

**KERATOKONUS HASTALIĐININ YAPAY ZEKA  
TEKNİKLERİYLE TEŐHİS EDİLMESİ**

**2016  
DOKTORA TEZİ  
BİLGİSAYAR MÜHENDİSLİĐİ**

**Murat UÇAR**

**KERATOKONUS HASTALIĐININ YAPAY ZEKA TEKNİKLERİYLE TEŐHİS  
EDİLMESİ**

**Murat UÇAR**

**Karabük Üniversitesi  
Fen Bilimleri Enstitüsü**

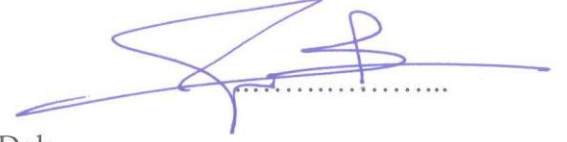
**Bilgisayar MühendisliĐi Anabilim Dalında  
Doktora Tezi  
Olarak Hazırlanmıştır**

**KARABÜK  
Mayıs 2016**

Murat UÇAR tarafından hazırlanan “KERATOKONUS HASTALIĞININ YAPAY ZEKÂ TEKNİKLERİYLE TEŞHİS EDİLMESİ” başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Baha ŞEN

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

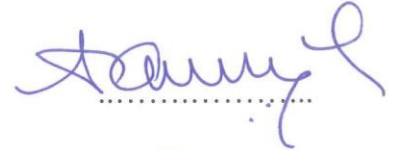


Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 29/04/2016

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Prof. Dr. Abdullah ÇAVUŞOĞLU (YÖK Üyesi)



Üye : Prof. Dr. Raif BAYIR (KBÜ)



Üye : Yrd. Doç. Dr. Baha ŞEN (YBÜ)



Üye : Yrd. Doç. Dr. İlhami Muharrem ORAK (KBÜ)



Üye : Yrd. Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ)



...../...../2016

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Nevin AYTEMİZ

Fen Bilimleri Enstitüsü Müdürü



*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

  
Murat UÇAR

## ÖZET

**Doktora Tezi**

### **KERATOKONUS HASTALIĞININ YAPAY ZEKA TEKNİKLERİYLE TEŞHİS EDİLMESİ**

**Murat UÇAR**

**Karabük Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:**

**Yrd. Doç. Dr. Baha ŞEN**

**Mayıs 2016, 107 sayfa**

Keratokonus gözün en önde yerleşimli saydam tabakasının yani korneanın, ilerleyici incelme ve sivrileşmesiyle görülen hastalığıdır. Bir başka deyişle gözün önünde yer alan ve bir kubbe bombeliğinde olması gereken kornea tabakasının bombeliğinin ilerleyici şekilde bozulması ve konik şekil almasıdır. Bu çalışmada amaç keratokonus hastalığının teşhisi için istatistiksel analizlere dayalı yeni bir sınıflandırma metodu tanımlamak ve hastalığın teşhisinin yüksek performanslı zeki bir sistem tarafından gerçekleştirilmesini sağlamaktır.

Çalışmada Ankara Atatürk Eğitim ve Araştırma Hastanesi'nden alınan 159 keratokonus hastasının 301 gözü ile kontrol grubu olarak 265 refraktif cerrahi adayının 394 gözü kullanılmıştır. Uygulamada öncelikle istatistiksel analizlere dayalı yeni bir sınıflandırma tanımlanmış ve keratokonus hastalığının teşhisi için yüksek

performanslı zeki bir sistem tasarlanmıştır. Çalışmada sınıflandırma konusunda en çok tercih edilen yöntemlerden yapay sinir ağları ve destek vektör makineleri kullanılmıştır. Çalışmanın amacı bu yöntemlerin doğruluk oranları ve performanslarını karşılaştırarak en uygun yöntemi bulmaya çalışmaktır.

**Anahtar Sözcükler :** Paralel programlama, CUDA, zeki sistemler, veri madenciliği.

**Bilim Kodu** : 902.1.014



## **ABSTRACT**

**Ph. D. Thesis**

### **DIAGNOSING KERATOCONUS DISEASE WITH ARTIFICIAL INTELLIGENCE TECHNIQUES**

**Murat UÇAR**

**Karabük University**

**Graduate School of Natural and Applied Sciences**

**Department of Computer Engineering**

**Thesis Advisor:**

**Assist. Prof. Dr. Baha ŞEN**

**May 2016, 107 pages**

Keratoconus is an eye disease characterised by progressive thinning of the cornea, which is a transparent layer of tissue in the front of the eye. In other words, it is the progressive distortion of the corneal layers into a conical shape. Vision progressively deteriorates as the cornea becomes more conical. The main focus of the study was to define a new classification method for detecting keratoconus disease based on statistical analysis and to diagnose the keratoconus disease by using high performance intelligent system.

In this study, 301 eyes of 159 patients were enrolled as the subject group and 394 eyes of 265 refractive surgery candidates were enrolled as the control group. In our application primarily a new classification which based on statistical analysis was defined and a high performance intelligent system was designed for predicting

keratoconus disease. Artificial neural networks and support vector machines, which are among the most preferred classification techniques, were used in this study. The aim of the study is finding the most appropriate method by comparing the accuracy and performance of these methods.

**Key Word** : Parallel programing, CUDA, intelligent systems, data mining.

**Science Code** : 902.1.014





## TEŞEKKÜR

Tezi hazırlama sürecimde, her aşamada bana yol gösteren, çalışmalarımı yönlendiren, bilgi, öneri ve yardımlarını hiçbir zaman esirgemeyen çok değerli danışman hocam Yrd. Doç. Dr. Baha ŞEN'e, sonsuz teşekkürlerimi sunarım.

Yönlendirmeleri ile tezin gelişimine katkı sağlayan tez izleme jüri üyesi sayın hocalarım Prof. Dr. Raif BAYIR ve Yrd. Doç. Dr. İlhami Muharrem ORAK'a yardımlarından dolayı teşekkür ederim.

Tez çalışmasının takibi sırasında değerli önerileriyle yardımcı olan sayın hocalarım Prof. Dr. Abdullah ÇAVUŞOĞLU, Prof. Dr. Fatih Vehbi ÇELEBİ ve Doç. Dr. Hasan Basri ÇAKMAK' a teşekkür ederim.

Çalışmalarım süresince bana hep destek olan sevgili eşim Emine UÇAR'a ve aileme manevi hiçbir yardımı esirgemediği yanımda oldukları için tüm kalbimle teşekkür ederim.

Bu çalışmamı, oğlum Ege UÇAR'a atfediyorum.

## İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLER DİZİNİ.....	xiii
ÇİZELGELER DİZİNİ .....	xv
SİMGELER VE KISALTMALAR DİZİNİ .....	xvi
BÖLÜM 1 .....	1
GİRİŞ .....	1
1.1. LİTERATÜR.....	3
BÖLÜM 2 .....	11
VERİ MADENCİLİĞİ.....	11
2.1. VERİ MADENCİLİĞİ TANIMI.....	11
2.2. VERİ MADENCİLİĞİNİN UYGULAMA ALANLARI .....	12
2.3. VERİ MADENCİLİĞİ SÜRECİ.....	12
2.4. VERİ HAZIRLAMA.....	13
2.4.1. Veri Temizleme .....	13
2.4.2. Veri Bütünleştirme.....	14
2.4.3. Veri İndirgeme.....	15
2.4.4. Veri Dönüştürme .....	16
2.4.5. Veri Madenciliği Yönteminin Uygulanması .....	16
2.4.6. Sonuçların Değerlendirilmesi Ve Sunulması.....	16

BÖLÜM 3 .....	17
SINIFLANDIRMADA KULLANILAN İSTATİSTİKSEL ANALİZ TEKNİKLERİ .....	17
3.1. FAKTÖR ANALİZİ .....	17
3.1.1. Veri Setinin Uygunluğunun Değerlendirilmesi .....	18
3.1.2. Faktör Türetme Modelleri Ve Uygun Faktör Analizi Yönteminin Seçimi .....	19
3.1.2.1. Temel Bileşen Faktörü (Principal Component Factoring) .....	19
3.1.2.2. Türetilecek Ortak Faktör Sayısının Belirlenmesi .....	20
3.1.2.3. Faktörlerin Döndürülmesi .....	21
3.2. KÜMELEME ANALİZİ .....	21
3.2.1. Kümeleme Analizinin Aşamaları .....	22
3.2.2. Kümeleme Analizinde Değişken Türlerine Göre Benzerlik Ve Uzaklık Ölçüleri .....	23
3.2.2.1. Aralık Ve Oransal Ölçekli Değişkenler .....	23
3.2.2.2. Nominal Ölçekli Değişkenler .....	26
3.2.2.3. Ordinal Ölçekli Değişkenler .....	28
3.2.3. Kümeleme Yöntemlerinin Sınıflandırılması .....	29
3.2.3.1. Hiyerarşik Yöntemler .....	29
3.2.3.2. Hiyerarşik Olmayan Kümeleme Yöntemleri .....	33
3.2.4. Küme Geçerliliği Teknikleri .....	34
BÖLÜM 4 .....	37
SINIFLANDIRMADA KULLANILAN ZEKİ SİSTEMLER .....	37
4.1. YAPAY SINIR AĞLARI .....	37
4.1.1. Yapay Sinir Ağlarının Yapısı Ve Temel Elemanları .....	38
4.1.1.1. Biyolojik Sinir Hücresinin Yapısı .....	38
4.1.1.2. Yapay Sinir Hücresinin Yapısı .....	39
4.1.2. Yapay Sinir Ağı Yapıları .....	40
4.1.2.1. İleri Beslemeli Ağlar .....	40
4.1.2.2. Geri Beslemeli Ağlar .....	40
4.1.3. Yapay Sinir Ağları Algoritmaları .....	41
4.1.3.1. Danışmanlı Öğrenme .....	41

	<b><u>Sayfa</u></b>
4.1.3.2. Danışmansız Öğrenme .....	41
4.1.3.3. Destekleyicili Öğrenme .....	42
4.1.4. Yapay Sinir Ağlarında Öğrenme Ve Test.....	42
4.1.5. Yapay Sinir Ağı Modelleri .....	43
4.1.5.1. Çok Katmanlı Algılayıcı .....	43
4.1.5.2. Radyal Tabanlı Fonksiyon .....	45
4.2. DESTEK VEKTÖR MAKİNELERİ.....	46
4.2.1. İkili DVM Sınıflandırma Algoritması .....	47
4.2.2. Çok Sınıflı DVM Sınıflandırma Algoritması .....	49
BÖLÜM 5 .....	52
CUDA .....	52
5.1. GRAFİK İŞLEM BİRİMİ (GPU).....	53
BÖLÜM 6 .....	56
UYGULAMA .....	56
6.1. VERİ.....	56
6.2. UYGULAMA.....	57
6.2.1. Faktör Analizi .....	57
6.2.2. Kümeleme Analizi .....	62
6.2.3. Sınıflandırma Yöntemleri .....	65
6.2.3.1. Çok Katmanlı Yapay Sinir Ağları.....	78
6.2.3.2. Radyal Tabanlı Fonksiyon Ağları .....	83
6.2.3.3. Destek Vektör Makineleri.....	85
6.2.4. CUDA Destekli Sınıflandırma Yöntemleri .....	86
6.2.4.1. CUDA Destekli MLP Yapay Sinir Ağı .....	87
6.2.4.2. CUDA Destekli RBF Yapay Sinir Ağı .....	88
6.2.4.3. CUDA Destekli DVM Yapay Sinir Ağı .....	89
6.2.5. Zeki Sistemin Hastalık Teşhisinde Kullanımı .....	90
6.2.5.1. Hasta Bilgileri Giriş .....	90
6.2.5.2. Teşhis İşlemi .....	90

	<b><u>Sayfa</u></b>
BÖLÜM 7 .....	92
SONUÇLAR VE DEĞERLENDİRME .....	92
KAYNAKLAR .....	97
ÖZGEÇMİŞ .....	107



## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 1.1. Normal ve keratokonuslu göz. ....	1
Şekil 2.1. Veri madenciliği süreci. ....	13
Şekil 3.1. Hiyerarşik kümeleme yöntemleri. ....	30
Şekil 4.1. Biyolojik sinir hücresi. ....	38
Şekil 4.2. Yapay sinir hücresi. ....	39
Şekil 4.3. Çok katmanlı yapay sinir ağı mimarisi. ....	44
Şekil 4.4. Radyal tabanlı fonksiyon yapay sinir ağı mimarisi. ....	45
Şekil 4.5. İki sınıflı veriyi ayıran bir alt düzlem, en iyi alt düzlem ve marjları. ....	48
Şekil 4.6. Farklı 5 sınıfın bire karşı bir ayrışımı kullanan çoklu sınıf DVM ile sınıflandırılması. ....	50
Şekil 5.1. GPU hesaplaması için programlama dillerinin tarihçesi. ....	54
Şekil 5.2. Gigaflop bağlamında tepe performansının tarihsel karşılaştırması ve mevcut en hızlı NVIDIA GPU lar ve Intel CPU lar için bant genişliği... 54	54
Şekil 6.1. Scree Test (Yığın grafiği). ....	59
Şekil 6.2. MATLAB'ta oluşturulan çok katmanlı algılayıcı YSA mimarisi. ....	65
Şekil 6.3. Trainlm eğitim fonksiyonu sonuçları. ....	66
Şekil 6.4. Trainlm eğitim fonksiyonu performans grafiği. ....	67
Şekil 6.5. Trainbfg eğitim fonksiyonu sonuçları. ....	68
Şekil 6.6. Trainbfg eğitim fonksiyonu performans grafiği. ....	68
Şekil 6.7. Traincgb eğitim fonksiyonu sonuçları. ....	69
Şekil 6.8. Traincgb eğitim fonksiyonu performans grafiği. ....	69
Şekil 6.9. Trainrp eğitim fonksiyonu sonuçları. ....	70
Şekil 6.10. Trainrp eğitim fonksiyonu performans grafiği. ....	71
Şekil 6.11. Trainbr eğitim fonksiyonu sonuçları. ....	72
Şekil 6.12. Trainbr eğitim fonksiyonu performans grafiği. ....	72
Şekil 6.13. Trainoss eğitim fonksiyonu sonuçları. ....	73
Şekil 6.14. Trainoss eğitim fonksiyonu performans grafiği. ....	73
Şekil 6.15. Traingd eğitim fonksiyonu sonuçları. ....	74

Şekil 6.16. Traingd eğitim fonksiyonu performans grafiği.....	74
Şekil 6.17. Traingda eğitim fonksiyonu sonuçları. ....	75
Şekil 6.18. Traingda eğitim fonksiyonu performans grafiği.....	75
Şekil 6.19. Traingdx eğitim fonksiyonu sonuçları. ....	76
Şekil 6.20. Traingdx eğitim fonksiyonu performans grafiği.....	76
Şekil 6.21. Traingdm eğitim fonksiyonu sonuçları.....	77
Şekil 6.22. Traingdm eğitim fonksiyonu performans grafiği.....	77
Şekil 6.23. Çok katmanlı algılayıcı yapay sinir ağı mimarisi. ....	80
Şekil 6.24. Keratokonus teşhis yazılımı giriş ekranı.....	80
Şekil 6.25. Yapay sinir ağı oluşturma ekran görüntüsü. ....	81
Şekil 6.26. Eğitim sonucunda oluşan ekran görüntüsü (YSA). ....	82
Şekil 6.27. Yapay sinir ağı listeleme ve seçim ekranı.....	82
Şekil 6.28. Radyal tabanlı fonksiyon ağları giriş ekranı. ....	83
Şekil 6.29. Eğitim ve test sonucunda oluşan ekran görüntüsü (RBF).....	84
Şekil 6.30. Radyal tabanlı fonksiyon ağı listeleme ve seçim ekranı.....	84
Şekil 6.31. Destek vektör makineleri giriş ekranı. ....	85
Şekil 6.32. Eğitim ve test sonucunda oluşan ekran görüntüsü (DVM).....	86
Şekil 6.33. Destek vektör makinesi listeleme ve seçim ekranı. ....	86
Şekil 6.34. CUDA destekli MLP yapay sinir ağı giriş ekranı.....	87
Şekil 6.35. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDAMLP).....	87
Şekil 6.36. CUDA destekli RBF yapay sinir ağı giriş ekranı.....	88
Şekil 6.37. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDARBF). ....	88
Şekil 6.38. CUDA destekli DVM yapay sinir ağı giriş ekranı.....	89
Şekil 6.39. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDADVM).....	89
Şekil 6.40. Hasta bilgileri giriş ekranı.....	90
Şekil 6.41. Hasta seçim formu ekranı. ....	91
Şekil 6.42. Hastalık teşhis formu ekranı. ....	91

## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 3.1. KMO uygunluk testi için önerilen kriterler. ....	19
Çizelge 3.2. İkili değişkenler için kontenjans tablosu. ....	27
Çizelge 3.3. Benzerlik ölçüleri.....	28
Çizelge 6.1. Uygulamada kullanılan bağımsız değişkenlerin listesi.....	57
Çizelge 6.2. KMO ve Barlett testi sonuçları. ....	58
Çizelge 6.3. Varyansa katılma (Öz Değer) kriteri sonuçları.....	58
Çizelge 6.4. Değişkenlerin ait olduğu faktör sonuçları.....	60
Çizelge 6.5. Güvenilirlik sonucu. ....	60
Çizelge 6.6. Değişkenlerin korelasyon değerleri. ....	61
Çizelge 6.7. VIF (Variance Inflation Factor) ve TV (Tolerance Value). ....	61
Çizelge 6.8. Aglomeratif çizelge .....	62
Çizelge 6.9. Kümeleme analizi sonuçları. ....	63
Çizelge 6.10. Altı kümeye ilişkin Wilks' Lamda değeri. ....	64
Çizelge 6.11. İstatistiksel sonuçlara göre keratokonus sınıflandırması. ....	65
Çizelge 6.12. Eğitim fonksiyonlarına ait başarı oranları ve eğitim süreleri. ....	78
Çizelge 7.1. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (MLP) .....	93
Çizelge 7.2. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (RBF) .....	93
Çizelge 7.3. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (DVM) .....	94
Çizelge 7.4. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDAMLP).....	94
Çizelge 7.5. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDARBF).....	95
Çizelge 7.6. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDADVM).....	95
Çizelge 7.7. Yöntemlerin doğruluk oranı ve zamana göre karşılaştırılması.....	96



## SİMGELER VE KISALTMALAR DİZİNİ

### KISALTMALAR

- CUDA : Compute Unified Device Architecture  
CLMI : The Cone Location and Magnitude Index  
PPK : Percent Probability Keratoconus  
KMO : Kaiser-Mayer-Olkin  
YSA : Yapay Sinir Ağları  
MLP : Multi Layer Perceptron (Çok Katmanlı Algılayıcı)  
RBF : Radial Basis Function (Radyal Tabanlı Fonksiyon)  
RTFA : Radyal Tabanlı Fonksiyon Ağları  
DVM : Destek Vektör Makineleri  
BKB : Bire Karşı Bir  
BKD : Bire Karşı Diğer  
GPU : Graphical Process Unit (Grafik İşlemci Birimi)  
CPU : Central Process Unit (Merkezi İşlemci Birimi)  
SPSS : Statistical Package for the Social Sciences  
VIF : Variance Inflation Factor

## BÖLÜM 1

### GİRİŞ

Keratokonus kelimesi; Latince kornea manasına gelen Kerato ve konus manasına gelen Konos kelimelerinin birleştirilmesiyle oluşturulmuştur. Keratokonusun bir hastalık olarak detaylı bir tanımı ve korneanın diğer ektaziyle seyreden hastalıklarından ayrımı ilk olarak 1854 yılında yapılmıştır [1]. Keratokonus; kornea stromasının ilerleyici olarak incilmesi sonucu korneanın protrüzyonuyla meydana gelen, korneanın bilateral, asimetric ve inflamatuvar olmayan bir dejenerasyonudur. Korneal incelme genellikle korneanın inferior, inferiotempral ve santral bölgelerinde meydana gelir [2]. Bunun yanında korneal incelmenin çok seyrek de olsa korneanın süperior kadranında görüldüğü keratokonus vakaları da mevcuttur [3]. Korneal incelme ve sonrasındaki dışa doğru eğim görme kalitesinde hafiften ağır derecelere kadar değişebilen bozulmalara neden olan ilerleyici düzensiz astigmatizma ve miyopiye sebep olur [4]. Gözün en ön tabakasını kornea denilen saydam tabaka oluşturur. Kornea tabakası; cisimlerin net olarak görülmesini sağlayan, gözün kırma derecesini belirleyen en önemli optik unsurdur. Kornea normalde belirli bir dışbükeyliğe sahiptir. Keratokonus hastalarında ise korneanın dışa doğru olan şişkinlik miktarı giderek artar, yani kornea öne doğru sarkar (Şekil 1.1).



Şekil 1.1. Normal ve keratokonuslu göz.

Doğuştan kazanılan bir özellik olarak kornea dışbükeyliğindeki artış, keratokonus hastalarında astigmata neden olur. Bu gruptaki hastalarda meydana gelen astigmat, kornea yüzeyinin üst ve alt kadranında simetrik olmadığı, kontakt lens ya da gözlükle tam düzeltilemediği için görme kalitesini ileri derecede etkileyebilir ve hayat kalitesini düşürebilir.

Keratokonuslu hastanın yakınmaları arasında ışığa duyarlılığın artması, çift görme, kamaşma, ışıkların çevresinde halkalar görme, özellikle araba kullanırken hayalet görüntülerin ortaya çıkması, yanma, batma ve kızarıklık olarak sıralanabilir. Keratokonus ilerleyici bir hastalık olduğundan bu yakınmalar giderek artma eğilimindedir. İleri yaşlara gelindiğinde ise kornea merkezinde kalıcı beyaz bir leke meydana gelir ki; bu dönemde artık keratoplasti (kornea nakli) kaçınılmazdır.

Genellikle ergenlik döneminde başlayan hastalıktan hastalar 20'li yaşlarında haberdar olurlar. 20–40 yaş arasında ilerleme gösterip 40 yaştan sonra durağan döneme girer. Günümüzde keratokonus hastalığı batı toplumlarında her 2000 kişiden birinde gözlemlenmektedir. Keratokonus'un görülme sıklığı her geçen yıl daha da artmaktadır. İlerleyen miyop ve astigmat, kornea incilmesi ve sivrileşmesi ile belirti vermeye başlayan keratokonus hastalığına, çok özel tetkiklerle erken dönemde teşhis konulabilir.

Hastalığın başlangıcından itibaren remisyon ve nükslerin olduğu bir ilerleme söz konusudur ancak bu ilerlemenin ne şiddette, ne kadar süre devam edeceğini tahmin etmek zordur. Çoğunlukla hastalığın 4. dönemde stabilize olduğu gözlenmiştir [5,6]. Hastalık başlangıçta bir gözde başlasa da zamanla diğer gözde de hastalığın belirtileri ortaya çıkar. Ancak diğer gözün ne zaman etkileneceğini tahmin etmek zordur. Rabinowitz ve arkadaşlarının yaptığı bir çalışmada, bir gözünde keratokonus olan hastaların %50'sinde 16 yıl içinde diğer gözde de keratokonus oluşacağı bildirilmektedir [7].

Keratokonus teşhisi günümüzde eskiye kıyasla çok daha önemli bir hale gelmiştir. Bunun başlıca nedenlerinden biri refraktif lazer cerrahisinin çok yaygınlaşmış olmasıdır. Refraktif cerrahiye aday hastalarda keratokonus görülme sıklığının genel

popülasyona oranla çok daha fazla olduğunu bildiren çalışmalar mevcuttur [8,9]. Refraktif cerrahiye aday hastalarda normal popülasyona göre miyopi ve astigmatizma görülme sıklığı daha fazla olduğundan aslında bu beklenen bir durumdur. Keratokonus hastalarında refraktif lazer cerrahisi sonrası ektazi görülme riski çok fazla olduğundan keratokonusun başlangıç dönemlerinde bile doğru teşhis edilebilmesi gerekmektedir. Refraktif lazer cerrahisi öncesi korneal topografi yapılmasına rağmen keratokonus teşhisi konulamayıp sonrasında ektazi gelişen vakalar olması, keratokonus teşhisinde yeni yöntemlerin geliştirilmeye çalışılmasının nedenini açıklamaktadır [10].

Bu çalışmada amaç keratokonus hastalığının teşhisi için istatistiksel analizlere dayalı yeni bir sınıflandırma metodu tanımlamak ve hastalığın teşhisinin zeki bir sistem tarafından gerçekleştirilmesini sağlamaktır. CUDA paralel işlemci mimarisi uygulamaya eklenerek daha iyi bir performans sağlanmıştır. Çalışma 6 ana bölüm halinde tasarlanmıştır.

Birinci bölümde keratokonus hastalığının genel tanımı ve sınıflandırması üzerinde durulmuştur. İkinci bölümde; veri madenciliği ve veri hazırlama süreci teorik olarak incelenmiştir. Üçüncü bölümde; sınıflandırmada kullanılan istatistiksel analiz teknikleri teorik olarak anlatılmıştır. Dördüncü bölümde uygulamada kullanılacak sınıflandırma teknikleri incelenmiştir. Beşinci bölümde paralel programlama yöntemleri teorik olarak anlatılmıştır. Altıncı bölümde keratokonus hastalığının zeki sistemler ve paralel programlama yöntemleri ile teşhisini sağlayan yüksek performanslı bir uygulama gerçekleştirilmesi sağlanmıştır.

Sonuç bölümünde ise bu uygulama sırasında edinilen deneyimler ve bulgular paylaşılmıştır. Bu bölümde sistemin keratokonus hastalığının teşhisinde kullanılabilirliği ve performansı değerlendirilmiştir.

## **1.1. LİTERATÜR**

Keratokonusun erken topografik değişiklikleri ilk olarak 1938 yılında Amsler tarafından tanımlanmıştır. Amsler; fotoğrafik plasido disk kullanarak, hastalarda

daha klinik ve biyomikroskopik deęişiklikler görülmeden bazı deęişiklikler olduğunu fark etmiştir. Böylelikle keratokonusu sadece plasido diskte deęişikliklerin görüldüğü erken evre ve klinik olarak belirgin özelliklerin görüldüğü geç evre olarak ikiye ayırmıştır. Erken evre keratokonusu da kendi içinde keratokonus fruste ve erken veya hafif keratokonus olarak iki kısma ayırmıştır [6].

1980'li yılların başında bilgisayar destekli videokeratoskopilerin kullanılmaya başlanması, keratokonus teşhis ve takibinde bir devrim niteliğinde olmuştur. Videokeratoskopilerde keratokonus teşhisi için gerekli olan üç görünüm tanımlanmıştır: Çevresi dikliği giderek azalan konsantrik halkalarla çevrili dik bir alan, korneanın inferior ve süperiorunda güç asimetrisi ve horizontal meridyenin üstünde ve altında en dik aksların çarpıklığı (SRAX) [6].

Wilson ve Kylce'in yaptığı bir çalışmada kornea topografisinde konuş benzeri görüntü sergileyen ama klinik olarak keratokonus özellikleri göstermeyen refraktif cerrahi adaylarının hepsinde sonradan keratokonus teşhisi konulduğu bildirilmiştir [8].

Raiskup-Wolf ve ark.'nın keratokonus olgularında maksimum 6 senelik takibi olan çapraz bağ tedavisinin sonuçlarını yayınladıkları çalışmalarında, 3 sene sonunda korneal eğimde ortalama 4.84 D'lik bir düşüş ve buna eşlik eden en iyi düzeltilmiş görme keskinliği değerinde artış olduğu belirtilmiştir [11].

