journal of Innovative Computing, Information and Control ICIC International 2009 ISSN 1349-4198 Volume 5, Number 2, February 2009.

Cheng, Y., D.Q., Feng, Q.R., Positive approximation and converse approximation in interval-valued fuzzy rough sets, Inform Sciences, 181, 2086-2110, 2011.

Cunningham, P., Delany, S.J., k-Nearest Neighbour Classifiers, University College Dublin Technical Report UCD-CSI-2007-4 March 27, 2007.

Das, S., Sil, S., Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm, Inform Sciences, 180, 1237-1256, 2010.

Di Nuovo, A.G., Missing data analysis with fuzzy C-Means: A study of its application in a psychological scenario, Expert Syst Appl, 38, 6793-6797, 2011.

Dunham, M.H., Data Mining Introductory and Advanced Topics. New Jersey: Pearson Education, Inc., 2003.

Dunn, J.C., A Fuzzy Relative of the Isodata Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics, 3: 32-57, doi:10.1080/01969727308546046, 1973.

Emel, G.G., Taşkın, Ç., Genetik Algoritmalar Ve Uygulama Alanları, Uludağ Üniversitesi, İktisadi Ve İdari Bilimler Fakültesi Dergisi, Cilt XXI, Sayı 1, S. 129-152, 2002.

Farhangfar, A., Kurgan, L.A., Pedrycz, W., A novel framework for imputation of missing values in databases, Ieee T Syst Man Cy A, 37, 692-709, 2007.

Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., Uthurusamy, R., "Advances in Data Mining and Knowledge Discovery", AAAI Pres, USA, 1994.

Feng, H.H., Chen, G.S., Yin, C., Yang, B.R., Chen, Y.M., A SVM regression based approach to filling in missing values, Knowledge-Based Intelligent Information and Engineering Systems, Pt 3, Proceedings, 3683, 581-587, 2005.

Fodor, I.K., A survey of dimension reduction techniques, Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.

Gao, S., Zhang, J., Wang, X., Tang, Z., Multi-Layer Neural Network Learning Algorithm Based On Random Pattern Search Method, International Journal of Innovative Computing, Information and Control ICIC International 2009 ISSN 1349-4198 Volume 5, Number 2, February 2009.

Garcia, A.J.T., Hruschka, E.R., Naive Bayes as an imputation tool for classification problems, Hybrid Intelligent Systems, 2005.

Genc, S., Boran, F.E., Akay, D., Xu, Z.S., Interval multiplicative transitivity for consistency, missing values and priority weights of interval fuzzy preference relations, Inform Sciences, 180, 4877-4891, 2010.

Ghosh, S., Mujumdar, P.P., Future rainfall scenario over Orissa with GCM projections by statistical downscaling, Curr Sci India, 90, 396-404, 2006.

Gil, M., Sarabia, E.G., Llata, J.R., Oria, J.P., Fuzzy c-means clustering for noise reduction, enhancement and reconstruction of 3D ultrasonic images, Emerging Technologies and Factory Automation, Proceedings, ETFA '99, 1999.

Goldberg, D.E., Sizing Populations for Serial and Parallel Genetic Algorithms, Proceedings of the Third International Conference on Genetic Algorithms, 70-79, 1989.

Guclu, N., Sotirofski, K., Bilgi Yönetimi, Türk Eğitim Bilimleri Dergisi, 4(4), 351-371, Güz 2006.

Hajek, J., Sidak, Z.K., Theory of rank tests, Academic Press, New York, 1967.

Han, J., Kamber, M., Data Mining Concepts and Techniques, Academic Press, 2001.

Hathaway, R.J., Bezdek, J.C., Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm, Pattern Recogn Lett, 23, 151-160, 2002.

Hengpraphrom, K., Na Wichian, S., Meesad, P., Missing Value Imputation Using Genetic Algorithm, Third International Symposium on Intelligent Informatics (ISII2010) First International Symposium on Information and Knowledge Management (ISIKM2010) September 5 – 7, 2010.

Hlalele, N., Nelwamondo, F., Marwala, T., Imputation of Missing Data Using PCA, Neuro-Fuzzy and Genetic Algorithms, Springerlink, Advances in Neuro-Information Processing, Lecture Notes in Computer Science, Volume 5507/2009, 485-492, 2009.

Huang, C.H., Kao, H.Y., Interval Regression Analysis with Soft-Margin Reduced Support Vector Machine, Next-Generation Applied Intelligence, Proceedings, 5579, 826-835, 2009.