Keratokonus tanısı ve sınıflandırmasındaki etkinliği artırmak amacıyla keratokonus tanısında ve takibinde altın standart olan korneal topografi cihazı ve aberrometre, pakimetre gibi çeşitli yardımcı cihazlar kullanılarak çeşitli nicel parametreler ve indeksler geliştirilmiştir. İlk olarak Rabinowitz; santral keratometri (Santral K), inferior ve süperior korneada eğim asimetrisi (I-S değeri) ve sağ-sol göz arasındaki korneal kırma gücü farkından oluşan üç değer kullanarak normal korneaları keratokonuslu kornealardan ayırmaya çalışmıştır [12]. Rabinowitz ve Mc Donnel keratokonus ayrımı için santral keratometri I-S değerlerini kullanarak bir metod belirlemişlerdir [13]. Rabinowitz'in bu çalışmalarına çeşitli itirazla gelmiştir.

Maeda ve ark. santral keratometrinin keratokonus tespitinde yetersiz bir kriter olduğunu; çünkü emetrop olup santral korneal kırıcılık güçleri 48-50 D olan bir çok aile gözlemlediklerini belirtmişlerdir. Yine bu yazarlar, korneal greftlerde ve refraktif cerrahiler sonrası kornealarda olduğu gibi bazı durumlarda da keratokonustakine benzer biçimde kornea inferior ve süperiorunda asimetri gözlenebileceğini ve keratokonusta dikleşmenin alt periferle sınırlı olmadığını belirtmişlerdir. Son olarak iki göz arasındaki kırma gücü farkının, sağlam göz cerrahi geçirmiş olabileceğinden ve sağlam gözün topografi bulguları olmayabileceğinden keratokonusun iyi bir belirleyicisi olmayacağını söylemişlerdir [14].

Li ve ark'nın yaptığı bir çalışmada da santral K değerinin keratokonus ayırımında iyi bir öngörü değeri olmadığı belirtilmiştir [15].

Simolek ve Kyle'in yaptıkları bir çalışmada keratokonus ayırımında, KCI ve modifiye Rabinowitz-Mc Donnel testinin duyarlıklarının hemen hemen aynı olduğu fakat bu ikisinin duyarlıklarının ortalama SimK'dan daha fazla olduğu, aynı şekilde bu iki testin özgünlüklerinin birbirlerine yakın olmakla beraber ortalama SimK'dan belirgin şekilde daha fazla olduğu bulunmuştur. Smolek ve Kyle'in yaptığı çalışmada ortalama SimK'nın kesim noktası 43,53D duyarlılığı 76% ve özgünlüğü 80% olarak belirlenmiştir [16].

Rabinowitz ve Rasheed, 2000 senesinde yayınlanan çalışmalarında keratokonus tanısı için kornea topografilerinde kullanılacak yeni bir kantitatif yöntem tanımlamışlardır. KISA% indeksi; santral K, I-S değeri, düzenli korneal astigmatizmayı temsil eden AST indeksi ve korneadaki düzensiz astigmatizmayı temsil eden SRAX indeksi kullanılarak hesaplanan bir indekstir [17].

Keratokonusta geçmişten günümüze yapılan wavefront aberasyonlarıyla alakalı çalışmalar keratokonik kornealar ile normal kornealar arasında bazı belirgin farklılıklar olduğunu ortaya koymuştur. Bu farklılıklardan keratokonus teşhisi ve sınıflandırılmasında yararlanılmaya çalışılmıştır. Şimdiye kadar yapılmış olan çalışmaların birçoğunda keratokonuslu gözlerde yüksek sıralı aberasyonların

özellikle de koma ve koma benzeri aberasyonların normal gözlerdekine kıyasla daha yüksek olduğu bulunmuştur [18,19].

Lagana, Cox ve Potvin, keratokonuslu gözlerde videoabereskopla total oküler aberasyon ölçümünün keratokonus teşhisinde ve progresyonunun izlenmesinde hassas ve güvenilir bir yöntem olduğunu iddia etmişlerdir [20].

Barbero ve ark. ise hem videoabereskopi hem de videokeratoskopinin keratokonus teşhisinde faydalı olduğunu iddia etmişlerdir [21].

Gobbe ve Guillon ise; aberasyonların büyük pupil çaplarındaki aberasyon ölçümünde; videokeratoskopların videoabereskoplara göre daha güvenilir olduğunu belirtmişlerdir [22].

Keratokonuslu gözlerdeki wavefront aberasyonlarının normal gözlere kıyasla farklılık göstermesinden yola çıkılarak, wavefront aberasyonları kullanılarak keratokonus teşhisi için indeksler oluşturulabileceği fikri oluşmuştur. Schwiegerling ve ark. korneal topografideki yükseklik verilerini Zernike polinomlarına uyarlayıp, üçüncü sıralı Zernike katsayılarından Z3 indeksini geliştirmişlerdir [23].

Twa ve ark. yine korneal topografideki yükseklik datalarının Zernike polinomlarına uyarlayıp, otomatik karar verme ağacıyla keratokonus için C4.5 indeksini geliştirmişlerdir [24].

Mahmoud ve ark. keratokonus tanısında kullanmak amacıyla, birçok topografi cihazındaki kornea ön yüz haritalarının hepsinde birden kullanılabilecek bir indeks geliştirmişlerdir. CLMI (The Cone Location and Magnitude Index), kornea topografi haritasında en dik alanın bulup bu en dik alanı haritanın geri kalan bölgeleriyle karşılaştırarak en dik alanın bir konuş olup olmadığına karar verir denebilir. CLMI büyüklüğü, en dik alanın ortalama eğiminin haritanın geri kalanının ortalama eğimine göre farkını temsil eder. Aksiyel haritadan elde edilen bu CLMI büyüklüğü kullanılarak hastanın keratokonus olma ihtimali hesaplanır (PPK: Percent Probability Keratoconus). Yapılan çalışmada PPK değeri 20%'den küçük çıkanlar normal kabul

edilirken, PPK deęeri 20% ile 45% arasında ıkanlar Őüpheli keratokonus, PPK deęeri 45% üzerinde ıkanlar ise keratokonus kabul edilmiŐtir [25].

Jafri ve arkadaŐları LASIK sonrası ektazi geliŐen hastaların oęunluęunun korneal topografi ile tanı konamamıŐ keratokonus hastaları olduęunu belirtmiŐlerdir [26]. Bu nedenle refraktif cerrahi adaylarının keratokonus hastalarından ayırımında, klinisyenin elinde hastanın keratokonus olma riskini sayısal olarak ifade edebileceęi ve deęerlendirebileceęi eŐitli parametrelerin ve indekslerin olması nemlidir.

Keratokonus genellikle izole grlen bir klinik durum olmasına raęmen literatrde birliktelik gsterdięi rapor edilen birok hastalık vardır. Bunlardan keratokonusla birliktelięi en sık rapor edilen hastalıklar arasında Down sendromu, Leber'in konjenital amorozisi, Marfan sendromu, mitral valv prolapsusu ve retinitis pigmentoza gibi hastalıklar sayılabilir [27,28]. Keratokonusla birliktelięi konusunda zerinde en ok durulan konulardan biri ise atopi ve kronik gz kaŐıma hikyesidir. Keratokonusla atopi hikyesi birliktelięinin ok sık olduęunu bildiren eŐitli yayımlar mevcuttur [29,30]. Ancak bazı yazarlar keratokonusla atopinin deęil atopinin neden olduęu kronik gz kaŐıma hikyesinin birliktelięinin sık grlen bir durum olduęunu ve kronik gz kaŐımanın keratokonusun etyopatogenezinde yer alan nemli bir durum olabileceęini belirtmiŐlerdir [31,32].

Ender GLEMEZ, EskiŐehir Osmangazi niversitesi Tıp Fakltesi'nden almıŐ olduęu verilerle yaptığı alıŐmasında normal bireyler ve keratokonus olgularında PENTACAM HR ve ORBSCAN II cihazlarının lm hassasiyetini incelemiŐtir [33].

Erdin UŐKUL, Dokuz Eyll niversitesi Tıp Fakltesi'nden almıŐ olduęu veriler zerinde wavefront analizr ile keratokonus taramasında artıŐ gsterdięi bilinen aberasyonları deęerlendirerek farklı grme rehabilitasyonu yntemlerini kendi aralarında ve normal kontrol grubu ile kıyaslamıŐtır [34].



Bülent ÇÜÇEN keratokonus tedavisinde kullanılan Crosslinking cihazının deneysel olarak üretilmesi, güvenilirliğinin ve etkinliğinin değerlendirilmesi üzerine çalışmıştır [35].

Arbalaez ve ark. yapmış oldukları çalışmada Placido korneal topografi ile birlikte Scheimpflug kamera tarafından sağlanan kornea ölçümlerine dayanarak keratokonus tanısı için destek vektör makinelerini kullanmışlardır [36].

Mesplie ve ark. keratokonus hastalığının ilerleme hızını yetişkin ve çocuk hastalar üzerinde 2 yıllık bir süre içerisinde incelemişler ve keratokonus hastalığının çocuk hastalarda yetişkinlere oranla daha hızlı arttığını görmüşlerdir [37].

Uçakhan ve ark. subklinik keratokonuslu, keratokonuslu ve normal gözler üzerinde Scheimpflug görüntüleme parametrelerinin etkinliğini belirlemeye çalışmışlardır [38].

Timuçin ve arkadaşları 2013 yılında yapmış oldukları çalışmada, kontakt lens kullanmayan keratokonus hastalarında kornea endotel hücre yoğunluğunu değerlendirmişlerdir [39].

Vega- Estrada ve arkadaşları 2013 yılında yapmış oldukları çalışmada, keratokonus tedavisi için görsel, refraktif ve aberrometric düşüklüğe dayanarak intrakorneal halka bölümleri sonucunu analiz etmişlerdir [40].

André ve arkadaşları 2013 yılında yapmış oldukları çalışmada mevcut GPU programlama stratejileri ve profil odaklı geliştirme ve gelecek trendler hakkında genel bir bilgi vererek süreci kolaylaştırmayı amaçlamışlardır [41].

Qi Li ve arkadaşları 2013 yılında yaptıkları çalışmada, Grafik İşlem Birimi (GPU) üzerinde aynı anda birden fazla eğitim işlemini çalıştıran yeni bir paralel destek vektör makinesi uygulaması gerçekleştirmişlerdir ve sonuç olarak paralel olarak çalışan SVM'yi LIBSVM aracı ile karşılaştırdıklarında duruma göre 10-100 kat daha hızlı olduğunu görmüşlerdir [42].

Bressan ve arkadaşları 2013 yılında, rastgele grafik üretimi için GPU üzerinde etkin ve verimli paralel bir çalışma gerçekleştirmişlerdir ve rastgele grafikler üreten kapsamlı deneysel değerlendirmeler sonucunda önemli bir hız kazancı elde etmişlerdir [43].

Upadhyaya 2013 yılında paralel yaklaşımların makine öğrenmesinde kullanılması ile ilgili kapsamlı bir araştırma yapmıştır. Sonuç olarak paralel mimarilerin görüntü işleme, oyun vb. gibi alanlarda hız kazanımının çok yüksek olduğunun çok açık olduğunu fakat makine öğrenmesi ve grafik geçişi gibi alanlarda da önemli oranda hız kazancı sağladığını görmüştür [44].

Kamiya ve arkadaşları 2014 yılında yapmış oldukları çalışmada keratokonuslu gözlerle ait keratometri, pakimetri ve korneal yükseklik değerlerini hastalığın klinik evrelerine göre değerlendirmişlerdir ve ön ve arka yükseklik değerlerini hastalığın teşhisi için en etkili parametreler olarak belirlemişlerdir [45].

Kanellopoulos ve arkadaşları 2014 yılında yapmış oldukları çalışmada keratokonus hastaları ve normal gözlerde ön segment optik koherens tomografi (AS-OCT) cihazı kullanılarak elde edilmiş epitel kalınlık dağılım karakteristiklerini araştırmışlar ve OCT epitel haritalamanın erken ve ilerleyen keratokonus teşhisinde kritik bir potansiyele sahip olduğunu belirtmişlerdir [46].

Oruçoğlu ve arkadaşlarının 2015 yılında yapmış oldukları çalışmada normal ve keratokonuslu hastalara ait ön ve arka kornea yüzey parametrelerini, keratokonus indekslerini ve Pentacam Scheimpflug korneal topografi cihazından elde edilen gelişmiş yükseklik harita verilerini kullanmıştır. Keratokonuslu gözleri normal gözlerden ayırt etmek için bu parametrelerin duyarlılık ve özgüllüğünü karşılaştırmışlardır [47].

Belin ve arkadaşları 2015 yılında yapmış oldukları çalışmada keratokonus hastalığı için yeni bir sınıflandırma yaklaşımı geliştirmişlerdir. 5 aşamalı olarak belirledikleri çalışmalarında standart ön yüz parametrelerine ek olarak arka eğrilik, ince pakimetri ve gelişmiş korneal görüntüleme parametrelerini de eklemişlerdir [48].

Gomes ve dünyanın dört bir tarafından diğerk oftalmoloji uzmanlarının katılmış oldukları 2015 yılında yapılan çalışmada keratokonus ve ektatik hastalıkların tanımı, klinik yönetimi ve cerrahi tedaviler konusu tartışılmıştır. Keratokonus ve ektatik hastalıkların yönetimi için ortak teşhis yöntemleri oluşturulmuştur. Keratokonus hastaları için cerrahi ve cerrahi olmayan tedavilerde aşamalı bir yaklaşım sunulmuştur [49].

Toprak ve arkadaşları 2015 yılında yapmış oldukları çalışmada Scheimpflug korneal topografî cihazından elde etmiş oldukları topografik ve pakimetrik parametrelerin keratokonus hastalığının teşhisi için önemini araştırmışlardır. Her ne kadar maksimum keratometri ve ince pakimetri değerleri hastalığın teşhisinde yüksek tanısallık yeteneğesi sahip görünse de tek parametre kullanımının ayırıcı tanıda yetersiz kaldığını değerlendirmektedirler [50].

Du ve arkadaşları 2015 yılında yapmış oldukları çalışmada keratokonus hastalığının ilerleme durumunu değerlendirmek için temel göstergelerin tanısallık etkinliğini değerlendirerek karakteristik göstergeler bulmaya çalışmışlardır. Hastalığın başlangıç aşamasında kalınlık ve arka alt üst değeri önemli tanısallık değerlere sahip iken orta evrede yükseklik değeri, hastalığın şiddetli olduğu aşamada ise ön eğrilik indeksleri en karakteristik göstergeler olarak tanımlanmıştır [51].

Cavas-Martínez ve arkadaşları 2016 yılında yapmış oldukları çalışmada korneanın morfolojik karakterizasyonu için kullanılan korneal topograf teknolojilerini ve keratokonus teşhisinde kullanılan ana indeksleri tanımlamışlardır [52].

## BÖLÜM 2

### VERİ MADENCİLİĞİ

#### 2.1. VERİ MADENCİLİĞİ TANIMI

Veri madenciliği, büyük ve karmaşık veri kümelerindeki ilişki ve örüntülerin açığa çıkarıldığı bir bilgi keşif sürecidir ve bir veri analiz tekniğidir. Kendi başına bir çözüm değildir. Aşağıda veri madenciliği ile ilgili çeşitli tanımlar yer almaktadır.

Büyük veri tabanlarında gizli kalmış örüntüleri çıkarma sürecine veri madenciliği adı verilmektedir. Geleneksel yöntemler kullanılarak çözülmesi çok zaman alan problemlere veri madenciliği süreci kullanılarak daha hızlı bir şekilde çözüm bulunabilir [53].

Veri Madenciliği, büyük miktardaki veri yığını içerisinde gelecekle ilgili tahmin yapmamızı sağlayacak, bağıntı ve kuralların bilgisayar programları kullanılarak aranması olarak da ifade edilebilir [54].

Veri madenciliği, ham verinin tek başına sunamadığı bilgiyi çıkaran veri analizi sürecidir [55].

Başka bir tanım ise şöyledir; eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır [56].

Fayyad'ın (2006) tanımlamasına göre ise veri madenciliği, veri tabanlarından geçerli, önceden bilinmeyen, güvenilir, potansiyel olarak kullanışlı ve nihayetinde anlaşılabilir örüntülerin çıkarılması işlemidir [57].

Veri Madenciliđi; geniş veritabanlarından bilgi çıkartabilmek amacıyla makine öğrenmesi, örüntü tanıma, istatistik, görselleştirme gibi alanların tekniklerini bir araya getiren disiplinler arası bir alandır [58].

## **2.2. VERİ MADENCİLİĐİNİN UYGULAMA ALANLARI**

Azalan bilgi işleme maliyeti, verinin toplanması ve saklanmasıdaki kolaylık, veritabanı yönetim sistemi teknolojilerindeki ilerlemeler, kullanılabilecek analitik araçların oldukça fazlaşmasıyla birlikte veri madenciliđi uygulamalarına olan ilgi artmaktadır [59].

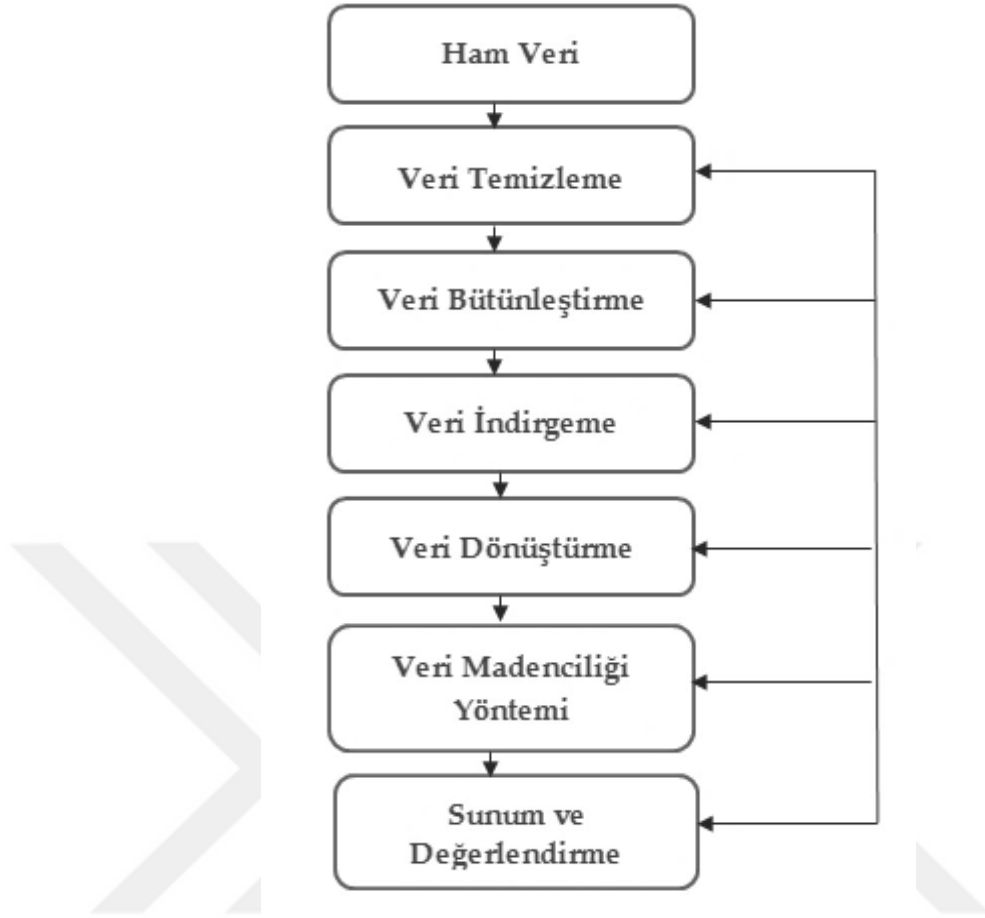
Veri madenciliđi yöntemini günümüzde karar verme sürecine ihtiyaç duyulan birçok alanda uygulamak mümkündür. Özellikle eğitim, pazarlama, bankacılık, sigortacılık, borsa, tıp, endüstri sektörlerinde yaygın olarak kullanılmaktadır.

## **2.3. VERİ MADENCİLİĐİ SÜRECİ**

Veri hazırlama ile ilgili işlemler veri madenciliđi kavramı içinde düşünülürse, veri madenciliđinin bir süreç olarak değerlendirmesi gerekir. Veri madenciliđi süreci aşağıda belirtilen adımları içermektedir [60]:

1. Veri temizleme
2. Veri bütünleştirme
3. Veri indirgeme
4. Veri dönüştürme
5. Veri madenciliđi algoritmasını uygulama
6. Sonuçları sunum ve değerlendirme

Veri madenciliđi sürecinin aşamaları Şekil 2.1’de ifade edilmiştir. Burada ham veri, henüz herhangi bir işleme tabi tutulmamış (ön işlem öncesi) veridir. Ham veri, sırasıyla temizleme, bütünleştirme, indirgeme ve dönüştürme gibi işlemlerden geçtikten sonra, veri madenciliđi algoritması uygulanarak sonuçlar elde edilmektedir.



Şekil 2.1. Veri madenciliği süreci.

## 2.4. VERİ HAZIRLAMA

Veri tabanları içindeki verinin ve bu veriye dayalı olarak elde edilen veri madenciliği sonuçlarının kalitesinin artırılması, veriyi analize hazırlarken dikkat edilmesi gereken en önemli noktadır. Veri madenciliği işlemlerini kolaylaştırmak ve verimliliği arttırmak için veri tabanındaki veriler, bir “ön işleme” aşamasından geçirilir [60]. Veri madenciliği öncesinde verinin hazırlanması süreci olarak da kabul edilen bu işlemler özellikle veri tabanındaki bozuk değerleri ve veriler arasındaki tutarsızlıkları kaldırmayı amaçlamaktadır.

### 2.4.1. Veri Temizleme

Kirli veriler veri madenciliğinde karmaşıklığa neden olmakta ve sonuçlardan kuşku duyulmasına, güvensizliğe neden olmaktadır. Bu nedenle uygulamaya geçmeden

önce veri ön işleme adımı olan veri temizleme işlemlerinden sonra verilerin kullanılması gerekmektedir. Veri analizlerine veriyi temizleyerek ve uygun olmayan verilerin ayıklanması suretiyle başlanır. Bu tür işlemlerin ardından veri analiz edilir ve sonuçlar elde edilir [61,62]. Veri temizleme sürecinde, kayıp olan değerlerin yerine yenilerinin bulunması, gürültülü (noisy) ve uyumsuz verilerin ayıklanması gibi işlemlere yer verilir. Veri tabanındaki kayıp verilerin telafi edilebilmesi için bazı yollara başvurulur. Veri kümesini kayıp değerlerden arındırmak için kullanılan yöntemlerden en önemlileri ve yaygın biçimde kullanılanları şunlardır [60]:

1. Kayıp verilerin yer aldığı kayıtlar tümüyle yok edilir. Bu şekilde yok edilen sorunlu kayıtların sayısı fazla ise bu yöntem kullanışlı olmaz.
2. Kayıp değerler için genel bir sabit kullanılır. Bu yöntemde “BİLİNMIYOR” veya “∞” gibi sabit bir değer tanımlanabilir. Ancak veri madenciliği programları bu değeri ortak bir değer gibi algılayabilir. Bu nedenle çok tercih edilen bir yaklaşım değildir.
3. Sayısal değerlere sahip nitelikler için hesaplanacak ortalama değer kayıp değerler yerine kullanılabilir.
4. Kayıp değer yerine aynı sınıfa ait tüm örnekler için değişkenin ortalaması kullanılabilir.
5. Kayıp değer mevcut verilere dayanarak en uygun değer kullanılarak tamamlanabilir. En uygun değer belirlenebilmesi için regresyon veya karar ağacı gibi teknikler kullanılabilir. Örneğin veri setindeki diğer müşteri verilerini kullanarak gelir ile ilgili kayıp değerleri öngörmek için bir karar ağacı oluşturmak gibi.

#### **2.4.2. Veri Bütünleştirme**

Veri madenciliği uygulamalarının kullandığı veri ambarının oluşturulması esnasında değişik kaynaklardan (veritabanları, veri küpleri, metin dosyaları, vb.) elde edilen veriler arasında uyum sağlamak için verilerin bütünleştirilmesi işlemlerinin yerine getirilmesi beklenir. Bu adım, bütün veri kaynaklarını karakteristiklerine, özelliklerine ve toplanma seviyelerini eşleştirerek tutarlılık sağlayarak bütünleştirilir.

Farklı kaynaklardaki verilerin bütünleştirilmesi, örneğin bir kaynakta müşteri numarasının bir başka kaynaktan gelen müşteri numarası ile aynı tanıma sahip olması amacıyla yapılır. Veri tabanları ve veri ambarlarında veri türlerinin tanımlandığı metadata bu sorunu çözüme yardımcı olur [60]. Başka örnek de, bir bankanın farklı veritabanları üzerinde kredi kartları ile ilgili hesaplara sahip olması üzerine verilebilir. Bu hesaplarda müşteri isimleri farklı biçimlerde kaydedilmiş olabilir [63]. Özetle ifade etmek gerekirse veri bütünleştirilmesi veri kalitesi ile ilgili sorunların çözümlenmesi için kullanılan bir yöntemdir [64].

### 2.4.3. Veri İndirgeme

Bilindiği gibi veri madenciliği uygulamalarının bazılarında milyarlarca kayıt ve binlerce nitelik ile karşılaşmak mümkün olabilmektedir. Bu durumda verinin temel özelliklerini kaybetmeden, bir kısmını atarak azaltma yoluna gidilip gidilmeyeceği sorunu ile karşılaşılır [65]. Bu sorunu azaltmanın yolu olan birleştirmede önemli noktalardan biri de veri indirgemedir.

Değişkenlerdeki tutarsızlıklar elde edilen veri setinde fazlalıklara neden olabilir. Bu fazlalıkları tespit etmek için korelasyon analizinden yararlanılmaktadır. Örneğin korelasyon analizi ile müşteri kimliği ile müşteri numarası arasında yüksek bir ilişki bulunursa değişkenlerden biri veri deposundan çıkarılarak indirgeme yapılır. Veri madenciliği yöntemleri hesaplama ağırlıklı olduğundan, veri indirgemesiyle doğal olarak maliyetlerin azalması sağlanabilmektedir. Veri indirgeme işlemleri sonucunda aşağıda belirtilen yararlar elde edilir [62]:

1. Öğrenme (model) maliyetleri azalır.
2. Öğrenme (model) performansı artar.
3. İlgisiz boyut sayısı azalır.
4. Gereksiz boyut sayısının azaltılır.

İlgisiz ve gereksiz boyut sayısı sorununun üstesinden gelmek için;

1. Nitelik seçimi,
2. Kayıt seçimi.



işlemlerine başvurulabilir. Veri madenciliği araştırmasında hangi niteliklerin kullanılacağına karar verme işlemi “nitelik seçimi” olarak isimlendirilir. Birbirini andıran niteliklerin hepsini kullanmak yerine benzer nitelikleri göz ardı edip diğerlerinden anlamlı derecede farklı olanları dikkate almak yeterli olabilir.

#### **2.4.4. Veri Dönüştürme**

Bazı durumlarda veriyi veri madenciliği çözümlmelerine aynen katmak uygun olmayabilir. Değişkenlerin ortalama ve varyansları birbirlerinden önemli ölçüde farklı olduğu takdirde büyük ortalama ve varyansa sahip değişkenlerin diğerleri üzerindeki baskısı daha fazla olur ve onların rollerini önemli ölçüde azaltır. Ayrıca değişkenlerin sahip olduğu çok büyük ve çok küçük değerler de çözümlmelerin sağlıklı biçimde yapılmasını engeller. Bu nedenle bir dönüşüm yöntemi kullanarak söz konusu değişkenlerin normalleştirilmesi ya da standartlaştırılması gerekmektedir [66].

#### **2.4.5. Veri Madenciliği Yönteminin Uygulanması**

Bu evrede sınıflandırma, kümeleme ve birliktelik madenciliği gibi veri madenciliği yöntemlerinden, çalışmanın amacına ve üzerinde çalışılan veri kümesine uygun olanı belirlenerek uygulanmaktadır. Burada yöntemler için geliştirilmiş algoritmaların özelliklerine dikkat edilmeli ve veriye en uygun olan algoritma seçilmelidir [66].

#### **2.4.6. Sonuçların Değerlendirilmesi Ve Sunulması**

Bu evrede ise veri madenciliği yöntemleri ile elde edilen sonuçların düzenlenmesi, yorumlanması ve sunuma hazır hale getirilmesi işlemleri gerçekleştirilmektedir. Sonuçları sunarken anlaşılabilirliğinin artırılması için veri görselleştirme tekniklerinden yararlanılabilmektedir [67].

## BÖLÜM 3

### SINIFLANDIRMADA KULLANILAN İSTATİSTİKSEL ANALİZ TEKNİKLERİ

Bu çalışmada amaç keratokonus hastalığının teşhisi için istatistiksel analizlere dayalı yeni bir sınıflandırma metodu tanımlamaktır. Hiyerarşik kluster (küme) analizi adı verilen istatistik yöntemini kullanarak benzer özelliğe sahip objeleri veya olayları gruplandırarak incelemek mümkündür [68]. Analize konu olan her olay birbirine benzeme ya da benzememe eğiliminde olduğundan, kümeleme analizi yöntemiyle birbirine benzer homojen grupların tespiti matematiksel olarak yapılabilir. Diğer istatistik tekniklerinde olduğu gibi bu yöntemde de analizler sırasında hangi veri tiplerinin kullanılacağı, mesafenin hangi matematiksel metotla hesap edileceği ve hangi kriterler kullanılarak grup sayısının belirleneceğine önceden karar verilmelidir [69].

Doğru seçilmeyen değişken yanlış sonuçların ortaya çıkmasına neden olacağı için kümeleme analizi sırasında hangi değişkenlerin kullanılacağına seçimi önemlidir. Ayrıca kullanılması gereken değişkenler analiz dışı bırakılırsa eksik ve yanlış sonuçlar ortaya çıkar. Bu nedenle kümeleme analizinden önce hangi değişkenlerin kullanılacağına karar verirken çok değişkenli istatistik tekniklerinden faktör analizi kullanılmıştır.

#### 3.1. FAKTÖR ANALİZİ

Faktör analizi; birbiriyle ilişkili çok sayıda değişkeni az sayıda, daha anlamlı, kolay anlaşılabilir ve birbirinden bağımsız faktörler haline getiren ve yaygın olarak kullanılan çok değişkenli istatistik tekniklerinden biridir.

Faktör analizi çok sayıda değişken arasından diğer analizlerde kullanılacak temsili değişkenleri belirlemeye yardım eder. Orijinal veri setinden sağlanan faktörler bağımlı yöntemlerden regresyon, korelasyon ve diskriminant analiz ve iç bağımlı yöntemlerden kümeleme analizi gibi geniş bir yelpazede kullanılmaktadır. Faktör analizi ölçülebilen ve gözlenebilen çok sayıdaki özellik arkasında yatan gerçek nedenleri yani gözlenemeyen ve ölçülemeyen gizli boyutları ortaya çıkarmaya yaramaktadır [70].

### **3.1.1. Veri Setinin Uygunluğunun Değerlendirilmesi**

Veri setinin faktör analizine uygun olup olmadığının değerlendirilmesi için üç yaklaşım vardır: Bunlardan birincisi korelasyon matrisi, ikincisi Kaiser-Mayer-Olkin (KMO) uygunluk testi, diğeri ise Barlett testidir.

Faktör analizinde korelasyon matrisi için istatistik temellerin yanında faktör analizinin uygulanabilirliğini kanıtlamak için korelasyon matrisinin yeteri kadar anlamlı korelasyonlara sahip olması gerekir çünkü değişkenler arasındaki korelasyonlar ne kadar yüksek ise, değişkenlerin ortak faktörler oluşturma olasılıkları o kadar yüksektir [71]. Korelasyon katsayıları %30'dan büyük olamayan değişkenlerin büyük bir olasılıkla faktör analizinden çıkartılması uygun olacaktır [70].

KMO testi, değişkenler arasındaki korelasyonları ve faktör analizinin uygunluğunu ölçen uygunluk testidir. Testin değeri 0-1 aralığında değişmektedir. KMO oranının 0.5'in üzerinde olması gerekir. Oran ne kadar yüksek olursa veri seti faktör analizi için o kadar iyidir denilebilir. KMO uygunluk testi için önerilen kriterler Çizelge 3.1'deki gibidir.

Barlett testi, korelasyon matrisinde değişkenlerin en azından bir kısmı arasında yüksek oranlı korelasyonlar olduğu olasılığını test eder. Analize devam edilebilmesi için "Korelasyon matrisi birim matristir" sıfır hipotezinin reddedilmesi gerekir. Eğer sıfır hipotezi reddedilirse, değişkenler arasında yüksek korelasyonlar olduğunu, başka bir deyişle veri setinin faktör analizi için uygun olduğunu gösterir.

Çizelge 3.1. KMO uygunluk testi için önerilen kriterler.

<b>KMO Ölçüsü</b>	<b>Önerilen Düzey</b>
0.90+	Mükemmel
0.80+	Çok İyi
0.70+	İyi
0.60+	Orta
0.50+	Kötü
0.50-	Kabul Edilemez

### 3.1.2. Faktör Türetme Modelleri Ve Uygun Faktör Analizi Yönteminin Seçimi

Faktör türetme modellerinin ilk bakışta iki temel amacı söz konusudur: Değişkenler arasındaki korelasyonları en iyi şekilde yeniden türetmek ve değişkenlerin varyansını en yüksek düzeyde açıklamaktır. Faktör türetme yönteminin seçimi araştırmanın amacına bağlıdır. Faktör analizi modelleri, Temel Bileşen Faktörü Yöntemi dışında Asal Eksen Faktörü, Görüntü Faktörü, Ağırlıksız En Küçük Kareler, Maksimum Olabilirlik ve Alfa Faktörü gibi yöntemlerdir.

Çalışmada gözlemlere ait değişkenler arasında ilişkisiz doğrusal bileşenlerin belirlenip bu bileşenlere göre yeni bir sınıflandırma yapmak amaçlandığı için Temel Bileşen Faktörü Modeli kullanılmış ve bu yöntem üzerinde durulmuştur.

#### 3.1.2.1. Temel Bileşen Faktörü (Principal Component Factoring)

Bu model, değişkenin açıklanan varyansının maksimum olmasını amaçlar. Modelde, değişken sayısına (p) eşit sayıda ve birbirinden bağımsız temel bileşenler, tüm değişkenlerin açıkladığı toplam varyansa maksimum katkıda bulunmaktadır. Model basit olarak:

$$x_j = b_{j1}F_1 + b_{j2}F_2 + b_{j3}F_3 + \dots + b_{jp}F_p \quad (j = 1, 2, \dots, p) \quad (3.1)$$

eşitliğindeki gibi yazılabilir.

Sonuçta p sayıda değişken, yine birbirinden bağımsız p tane temel bileşenle temsil edilir. Bu temel bileşenler sırasıyla toplam varyansa maksimum katkıda bulunurlar. Yani birinci temel bileşen en çok, diğer bileşenler ise gittikçe azalan miktarlarda toplam varyansa katılırlar. Bu sebeple, toplam varyansın büyük bir kısmı az sayıda temel bileşenle açıklanabilmektedir [72].

Bu yöntem, tahmin edilen ortak varyansların bire eşit olduğunu varsaymaktadır. Yani temel bileşen modeli, başta ortak faktör sayısı ile değişken sayısının eşit olduğunu, bu ortak faktörlerden birkaçının toplam varyansın önemli bir kısmını açıklayacağını ve geriye kalan diğer faktörlerin ise daha az açıklayıcılığı bulunan düşük varyansları göstereceğini varsaymaktadır [71].

### **3.1.2.2. Türetilen Ortak Faktör Sayısının Belirlenmesi**

Faktör analizinde üretilebilecek faktör sayısı en fazla değişken sayısı kadar olabilir. Değişken sayısı kadar faktör türetilirse her değişken bir faktörle temsil edilecek demektir ki bu durumda bir şey kazanılmaz. Burada amaç, değişkenler arasındaki ilişkileri en yüksek derecede temsil edecek az sayıda faktör elde etmektir [72]. Gerekli faktör sayısını belirlemek için toplam varyansın her bir faktör tarafından yüzde kaçının açıklandığına bakılması gerekir. Faktör analizinde değişkenler standardize edildiğinden toplam varyansa değişken sayısına eşittir. Faktörlerin açıkladığı varyans miktarına göre faktör sayısını belirleyen çeşitli kriterler vardır. Bu kriterler aşağıdaki gibi sıralanabilir:

1. *Varyansa Katılma (Kaiser, Öz değer, Eigenvalue) Kriteri:* Öz değer, hem faktörlerce açıklanan varyansı hesaplamada, hem de önemli faktör sayısına karar vermede dikkate alınan bir katsayıdır [73]. Bu kriter göre, öz değeri 1 ya da 1'den büyük olan faktörler önemli, 1'den küçük olan faktörler önemsiz varsayılmaktadır.
2. *Scree Test Kriteri:* Varyansa katılma (Kaiser) kriterinin sakıncasını ortadan kaldırmak için Cattell tarafından geliştirilmiştir [70]. Yöntem, faktörlerin öz değerlerine dayalı olarak çizilen yığın grafiğinin incelenmesine dayanır.

Grafikte dikey eksen öz deęer miktarlarını, yatay eksen ise faktörleri gösterir. Grafik, faktörlerin öz deęerleriyle eşleştirilmesi sonucunda bulunan noktaların birleştirilmesiyle elde edilir. Grafikte yüksek ivmeli, hızlı düşüşlerin yaşandıęı faktör, önemli faktör sayısını verir [73].

3. *Toplam Varyansın Yüzdesi Yöntemi*: Bu yöntemde her ilave faktörün toplam varyansın açıklanmasına katkısı %5'in altına düştüğünde maksimum faktör sayısına ulaşılmış demektir.
4. *Joliffe Kriteri*: Bu kritere göre 0.7'nin altındaki tüm faktörler modelden çıkarılır.
5. *Açıklanan Varyans Kriteri*: Bu kritere göre varyansın %90'ını açıklayan faktör sayısı yeterli kabul edilir.
6. *Faktör Sayısının Araştırmacı Tarafından Belirlenmesi*: Araştırmacının faktör sayısına kendisinin karar vermesidir [71].

### **3.1.2.3. Faktörlerin Döndürülmesi**

Faktör döndürülmesinde amaç, isimlenebilir ve yorumlanabilir faktörler elde etmektir. Döndürmede en çok kullanılan yöntem ortogonal (dik) döndürmedir. Ortogonal döndürmede elde edilen faktörler birbirleri ile korelasyon içinde deęildir. Ortogonal (dik) döndürme yöntemleri varimax (en çok kullanılan), equamax ve quartimax'tır. Promax ve direct oblimin yöntemleri ise oblik (eęik) döndürme yapılmak istendiğinde kullanılan tekniklerdir. Veri seti çok büyük olduğunda Promax döndürme, Direct oblimin döndürmeye tercih edilir.

## **3.2. KÜMELEME ANALİZİ**

Çok deęişkenli istatistiksel tekniklerden birisi olan kümeleme analizi, grup sayısı bilinmeyen ve gruplandırılmamış verilerin benzerliklerine göre sınıflandırılması amacıyla kullanılmaktadır. Kümeleme analizi verilerin birimlere veya deęişkenlere

göre birbirlerine benzerlikleri bakımından ayrı kümelerde toplanmasını sağlayan bir tekniktir. Kümeleme analizi birbirine benzer olan bireylerin aynı gruplarda toplanmasını amaçlaması bakımından diskriminant analizi ile birbirine benzer değişkenlerin aynı gruplarda toplanmasını amaçlaması nedeniyle de faktör analizi ile benzerlik göstermekte olup veri indirgeme özelliği vardır [74].

Diğer çok değişkenli istatistik analizlerde önemli olan verilerin normalliği varsayımı, kümeleme analizinde çok önemli olmayıp uzaklık değerlerinin normalliği yeterli görülmektedir [75].

Kümeleme işlemi belirlenen amaca göre, iki gözlemin veya iki değişkenin benzerlik(yakınlık) veya uzaklık ölçüsüne bakılarak yapılır.

Kümeleme teknikleri, araştırma problemlerinde geniş bir alanda uygulanır [76]. Tıpta hastalıkları, hastalık tedavilerini sınıflandırmada kümeleme çok kullanışlı bir yöntemdir. Böylelikle “herhangi bir hastalık hangi grupta, o grubun belirtileri nelerdir ve tedavi yöntemleri nelerdir?” bunları bilmek tedavi sürecinde fayda sağlar. Tıbbın psikolojiyle ortak alanı olan psikiyatri dalında, paranoya, şizofren, manik depresif gibi önemli hastalıkların belirtilerinin doğru kümelerde teşhis edilmesi, doğru tedavi için gereklidir. Sosyal bilimlerde de kümeleme analizi teknikleri uygulanmaktadır. Örneğin suç istatistiklerinde suçları sınıflandırmada kullanılabilir. Kümeleme analizinin en son uygulama alanı ise veri madenciliğidir, veri madenciliğinde büyük veri yığınlarından kurtulup özet veriler elde etmek hedeflenir. Bu aşamada kümeleme analizi kullanılır.

### **3.2.1. Kümeleme Analizinin Aşamaları**

Kümeleme Analizinin uygulama aşamaları aşağıdaki gibidir [77];

1. Değişkenlerin seçilmesi ve veri matrisinin belirlenmesi
2. Birimlerin birbirleriyle olan benzerlik ya da uzaklıklarını gösterecek uygun bir benzerlik/uzaklık ölçüsü ile benzerlik/uzaklık matrisinin oluşturulması
3. Uygun bir kümeleme tekniği ile benzerlik/uzaklık matrisine göre birimlerin uygun sayıda kümelere ayrılması

4. Oluşturulan bu kümelerin yorumlanması, kümelerin yapılarının kurulan hipotezlerle test edilmesi ve gerekli analitik yöntemlerin uygulanması.

### **3.2.2. Kümeleme Analizinde Değişken Türlerine Göre Benzerlik Ve Uzaklık Ölçüleri**

Bir veri setinde yer alan birimlerin kümelenmesi, temel bileşenler analizi, çok boyutlu ölçekleme ve kendinden düzenlenen haritalar gibi boyut azaltma işlemlerinin yapılabilmesi, bu birimlerin birbirleriyle olan benzerlikleri (similarity) ya da birbirine olan uzaklıkları (dissimilarity) kullanılarak gerçekleştirilmektedir [78].

Benzerlik ölçüleri maksimum 1 değerini benzerlik değeri olarak alabilirler. Benzerlik ölçülerinin değerleri arttıkça birimler arasındaki benzerlikler artar, azaldıkça da birimler arasındaki benzerlikler azalır. Uzaklık ölçüleri ise benzerlik ölçülerinin tam tersi bir yaklaşım sergilerler. Uzaklık ölçülerinin küçük olması birimlerin birbirilerine benzer olduğunu gösterir. Uzaklık ölçülerinde 0 maksimum benzerliği ifade eder. Uzaklık ölçülerinin değerleri arttıkça birimler arasındaki benzerlik azalır, azaldıkça da birimler arasındaki benzerlik artar [79].

Değişkenlerin kesikli ya da sürekli olmalarına ya da değişkenlerinin nominal, ordinal, aralık ya da oransal ölçekte olmalarına göre hangi uzaklık ölçüsünün ya da benzerlik ölçüsünün kullanılacağına karar verilir. Aşağıda değişken türlerine göre kullanılan uzaklık veya benzerlik ölçüleri anlatılmıştır.

#### **3.2.2.1. Aralık Ve Oransal Ölçekli Değişkenler**

Aralık ölçeği sayısal olarak ifade edilebilen, toplama ve çıkarma gibi matematiksel işlemleri mümkün kılan ölçeklerdir. Aralık ölçekte gerçek anlamda bir sıfır yoktur. Ölçüm farklılıklarının ve düzenlerinin önemli olduğu bir ölçektir. Aralık ölçeğine ısı ölçümlerinde kullanılan Celcius ölçeğini örnek verebiliriz. Bilindiği gibi suyun donma derecesi Celcius ölçeğinde  $0^{\circ}\text{C}$ ' dir.  $0^{\circ}\text{C}$  hiçlik anlamına gelmemektedir. 0 ısının yok olduğunu göstermeyen keyfi bir değerdir. Aralık ölçeklerinin oranları da herhangi bir anlam taşımaz.  $4^{\circ}\text{C}$ ,  $8^{\circ}\text{C}$  iki katı değildir, ancak ondan daha düşük bir



ıyı ifade etmektedir. Oran ölçeği ise, aralık ölçeğinin özelliğini taşıdığı gibi belli bir sıfır değerine sahip olan ölçeklerdir. Ölçümler arasında düzen ve uzaklık olduğu gibi ölçümler arasındaki oranda önemlidir. Örneğin, 20 metrekarelik bir alan 10 metrekarelik bir alanın 2 katı olduğu gibi. Aralık ve oran ölçekli veriler arasındaki uzaklık ve benzerlikler aşağıdaki gibi hesaplanır [80].

*Öklidyen ya da karesel öklit uzaklık ölçüsü:* Öklid uzaklığı en sık kullanılan uzaklık ölçüsüdür. Basit olarak çok boyutlu uzayda geometrik uzaklıktır ve

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (3.2)$$

biçiminde hesaplanır. Burada

$d_{ij}$  : i. ve j. birimin birbirine olan uzaklığı

$x_{ik}$  : i. birimin k. değişken değeri

$x_{jk}$  : j. birimin k. değişken değeri

$i = 1, \dots, n; \quad j = 1, \dots, n$  ve  $k = 1, \dots, p$ 'dir.  $n$  birim ve  $p$  değişken sayısıdır.

Karesel Öklid uzaklığı ise

$$d_{ij}^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2 \quad (3.3)$$

biçimindedir [67].

*Pearson Uzaklık Ölçüsü:* Pearson uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2 / s_k^2} \quad (3.4)$$

formülü ile hesaplanır. Bu formülde kullanılan  $s_k$ , uzaklığın hesaplandığı değişkene ait standart sapmadır. Bununla birlikte farklı gruplar hakkında önceden bilgi sahibi olunmadığı için, uzaklık hesaplanmasında  $s$  değerinin kullanılması doğru olmaz. Bu nedenle Pearson uzaklık ölçüsü yerine genellikle Öklidyen uzaklık ölçüsü tercih

edilir. Kümeleme analizinde kullanılacak olan değişkenler belirli önem derecelerine göre ağırlıklandırılmışlarsa, Pearson uzaklık ölçüsü formülü aşağıdaki gibi olur.

$$d(i, j) = \sqrt{\sum_{k=1}^p w_k (x_{ik} - x_{jk})^2 / s_k^2} \quad (3.5)$$

Pearson uzaklık ölçüsüne, “karesel Pearson uzaklık” ya da “standardize öklid uzaklığı” adı da verilir.

*City Block (Manhattan) Uzaklık Ölçüsü:* City-Block uzaklık ölçüsü, birimler arasındaki mutlak uzaklıkların toplamını alarak hesaplayan bir uzaklık ölçüsüdür ve

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}| \quad (3.6)$$

biçiminde ifade edilir. City-Block uzaklık ölçüsü uygulamada bazı sorunlara yol açmaktadır. Bu sorunlardan en belirginini City-Block uzaklık ölçüsünün değişkenler arasında korelasyon (ilişki) olmadığını varsaymasıdır. Eğer araştırma konusunda değişkenler arasında korelasyon varsa City-Block uzaklık ölçüsüyle hesaplanan uzaklık ölçüleri baz alınarak yapılan kümeleme anlamlı olmayacaktır. Sorunlardan bir diğeri de ölçüm yapılan değişkenlerin birimleri farklı olması durumunda standartlaştırılmış Karesel Öklid uzaklığıyla karşılaştırıldığında City-Block uzaklık ölçüsünün anlamlı sonuçlar vermediğidir [81].

*Minkowski Uzaklık Ölçüsü:* Uzaklıkların belirlenmesinde bir diğer kullanılan ölçüde Minkowski olarak bilinen uzaklık ölçüsüdür. Minkowski uzaklık ölçüsü

$$d_{ij} = \left[ \sum_{k=1}^p |x_{ik} - x_{jk}|^q \right]^{1/q} \quad (3.7)$$

olarak tanımlanmıştır. Minkowski uzaklık ölçüsü  $q=1$  için City-Block uzaklık ölçüsüne  $q=2$  için ise öklid uzaklık ölçüsüne eşit olacaktır. Belirtilmesi gereken bir durumda formülden de görüleceği gibi Minkowski uzaklık ölçüsü genel bir uzaklık ölçüsü olup, Öklid ve City-Block uzaklık ölçüleri Minkowski uzaklık ölçüsünün özel bir durumudur [82].

*Mahalanobis uzaklığı:* Kullanılan diğer bir uzaklık ölçüsü de, doğrudan birleştirme yapan, standart bir yöntem olan Mahalanobis Uzaklık ölçüsüdür. İki değişken arasında bir ilişki mevcut ise, bu iki değişken arasındaki kovaryans veya korelasyonu göz önüne alan Mahalanobis uzaklığının kullanılması gerekmektedir.

p değişkenli bir analizde i ve k gözlemleri arasındaki Mahalanobis uzaklık ölçüsü;

$$Md_{ik} = (x_i - x_k)S^{-1}(x_i - x_k) \quad (3.8)$$

biçimindedir, burada yer alan S, p×p tipinde örneklem kovaryans matrisini göstermektedir [83]. Mahalanobis uzaklığının avantajı, aykırı noktaları da hesaplamasıdır. Bu yönleriyle Mahalanobis uzaklığı, uzaklık ölçüleri arasında en avantajlı olanıdır denilebilir.

*Açısal Benzerlik Ölçüsü (Cosine Similarity Measure) :* Açısal benzerlik ölçüsü iki veri noktasının özellik vektörleri arasındaki açının kosinüsüdür. [+1, -1] arasında değerler alır. İki vektör arasındaki açısal benzerlik ölçüsü aşağıdaki gibi belirlenir.

$$s_{ij} = \frac{x_i^T x_j}{\sqrt{x_i^T x_i} \sqrt{x_j^T x_j}} \quad (3.9)$$

*Korelasyon Benzerlik Ölçüsü (Correlation Similarity Measure):* İki özellik vektörü arasındaki korelasyon değeri bu iki vektörün birbirine olan benzerlik derecesini gösterir. [+1, -1] aralığında değerler alır. Herhangi iki vektör arasındaki korelasyon benzerlik ölçüsü aşağıdaki gibi belirlenir.

$$s_{ij} = \frac{(x_i - \bar{x})^T (x_j - \bar{x})}{\sqrt{(x_i - \bar{x})^T (x_i - \bar{x})} \sqrt{(x_j - \bar{x})^T (x_j - \bar{x})}} \quad (3.10)$$

### 3.2.2.2. Nominal Ölçekli Değişkenler

Bu ölçekte kullanılan rakamlar veya isimler birimleri sınıflara veya kategorilere ayırmaktadır. Örneğin nüfusu cinsiyet özelliğine göre sınıflandırırken kadın ve erkek

gibi şıklar kullanılabilmesi gibi kadınlar için 1, erkekler için 0 rakamları birimlerin sınıflandırılması için de kullanılabilir. Bu ölçekte bulunan verilerin toplamları hiçbir anlam taşımaz. Nominal ölçekli değişkenler ikili (binary), ikili olmayan ölçekli değişkenler olarak ikiye ayrılırlar [80].

### *İkili Nominal Değişkenler*

Erkek - Kadın, Evet – Hayır, Olumlu – Olumsuz gibi iki seçenek alabilen değişkenler nominal ölçekli değişkenlerdir. İkili nominal değişkenlerde, aralık ve oransal ölçekli verilerde kullanılan Pearson, Öklidyen, Manhattan (City block) Minkowski gibi birimler arası uzaklıklarının kullanılması uygun değildir.

İkili nominal ölçekli değişkenlerde dört gözlü kontenjans tablolarından yararlanarak benzerlik ölçüleri elde edilebilir. Dört gözlü kontenjans tablosundan elde edilen 1-1, 0-0, 0-1, 1-0 eşleşmelerin frekansları kullanılarak sözü edilen benzerlik ölçüleri hesaplanır. Çizelge 3.2’ de dört gözlü kontenjans tablosu yer almaktadır.

Çizelge 3.2. İkili değişkenler için kontenjans tablosu.

		j. gözlem		
i. gözlem		1	0	Toplam
	1	a	b	a+b
	0	c	d	c+d
	Toplam	a+c	b+d	p= a+b+c+d

Çizelge 3.2’ de yer alan kontenjans tablosundaki frekans değerleri kullanılarak Çizelge 3.3’ de yer alan 6 adet benzerlik ölçüsü elde edilir.

Çizelge 3.3. Benzerlik ölçüleri.

Katsayı	Benzerlik Ölçüleri
Jaccard Benzerlik Katsayısı	$\frac{a}{a+b+c}$
Ochiai Benzerlik Katsayısı	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Rao Benzerlik Katsayısı	$\frac{a}{a+b+c+d}$
Basit Eşleşme Benzerlik Katsayısı	$\frac{a+d}{a+b+c+d}$
Binary Öklid Uzaklığı	$\sqrt{b+c}$
Binary Karesel Öklid Uzaklığı	$b+c$

### İkili Olmayan Nominal Değişkenler

İki seçenekten daha fazla seçeneğe sahip olan değişkenlerin uzaklıklarının hesaplanması ise şu formülle hesaplanır.

$$d(i, j) = \frac{p-m}{p} \quad (3.11)$$

p toplam değişken sayısı, m eşleşen değişken sayısı olmak üzere eşleşmeyen değişken sayısı (p-m) toplam değişken sayısına oranlanarak iki birim arasındaki uzaklık bulunur. Bu uzaklık 1' den çıkarılırsa, iki değişken arasındaki benzerlik katsayısı bulunur.

### 3.2.2.3. Ordinal Ölçekli Değişkenler

Bu ölçekte rakamlar, büyüklük, tercih gibi çeşitli özelliklerin sıralanmasında kullanılır. Markaların en çok beğenilenden en az beğenilene doğru 1' den başlayarak sıralanmasında ordinal ölçek kullanılmaktadır. Burada bireylerin markalardan ne kadar memnun olduğu anlaşılammakta, birinci sıradaki markanın dördüncü sıradakinden 4 katı kadar beğenildiği anlamı çıkartılamamaktadır [83]. Ordinal ölçekli değişkenlere sahip birimlerin uzaklık ölçüsü aşağıdaki adımları takip ederek elde edilir.

1.  $x_f$  değişkenine ait i. birimin sıralaması M birim içerisinde  $r_{if}$  olur.

2. Her bir ordinal deęişken farklı sayıda durumlara sahip olacaęından, her bir deęişken ařaęıdaki formül ile  $[0,1]$  aralıęına indirgenir.

$$z_{if} = \frac{r_{if}-1}{M_f-1} \quad (3.12)$$

3.  $[0,1]$  aralıęına indirgenen deęişkenlere ait birimler arasındaki uzaklıklar oransal ve aralık ölçekte kullanılan uzaklık ölçüleri ile bulunur.

### 3.2.3. Kümeleme Yöntemlerinin Sınıflandırılması

Kümeleme yöntemleri; uzaklık matrisi ya da benzerlik matrisinden yararlanarak birimler ya da deęişkenleri kendi içinde homojen ve kendi aralarında heterojen gruplamalar oluşturmaya saęlayan yöntemlerdir [84]. En çok bilinen ya da kabul gören kümeleme yöntemleri; hiyerarşik ve hiyerarşik olmayan yöntemler biçiminde iki grupta toplanmaktadır.