Kaya, Y., Tekin, R., Veritabanı ve Uygulamaları, Papatya Yayıncılık Eğitim, İstanbul, 2007.

Kayaalp, K., Asenkron Motorlarda Veri Madenciliği İle Hata Tespiti, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Eğitim Anabilim dalı, Yüksek Lisans Tez, 2007.

Kohavi, R., A study of cross validation and bootstrap for accuracy estimation and model selection. The Fourteenth International Joint Conference on Artificial Intelligence, San Francisco, CA:Morgan Kaufman , pp: 1137-45, 1995.

Li, D., Deogun, J., Spaulding, W., Shuart, B., Towards missing data imputation: A study of fuzzy K-means clustering method, Rough Sets and Current Trends in Computing, 3066, 573-579, 2004.

Li, Y., Yang, Y., Zhou, L., Zhu, R., Observations On Using Problem-Specific Genetic Algorithm For Multiprocessor Real-Time Task Scheduling, International Journal of InnovativeComputing, Information and Control ICIC International 2009 ISSN 1349-4198 Volume 5, Number 9, September 2009.

Li, D., Gu, H., Zhang, L.Y., A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data, Expert Syst Appl, 37, 6942-6947, 2010.

Liao, Z., Lu, X., Yang, T., Wang, H., Missing Data Imputation: A Fuzzy K-means Clustering Algorithm over Sliding Window, Fuzzy Systems and Knowledge Discovery, 14-16 133 – 137, Aug 2009.

Lim, C.P., Leong, J.H., Kuan, M.M., A hybrid neural network system for pattern classification tasks with missing features, Ieee T Pattern Anal, 27, 648-653, 2005.

Little, R.J.A., Rubin, D.B., Statistical analysis with missing data, Wiley, New York, 1987.

Mahmoud, M.S., Emzir, M.F., State estimation with asynchronous multi-rate multi-smart sensors, Inform Sciences, 196, 15-27, 2012.

Maimon, O., Rokach, L., Data Mining and Knowledge Discovery Handbook, Springer, 2010.

Malhotra, Y., Knowledge Management in Inquiring Organizations. Proceedings of 3rd Americas Conference on Information Systems. Indianapolis, August, 239-295, 1997.

Marwala, T., Chakraverty, S., Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm, Curr Sci India, 90, 542-548, 2006.

Marwala, T., Computational intelligence for missing data imputation, estimation and management:knowledge optimization techniques, Information Science Reference, Hershey PA, 2009.

Meng, Z.Q., Shi, Z.Z., Extended rough set-based attribute reduction in inconsistent incomplete decision systems, Inform Sciences, 204, 44-69, 2012.

Mitchell, T.M., Machine Learning, McGraw Hill, 1997.

Mohamed, S., Marwala, T., Neural Network Based Techniques for Estimating Missing Data in Databases, 16th Annual Symposium of the Patten Recognition Association of South Africa, Langebaan, pp.27-32, 2005.

Mohamed, A.K., Nelwamondo, F.V., Marwala, T., Estimating Missing Data Using Neural Network Techniques, Principal Component Analysis and Genetic Algorithms, PRASA proceedings, 2007.

Muller K.R., Smola A., Ratch G., Scholkopf B., Kohlmorgen J., Vapnik V.N., Using support vector support machines for time series prediction, Image Processing Services Research Lab, AT&T Labs, 2000.

Nandhini, T., Sakthivel, N.K., A Smart Hybrid Data Mining Technique for Improving Classification Accuracy in Complex Data Sets, European Journal of Scientific Research ISSN 1450-216X Vol.71 No.2, pp. 233-242, 2012.

Narang, R., Database Manegement Systems, Prentice-hall of India Private Limited, New Delhi, 2004.

Nelwamondo, F.V., Mohamed, S., Marwala, T., Missing data: A comparison of neural network and expectation maximization techniques, Curr Sci India, 93, 1514-1521, 2007.

Nelwamondo, F., Marwala, T., Fuzzy ARTMAP and Neural Network Approach to Online Processing of Inputs with Missing Values, SAIEE Africa Research Journal, 93(11), pp. 1514-1521, 2007.

Nelwamondo, F.V., Marwala, T., Techniques For Handling Missing Data: Applications To Online Condition Monitoring, International Journal of Innovative Computing, Information and Control ICIC International 2008 ISSN 1349-4198 Volume 4, Number 6, June 2008.