#### 3.2.3.1. Hiyerarşik Yöntemler

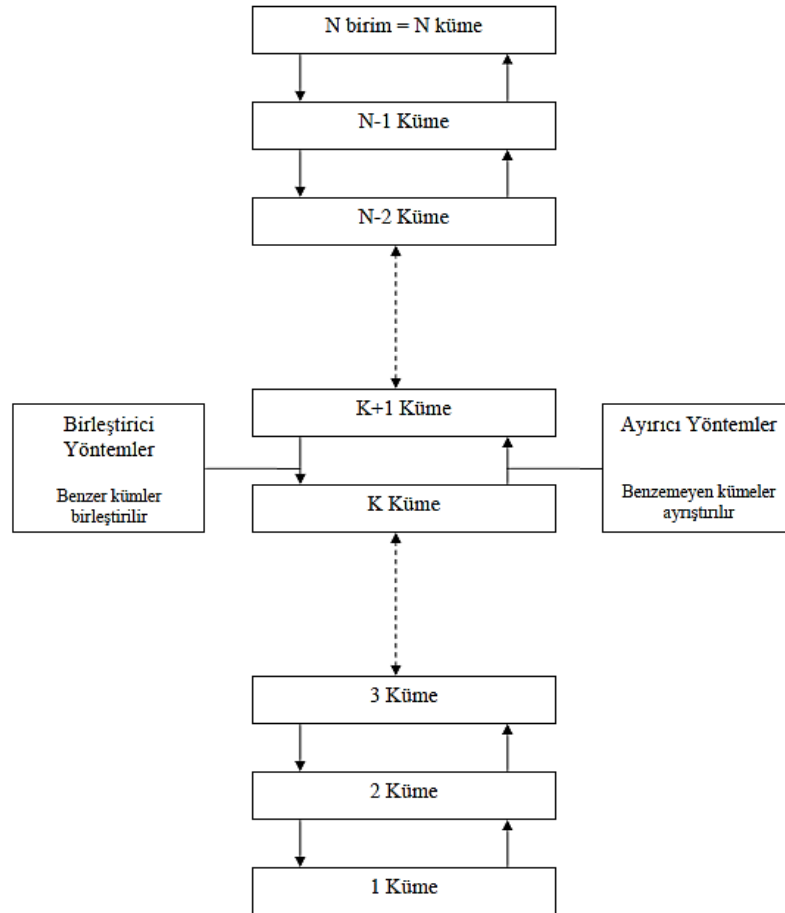
Hiyerarşik kümeleme yönteminde öncelikle  $n$  tane küme olduęu kabul edilir. İlk adım olarak en yakın iki küme birleştirilir. İkinci adımda küme sayısı bir indirgenerek yenilenmiş uzaklıklar matrisi bulunur. Bu iki adım  $n-1$  defa tekrarlanır [85].

Hiyerarşik kümeleme yöntemleri, hiyerarşik ayrışmanın ařaęıdan yukarıya veya yukarıdan ařaęıya doęru olmasına göre birleştirici (agglomerative) ve ayırıcı (divisive) hiyerarşik kümeleme olmak üzere iki ana gruba ayrılır [86].

Birleştirici hiyerarşik kümeleme yöntemleri, başlangıçta tüm birimleri tek başlarına ayrı birer küme oluşturduęunu kabul ederek,  $n$  birimi aşamalı olarak sırasıyla  $n, n-1, n-2, \dots, n-r, \dots, 3, 2, 1$  kümeye ayırmayı amaçlar. Ayırıcı hiyerarşik kümeleme yöntemleri ise başlangıçta tüm birimleri birer küme olarak kabul ederek birimleri aşamalı olarak  $n$  birimi sırasıyla  $1, 2, 3, \dots, n-r, \dots, n-3, n-2, n-1, n$  kümeye ayırmayı

amaçlar. Ayırıcı hiyerarşik kümeleme yöntemi birleştirici hiyerarşik kümelemenin tersi bir yaklaşım kullanır [84].

Şekil 3.1 birleştirici ve ayırıcı hiyerarşik kümeleme yöntemlerinin sergilediği kümeleme yaklaşımlarını güzel bir şekilde anlatmaktadır. Görüldüğü gibi hiyerarşik yöntemler iteratif yöntemlerdir. “Bu yöntemlerin en büyük olumsuzluğu, bir adım gerçekleştirildikten sonra bir daha tekrar aynı adıma geri dönülemezdir. Bu yüzden de yanlış kararları düzeltme imkânı vermemektedir [87].” Ayrıca hiyerarşik kümeleme analizleri büyük veri setlerinde hesapsal karışıklıktan ve bellek gereksinimlerinden dolayı kullanışlı değildir. Bunun yanında bir çok çalışmada doğru sayıda küme seçiminde yol gösterici olabilmektedirler [86]. Uygulamalarda çoğunlukla birleştirici hiyerarşik kümeleme yöntemi kullanılmaktadır. Birleştirici hiyerarşik yöntemler arasında en çok tek bağlantılı, tam bağlantılı, grup ortalama yöntemi ve Ward yöntemi kullanılmaktadır.



Şekil 3.1. Hiyerarşik kümeleme yöntemleri.

*Tek Bağlantı Tekniği:* En yakın komşuluk olarak da bilinen tek bağlantı tekniği, uzaklıklar matrisini kullanarak birbirine en yakın (uzaklık değerleri en küçük) birey ya da nesnelere birleştirmeye dayanmaktadır [81]. Bu teknikte önce birbirine en yakın iki birim (gözlem) bir kümeyle yerleştirilir. Daha sonra diğer en yakın uzaklık tespit edilerek ilk oluşturulan kümeyle bu gözlem eklenir veya iki gözlemden oluşan yeni bir küme oluşturulur. İşlem tüm gözlemlerin bir kümeyle yerleştirilmesine kadar devam eder.

Bu teknikte eğer  $i$  ve  $j$  nci birimler birleştirilmiş ise birleştirilen kümenin  $k$  ıncı küme ile ilişkisi uzaklık ölçütü olarak,

$$d_{k(i,j)} = \text{Min}(d_{ki}, d_{kj}) \quad (3.13)$$

biçiminde ifade edilmektedir.

Eşitlikte;

$d_{k(i,j)}$  :  $k$ . kümenin daha önce oluşan  $i$ . ve  $j$ . kümelerle olan uzaklığını.

$d_{kj}$  :  $k$ . kümenin  $j$ . kümeyle olan uzaklığını.

$d_{ki}$  :  $k$ . kümenin  $i$ . küme ile olan uzaklığını göstermektedir.

*Tam Bağlantı Tekniği:* Bu yöntem, en uzak komşuluk olarak da bilinmektedir. Tek bağlantı tekniğine çok benzemekle birlikte bu teknikteki tek farklılık her kümedeki eleman çiftleri arasındaki uzaklığın maksimum olanının ele alınmasıdır. Bu tekniğe tam bağlantı tekniği denmesinin nedeni, bir küme içindeki tüm birimlerin birbirlerine maksimum uzaklık veya minimum yakınlığa bağlı olmasıdır [88]. Tam bağlantı tekniğindeki uzaklıklar,

$$d_{k(i,j)} = \text{Min}(d_{ki}, d_{kj}) \quad (3.14)$$

biçiminde gösterilmektedir.

*Ortalama Bağlantı Tekniği:* Bu teknikte de işleme tek bağlantı ve tam bağlantı tekniklerinde olduğu gibi başlanır. Ancak kümeleme kriteri olarak bir küme içindeki



birim ile diğer küme içindeki birimler arasındaki ortalama uzaklıklar kullanılır. Ortalama bağlantı tekniğinde kümeler küçük varyanslar ile birbirlerine bağlıdır. Bu teknik tek bağlantı ve tam bağlantı teknikleri arasında sonuçlar vermesi nedeniyle bir alternatif yöntem olarak önerilmektedir [89].

*Ward Yöntemi:* Ward yönteminde, grup bağlantılarından çok grup içi kareler toplamı işlenmektedir [90]. Yönteme her birinin içinde tek bir birim bulunan n tane küme ile başlanır. Yöntemin ilk basamağında her gözlem bir küme olduğundan Hata Kareler Toplamı sıfır olmaktadır [91]. Her aşamada iki alt küme bir sonraki seviyeyi oluşturmak için birleştirilir. Bu durumda k(k-1) alt grup olduğu varsayılır. k kümesinde yer alan  $n_i$  noktanın k kümesinin ortalamalar vektörüne olan Öklid uzaklıkları toplamı, hata kareler toplamıdır ve  $W_k$  olarak ifade edilir;

$$W_k = \sum_{i=1}^p \sum_{j=1}^{n_k} (x_{ijk} - \bar{x}_{ik})^2 = \sum_{i=1}^p \sum_{j=1}^{n_k} x_{ijk}^2 - n_k \sum_{i=1}^p \bar{x}_{ik}^2 \quad (3.15)$$

biçiminde hesaplanır. Burada  $W_k$  değeri  $k= 1,2, \dots, n$  kümelerde hesaplanarak küme içi hata kareler toplamı,

$$W = \sum_{i=1}^n W_k \quad (3.16)$$

şeklinde hesaplanır ve  $W$ 'de en küçük artışa sahip olan p ve q kümeleri birleştirilerek t kümesi elde edilir.  $W$ 'deki bu artış;

$$DW_{pq} = W_t - W_p - W_q \quad (3.17)$$

eşitliği ile hesaplanır. Böylece n birim (n-1) kümeye ayrılmış olur. Böylelikle küme sayısı k=1 oluncaya kadar  $W$  artış değerleri bulunarak birimlerin aşamalı biçimde birbirlerine bağlanmaları sağlanır.

Analiz sonucunda Ward Yöntemi kullanıldığında, birimler değişik seviyelerde başarılı bir şekilde bir araya geldikleri “dendrogram” adlı şemada gösterilir [92]. Ward yöntemi aykırı noktalara duyarlı bir yöntemdir [91].

### 3.2.3.2. Hiyerarşik Olmayan Kümeleme Yöntemleri

Bazı durumlarda küme sayısı önceden bellidir ve arařtırmacı bu küme sayısına göre çözümler üretmek durumundadır. Küme sayısı konusunda ön bilgi varsa veya arařtırmacı anlamlı olacak küme sayısına karar vermiş ise bu durumda, çok uzun zaman alan hiyerarşik yöntemler yerine hiyerarşik olmayan yöntemler kullanılmaktadır [92]. Hiyerarşik olmayan yöntemlerde prosedür řu şekildedir; ilk kısım veya belirli bir başlama noktası ile başlanır. Şayet ilk kısım ile başlanırsa çalışılan ana yığından belirli bir örneklem seçilir ve ilk kısmı elde edebilmek için küme üyeleri düzeltilir.

Hiyerarşik olmayan kümeleme yöntemlerinin temel dezavantajı küme sayılarının daha önceden belirlenmesi ve küme seçimlerinin keyfi olmasıdır [93]. Hiyerarşik olmayan kümeleme yöntemleri başlığı altında birçok teknikten söz edilebilir ancak bunlardan en sık kullanılan iki tanesi  $k$ -ortalama yöntemi ve en çok olabilirlik yöntemidir.

*K-Ortalama Yöntemi:* MacQueen “ $k$  – ortalama” terimini her bir birimin en yakın merkezli (ortalama) kümeye atanması süreci anlamında kullanmıştır.  $k$  – ortalama tekniđi, gözlemleri kümelerin önceden belirlenen sayısına göre gruplandırmakla işleme baslar. Böylece her biri tek gözlemden oluşan  $k$  tane küme ile işleme başlanır ve her bir yeni gözlem en yakın ortalamalı gruba eklenir. Gruba yeni bir gözlem eklendikten sonra küme ortalaması yeniden hesaplanır. Bu süreç tüm gözlemler gruplara atanıncaya kadar devam eder. Tüm gözlemler gruplara atandıktan sonra atandıkları küme ortalamasından daha yakın küme ortalaması varsa, gözlemlerin yerleri deđiştirilmektedir. Amaç diđer kümeleme yöntemlerinde olduđu gibi, gerçekleştirilen kümeleme işlemi sonucunda elde edilen kümelerin, küme içi benzerliklerinin maksimum, kümeler arası benzerliklerinin ise minimum olmasını sağlamaktır. Küme benzerliđi, kümenin ađırlık merkezi kabul edilen bir birim ile kümedeki diđer birimler arasındaki uzaklıkların ortalama deđeri ile ölçülmektedir [94].

Yöntemin işleyişi aşağıda özetlenmiştir:

*Adım 1:* İlk  $k$  gözlem, her biri bir elemanlı küme olarak alınır ve bunların her biri birer küme ortalaması olarak kabul edilir. Tüm birimlerin küme ortalamalarına olan uzaklıkları hesaplanır.

*Adım 2:* Geriye kalan  $(n - k)$  birimin her biri en yakın küme ortalaması olan kümeye atanır. Her atamadan sonra küme ortalaması yeniden hesaplanır.

*Adım 3:* Bütün birimler 2.adımda  $k$  kümeye atandıktan sonra küme ortalamaları yeni çekirdek nokta olarak alınır ve en yakın ortalamaya göre atama işlemi küme elemanları yerlerinin değişmez olmasına kadar tekrarlanır.

*En Çok Olabilirlik Yöntemi:* Diskriminant analizinde de kullanılan en çok olabilirlik yönteminde her bir gözlem en büyük olabilirlik değerini verecek biçimde daha önceden belirlenen kümelere atanır. Kuramsal dayanağı güçlü olmakla birlikte en çok olabilirlik yöntemi yaygın olarak kullanılmamaktadır.

### **3.2.4. Küme Geçerliliği Teknikleri**

Kümeleme analizi uygulamalarında doğru küme sayısı çoğunlukla bilinemez. Kümeleme analizinin sonuçlarının kalitesini değerlendirmek için küme geçerliliği tekniklerine ihtiyaç vardır. Bu teknikler arasında en çok kullanılanları Silhouette indeksi, Calinski ve Harabazs indeksi, Krzanowski ve Lai indeksi olarak sayılabilir. Bu indeksler küme içi değişim ve kümeler arası değişim ölçüleri arasındaki ilişkilere dayanmaktadır. İyi bir kümeleme ile aynı kümede yer alan birimler arasındaki benzerliğin olabildiğince fazla, farklı kümelerde yer alan birimler arasındaki benzerliğin ise olabildiğince az olması amaçlanmaktadır. Küme geçerliliği teknikleri küme içi ve kümeler arası değişimlerin farklı açılardan değerlendirilmesi ile birbirinden ayrılmaktadır [95]. Bu indekslerin yanı sıra çok değişkenli hipotezlerin değerlendirilmesinde göz önüne alınan Wilks Lamda ölçütü de küme sayısının belirlenmesinde kullanılır.

*Silhouette İndeksi*: Rousseeuw (1987), her bir birimin kendi kümesine uygunluğunu tanımlayacak bir Silhouette indeksi önermiştir.  $a(i)$  ;  $i$ . birimin kendi kümesindeki tüm noktalara olan ortalama uzaklıklarını (benzerliğini) ve  $b(i)$  ;  $i$ . birimin diğer kümelerdeki tüm noktalara olan ortalama uzaklıkların minimumunu gösterebilir. Buradan  $i$ . birim için Silhouette indeksi;

$$sil(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.18)$$

olarak tanımlanır. Eğer  $sil(i)$  değeri 1'e yaklaşırsa  $i$ . birimin atandığı kümeye daha uyduğu,  $sil(i)$  değeri 0'a yaklaşırsa veya negatif olursa  $i$ . birimin atandığı kümeye uygun olmadığı sonucuna varılır. Negatif değerler yalnızca bir birim en uygun kümesine atanmadığında ortaya çıkar. Tüm kümelemenin kalitesi (geçerliliği) için doğal bir ölçü;

$$sil(C) = \frac{1}{n} \sum_{s_i \in S} sil(i) \quad (3.19)$$

tüm birimler için ortalama Silhouette değeri olarak tanımlanabilir. Bu kritere göre, maksimum ortalama Silhouette değerine ulaşılan küme sayısı uygun küme sayısı olarak alınır [96].

*Calinski ve Harabazs İndeksi*: Calinski ve Harabazs (1974),  $k$  kümeye sahip bir kümelemenin kalitesini (geçerliliğini) değerlendirebilmek için

$$CH(k) = \frac{BSS(k)/(k-1)}{WSS(k)/(n-k)} \quad (3.20)$$

indeksini önermişlerdir. Burada,

$$WSS(k) = \frac{1}{2} \sum_{l=1}^k \sum_{i,j \in C_l} d(i,j) \quad (3.21)$$

$$BSS(k) = \frac{1}{2} \sum_{l=1}^k \sum_{i \in C_l, j \notin C_l} d(i,j) \quad (3.22)$$

olmak üzere,  $WSS(k)$  ve  $BSS(k)$  sırasıyla, kümeler içi ve kümeler arası kareler toplamlarıdır.  $WSS(k)$  ve  $BSS(k)$  hesaplanırken uzaklık ölçüleri kullanılmaktadır. Bu kritere göre, maksimum  $CH$  indeks değerine ulaşılan küme sayısı, uygun küme sayısı olarak alınır [97].

*Krzanowski ve Lai İndeksi:* Krzanowski and Lai (1985), küme içi kareler toplamı ( $WSS$ ) değerinin azalışı tabanında bir indeks önermişlerdir [98]. Öncelikle Krzanowski ve Lai

$$DIFF(k) = (k - 1)^{2/p} WSS(k - 1) - k^{\frac{2}{p}} WSS(k) \quad (3.23)$$

istatistiğini tanımlamışlar ve bu istatistiğe bağlı olarak

$$KL(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right| \quad (3.24)$$

indeksini tanımlamışlardır.

$$DIFF(k) = k^{\frac{2}{p}} WSS(k) \quad (3.25)$$

alınarak da  $KL$  indeksi hesaplanmaktadır.  $KL$  indeks değerinin maksimum olması amaçlanmaktadır.  $K$ ,  $KL$  indeksi için uygun küme sayısı olsun.  $KL$  indeksinde uygun küme sayısının yani  $k$ 'nın belirlenmesindeki temel düşünce,  $WSS(k)$  değerinin uygun küme sayısına kadar hızlı bir şekilde azaldığı ve uygun küme sayısından sonra yavaş bir şekilde azaldığıdır. Yani  $WSS(k)$  değerinin en hızlı azalış değerine ulaştığı küme sayısı uygun küme sayısı olarak alınmaktadır.

## BÖLÜM 4

### SINIFLANDIRMADA KULLANILAN ZEKİ SİSTEMLER

Çalışmada lojistik regresyon, yapay sinir ağları ve destek vektör makineleri olmak üzere iki sınıflandırma yöntemi kullanılmıştır. Yapay sinir ağları yönteminde çok katmanlı algılayıcı ve radyal tabanlı fonksiyon olmak üzere iki farklı ağ mimarisi kullanılmıştır. Bu yöntemlerin seçilmesinin nedeni sınıflandırma türü tahmin problemlerindeki üstün başarılarıdır. Ayrıca çeşitli alanlarda yapılan bilimsel araştırmalarda, özellikle sağlık bilimlerinde incelenen olayların karmaşık, olayı açıklamada kullanılacak değişkenlerin fazla sayıda olması nedeniyle, birden fazla değişkeni inceleyip, değişkenler arasındaki karmaşık ilişkilerin yorumlanmasında kolaylıklar sağlayan çok değişkenli istatistiksel analizlerin kullanılmasının daha iyi sonuç verdiği görülmüştür.

#### 4.1. YAPAY SİNİR AĞLARI

Yapay sinir ağları (artificial neural networks), canlı organizmaların karşılıklı bağlantılı sinir hücrelerinin oluşturduğu karmaşık öğrenme algoritmalarından esinlenilerek geliştirilmiştir [99]. Yapay sinir ağları, veri madenciliği çalışmalarında hem tanımlama hem de öngörü modellerini kapsayan birçok amaç için kullanılabilir [100]. Yapay sinir ağları, veri madenciliğinin yapay öğrenme bileşeni açısından en yaygın kullanılan araçları haline gelmektedir. Sınıflandırma probleminin yanı sıra, tahmin ve ilişki analizi gibi amaçlarla da kullanılabilir. Yapay sinir ağlarının en büyük dezavantajı, elde edilen modelin anlamlandırılması ve açıklanabilirliğinde yaşanan güçlük olarak görülmektedir. Genellikle, süreçte kullanılan bağlantı ağırlıklarının ifade ettiği ilişkiyi anlamak mümkün değildir. Bu durum, sonuçların nasıl ortaya çıktığını kavramamızı imkânsız hale getirir. Bu yüzden, sonuçların ortaya çıkışındaki fonksiyonel formdan çok, sonuçların kendisinin önemli olduğu durumlarda uygulanması uygun bulunmaktadır.

Yapay sinir ağıları, bu yapısından dolayı kara kutu (black box) olarak da anılmaktadır. Ancak, birçok sinir hücresi barındırmaları nedeniyle, bozuk veri yapıları üzerinde dahi öğrenme kabiliyetine sahiptirler. Bu açıdan, veri madenciliğinin sınıflandırma fonksiyonu için kullanımı giderek yaygınlaşmıştır.

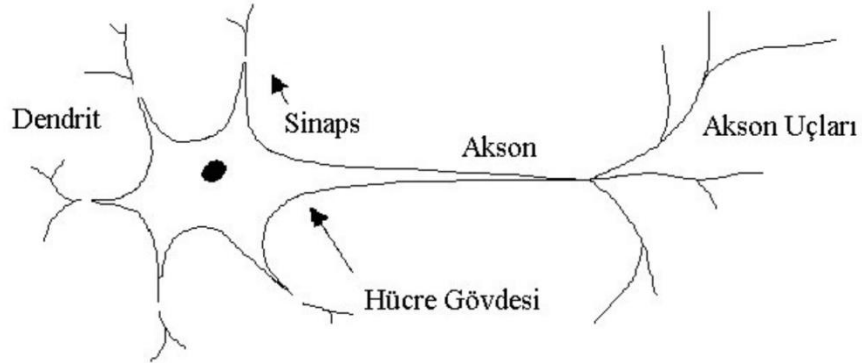
#### 4.1.1. Yapay Sinir Ağlarının Yapısı Ve Temel Elemanları

Yapay sinir ağıları, biyolojik sinir sisteminden esinlenilerek geliştirilmiştir. Bu nedenle işleyişinin tam olarak anlaşılabilmesi için önce biyolojik sinir sistemi hakkında bilgi sahibi olmak gerekir.

##### 4.1.1.1. Biyolojik Sinir Hücresinin Yapısı

Beynimizde 10 milyar adet sinir hücresi ve bunlarda altı trilyondan fazla sayıda bağlantının olduğu söylenmektedir [101].

Sinir sistemi, birbiri ile iletişim halinde olan sinir hücrelerinden oluşur. Şekil 4.1’de bir sinir hücresinin sahip olduğu temel elemanlar gösterilmiştir.



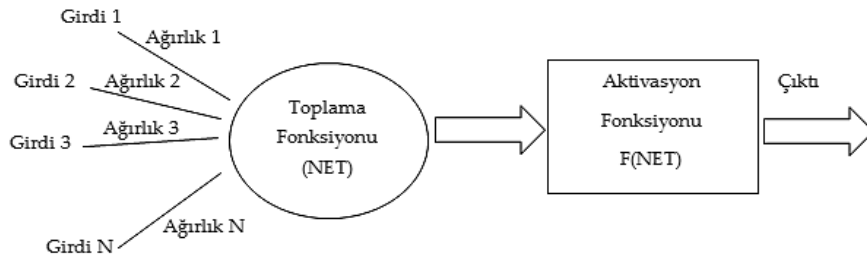
Şekil 4.1. Biyolojik sinir hücresi.

Sinaps’ler, bir sinir hücresinden diğerine elektrik sinyallerinin geçişini sağlayan boşluklardır. İki hücre arasındaki fiziksel olmayan bağlantılar olarak görülebilirler. Sinaps’lerden geçen sinyaller hücre gövdesine giderler. Hücre gövdesi sinyalleri işleme tabi tutar, sinir hücresi kendi elektrik sinyalini akson aracılığı ile dendrit’lere gönderir. Sinapsler dendritler tarafından sinapslere gönderilerek diğer hücrelere

iletilir. İki hücre arasındaki bilgi alış verişi, sinaptik bağlantılardaki nörotransmitter maddeler sayesinde sağlanır. Bu özellikte milyarlarca sinir hücresinin bir araya gelerek sinir sistemini oluşturur. Yapay sinir ağlarının geliştirilmesinde biyolojik hücrelerin bu özelliklerinden yararlanır [101].

#### 4.1.1.2. Yapay Sinir Hücresinin Yapısı

Biyolojik sinir hücrelerinden esinlenilerek oluşturulan ve işlem elemanı olarak da adlandırılan yapay sinir hücrelerinin beş temel elemanı vardır. Bunlar Şekil 4.2’de gösterildiği gibi girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktıdır [101].



Şekil 4.2. Yapay sinir hücresi.

*Girdiler:* Bir yapay sinir hücresine dış dünyadan veya başka hücrelerden gelen bilgilerdir. Bunlar, ağa öğretilmek istenen örnekler tarafından belirlenir [101].

*Ağırlıklar:* Ağlar, öğrenme süresince girdi vektöründeki her bir bilginin önemini temsil eden ağırlıkları belirlemeye çalışırlar. Ağırlıkların değerinin büyük ya da küçük, pozitif ya da negatif olması önemli veya önemsiz olduğu anlamına gelmez ancak sıfır olması etkisinin olmadığını gösterir [101]. Ağın öğrendiği bilgiler bu ağırlık değerlerinde dağıtık olarak saklanır. Ağırlıkların güncellenmesi eğitim bitene kadar yapılır, test aşamasında ağırlıklar üzerinde değişiklik yapılmaz.

*Toplama Fonksiyonu:* Hücreye gelen net girdiyi hesaplayan fonksiyondur. Bu işi yapan çeşitli toplama fonksiyonları bulunmaktadır. Her bir işlem elemanının aynı toplama fonksiyonunu kullanabileceği gibi, ağ içinde bazı işlem elemanlarının farklı



toplama fonksiyonu kullanması da mümkündür [101]. En yaygın kullanılan toplama fonksiyonu kümülatif toplamdır.

*Aktivasyon fonksiyonu:* Hücreye gelen net girdiyi işleyerek üretilecek çıktıyı belirleyen fonksiyondur. Bu fonksiyonların da farklı çeşitleri bulunmaktadır. Bazı ağ modellerinde aktivasyon fonksiyonu olarak türevi alınabilir bir fonksiyonun seçilmesi gereklidir [101]. En yaygın kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur.

*Çıktılar:* Ağın çıktısı aktivasyon fonksiyonu tarafından belirlenir ve dış dünyaya veya girdi olarak tekrar ağa gönderilir [101].

#### **4.1.2. Yapay Sinir Ağı Yapıları**

Yapay sinir ağları yapılarına göre ileri beslemeli ve geri beslemeli olmak üzere iki grupta sınıflandırılırlar.

##### **4.1.2.1. İleri Beslemeli Ağlar**

İleri beslemeli yapay sinir ağları, sinyallerin yalnızca girdi katmanından çıktı katmanına doğru tek yönlü olarak taşınabilmesine olanak verir. Geri besleme döngüleri oluşmaz; örneğin bir katmanın çıktısı aynı katman üzerinde etkili olamaz. Yaygın olarak örüntü tanımlama problemlerinde kullanılırlar. Bu tür ağ yapılarından aynı zamanda “aşağıdan yukarı” veya “yukarıdan aşağı” olarak da söz edilir [102].