Nelwamondo, F.V., Golding, D., Marwala, T., A dynamic programming approach to missing data estimation using neural Networks, Information Sciences, doi:10.1016/j.ins.2009.10.008, 2009.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S., A Bayesian missing value estimation method for gene expression profile data. Bioinformatics, 19, 2088-2096, doi: 10.1093/bioinformatics/btg287, 2003.

Ogawa, N., DeRisi, J., Brown, P.O., New components of a system for phosphate accumulation and polyphosphate metabolism in Saccharomyces cerevisiae revealed by genomic expression analysis, MolBiol Cell, 11(12):4309-4321, 2000.

Oğuzlar, A., Veri Önişleme. Erciyes Üniversitesi iktisadi ve idari Bilimler Fakültesi Dergisi, 21, 67-76, 2003.

Özdemir, A., Aslay, F., Çam, H., "Veri tabanında Bilgi Keşfi Süreci: Gümüşhane Devlet Hastanesi Uygulaması", Selçuk Üniversitesi İktisadi ve İdari Bilimler Fakültesi, Sosyal ve Ekonomik Araştırmalar Dergisi, Cilt 14., ss. 347-367, 2010.

Patil, D.V., Multiple Imputation of Missing Data with Genetic Algorithm based Techniques, IJCA Special Issue on Evolutionary Computation for Optimization Techniques ECOT, 2010.

Pelckmans, K., De Brabanter, J., Suykens, J.A.K., De Moor, B., Handling missing values in support vector machine classifiers, Neural Networks, 18, 684-692, 2005.

Piramuthu, S., Evaluating Feature Selection Methods for Learning in Data Mining Applications, European Journal of Operational Research, Article In Press, pp.1-11, 2004.

Qiao, W., Gao, Z., Harley, R.G., Continuous on-line identification of nonlinear plants in power systems with missing sensor measurements, Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vols 1-5, 1729-1734, 2005.

Salama, A.S., Topological solution of missing attribute values problem in incomplete information tables, Inform Sciences, 180, 631-639, 2010.

Schafer, J.L., Graham, J.W., Missing data: Our view of the state of the art. Psychological Methods, 7 (2), 147–177, 2002.

Schneider, T., Analysis of incomplete climate data: Estimation of Mean Values and covariance matrices and imputation of Missing values. Journal of Climate 14, 853-871, 2001.

Sehgal, M.S., Gondal, L., Dooley, L.S., Collateral Missing value imputation: a new robust missing value estimation algorithm for microarray data, Bioinformatics, vol. 21, pp. 2417-2423, 2005.

Sehgal, M.S.B., Gondal, I., Dooley, L.S., Coppel, R., Ameliorative missing value imputation for robust biological knowledge inference, J Biomed Inform, 41, 499-514, 2008.

Sever, H., Oğuz, B., Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım:Kısım I: Eşleştirme Sorguları ve Algoritmalar. Bilgi Dünyası, vol. 3, n. 2, pp. 173-204, 2002.

Shahi, A., Atan, R.B., Sulaiman, M.N., Detecting Effectiveness of Outliers and Noisy Data on Fuzzy System Using FCM, European J. of Sci. Research, vol. 36, issue 4, pp. 627-638, 2009.

Sidak, Z., Sen, P.K., Hajek, J., Theory of rank tests (probability and mathematical statistics). Academic Press, 1999.

Silva-Ramirez, E.L., Pino-Mejías, R., López-Coello, M., Cubiles-de-la-Vega, M.D., Missing value imputation on missing completely at random data using multilayer perceptrons, Neural Networks , 2011.

Sim, K., Liu, G.M., Gopalkrishnan, V., Li, J.Y., A case study on financial ratios via cross-graph quasi-bicliques, Inform Sciences, 181, 201-216, 2011.

Smola, A.J., Scholkopf, B., A tutorial on support vector regression, Stat Comput, 14, 199-222, 2004.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B., Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization, Molecular Biology of the Cell, Vol. 9, Issue 12, 3273-3297, December 1998.

Sun, Y., Braga-Neto, U., Dougherty, E.R., Impact ofMissing Value Imputation on Classification for DNAMicroarray Gene Expression Data—AModel-Based Study, EURASIP Journal on Bioinformatics and Systems Biology Volume, 2009.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., Least Squares Support Vector Machines, http://www.esat.kuleuven.be/sista/lssvmlab/ last accessed 19 apr 2013, World Scientific, Singapore, 2002.

Timm, H., Doring, C., Kruse, R., Different approaches to fuzzy clustering of incomplete datasets, Int J Approx Reason, 35, 239-249, 2004.