##### **4.1.2.2. Geri Beslemeli Ağlar**

Geri beslemeli ağlar, sinyallerin her iki yönde de iletilerek ağda döngüler oluşturmasına imkân verirler. Bu tür ağlar dinamiktir, yani kararlı hale gelene kadar durumları sürekli değişir. Kararlı hale geldiklerinde ise girdiler değişene kadar kararlılık noktasında kalırlar. Bu tür ağ yapılarından aynı zamanda “interaktif” veya “tekrarlı” olarak da söz edilir [102]. Bir katmanın çıktı değerinin aynı katmana veya diğer katmanlara girdi olarak beslenebildiğinden dolayı, ağın herhangi bir andaki

çıkış değeri hem o andaki hem de daha önceki giriş değerlerine göre belirlenir. Geri beslemeli ağlar dinamik yapılarından dolayı yaygın olarak tahminleme uygulamalarında kullanılırlar [103]. MLP, LVQ, RBFNN, PNN, GNN ileri beslemeli ağlara, SOM, BAM, ART, Elman, Jordan, RecurrentBP, TDNN geri beslemeli ağlara örnektir.

### **4.1.3. Yapay Sinir Ağları Algoritmaları**

#### **4.1.3.1. Danışmanlı Öğrenme**

Bu öğrenme yönteminde eğitim sırasında ağa hem girdiler, hem de o girdiler karşılığında oluşturulması gereken çıktılar verilerek ağın girdi ve çıktılar arasındaki ilişkileri öğrenmesi amaçlanır. Öğrenme süresince ağın ürettiği çıktılar beklenen çıktı değerleri ile karşılaştırılarak hata oranı hesaplanır. Ağırlık değerleri hesaplanan hata oranına göre yeniden düzenlenir [104]. Böylece bir sonraki adım veya döngüde beklenene daha yakın bir çıktı değerinin üretilmesi sağlanır. Eğitim aşaması genellikle uzun zaman almaktadır. Yapay sinir ağlarının birçoğu danışmanlı öğrenme yöntemini kullanır. Bu yöntemi kullanan en yaygın ağ modeli MLP (Çok Katmanlı Algılayıcı) ağıdır.

#### **4.1.3.2. Danışmansız Öğrenme**

Bu öğrenme yönteminde ağa sadece girdi değerleri gösterilir ve parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir.

Öğrenme, ağın her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturması şeklinde gerçekleşir. Ağ, ağırlık değerlerini aynı özellikte olan örnekleri ayırabilecek şekilde düzenler [104]. SOM (Özörgütlemeli Harita) Ağları danışmansız öğrenme stratejisini kullanmaktadır [101].

#### 4.1.3.3. Destekleyicili Öğrenme

Bu tür öğrenmede sisteminde girdi değerleri için üretilmesi gereken çıktı değerleri sisteme gösterilmez, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesi beklenir ve ağa ürettiği çıktının doğru veya yanlış olduğuna dair bilgi verilir. Bu bilgi dikkate alınarak ağ yeniden düzenlenir. LVQ (Vektör Nicelemeli Öğrenme) Ağı bu stratejiyi kullanan sistemlere bir örnektir [101].

#### 4.1.4. Yapay Sinir Ağlarında Öğrenme Ve Test

Yapay sinir ağlarında işlem elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesi işlemine ağın eğitilmesi denir. Başlangıçta bu ağırlıklar rastgele atanır. Ağ, kendisine örnekler gösterildikçe doğru çıktıları üretecek ağırlık değerlerini bulmak amacıyla bu ağırlık değerlerini günceller. En doğru ağırlık değerlerinin bulunabilmesi için örnekler ağa defalarca gösterilir. Ağ doğru ağırlık değerlerine ulaştığında örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşmasına *ağın öğrenmesi* denir. Ağırlık değerlerinin değişmesi *öğrenme kurallarına* göre yürütülür [101].

Öğrenme olayı iki aşamada gerçekleşir. Bunlardan ilki ağın gösterilen örnek için üreteceği çıktının belirlenmesidir. İkincisi ise bu çıktının doğruluk derecesine göre ağın bağlantılarındaki ağırlıkların değiştirilmesidir [101].

Ağın eğitimi tamamlandıktan sonra performansının ölçülmesi için öğrenme sırasında görmediği örnekler ile yapılan denemelere *ağın test edilmesi* denir. Ağ eğitimde belirlenen ağırlıkları kullanarak çıktı üretir. Test sırasında ağırlıkların değerleri değiştirilmez. Yapay sinir ağlarının bilinen örneklerden belirli bilgileri çıkararak bilinmeyen örnekler hakkında yorum veya genelleme yapabilme yeteneğine *adaptif öğrenme* denir. Ağ daha önce hiç görmediği örneklere kabul edilir doğrulukta cevaplar veriyorsa ağın performansı iyi kabul edilir. Performansın yetersiz sayıldığı durumlarda ağın tekrar eğitilmesi veya eğitim için yeni örneklerin belirlenmesi yoluna gidilir [101].

#### 4.1.5. Yapay Sinir Ağı Modelleri

Bir yapay sinir ağının modeli, işlem elemanlarının bağlanması sonucu oluşan topoloji, toplama fonksiyonu, aktivasyon fonksiyonu, öğrenme stratejisi ve öğrenme kuralı ile belirlenir [101]. Tez projesi kapsamında Çok Katmanlı Algılayıcı (MLP) ve Radyal Tabanlı Fonksiyon (RBF) ağ modeli kullanılmıştır. Bu bölümde bu ağ modellerinin yapıları hakkında bilgiler verilecektir.

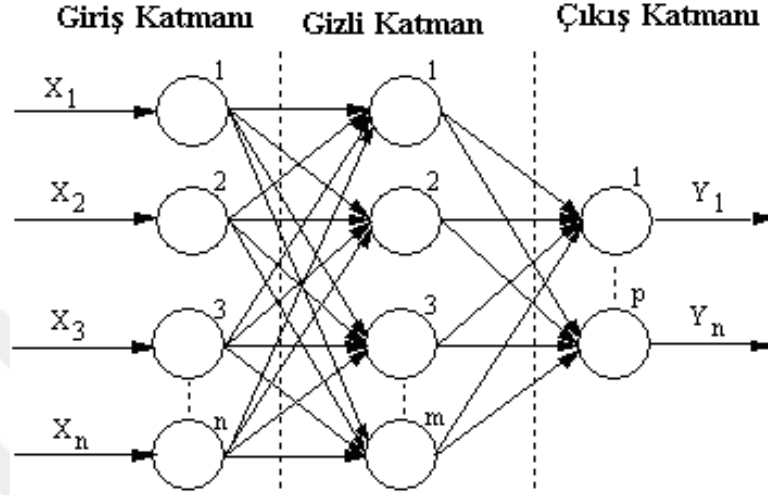
##### 4.1.5.1. Çok Katmanlı Algılayıcı

Çok katmanlı algılayıcı, denetimli öğrenme yöntemini kullanan ve genellikle akademik ya da deneysel sınıflandırma amaçları için kullanılan bir yöntemdir. Çok katmanlı algılayıcının ileri beslemeli doğası, bilginin giriş katmanından çıkış katmanına doğru akmasını ifade etmektedir. Giriş katmanındaki sinir hücreleri gizli katmandaki hücrelere, gizli katmandaki sinir hücreleri ise çıkış katmanındaki hücrelere bağlanarak öngörülerini üretirler (Şekil 4.3). Çok Katmanlı Algılayıcı modeline aynı zamanda “*hata yayma modeli*” ve “*geri yayım modeli*” (ing. back propagation) de denmektedir. Öğrenme yöntemi olarak “*delta öğrenme kuralı*” kullanılır. Ağın beklenen çıktısı ile ürettiği çıktı arasındaki hatayı en aza indirmek amacıyla hatayı ağa yayarak çalışmasını sürdürür [101]. Çok Katmanlı Algılayıcı, ileriye doğru bağlantılı üç katmandan oluşur:

*Girdi Katmanı:* Dış dünyadan gelen bilgileri işlemeden ara katmana gönderir. Her bir işlem elemanının sadece bir girdisi ve bir çıktısı vardır. Bu çıktı bir sonraki katmanda bulunan bütün işlem elemanlarına gönderilir. Girdi katmanındaki her işlem elemanı bir sonraki katmanda bulunan bütün işlem elemanlarına bağlıdır.

*Ara Katmanlar:* Girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderirler. Ağda birden fazla ara katman ve bir ara katmanda birden fazla işlem elemanı bulunabilir. Ara katmandaki her işlem elemanı bir sonraki katmanda bulunan bütün işlem elemanlarına bağlıdır.

**Çıktı Katmanı:** Ara katmandan gelen bilgileri işleyerek ürettiği çıktıları dış dünyaya gönderir. Bir çıktı katmanında birden fazla işlem elemanı bulunabilir. Her işlem elemanı bir önceki katmanda bulunan bütün işlem elemanlarına bağlıdır. Her işlem elemanının sadece bir çıktısı vardır.



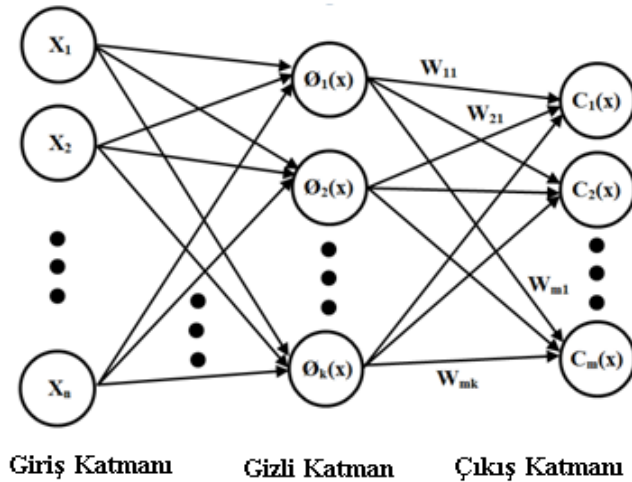
Şekil 4.3. Çok katmanlı yapay sinir ağı mimarisi.

Geri yayılım algoritması, YSA' lar açısından en önemli tarihsel gelişmelerden biridir. Geri yayılım algoritması veya bir başka adıyla genelleştirilmiş Delta Algoritması uygulamada en çok kullanılan öğretim algoritmasıdır. Hata, ağdaki ağırlıkların bir fonksiyonu olarak görülür ve hataların kareleri ortalaması delta kuralında olduğu gibi dereceli azaltma (gradient descent) yöntemi kullanılarak, minimize edilmeye çalışılır. Bu algoritma, hataları çıkıştan girişe geriye doğru azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Geri yayılım algoritması, ÇKA (Çok Katmanlı Algılayıcı)'ları eğitmede en çok kullanılan temel bir öğrenme algoritmasıdır. Eğitim işlemi ve eğitimden sonraki test işlemi bu akışa göre gerçekleştirilir. Geri yayılım algoritması, danışmanlı öğrenme yöntemini kullanır. Örnekler ağa öğretilir ve ağa hedef değeri verilir. Öğrenme, her örnek için ağın çıktı değeri ile hedef değerinin karşılaştırılmasıdır. Hata değeri, ağa tekrar geri besleme şeklinde verilir. Örnek setindeki hata kareleri toplamını azaltmak için nöronlar arasındaki bağlantı ağırlıkları değiştirilir. Tipik çok katlı geri yayılım ağı, daima; bir giriş katmanı, bir çıkış katmanı ve en az bir gizli katmana sahiptir. Gizli katmanların sayısında teorik olarak bir sınırlama yoktur. Fakat genel olarak bir veya iki tane bulunur. Desen sınıflandırma problemlerini çözmek için maksimum dört

katmana (üç gizli ve bir çıkış tabakası) gereksinim duyulduğuna işaret eden bazı çalışmalar yapılmıştır. Geri yayılım algoritması, gradyan azalan ve ÇKA' ları eğitmede en çok kullanılan temel bir algoritmadır [105,106].

#### 4.1.5.2. Radyal Tabanlı Fonksiyon

Radyal Tabanlı Fonksiyon (RTF), ilk olarak çok değişkenli gerçek bir interpolasyon probleminin çözümünde ortaya atılmıştır. Bu konudaki ilk çalışmalar Powell (1985) tarafından yapılmıştır. RTF, YSA tasarımında ilk olarak Broomhead ve Lowe (1988) tarafından kullanılmıştır. Moody ve Darken (1989), Renals (1989) ve Poggio ve Grosi (1990) yaptıkları çalışmalar ile radyal tabanlı fonksiyon ağı (RTFA) teorisi, tasarımı ve uygulamaları hakkında önemli katkılarda bulunmuşlardır. RTFA tasarımı çok boyutlu uzayda eğri uydurma (curve fitting) yaklaşımıdır. Bu nedenle RTFA'nın eğitimi, çok boyutlu uzayda verilere en iyi uyumu sağlayan uygun bir yüzey bulma problemi olarak ele alınır. RTFA Şekil 4.4'de görüldüğü üzere üç katmandan oluşmaktadır. İlk katman giriş katmanı olup kaynak düğümlerden (source nodes) oluşur. İkinci katman, yeteri kadar yüksek boyuta sahip ve aktivasyon olarak RTF kullanan gizli katmandır. Üçüncü katman ise giriş katmanından ağa gösterilen bir girdi vektörüne göre ağın çıktısının üretildiği çıkış katmanıdır. RTFA'da giriş katmanından gizli katmanına doğrusal olmayan, gizli katmandan çıkış katmana doğrusal bir dönüşüm söz konusudur [99].



Şekil 4.4. Radyal tabanlı fonksiyon yapay sinir ağı mimarisi.

Çıkış katmanı doğrusal olduğundan ağırlıklar, eğitim düşme ya da doğrusal en iyileme yöntemleri ile kolayca bulunabilir. Merkezler, girişler arasından rastgele ve sabit olarak seçilebilmekle birlikte RTFA'nın performansını iyileştirmek amacıyla merkez vektörlerinin ve genişliğin uyarlanması için çeşitli yöntemler geliştirilmiştir. Merkez vektörleri, eğitim düşme yöntemine göre danışmanlı öğrenme algoritması ile uyarlanarak, dik en küçük kareler yöntemi ile ya da kendiliğinden düzenlemeli yöntemle giriş örneklerinden öbekleme yapılarak belirlenebilir. RTFA'nın genel matematiksel ifadesi aşağıda verilmiştir.

$$G(x) = \sum_{i=1}^N W_{ij} \varphi_i(x) \quad (4.1)$$

Burada  $W_{ij}$  saklı katmandaki  $i$ . nöronun çıkış katmanındaki  $j$ . nöron arasındaki ağırlığı,  $\varphi_i(x)$  ise aktivasyon fonksiyonunu gösterir.

RTFA'da saklı katman aktivasyon fonksiyonu olarak genellikle Gauss fonksiyonu kullanılır. Gauss Fonksiyonu  $x$  giriş vektörünü,  $c_i$  merkezi,  $\|x - c_i\|$  standart Öklid uzaklığını,  $\sigma_i$  de genişliği göstermek üzere aşağıdaki şekilde ifade edilir [107].

$$\varphi(r) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \quad (4.2)$$

RTFA eğitimi iki aşamada yapılmaktadır. Birinci aşamada gizli katmandaki merkez vektörlerinin ve yayılım parametrelerinin bulunması söz konusudur. İkinci aşamada ise, gizli katman ile çıkış katmanı arasındaki uygun ağırlık değerlerinin ayarlanması yapılmaktadır [108].

## 4.2. DESTEK VEKTÖR MAKİNELERİ

Destek vektör makineleri (DVM), Vapnik tarafından geliştirilen ve istatistiksel öğrenme teorisine dayanan oldukça yeni bir yöntemdir [109]. İlk olarak optik karakter tanıma alanına uygulanan DVM, son otuz yılda Vapnik ve çalışma arkadaşları (Chervonenkis, Boser, Guyon, Cortes, Schölkopf) tarafından bugünkü haline getirilmiştir [110]. DVM yönteminde orijinal giriş uzayı, özellik uzayı olarak

adlandırılan ve içerisinde sınıflandırıcının genelleme kabiliyetini arttırmak için en uygun aşırı düzlemlerin belirlendiği yüksek boyutlu bir iç çarpım uzayı üzerinde haritalandırılır. En uygun ayrıştırıcı düzlemler, matematiğin bir dalı olan optimizasyon teorisi ve bununla ilişkili olan istatistiksel öğrenme teorisini kullanarak bulunur [109]. DVM yöntemi son yıllarda birçok alanda kullanılmaya başlanmıştır. El yazımı rakamların tanınması, resim ve nesne tanıma, ses tanımlama ve zaman serilerinin tahmin edilmesi DVM'nin kullanıldığı bazı alanlardır [111]. Bu uygulama alanlarının yanı sıra özellikle son yıllarda arıza tipi ve yerinin bulunması için de kullanılmışlardır [112,113]. DVM yönteminin YSA'ya göre bazı üstün yönleri aşağıda belirtilmiştir.

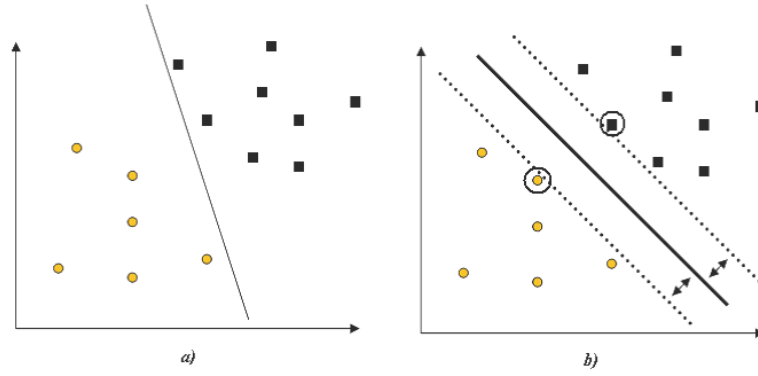
1. DVM çok az giriş ile öğrenilebilir [114]. YSA'da ise giriş boyutunun fazla olması öğrenmeyi olanaksız kılar ya da yavaşlatır. Çoğu DVM uygulamasında veri sayısını azaltmak için özellik çıkarma işlemine gerek duyulmaz [109].
2. DVM' de YSA'da görülen yerel minimum noktaları bulunmaz [115].
3. DVM yapısal hatalı sınıflandırma riski olarak adlandırılan hatayı en aza indirmeye çalıştığı için YSA'ya göre daha iyi bir genelleme özelliğine sahiptir [115].

DVM temelde ikili sınıflandırma için geliştirilen bir yöntem olmasına rağmen, arıza tipi belirlemede olduğu gibi çoğu uygulamada birden fazla sınıf mevcuttur. İki den fazla farklı kümeyi sınıflandırmak için, birden fazla ikili DVM kullanan çok sınıflı DVM yöntemi geliştirilmiştir [110,115,116].

#### **4.2.1. İkili DVM Sınıflandırma Algoritması**

Doğrusalca ayrılabilir veri durumunda her biri  $y_i = \{-1, 1\}$  ile gösterilen sınıflardan birine ait olan,  $R^n$ 'in elemanı olan  $x_i$ 'ler,  $i=1, \dots, N$ , kümesi  $S$  verilmiştir. Amaç, veri kümesini verilen etiketlere göre bir alt düzlemlerle ayırıp, aynı sınıfa ait bütün veri noktalarını alt düzlemin aynı tarafında bırakmaktır.





Şekil 4.5. a) İki sınıflı veriyi ayıran bir alt düzlem, b) En iyi alt düzlem ve marjları.

Bir  $x_i$ 'ler veri kümesi, eğer  $i=1, \dots, N$  için

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad (4.3)$$

koşulunu sağlayan bir  $\mathbf{w}$  varsa doğrusalca ayrılabilir. Burada  $(\mathbf{w}, b)$  aşağıdaki denkleme sahip bir alt düzlem tanımlamaktadır;

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (4.4)$$

ve ayıran alt düzlem olarak adlandırılır ve denklem (4.4)'deki çarpım, veri noktası ile etiketinin alt düzlemin aynı tarafında olmasını belirler. Şekil 4.5(a)'da iki sınıflı ayıran alt düzlemlerden biri gösterilmiştir. Tabii ki, iki sınıflı ayırabilen sonsuz sayıda alt düzlem vardır. Alt düzlemden bir veri noktası  $x_i$ 'ye olan bir  $d_i$  uzaklık ölçüsü tanımlanacak olursa:

$$d_i = \frac{\mathbf{w} \cdot \mathbf{x}_i + b}{\|\mathbf{w}\|} \quad (4.5)$$

Eğer denklemler birleştirilirse aşağıdaki denklem elde edilir:

$$y_i d_i \geq \frac{1}{\|\mathbf{w}\|} \quad (4.6)$$

Burada  $1/\|\mathbf{w}\|$ 'nin  $(\mathbf{w}, b)$  alt düzlemi ile  $x_i$  veri noktası arasındaki uzaklığın bir alt sınırı olduğunu gözlenir. Dolayısıyla, eğer veri noktaları kümesi içinde bu alt sınırı

eşitlikle sağlayan bir  $x_i$  noktası bulunabilirse bu, alt düzleme en yakın noktanın bulunduğu anlamına gelir. Aynı zamanda, en iyi alt düzlemin elde edilmesi için alt düzlemle en yakın veri noktası arasındaki uzaklığı en büyütmek gerekir ve bu marj gerçekten de  $1/\|w\|$  değerine karşılık gelir. Dahası, bu marjı en büyütmek  $w$ 'nin düzgesi  $\|w\|$ 'i en küçültmeye karşılık gelir. Bu marjı en küçültmek ile elde edilen  $w$  ve  $b$  kullanarak sınıflandırma problemi,  $x$  gelen veri olmak üzere,

$$\underline{w} \cdot x + \underline{b} \quad (4.7)$$

yukarıdaki denklemin işaretini bulmaya indirgenir. Böylelikle, karar fonksiyonu;

$$f(x) = \text{sign}(\underline{w} \cdot x + \underline{b}) = \text{sign}(\langle \underline{w}, x \rangle + \underline{b}) \quad (4.8)$$

olur.

#### 4.2.2. Çok Sınıflı DVM Sınıflandırma Algoritması

DVM yöntemi ilk olarak ikili sınıflandırma için geliştirilmiştir. Uygulamalardaki sınıflandırma problemlerinin büyük çoğunluğu ikiden fazla sınıftan oluştuğu için, çok sınıflı bir DVM sınıflandırıcısına ihtiyaç duyulmaktadır. Çok sınıflı sınıflandırma, ikili sınıflandırıcıların birleştirilmesiyle elde edilebilir [109]. Sınıflandırılabilir  $k$  - sınıflı bir problem ikili sınıflandırıcılar kullanılarak çözülebilir. Çok sınıflı sınıflandırma için önerilen pek çok DVM ayrıştırma yöntemi vardır. Bunlardan en önemlileri, bire-karşı-diğer (BKD) ve bire-karşı-bir (BKB) yöntemleridir [114,117].

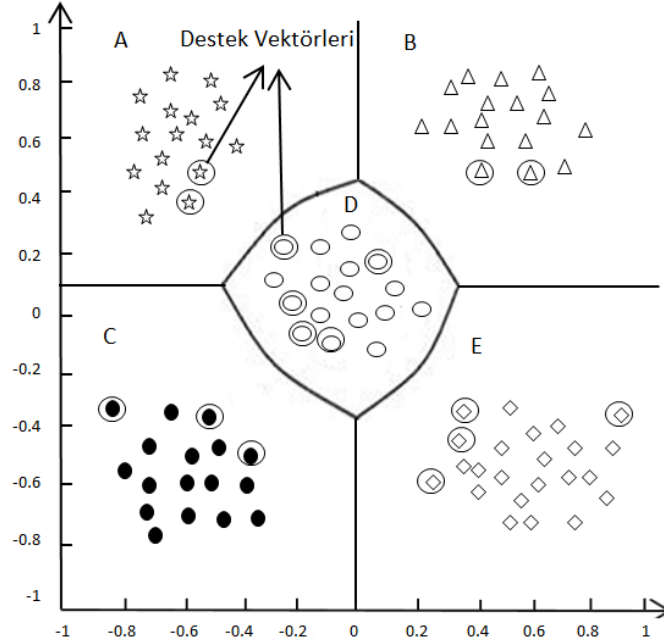
BKD yönteminde, eğitim verisini ikili DVM ile ayırabilen  $k$  adet sınıflandırıcı oluşturulur. Bu sınıflandırıcılardan her biri, bir sınıfı geriye kalan diğer sınıflardan ayırır [114].

Bu işleme, tüm sınıflandırıcılar kullanılana kadar devam edilir ve tüm sınıflandırıcıların ikili sınıflandırma sonuçları birleştirilerek karar işlevi yerine getirilir. BKB yönteminde ise,  $k$ -sınıflı bir sınıflandırma problemi için  $M = k(k-1)/2$

adet sınıflandırıcı oluşturulur. Bu sınıflandırıcılar  $i$ . ve  $j$ . sınıflara ait veriler ile eğitilir. BKB yönteminde ikili sınıflandırma problemi,

$$\begin{aligned} \min_{w^{ij}, b^{ij}, \xi_n^{ij}} & \frac{1}{2} (w^{ij})^T w^{ij} + C \sum_t \xi_n^{ij} (w^{ij})^T \\ (w^{ij})^T K(x, z) + b^{ij} & \geq 1 - \xi_n^{ij}, \quad y_n = i \text{ ise,} \\ (w^{ij})^T K(x, z) + b^{ij} & \leq 1 - \xi_n^{ij}, \quad y_n = j \text{ ise,} \\ \xi_n^{ij} & \geq 0 \end{aligned} \quad (4.9)$$

çözülerek elde edilir [116]. Buradaki  $K(x, z)$  çekirdek fonksiyonu,  $(x_n, y_n)$   $i$ . ya da  $j$ . eğitim örneği,  $b$  eşik değeri ve  $C$  ise düzenleme faktörüdür. Eğitim örneklerinin hangi sınıfa ait olduğunu belirlemek için bir yarışma stratejisi uygulanır. Eğer bir  $x$  eğitim örneğinin  $i$ . sınıfa ait olduğuna karar verilirse,  $i$ . sınıfın puanı 1 artırılır. Aksi durumda  $j$ . sınıfın puan hanesine 1 eklenir. Bu yarışma stratejisi sonucunda, hangi sınıfın puanı daha fazla ise  $x$  eğitim örneğinin kesin olarak o sınıfa ait olduğuna karar verilir [109]. Şekil 4.6'da 5 farklı sınıfın BKB yöntemi ile sınıflandırılması gösterilmiştir [117].



Şekil 4.6. Farklı 5 sınıfın bire karşı bir ayrışımı kullanan çoklu sınıf DVM ile sınıflandırılması.

Şekil 4.6'daki 5 farklı sınıftan oluşan sınıflandırma problemini çözmek için BKB yöntemi  $M = k(k - 1) / 2 = 10$  adet sınıflandırıcı kullanmıştır. BKB yönteminde kullanılan sınıflandırıcı sayısı  $k \geq 3$  olması durumunda BKD yöntemine göre daha fazla olmasına rağmen ( $M = k(k - 1) / 2 \geq k$ ), BKD yöntemi her örnek için geriye kalan tüm veri setini eğittiğinden eğitimi daha yavaştır [109].



## BÖLÜM 5

### CUDA

CUDA, GPU'nun (grafik işlem birimi) gücünden yararlanarak hesaplama performansında önemli artışlara olanak veren NVIDIA paralel hesaplama mimarisidir. Programcılar, C ve C++ programlama dillerine yapacakları bir eklenti sayesinde CUDA'yı edinebilir ve kullanabilirler.

Yazılım geliştiriciler, bilim adamları ve araştırmacılar, bugüne kadar milyonlarca satan CUDA-etkinleştirilmiş GPU'larla CUDA için görüntü ve video işlem, hesaplama dayalı biyoloji ve kimya, akışkan dinamiği simülasyonu, CT görüntü oluşturma dâhil olmak üzere geniş kullanım alanları bulmaktadırlar [118].

CUDA'nın ana özellikleri şunlardır:

*Paylaşılan bellek:* CUDA kullanabilen GPU'lardaki çoklu işlemcilerin her biri 16KB'lık bir paylaşılan belleğe sahiptir. Bu, farklı iş parçacıklarının birbirleri ile iletişim kurmalarına ve veri paylaşmalarına olanak verir. Paylaştırılmış bellek aynı zamanda yazılımsal olarak yönetilebilen tampon bellek olarak değerlendirilebilir. Bu da ana belleğe ciddi miktarda bant genişliği kazandırarak önemli ölçüde hız artışı sağlar. Lineer cebir, hızlı Fourier dönüşümleri ve görüntü işleme filtreleri gibi yaygın uygulamalar bu özellikten faydalanabilirler.

*Rastgele Yazma-Okuma (Toplama-dağıtma):* Grafik API'lerde parça programlar belirli bir bellek adresine 32 float (RGBA x 8 render hedefi) çıktı vermekle sınırlı iken, CUDA dağınık şekilde yazabilme (herhangi bir bellek adresine sınırsız sayıda yazma) özelliğine sahiptir. Bu, grafik API'lerini kullanılarak yapılamayacak yeni logaritmaların kullanılmasına olanak tanır.

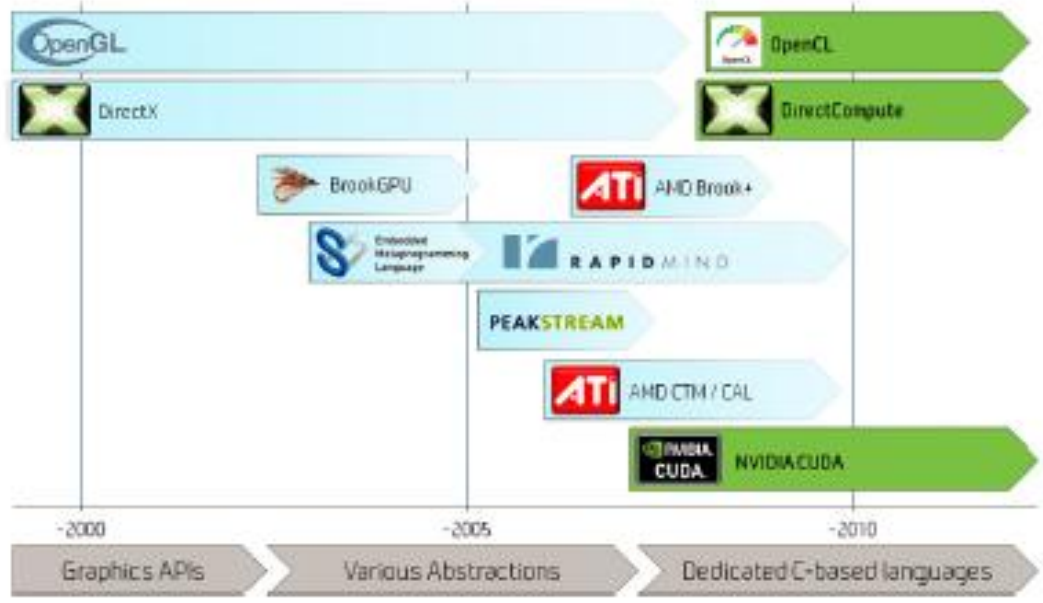
*Diziler ve Tamsayı Adreslemeleri:* Grafik API'leri, kullanıcıyı verileri doku olarak depolamaya yöneltirler. Bu da uzun dizileri 2D dokular halinde gruplandırmayı mecbur kılar, dolayısı ile kullanışsızdır ve fazladan adresleme hesapları gerektirir. CUDA ise verilerin standart diziler halinde depolanmasına olanak verir ve herhangi bir adresten birçok işlemi yapabilir.

*Dokulandırma Desteği:* CUDA, otomatik arabellek (cache), serbest filtreleme ve tamsayı adresleme özellikleri ile birlikte optimize edilmiş doku erişimi sağlar.

*Bütünleştirilmiş Bellek Yığınları ve Depolanması:* CUDA, birden fazla bellek yığın isteğini ve yine birden fazla depolama isteğini birlikte gruplandırır, böylece bellekteki verileri parçalar halinde okuyup yazabilir. Bellek bant genişliğini maksimum düzeyde kullanır.

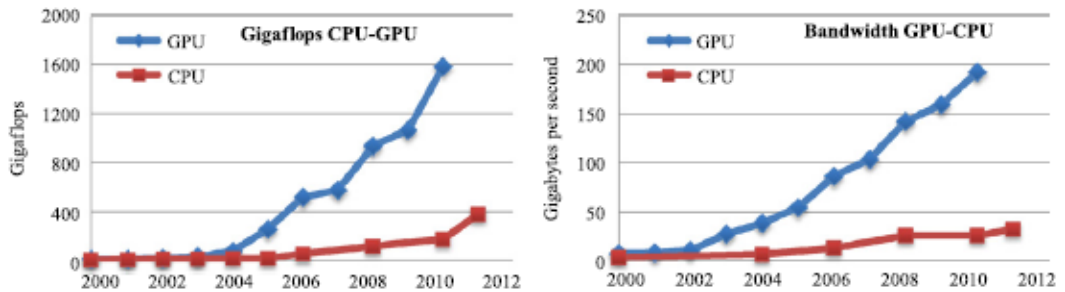
## **5.1. GRAFİK İŞLEM BİRİMİ (GPU)**

Grafik işleme üniteleri, GPU hesaplama veya GPGPU olarak adlandırılan genel amaçlı hesaplamalar için on yıldan fazla bir süredir kullanılmaktadır [119]. İlk GPU programları yazıldığında GPU daha çok bir hesap makinesi gibi kullanılmıştır: ve istenen sonuçları elde etmekte faydalanan bir takım sabit işlemlere sahiptir. GPU üç boyutlu gerçek dünyadan iki boyutlu (2B) resim elde etmek için tasarlandığından gerçekleştirebildiği işlemler temel olarak grafiklerle bağıntılıydı ve ilk GPU programları üçgen gibi grafiksel yapı taşları üzerinde gerçekleştirilen operasyonlar olarak ifade edilmiştir. Bu programları geliştirmek, hatalarını ayıklamak (debug) ve optimize etmek zordu ve sıklıkla derleyici hatalarıyla (bug) karşılaşılıyordu. Ancak, GPU'lar üzerine yapılan araştırmalar, GPU kullanımının merkezi işlem birimi (CPU) üzerinde belli algoritmalar için çarpıcı bir hız artışı sağladığını göstermiş [120,121] ve grafiklerden soyutlanmış üçüncü parti yüksek seviyeli dillerin gelişmesine yol açmıştır. Ancak bu diller, donanım üreticilerinin genel amaçlı hesaplamalarda GPU kullanımını mümkün kılan grafik olmayan dilleri çıkarmasıyla hızlı bir şekilde terk edilmiştir (Şekil 5.1).



Şekil 5.1. GPU hesaplaması için programlama dillerinin tarihçesi.

GPU lar grafik olmayan uygulamalarda ilk kullanılmaya başlandığında programcılar bir uygulamayı OpenGL veya DirectX gibi diller kullanarak grafiksel yapı taşları üzerinde tekrar yazmak durumundaydı. Bu bir külfet ve hata oluşturmaya meyilli olduğundan grafikleri soyutlayan bazı akademik ve üçüncü parti diller ortaya çıkmıştır. 2007'den beri üreticiler GPU programlama için AMD Close-to-Metal(CTM) ve NVIDIA CUDA gibi genel amaçlı diller yayımlamaya başlamışlardır. Günümüzde baskın olan genel amaçlı diller NVIDIA CUDA, DirectCompute ve OpenCL'dir. GPU hesaplama başarısı, CPU ile karşılaştırıldığında kendisinde bulunan muazzam performanstır (Şekil 5.2).



Şekil 5.2. Gigaflop bağlamında tepe performansının tarihsel karşılaştırması ve mevcut en hızlı NVIDIA GPU lar ve Intel CPU lar için bant genişliği.

Günümüzde bu ikisi arasında teorik tepe bant genişliği ve gigaflop performansı karşılaştırıldığında kabaca 7 kat performans farkı bulunmaktadır. Bu performans farkı iki işlemci arasındaki fiziksel çekirdeğe özgü kısıtlamalar ve mimari farklılıklardan ileri gelir. CPU esasında bir seri Von Neumann işlemcisidir ve seri işlemleri sırasıyla yerine getirmek için en uygun hale getirilmişlerdir. CPU nun temel performans faktörlerinden birisi geleneksel olarak giderek artan frekansıdır. Eğer frekans ikiye katlanırsa performansı ikiye katlanmış olur ve frekansın üstel büyümesinde uzun süreli bir eğilim oluşur. Ancak 2000’li yıllarda bu artış *güç duvarına* çarptığından beklenmedik bir durma noktasına gelmiştir [122]: çünkü CPU nun güç tüketimi frekansın küpüyle orantılıdır [121], güç yoğunluğu neredeyse bir nükleer reaktör çekirdeğinkine yaklaşmıştır [123]. Bu tür çipleri yeterince soğutmak mümkün değildir, frekansın üstel olarak artış eğilimi 4.0 Ghz’in hemen altında durmuştur ve CPU lar performanslarını çok çekirdek ve vektör komutları ile artırmaya başlamışlardır. CPU ların seri işleme performansında tavan yaptığı sıralarda GPU lar muazzam paralellikten dolayı performans olarak üstel biçimde büyümüşlerdir: ekrandaki bir pikselin rengini hesaplamak diğer piksellerden bağımsız olarak gerçekleştirilebildiğinden paralellik GPU ların performansını artırmak için doğal bir yol olmuştur. Paralellik performansı artırmanın sürdürülebilir bir yolu olarak görülmektedir ve GPU lar için en mükemmel şekilde uyumlu hale getirilmiş paralel iş yüklerini gösteren birçok uygulama bulunmaktadır.

Günümüzde üç ana GPU üreticisi bulunmaktadır. Bunlardan en büyüğü Intel’dir. Ancak, Intel sadece entegre edilmiş ve düşük performans pazarlara hâkim olabilmıştır. Yüksek performans ve ayrık grafikler için AMD ve NVIDIA önde gelen sağlayıcılarıdır. Akademik ve endüstriyel çevrelerde NVIDIA açık şekilde baskın sağlayıcıdır ve bu sebeple tez çalışmasında NVIDIA nın GPU ları üzerinde çalışılmıştır.



## BÖLÜM 6

### UYGULAMA

Çalışmada kullanılacak olan veri Ankara Atatürk Eğitim ve Araştırma Hastanesi'nden alınmıştır. Hastalara ait cinsiyet, yaş gibi genel bilgilerin yanı sıra hastalık teşhisine yönelik refraksiyon kusurları, görme keskinlikleri, Sim K1, SimK2, ortalama ve santral keratometrileri (Ort K, Santral K), en ince kornea kalınlıklarından oluşan pakimetri değerleri, korneal wavefront aberasyonları (total yüksek sıralı aberasyonlar (HOA), koma ve koma benzeri aberasyonlar, vertikal koma (3,-1), horizontal koma (3,1), sferik aberasyon (4,0)) ve aksiyel haritadan elde edilen CLMI büyüklükleri gibi bilgiler de alınmıştır. Alınan 695 kayıttan 366 tanesi erkek, 329 tanesi bayan hastaya aittir. 159 keratokonus hastasının 301 gözü ile kontrol grubu olarak 265 refraktif cerrahi adayının 394 gözünün verisi alınmıştır. İstatistiksel yöntemlere başlanılmadan önce 159 keratokonus hastasına ait 301 kayıt içerisinde null değerler içeren 17 kayıt ve 265 refraktif cerrahi adayına ait 394 kayıt içerisinde null değerler içeren 79 kayıt silinmiştir ve toplamda 599 kayıt kullanılmıştır.

#### 6.1. VERİ

Bu çalışmada amaç keratokonus hastalığının teşhisi için istatistiksel analizlere dayalı yeni bir sınıflandırma metodu tanımlamak ve hastalığın teşhisi için zeki bir sistem tasarlamak olduğu için uygulamada öncelikle faktör analizi kullanılarak birbiriyle ilişkili çok sayıda değişkenden, daha anlamlı, kolay anlaşılabilir ve birbirinden bağımsız faktörler elde edilmiştir. Daha sonra bu faktör üzerinde kümeleme analizi teknikleri kullanılarak yeni bir sınıflandırma metodu tanımlanmıştır.

Uygulamada kullanılan değişkenler ve bu değişkenlere ait açıklamalar Çizelge 6.1'de görüldüğü gibidir.

Çizelge 6.1. Uygulamada kullanılan bağımsız değişkenlerin listesi.

ALAN ADI	TİPİ	AÇIKLAMASI
Yas	Sayı	Hastaların yaşı
SimKFlat	Sayı	Keratometri Değerleri
SimKSteep	Sayı	
CentralK	Sayı	
CLMIAxis	Sayı	CLMI (The Cone Location Magnitude Index) Ölçümleri
CLMIuzak	Sayı	
CLMIindex	Sayı	
Ho RMS	Sayı	Aberrasyonlar
(3,1)	Sayı	
(3,-1)	Sayı	
(4,0)	Sayı	
Coma	Sayı	
Pach	Sayı	Pakimetri kornea kalınlık ölçümü
Spher	Sayı	Refraksiyon Düzeyi (Kırma kusuru)
Cylin	Sayı	
Logbc	Sayı	Görme Düzeyi

## 6.2. UYGULAMA

### 6.2.1. Faktör Analizi

Faktör analizi SPSS Statistics 20 kullanılarak, 16 değişken üzerinde yapılmıştır. Faktör analizinde faktör türetme modeli olarak temel bileşenler yöntemi kullanılmıştır. Faktörlerin sayısını belirlemek için hem öz değer istatistiği 1'den büyük olan faktörlere bakılmış hem de yığın grafiği incelenmiştir.

Veri setinin faktör analizine uygun olup olmadığının değerlendirilmesi için Kaiser-Meyer-Olkin (KMO) ölçümü ve Barlett testi kullanılmıştır. Bu testlere ilişkin sonuçlar Çizelge 6.2'deki gibidir.

Çizelge 6.2. KMO ve Bartlett testi sonuçları.

KMO measure of sampling adequacy		0.806
Bartlett's test of sphericity	Approximated Chi-Square	1963.727
	Df	120
	Significance level	0.000

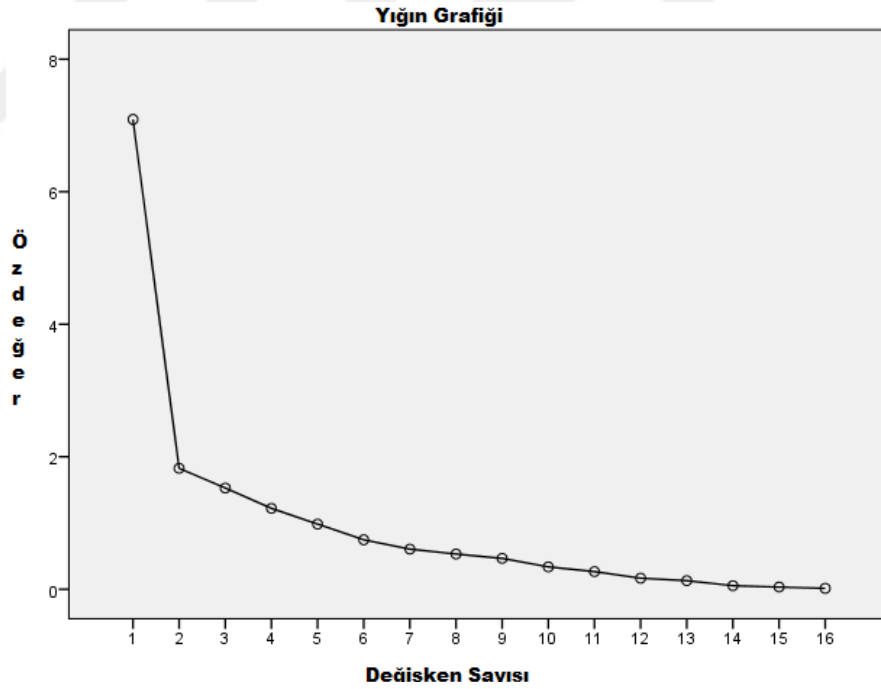
Çizelge 6.2'de görüleceği üzere KMO (0.806); bu örnekteki 695 kayıtlık örneklem büyüklüğünün yeterli olduğunu ve Bartlett's testi sonuçları da ( $p < 0.01$ ); verilerin normal dağılıma sahip olduğunu göstermektedir. Bu çalışmada; türetilcek faktör sayısı, Varyansa Katılma (Öz değer) kriteri ve Scree Test Kriteri (Yığın Grafiği) kullanılarak belirlenmiştir. Bu kriterlere göre türetilen faktör sayısı 4'tür. Varyansa katılma kriterine göre, öz değer istatistiği 1'den büyük olan bileşenler toplam faktör sayısını belirtmektedir, bu çalışmada öz değeri 1'den büyük olan 4 faktör vardır ve bu 4 faktör toplam varyansın %72,908'ini açıklamaktadır (Çizelge 6.3).

Çizelge 6.3. Varyansa katılma (Öz Değer) kriteri sonuçları.

	Başlangıç Özdeğerleri			Döndürülmüş Toplamları		
	Toplam	Varyans (%)	Kümülatif (%)	Toplam	Varyans (%)	Kümülatif (%)
1	7.091	44.320	44.320	5.209	32.555	32.555
2	1.825	11.407	55.728	3.524	22.028	54.582
3	1.528	9.549	65.277	1.610	10.062	64.644
4	1.221	7.631	72.908	1.322	8.264	72.908
5	0.984	6.149	79.057			
6	0.747	4.666	83.723			
7	0.606	3.785	87.508			
8	0.531	3.322	90.829			
9	0.466	2.912	93.741			
10	0.338	2.110	95.851			
11	0.267	1.666	97.517			
12	0.167	1.043	98.560			
13	0.129	0.807	99.367			
14	0.054	0.338	99.705			
15	0.033	0.208	99.913			
16	0.014	0.087	100.000			

Çalışmada birinci faktörün açıkladığı varyans % 44.320'dir. İkinci, üçüncü ve dördüncü faktörlerin açıkladıkları varyans oranları (% 11.407, % 9.549, %7.631), (Birinciye göre) hem azalmakta, hem de bu üç faktörün kendi aralarındaki yüzdeler oranları küçülmektedir. Benzer durum, özdeğerlerde de görülmektedir (7,091, 1,825, 1,528, 1,221). Bu tablo bilgileri, ölçeğin tek faktörlü (boyutlu) olma ihtimalini güçlendirmektedir fakat buna karar vermek için yığın grafiğini de incelemek gerekmektedir.

Yığın grafiği faktörlerin türetme sırasına göre açıkladıkları varyans miktarlarını göstermektedir. Elde edilen noktaların birleştirilmesiyle negatif eğimli bir eğri elde edilir ve yığın grafiği testi bu eğrinin yatay hale gelmeye başladığı noktada faktör türetme işlemine son verir. Yatay hale gelmesinin, varyansa katılma değerinin de 1'den az olmaya başladığı nokta olduğuna dikkat çekilmelidir. Grafikteki bilgiler de ölçeğin tek boyutlu olarak kabul edilebileceğini göstermektedir (Şekil 6.1).



Şekil 6.1. Scree Test (Yığın grafiği).

Bir değişken hangi faktör altında mutlak değer olarak büyük ağırlığa sahipse o değişken o faktör ile yakın ilişki içindedir. Çizelge 6.4 hangi değişkenin hangi faktöre ait olduğunu göstermektedir.

Çizelge 6.4. Değişkenlerin ait olduğu faktör sonuçları.

	Bileşenler			
	1	2	3	4
SimKsteep	<b><u>0.934</u></b>	-0.177	-0.039	-0.037
Coma	<b><u>0.900</u></b>	0.331	-0.015	0.000
Ho RMS	<b><u>0.900</u></b>	0.317	0.059	-0.028
Central K	<b><u>0.864</u></b>	-0.317	-0.057	-0.089
(3.-1)	-0.860	-0.332	<b><u>0.188</u></b>	-0.080
SimKflat	<b><u>0.836</u></b>	-0.277	0.065	0.174
LOGBC	<b><u>0.711</u></b>	-0.168	0.314	0.017
CLMIindex	<b><u>0.672</u></b>	0.489	0.044	-0.070
cylin	-0.566	-0.123	0.206	<b><u>0.327</u></b>
Pach	-0.509	<b><u>-0.038</u></b>	-0.097	-0.275
(4.0)	-0.578	<b><u>0.701</u></b>	0.134	0.137
CLMI uzak	-0.424	<b><u>0.589</u></b>	0.526	0.037
CLMIaxis	0.051	0.084	-0.766	<b><u>0.472</u></b>
spher	-0.494	<b><u>0.340</u></b>	-0.571	-0.050
(3.1)	0.211	0.040	0.052	<b><u>0.727</u></b>
yas	-0.226	-0.233	0.347	<b><u>0.461</u></b>

Faktör analizi ile belirlenen faktörlerin veya önceden belirlenen faktörlerin iç güvenilirliğini ölçmek için güvenilirlik analizi kullanılmıştır. Uygulamada güvenilirlik analizi 1. Faktöre ait değerler üzerinde uygulanmıştır ve sonuçlar Çizelge 6.5'deki gibidir.

Çizelge 6.5. Güvenilirlik sonucu.

Cronbach's Alpha	N of Items
0.884	7

Güvenilirlik katsayısı  $\alpha = 0.884$ , %88.4 ile faktörün oldukça güvenilir olduğunu göstermiştir.

Faktör analizi sonucunda elde edilen değişkenler arasında çoklu doğrusal bağlantı problemi olup olmadığını anlamak için öncelikle basit korelasyon katsayılarına bakılmıştır (Çizelge 6.6).

Çizelge 6.6. Değişkenlerin korelasyon değerleri.

		SimKFlat	SimKSteep	CentralK	CLMlind.	HoRMS	LOGBC	Coma
<b>SimKFlat</b>	Pearson Cor.	1	0.879	0.793**	0.439**	0.658**	0.628**	0.636**
	Sig.2-tailed		0.000	0.000	0.000	0.000	0.000	0.000
	N	284	284	284	284	284	284	284
<b>SimKSteep</b>	Pearson Cor.	0.879	1	0.895**	0.518**	0.779**	0.671**	0.736**
	Sig.2-tailed	0.000		0.000	0.000	0.000	0.000	0.000
	N	284	284	284	284	284	284	284
<b>Central K</b>	Pearson Cor.	0.793**	0.895**	1	0.468**	0.736**	0.591**	0.682**
	Sig.2-tailed	0.000	0.000		0.000	0.000	0.000	0.000
	N	284	284	284	284	284	284	284
<b>CLMlind.</b>	Pearson Cor.	0.439**	0.518**	0.468**	1	0.764**	0.351**	0.794**
	Sig.2-tailed	0.000	0.000	0.000		0.000	0.000	0.000
	N	284	284	284	284	284	284	284
<b>HoRMS</b>	Pearson Cor.	0.658**	0.779**	0.736**	0.764**	1	0.579**	0.943**
	Sig.2-tailed	0.000	0.000	0.000	0.000		0.000	0.000
	N	284	284	284	284	284	284	284
<b>LOGBC</b>	Pearson Cor.	0.628**	0.671**	0.591**	0.351**	0.579**	1	0.519**
	Sig.2-tailed	0.000	0.000	0.000	0.000	0.000		0.000
	N	284	284	284	284	284	284	284
<b>Coma</b>	Pearson Cor.	0.636**	0.736**	0.682**	0.794**	0.943**	0.519**	1
	Sig.2-tailed	0.000	0.000	0.000	0.000	0.000	0.000	
	N	284	284	284	284	284	284	284

\*\* Correlation is significant at the 0.01 level(2-tailed)

Yüzeysel olarak, iki bağımsız değişken arasındaki basit korelasyon katsayısı oldukça anlamlı ( $r > \%75$ ) ise, bu durum çoklu doğrusal bağlantı problemine yol açabileceğinden SimKSteep değişkeninin hem SimKFlat hem de CentralK değişkeniyle arasındaki basit korelasyon katsayısının oldukça anlamlı olduğu görülmüştür. Fakat istatistik açıdan anlamlı korelasyonlar her zaman çoklu doğrusal bağlantı problemine yol açmayacağı için bir de varyans arttırıcı faktör (VIF=Variance Inflation Factor) ve Tolerance Value kullanılmıştır (Çizelge 6.7).

Çizelge 6.7. VIF (Variance Inflation Factor) ve TV (Tolerance Value).

Model	Standartlaştırılmamış Katsayılar		Standart Katsayılar	t	Sig.	Doğrusallık istatistikleri		
	B	Std. Hata	Beta			T. V.	VIF	
1	Sabit	2.073	0.505		4.103	0.000		
	SimKflat	0.003	0.021	0.018	0.156	0.876	0.227	4.408
	SimKsteep	0.087	0.023	0.591	3.760	0.000	0.121	<b>8.259</b>
	Central K	-0.089	0.013	-0.841	-6.846	0.000	0.198	5.045

Çizelge 6.7 incelendiğinde yüksek VIF ve düşük tolerans değerinde sahip olan SimKSteep değişkeni uygulamadan çıkarılarak kümeleme analizi yapılmıştır.

### 6.2.2. Kümeleme Analizi

Kümeleme analizi uygulamasında hiyerarşik kümeleme yöntemlerinden tek bağlantı tam bağlantı, ortalama bağlantı ve Ward's teknikleri ile hiyerarşik olmayan kümeleme yöntemlerinden k-ortalama yöntemi kullanılmıştır. Uzaklık ölçüsü olarak Kareli Öklid uzaklığı kullanılmıştır.

Keratokonus hastalığının teşhisi için yeni bir sınıflandırma yaklaşımı bulmak amacıyla standardize edilen değerler Aglomeratif Hiyerarşik yöntem kullanılarak gruplandırılmıştır. Aglomeratif çizelgede ise katsayı (coefficients) kullanılarak birbirlerine en çok benzeyen kayıtlar eşleşmiştir. Bu kayıtlar Çizelge 6.8'de gösterilmektedir. Bu metodun sonuçlarına göre birbirlerine en çok benzeyen (0.05) kayıtlar, 1. sırada Kayıt29 ile Kayıt190 olurken 2. sırada Kayıt181 ile Kayıt256, 3. sırada ise Kayıt20 ile Kayıt280 olmuştur. Birbirine en az benzeyen kayıtlar ise Kayıt1 ile Kayıt3 olmuştur. Katsayılardan da görüleceği üzere birbirlerine en çok benzeyen kayıtlar ilk basamaklarda bir araya gelirlerken, birbirlerine daha az benzeyen kayıtlar ise daha sonraki basamaklarda bir araya gelmektedirler.

Çizelge 6.8. Aglomeratif çizelge.

Evre	Birleştirilmiş Kümeler		Katsayılar	Kümelemenin İlk Evresi		Sonraki Evre
	Küme1	Küme2		Küme1	Küme2	
1	29	190	0.005	0	0	61
2	181	256	0.017	0	0	54
3	20	280	0.030	0	0	26
4	143	196	0.048	0	0	72
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
280	1	9	676.615	277	273	282
281	3	5	887.435	279	278	283
282	1	26	1125.211	280	276	283
283	1	3	1981.000	282	281	0

Kümeleme analizi sonucunda elde edilen sonuçlar Çizelge 6.9'da yer almaktadır.

Çizelge 6.9. Kümeleme analizi sonuçları.