Thompson, B.B., Marks, R.J., El-Sharkawi, M.A., On the contractive nature of autoencoders: Application to sensor restoration, Proceedings of the International Joint Conference on Neural Networks, Vol. 4, pp. 3011–3016, 2003.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R.,Botstein, D., Altman, R.B, Missing value estimation methods for DNA microarrays, Bioinformatics 17, 520–525, 2001.

Twala, B., An empirical comparison of techniques for handling incomplete data using decision trees. Applied Artificial Intelligence 23, 373-405, 2009.

Van Hulse, J., Khoshgoftaar, T.M., Incomplete-case nearest neighbor imputation in software measurement data, IRI 2007: Proceedings of the 2007 IEEE International Conference on Information Reuse and Integration, 630-637, 2007.

Vapnik, V., Golowich, S.E., Smola, A., Support vector method for function approximation, regression estimation, and signal processing, Adv Neur In, 9, 281-287, 1997.

Wang, P.H., Pattern-Recognition with Fuzzy Objective Function Algorithms - Bezdek,Jc, Siam Rev, 25, 442-442, 1983.

Wang, X., Li, A., Jiang, Z.H., Feng, H.Q., Missing value estimation for DNA microarray gene expression data by Support Vector Regression imputation and orthogonal coding scheme, Bmc Bioinformatics, 7, 2006.

Wu, C.H., Wei, C.C., Su, D.C., Chang, M.H., Ho, J.M., Travel Time Prediction with Support Vector Regression, Intelligent Transportation Systems, IEEE Transactions on (Volume:5 , Issue: 4 ), Dec. 2004.

Wu, X., Wu, B., Deng, Y., Zhao, J., The Fuzzy Learning Vector Quantization with Allied Fuzzy c-means Clustering for Clustering Noisy Data, Journal of Information & Computational Science 8: 9, 1713-1719, 2011.

Xie, X.L., Beni, G., A Validity Measure for Fuzzy Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 841-847, doi:10.1109/34.85677, Aug. 1991.

Yang, Haiqin., Margin Variations in Support Vector Regression for the Stock Market Prediction, Chinese University of Hong Kong, Department of Computer Science & Engineering, Degree of Master of Philosophy Thesis, 2003

Yang, L.Y., Xu, L.S., Topological properties of generalized approximation spaces, Inform Sciences, 181, 3570-3580, 2011.

Yozgatligil, C., Aslan, S., İyigün, C., Batmaz, İ., Comparison of Missing Value Imputation Methods in Time Series: The Case of Turkish Meteorological Data. Theoretical and Applied Climatology, Volume 112, Issue 1-2, pp 143-167, April 2013.

# ÖZGEÇMİŞ

## KİŞİSEL BİLGİLER

**Adı Soyadı** : İbrahim Berkan AYDİLEK
**Uyruğu** : T.C.
**Doğum Yeri ve Tarihi** : Şanlıurfa 1981
**Telefon** : 0 507 437 36 76
**E-mail** : Berkan@selcuk.edu.tr, BerkanAydilek@hotmail.com

## EĞİTİM

| **Derece** | | **Adı, İlçe, İl** | **Bitirme Yılı** |
|---|---|---|---|
| Lise | : | Şanlıurfa Anadolu Lisesi | 1999 |
| Üniversite | : | Selçuk Üniversitesi Bilgisayar Mühendisliği | 2003 |
| Yüksek Lisans | : | Selçuk Üniversitesi Bilgisayar Mühendisliği | 2006 |
| Doktora | : | Selçuk Üniversitesi Bilgisayar Mühendisliği | 2013 |

## İŞ DENEYİMLERİ

| **Yıl** | **Kurum** | **Görevi** |
|---|---|---|
| 2004 | Selçuk Üniversitesi Bilgi İşlem | Okutman |

## UZMANLIK ALANI

Veri Madenciliği, Yapay Zeka, Makine Öğrenmesi, Uzaktan Eğitim

## YABANCI DİLLER

İngilizce

## YAYINLAR

I.B. Aydilek, A. Arslan, A Novel Hybrid Approach to Estimating Missing Values in Databases Using K-Nearest Neighbors and Neural Networks, Int J Innov Comput I, 8 (2012) 4705-4717 (Doktora tezinden türetilmiştir)

I.B. Aydilek, A. Arslan, A hybrid method for imputation of missing values using optimized Fuzzy c-means with Support Vector Regression and a Genetic Algorithm, Information Sciences, Volume 233, 1 June 2013, Pages 25–35 (Doktora tezinden türetilmiştir)