Kümeleme Yön.	Küme Sayısı	1	2	3	4	5	6	7	8	9	10
<b>WARD'S</b>	2 Küme	134	150								
	3 Küme	120	150	14							
	4 Küme	55	150	65	14						
	5 Küme	55	150	65	9	5					
	6 Küme	55	89	65	61	9	5				
	7 Küme	55	89	64	61	9	5	1			
	8 Küme	52	89	64	61	9	5	3	1		
	9 Küme	31	89	64	61	21	9	5	3	1	
	10 Küme	31	62	64	61	27	21	9	5	3	1
	<b>TEK BAĞLANTI</b>	2 Küme	283	1							
3 Küme		282	1	1							
4 Küme		281	1	1	1						
5 Küme		278	1	1	3	1					
6 Küme		277	1	1	3	1	1				
7 Küme		276	1	1	1	1	3	1			
8 Küme		276	1	1	1	1	2	1	1		
9 Küme		275	1	1	1	1	2	1	1	1	
10 Küme		275	1	1	1	1	1	1	1	1	1
<b>TAM BAĞLANTI</b>		2 Küme	270	14							
	3 Küme	269	14	1							
	4 Küme	269	9	5	1						
	5 Küme	61	208	9	5	1					
	6 Küme	61	208	9	2	1	3				
	7 Küme	58	208	9	2	3	1	3			
	8 Küme	58	152	56	9	2	3	1	3		
	9 Küme	57	152	56	9	2	3	1	1	3	
	10 Küme	57	152	56	8	2	3	1	1	1	3
	<b>ORTALAMA BAĞLANTI</b>	2 Küme	283	1							
3 Küme		269	14	1							
4 Küme		269	13	1	1						
5 Küme		269	6	7	1	1					
6 Küme		269	6	4	3	1	1				
7 Küme		268	6	4	3	1	1	1			
8 Küme		268	6	1	3	1	1	3	1		
9 Küme		268	5	1	3	1	1	1	3	1	
10 Küme		268	5	1	3	1	1	1	2	1	1
<b>K-MEANS</b>		2 Küme	203	81							
	3 Küme	206	77	1							
	4 Küme	149	37	97	1						
	5 Küme	109	60	1	102	12					
	6 Küme	50	1	8	120	62	43				
	7 Küme	2	78	14	117	4	68	1			
	8 Küme	67	1	41	39	43	83	2	8		
	9 Küme	77	16	6	48	40	93	1	2	1	
	10 Küme	63	65	49	30	20	50	1	4	1	1



Uygulamada Wards yöntemi kullanılarak yapılan gruplamanın daha iyi olduğu görülmüş ve son olarak kümeleme analizi sonuçlarının denetlenmesi amacıyla diskriminant analizi kullanılmıştır. Elde edilen farklı küme sayılarına göre uygulanan diskriminant analizi ile hesaplanan Wilk's Lamda değerine göre, en yüksek önemliliğe sahip küme sayısı kararlaştırılmıştır.

Diskriminant analizi ile öncelikle en düşük grup ya da küme sayısı, yani küme sayısının iki olması durumu için, küme sayısının bir eksiği yani birinci diskriminant fonksiyonunun anlamlılığı denetlenmiştir. Ardından aynı işlem üç küme ya da iki diskriminant fonksiyonu, dört küme ya da üç diskriminant fonksiyonu, beş küme ya da dört diskriminant fonksiyonu ve altı küme ya da beş diskriminant fonksiyonu için tekrarlanmıştır. Yapılan diskriminant analizi sonucunda, beşinci diskriminant fonksiyonunun ( $\alpha = 0.087$ ) düzeyinde anlamlı olduğu görülmüştür (Çizelge 6.10).

Çizelge 6.10. Altı kümeye ilişkin Wilks' Lamda değeri.

<b>Wilks' Lambda</b>				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 5	0.031	962.199	30	0.000
2 through 5	0.345	295.052	20	0.000
3 through 5	0.669	111.184	12	0.000
4 through 5	0.904	28.105	6	0.000
5	0.983	4.873	2	0.087

Sonuç olarak, yapılan diskriminant analiziyle, dört diskriminant fonksiyonunun gruplandırma esas alınması yeterli görülmüş ve beş grup altında toplanmasına karar verilmiştir. Böylece, araştırma kapsamındaki 284 kayıttan 55 tanesi birinci, 150 tanesi ikinci, 65 tanesi üçüncü, 9 tanesi dördüncü ve 5 tanesi ise beşinci grupta yer almıştır. Uzman görüşü alınarak istatistiksel yöntemler ile oluşturulan bu beş grubun hastalığın ilerleyişine göre belirlenmiş seviyeleri Çizelge (6.11) de gösterilmiştir.

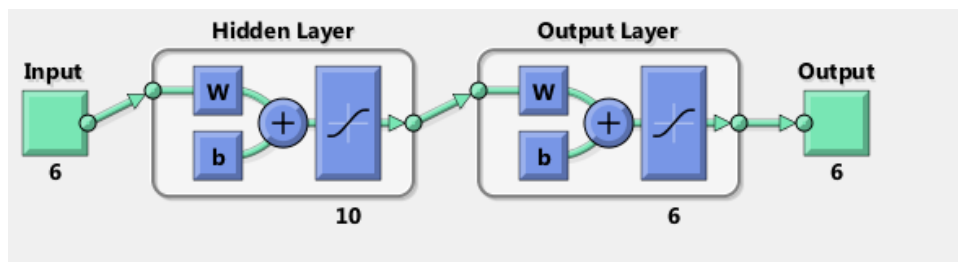
Çizelge 6.11. İstatistiksel sonuçlara göre keratokonus sınıflandırması.

Keratoconus şiddeti ve kayıt sayısı	CLMI Index		Ho RMS		Central K		SimKflat		Log BC		Coma	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Birinci seviye (150)	4.317	2.722	1.339	0.641	46.973	4.373	44.809	2.160	0.150	0.180	1.074	0.600
İkinci seviye (65)	12.071	5.945	3.267	0.653	49.893	4.406	45.250	2.144	0.327	0.248	2.954	0.618
Üçüncü seviye (55)	11.693	3.998	4.018	1.074	59.947	2.822	50.725	2.432	0.665	0.510	3.435	1.096
Dördüncü seviye (5)	16.762	10.367	7.396	2.121	70.526	7.083	62.658	5.762	3.000	0E-7	5.690	2.547
Beşinci seviye (9)	22.422	4.106	7.840	1.284	64.814	4.699	53.963	5.244	0.813	0.428	6.468	0.980

### 6.2.3. Sınıflandırma Yöntemleri

Kümeleme analizi sonrasında elde edilen gruplara sırasıyla yapay sinir ağları ve destek vektör makineleri yöntemleri uygulanmıştır.

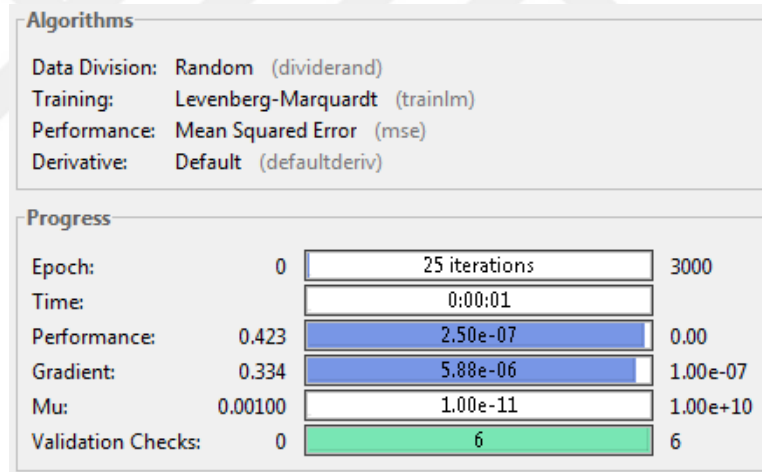
Çalışmada kullanılacak çok katmanlı algılayıcı eğitim fonksiyonunun seçimi için öncelikle MATLAB nntool araç kutusu kullanılmıştır. Yapay Sinir Ağı için en uygun değerlerin bulunması için gizli katman nöron sayısı, öğrenme oranı, momentum sabiti ve eğitim fonksiyonunun farklı denemeler sonucu verdiği en iyi değerler bulunmuştur. Eğitim ve test sonuçlarına göre gizli katman nöron sayısının 10 olmasına karar verilmiştir. En iyi öğrenme oranının 0.3 değeri olduğu eğitim ve test sonuçları neticesinde bulunmuştur ve momentum sabiti eğitim ve test sonuçlarına bakılarak 0.7 olarak seçilmiştir. Matlab ortamında oluşturulan çok katmanlı algılayıcı yapay sinir ağı Şekil 6.2’de gösterilmiştir.



Şekil 6.2. MATLAB ortamında oluşturulan çok katmanlı algılayıcı YSA mimarisi.

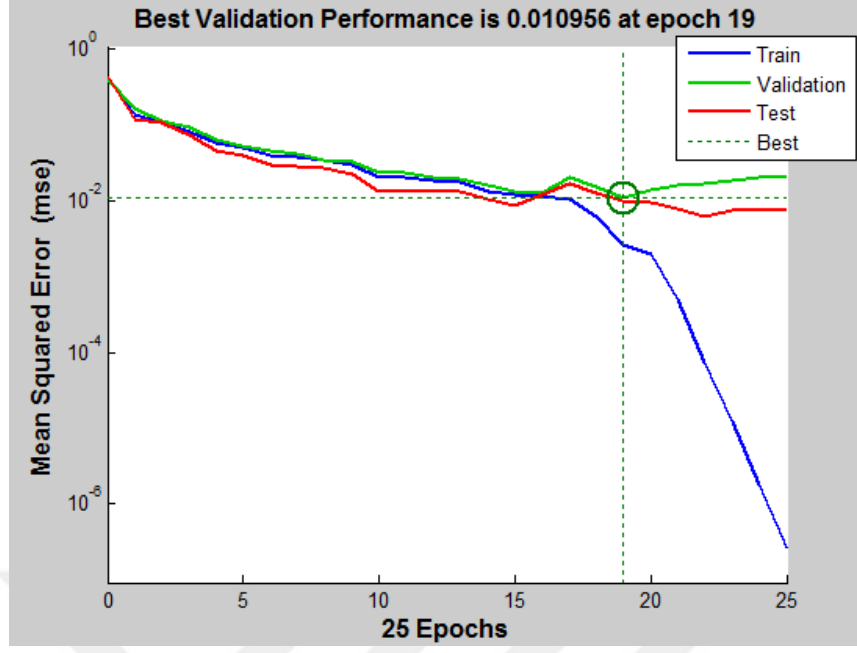
Eđitim fonksiyonlarına gre ok katmanlı yapay sinir ađının performansı ve test sonuçları ařađıda sıralanmıřtır.

*Trainlm fonksiyonu:* Levenberg-Marquardt (LM) algoritması, Newton metodunun hızıyla, adım dřme metodunun sađlamlıđının bileřkesidir. ok sayıda komřuluk fikri zerine dayanan LM algoritması, en kk kareler yaklařımı (least square estimation) metodudur [124, 125]. Levenberg-Marquardt algoritmasının en nemli avantajlarından biri, hızlı yakınsama zelliđidir. Sonu olarak LM đrenme algoritması ok hızlı olarak zme ulařmasına rađmen ok fazla bellek gerektirmektedir. Eđitim fonksiyonu olarak trainlm (Levenberg-Marquardt) fonksiyonu kullanıldıđında elde edilen sonular řekil 6.3'deki gibidir. Sonular incelendiđinde eđitim 25 iterasyonda tamamlanmıřtır ve eđitim iřlemi 1 saniye srmřtr.



řekil 6.3. Trainlm eđitim fonksiyonu sonuları.

Performans grafiđi incelendiđinde en iyi dođrulamanın 19. iterasyonda gerekleřtiđi grlmektedir. Grafikte mavi izgi hatanın karesinin ortalamasına gre eđitimi, kırmızı izgi test sonularını, yeřil izgi ise dođrulama miktarını gstermektedir (řekil 6.4).



Şekil 6.4. Trainlm eğitim fonksiyonu performans grafiği.

*Trainbfg fonksiyonu:* Newton yöntemi, en uygun noktayı hızlı bulmasından dolayı eşlenik gradyan yöntemine alternatif bir yöntemdir. Newton yöntemi, genellikle eşlenik gradyan yönteminden daha hızlı en uygun değere ulaşmaktadır. Ancak, çok katmanlı algılayıcı ağlar için Hessian matrisini hesaplamak yöntemi yavaşlatmaktadır. Newton yöntemine dayanan bazı yöntemler ile ikinci türevleri hesaplamak gerekmemektedir. Bu yöntemlere yaklaşık Newton yöntemleri denir ve her yaklaşımda yaklaşık Hessian matrisi güncellenir (güncelleme gradyanın fonksiyonudur). Bu yaklaşık Newton yöntemleri içinde en başarılı olanı Broyden, Fletcher, Goldfarb ve Shanno (BFGS) güncellemesidir. Eşlenik gradyan yöntemine göre daha az yaklaşımda en uygun noktayı bulmasına rağmen her yaklaşımda daha fazla hesaplama yapmakta ve daha fazla belleğe ihtiyaç duymaktadır. Eğitim fonksiyonu olarak trainbfg (BFGS Quasi-Newton) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.5'deki gibidir. Sonuçlar incelendiğinde eğitim 75 iterasyonda tamamlanmıştır ve eğitim işlemi 1 saniye sürmüştür.

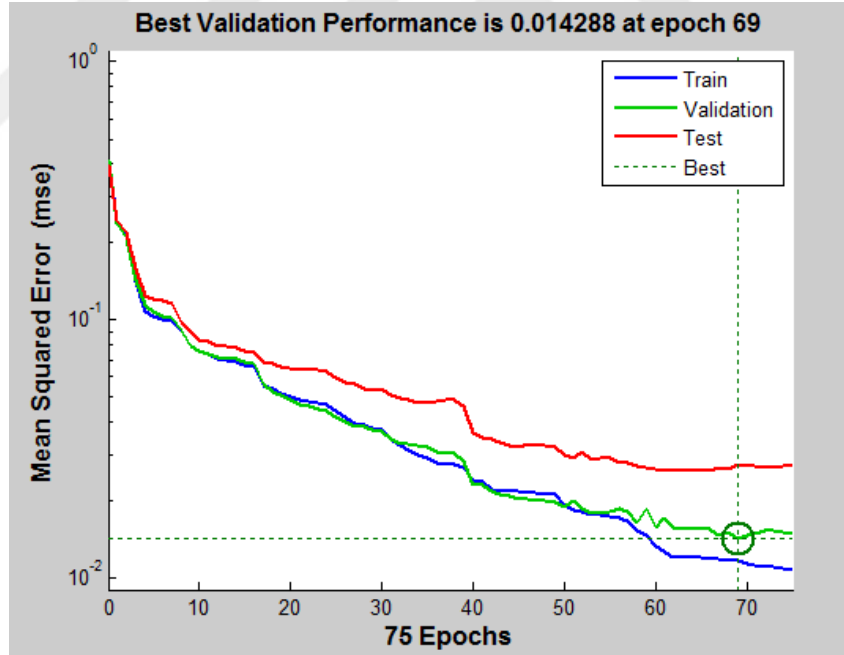
Algorithms			
Data Division:	Random	(dividerand)	
Training:	BFGS Quasi-Newton	(trainbfg)	
Performance:	Mean Squared Error	(mse)	
Derivative:	Default	(defaultderiv)	

Progress			
Epoch:	0	75 iterations	3000
Time:		0:00:01	
Performance:	0.409	0.0108	0.00
Gradient:	0.220	0.0104	1.00e-06
Validation Checks:	0	6	6
Resets:	0.00	0.00	4.00

Şekil 6.5. Trainbfg eğitim fonksiyonu sonuçları.

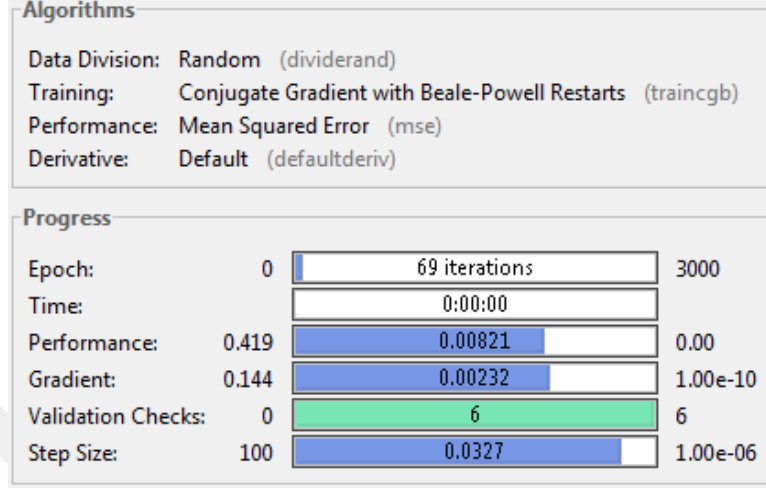
Performans grafiği incelendiğinde en iyi doğrulamanın 69. iterasyonda gerçekleştiği görülmektedir (Şekil 6.6).



Şekil 6.6. Trainbfg eğitim fonksiyonu performans grafiği.

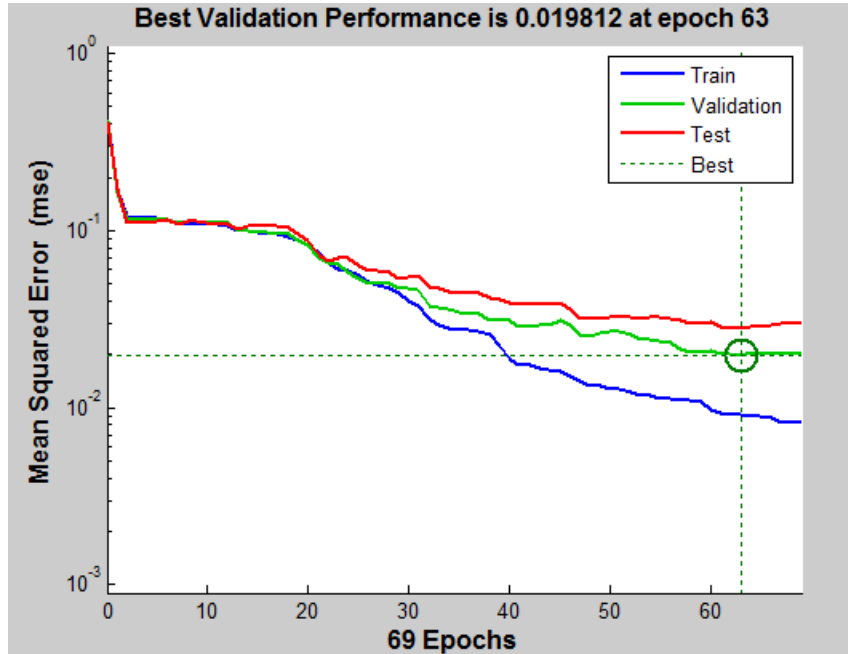
*Traincgb fonksiyonu:* Minimuma giden eşlenik yönlerin bulunmasında gradiyent vektörünün kullanılması ile bu metot ortaya çıkmıştır. Eğitim fonksiyonu olarak traincgb (Conjugate Gradient with Beale-Powell Restarts) fonksiyonu kullanıldığında

elde edilen sonuçlar Şekil 6.7'deki gibidir. Sonuçlar incelendiğinde eğitim 69 iterasyonda tamamlanmıştır ve eğitim işlemi yaklaşık 1 saniye sürmüştür.



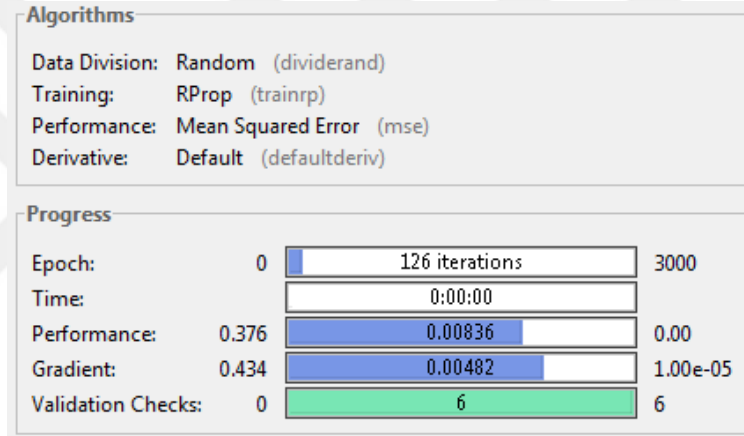
Şekil 6.7. Traincgb eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 63. iterasyonda gerçekleştiği görülmektedir (Şekil 6.8).



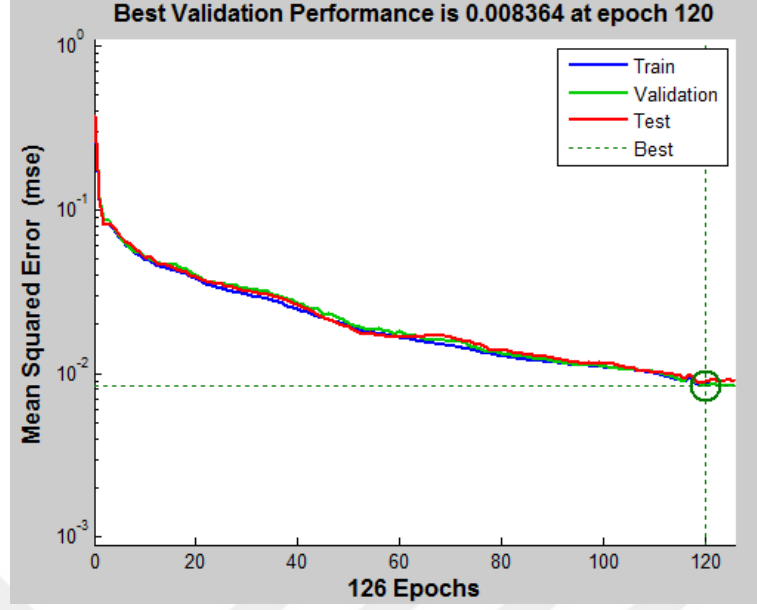
Şekil 6.8. Traincgb eğitim fonksiyonu performans grafiği.

*Trainrp fonksiyonu:* Resilient yayılım algoritması, diğer gradiyenti kullanan eğitim algoritmalarından farklı olarak gradiyentin genliğini kullanmaz. Gradiyentin bölgesel değişim işaretine göre belirlenen ağırlık adımı mevcut ağırlıklara eklenerek eğitim işlemi gerçekleştirilir [126]. Her ağırlık için yenileme değeri ardışık iki iterasyondaki kısmi türevin işaretinin değişip değişmemesine göre belirlenir. Bu öğrenme algoritması genel olarak standart eğitim azaltma algoritmalarından çok hızlıdır. Momentumlu öğrenmeye göre çok daha az hafıza gerektirir. Eğitim fonksiyonu olarak trainrp (resilient backpropagation) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.9'daki gibidir. Sonuçlar incelendiğinde eğitim 126 iterasyonda tamamlanmıştır ve eğitim işlemi yaklaşık 1 saniye sürmüştür.



Şekil 6.9. Trainrp eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 120. iterasyonda gerçekleştiği görülmektedir (Şekil 6.10).



Şekil 6.10. Trainrp eğitim fonksiyonu performans grafiği.

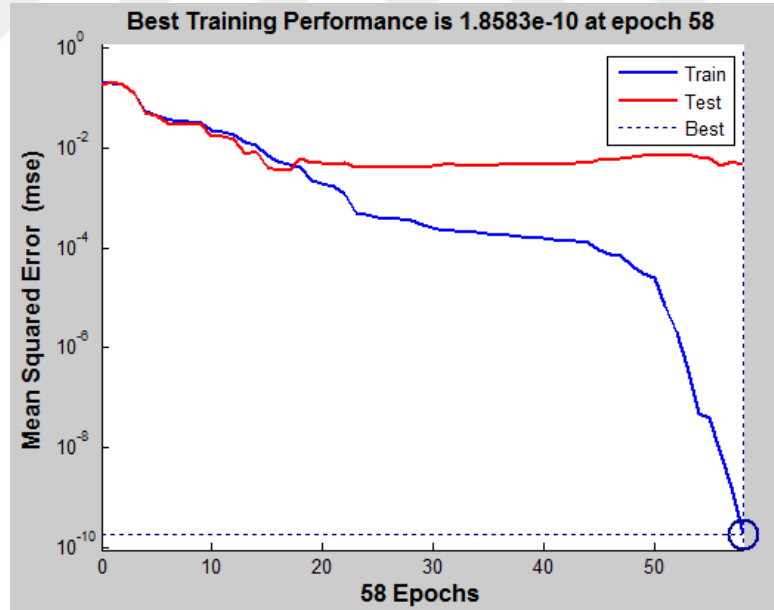
*Trainbr fonksiyonu:* Bayesian regülasyonu Levenberg Marquardt optimizasyonuna göre ağırlık ve bias değerlerini günceller. Karesel hata ve ağırlıkların kombinasyonunu minimize eder ve ağı üretmek için doğru kombinasyonu belirler. Daha iyi genelleştirme yeteneğine sahip bir ağ oluşturmak için Mackay regülasyonu ağ parametrelerinin boyutunu sınırlayan yöntem önermiştir. Regülasyon teknikleri ağırlık değerlerinin daha küçük değerlerde kalması için ağı zorlar. Bu ağın cevabının daha yumuşak olmasına, ağın ezberleme (over fitting) olasılığının azalmasına ve gürültüyü yakalamasına neden olur [127]. Eğitim fonksiyonu olarak trainbr (bayesian regulation) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.11'deki gibidir. Sonuçlar incelendiğinde eğitim 58 iterasyonda tamamlanmıştır ve eğitim işlemi 3 saniye sürmüştür.



Algorithms			
Data Division:	Random	(dividerand)	
Training:	Bayesian Regulation	(trainbr)	
Performance:	Mean Squared Error	(mse)	
Derivative:	Default	(defaultderiv)	
Progress			
Epoch:	0	58 iterations	3000
Time:		0:00:03	
Performance:	0.192	1.86e-10	0.00
Gradient:	0.346	1.42e-08	1.00e-07
Mu:	0.00500	0.0500	1.00e+10
Effective # Param:	146	83.1	0.00
Sum Squared Param:	77.9	4.29e+03	0.00

Şekil 6.11. Trainbr eğitim fonksiyonu sonuçları.

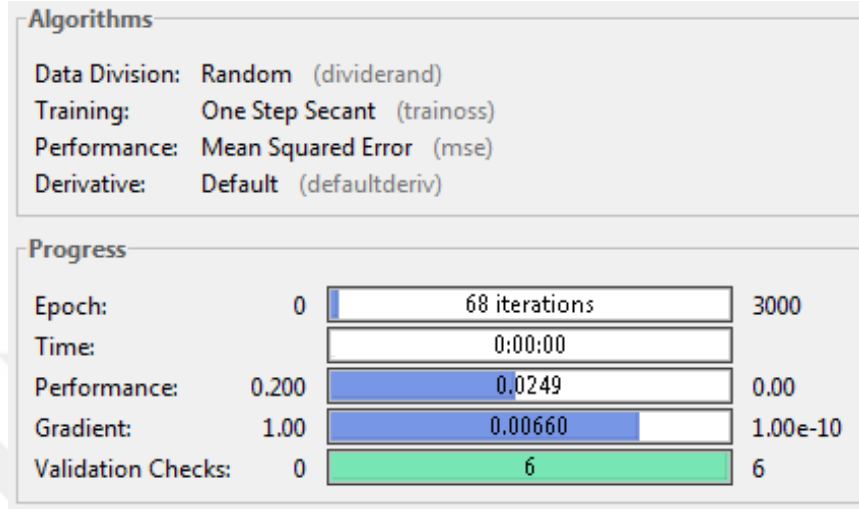
Performans grafiği incelendiğinde en iyi eğitim performansının 58. iterasyonda gerçekleştiği görülmektedir (Şekil 6.12).



Şekil 6.12. Trainbr eğitim fonksiyonu performans grafiği.

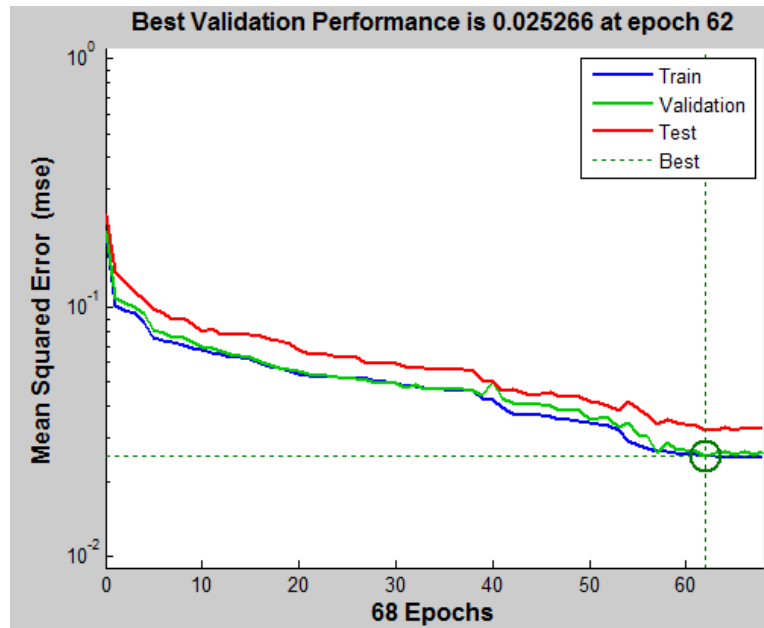
*Trainoss fonksiyonu:* Bazı yöntemlerin her bir iterasyonda ihtiyaç duyduğu hafıza ve işlem zamanını azaltmak için bir sekant yaklaşımına ihtiyacı vardır. Tek Adım Sekant Öğrenme Yöntemi bu alandaki yöntemler arasındaki eksikliklere cevap vermek amacıyla geliştirilen bir algoritmadır. Eğitim fonksiyonu olarak trainoss (one

step secant) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.13'deki gibidir. Sonuçlar incelendiğinde eğitim 68 iterasyonda tamamlanmıştır ve eğitim işlemi yaklaşık 1 saniye sürmüştür.



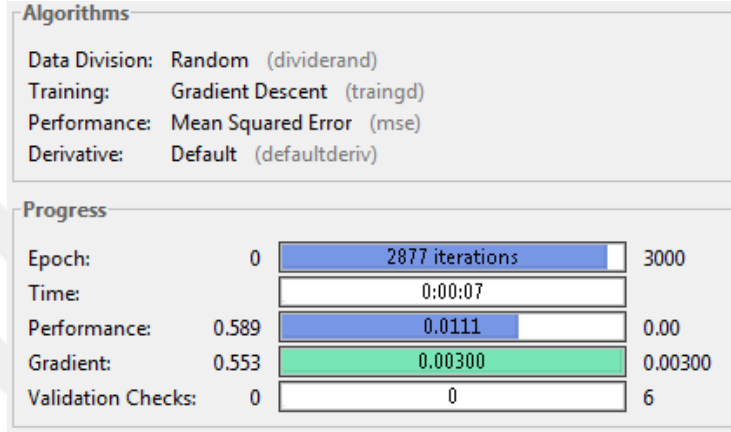
Şekil 6.13. Trainoss eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 62. iterasyonda gerçekleştiği görülmektedir.



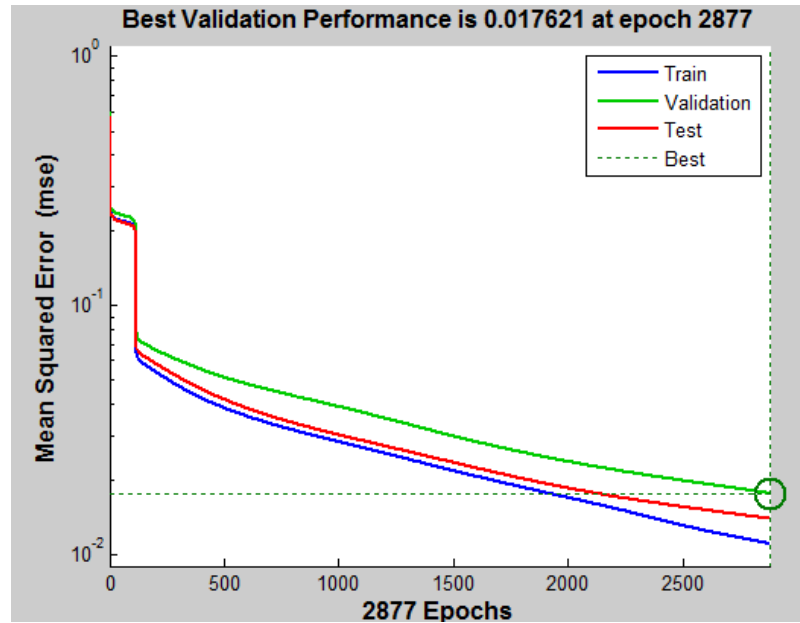
Şekil 6.14. Trainoss eğitim fonksiyonu performans grafiği.

*Traingd fonksiyonu:* En basit geriye yayılım öğrenme algoritması gradyeni azaltması algoritmasıdır. Bu algoritmada ağırlıklar, performans fonksiyonunun azaltılması yönünde ayarlanır. Fakat bu yöntem, pek çok problem için çok yavaş kalmaktadır. Eğitim fonksiyonu olarak traingd (gradient descent) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.15'deki gibidir. Sonuçlar incelendiğinde eğitim 2877 iterasyonda tamamlanmıştır ve eğitim işlemi 7 saniye sürmüştür.



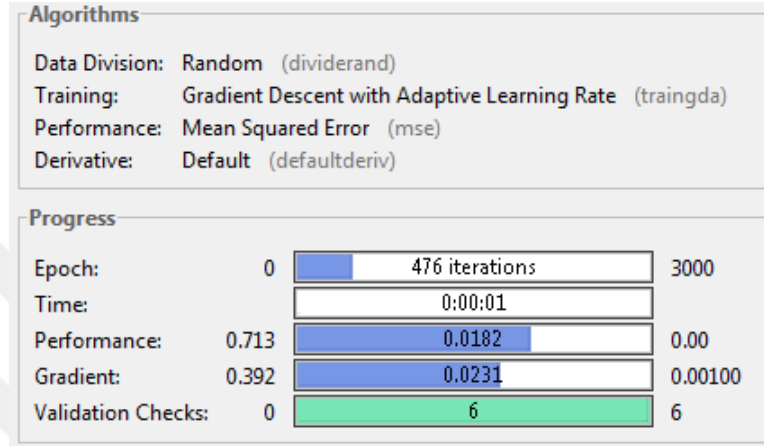
Şekil 6.15. Traingd eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 2877. iterasyonda gerçekleştiği görülmektedir (Şekil 6.16).



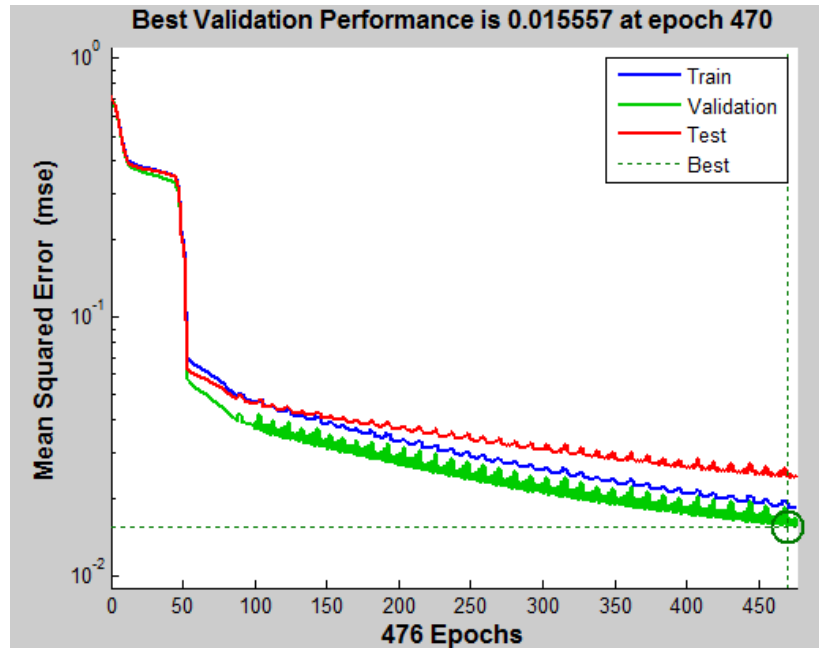
Şekil 6.16. Traingd eğitim fonksiyonu performans grafiği.

*Trainda fonksiyonu:* Gradyen azalma yöntemi gibidir yalnızca bu metotta öğrenme oranı eğitim süresince ayarlanabilir sabit değildir. Eğitim fonksiyonu olarak trainda (gradient descent with adaptive learning rate) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.17'deki gibidir. Sonuçlar incelendiğinde eğitim 476 iterasyonda tamamlanmıştır ve eğitim işlemi 1 saniye sürmüştür.



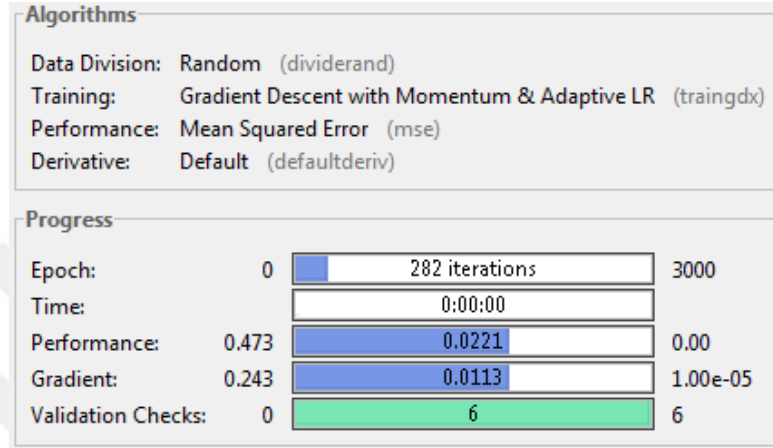
Şekil 6.17. Trainda eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 470. iterasyonda gerçekleştiği görülmektedir (Şekil 6.18).



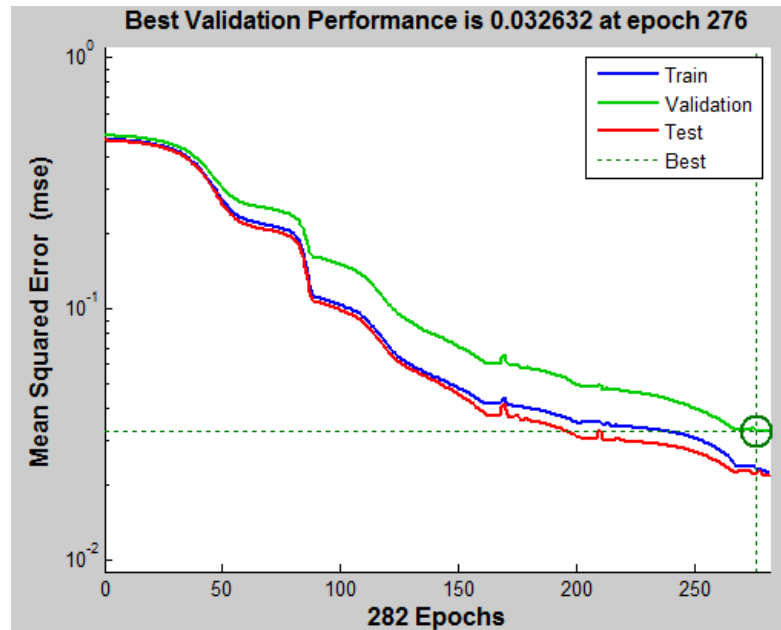
Şekil 6.18. Trainda eğitim fonksiyonu performans grafiği.

*Trainidx* fonksiyonu: Trainda fonksiyonuna momentum sabitinin de eklenmesi halidir. Eğitim fonksiyonu olarak *trainidx* (gradient descent with momentum & adaptive learning rate) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.19'daki gibidir. Sonuçlar incelendiğinde eğitim 282 iterasyonda tamamlanmıştır ve eğitim işlemi yaklaşık 1 saniye sürmüştür.



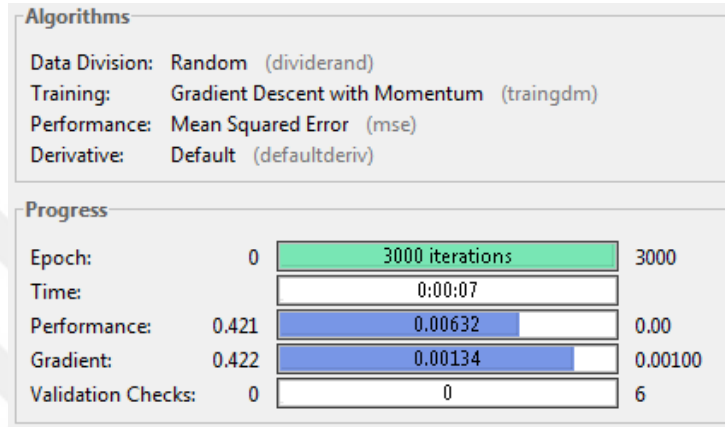
Şekil 6.19. Trainidx eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 276. iterasyonda gerçekleştiği görülmektedir (Şekil 6.20).



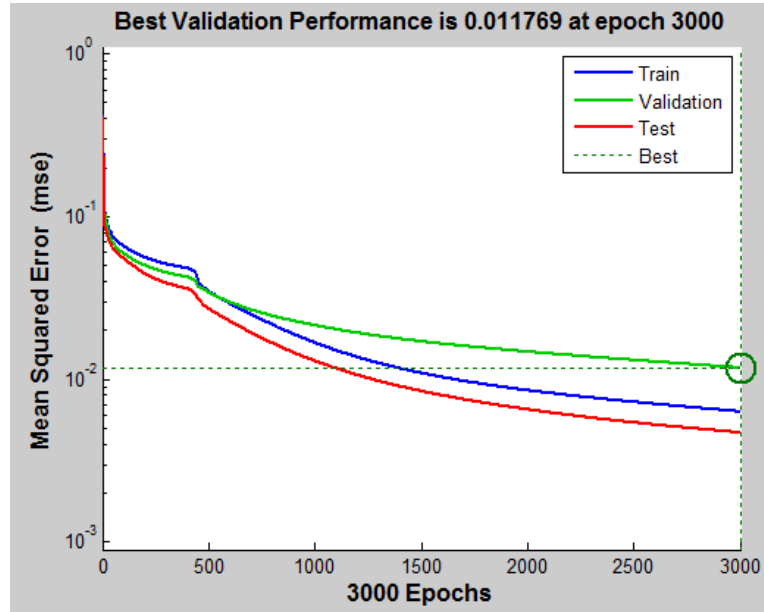
Şekil 6.20. Trainidx eğitim fonksiyonu performans grafiği.

*Traingdm fonksiyonu:* Gradyen azalma yöntemine momentum sabitinin de eklenmiş halidir eğitim ve test işlemi süresince öğrenme oranı ve momentum değerleri başlangıçta verilen değerlerdir. Eğitim fonksiyonu olarak traingdm (gradient descent with momentum) fonksiyonu kullanıldığında elde edilen sonuçlar Şekil 6.21'deki gibidir. Sonuçlar incelendiğinde eğitim 3000 iterasyonda tamamlanmıştır ve eğitim işlemi 7 saniye sürmüştür.



Şekil 6.21. Traingdm eğitim fonksiyonu sonuçları.

Performans grafiği incelendiğinde en iyi doğrulamanın 3000. iterasyonda gerçekleştiği görülmektedir (Şekil 6.22).



Şekil 6.22. Traingdm eğitim fonksiyonu performans grafiği.

Eđitim fonksiyonlarına ait başarı oranları ve eğitim süreleri Çizelge 6.12’de verilmiştir.

Çizelge 6.12. Eğitim fonksiyonlarına ait başarı oranları ve eğitim süreleri.

EĐİTİM FONKSİYONU	BAŞARI ORANI	SÜRE	İTERASYON SAYISI
TRAINLM	%96.39	1 sn.	25
TRAINBFG	%89.85	1 sn.	75
TRAINCGB	%89.58	1 sn.	69
TRAINRP	%96.17	1 sn.	126
TRAINBR	%98.02	3 sn.	58
TRAINOSS	%87.80	1 sn.	68
TRAIINGD	%95.18	7 sn.	2877
TRAIINGDA	%90.87	1 sn.	476
TRAIINGDX	%92.13	1 sn.	282
TRAIINGDM	%98.38	7 sn.	3000

Matlab ortamında yapılan eğitim ve test işlemleri sonucunda, hastalık teşhis edilirken ađı eğitim için programın bir kez çalıştırılmasının yeterli olması sebebiyle en yüksek doğruluk oranına sahip traingdm eğitim fonksiyonunun kullanılması uygun bulunmuştur ve çok katmanlı yapay sinir ađı Visual Studio 2010 platformunda C# dilinde hazırlanmıştır.

#### 6.2.3.1. Çok Katmanlı Yapay Sinir Ağları

Tahmin için kullanılan yapay sinir ağları içinde en yaygın olarak kullanılanı geri yayılım algoritmasıdır. Geri yayılım algoritması gradyen azalan ve çok katmanlı algılayıcıları eğitimde en çok kullanılan temel bir algoritmadır.

Ara katman sayısı ađın performansını yakından ilgilendirmektedir. Deneyerek istenilen ađ yapısına dolayısıyla gizli katman sayısına ulaşılır. Farklı gizli katman sayısına göre ađın performansı test edilip en iyi performansı veren yapı seçilir.

Başlangıç değerleri kadar öğrenme ve momentum katsayılarının belirlenmesi de ađın öğrenme performansı ile yakından ilgilidir. Öğrenme katsayısı ağırlıkların deđişim

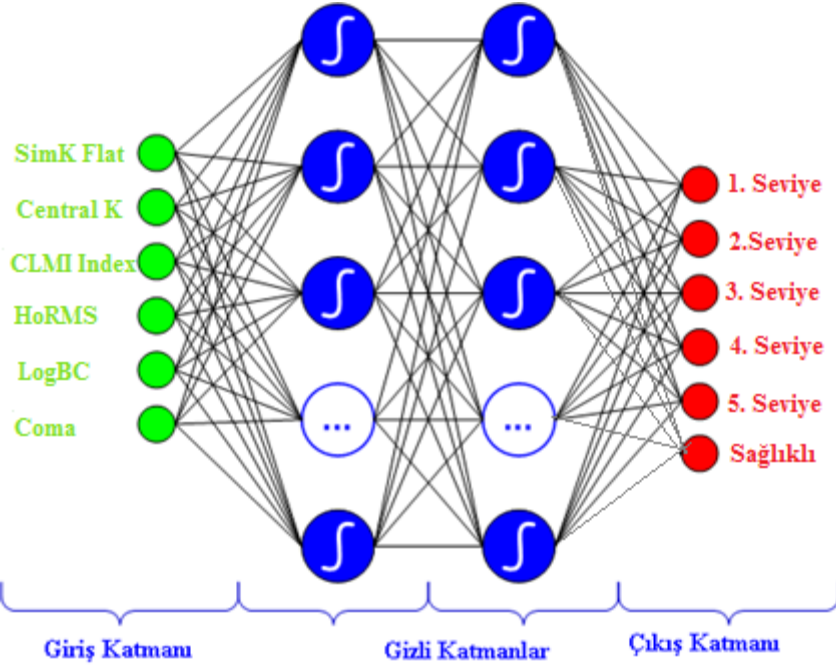
miktarını belirlemektedir. Eđer büyük deęerler seilirse o zaman yerel ozmler arasında aęın dolařması sz konusu olmaktadır. Kk deęerler seilmesi ęrenme zamanını artırmaktadır. Bu deęer denemeler sonucunda bulunmektedir.

Benzer řekilde momentum sabiti de ęrenmenin performansını etkiler. Momentum katsayısı bir nceki iterasyondaki deęiřimin belirli bir oranının yeni deęiřim miktarına eklenmesidir. Bu deęerin kk olması ozmn bulunmasını zorlařtırır. Bk olması ise tek bir ozmn bulunmasında sorunlar yařatabilmektedir.

ok katmanlı algılayıcı aęda aęın eęitilmesi kadar gereęinden fazla eęitilmesi de nemlidir. nk eęitilmek istenen bir aę problem uzayına ozm retecek aęırlıkları bulduktan sonra eęitime devam edilirse bu aęın aęırlıklarında daha fazla deęiřikliklere neden olur. Bylece en iyi ozm reten bir aę tekrar performansı dřk aęlara veya ęrenmeyen aęlara dnřebilir.

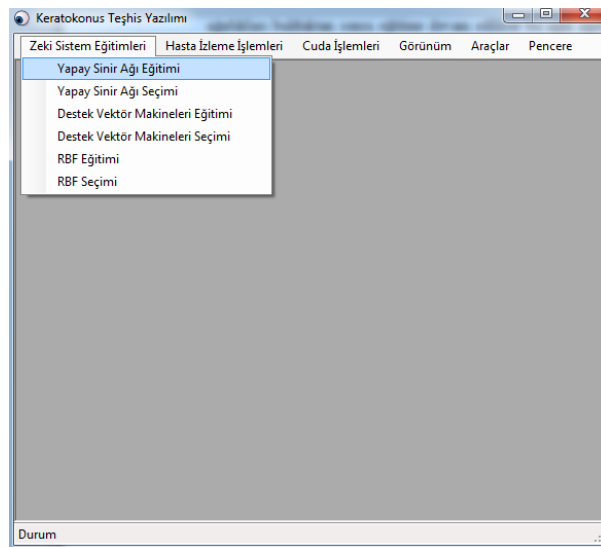
ok katmanlı algılayıcı ile yapılan eęitim denemelerinde hem sigmoid, hem lineer hem de tanjant hiperbolik aktivasyon fonksiyonları kullanılmıř ve en iyi sonu veren fonksiyon eęitimde kullanılmıřtır. ok katmanlı algılayıcıda aęın ilk aęırlık deęerleri ( $w$ ), deneme yolu ile  $[-1,1]$  aralıęında rastgele verilmiřtir. Aęırlıęın rastgele verilmesi nedeniyle aynı zelliklere sahip yapay sinir aęlarının farklı ama yakın hata sonuları rettięi gzlemlenmiřtir. Momentum ve ęrenme oranı ayarlanabilen bir geri yayılım algoritması kullanılmıřtır. Eęitimde ęrenme oranı 0.3 ve momentum 0.7 seildięinde en iyi sonular alınmıřtır. Eęitim iřlemi 3000 adımda tamamlanmıřtır. Uygulamada ok katmanlı algılayıcının giriř katmanında 6 adet giriř deęiřkeni olması nedeniyle 6 nron bulunmaktadır. Gizli katman nron sayısı 10-30 arasında seilmiřtir. 6 adet ıkıř stunu olması nedeniyle ıkıř katmanında 6 nron bulunmaktadır (řekil 6.23).





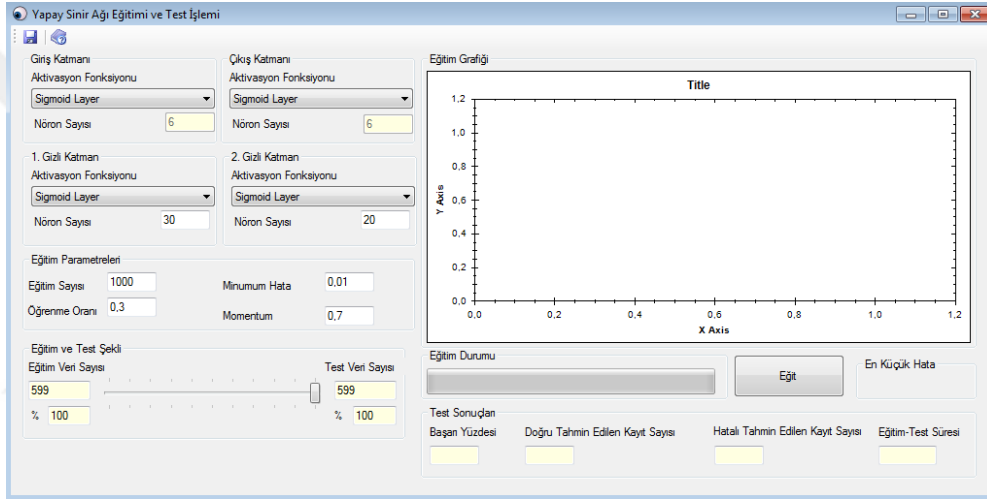
Şekil 6.23. Çok katmanlı algılayıcı yapay sinir ağı mimarisi.

Çalışmada kullanılan yazılım Visual Studio 2010 platformunda C# dilinde hazırlanmıştır. Eğitimde ve test aşamasında kullanılacak veriler ve eğitim sonuçları Access veri tabanında tutulmaktadır. Hazırlanan yazılımın giriş ekranı Şekil 6.24'deki gibidir. Burada Zeki Sistem Eğitimleri Menüsü altında bulunan Yapay Sinir Ağı Eğitimi seçilerek ağ oluşturma ekranına geçilir.



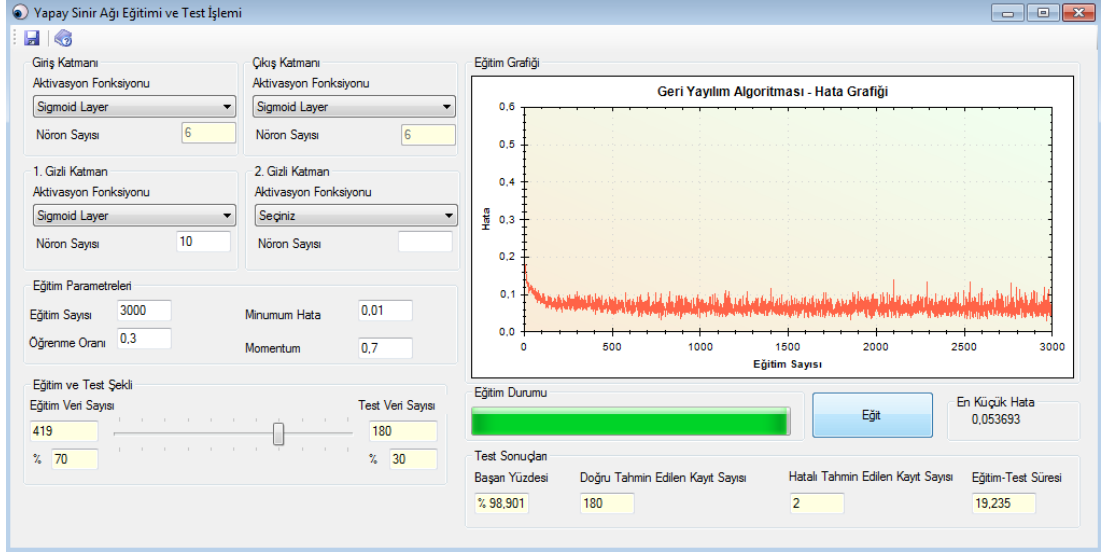
Şekil 6.24. Keratokonus teşhis yazılımı giriş ekranı.

Gelen ekranda giriş ve çıkış katmanına ait nöron sayıları veri tabanında bulunan değişkenlere göre program tarafından otomatik olarak atandığı için değiştirilememektedir. Fakat gizli katmanlara ait nöron sayıları kullanıcı tarafından değiştirilebilir. Ayrıca modelde kullanılacak olan gizli katman sayısı da kullanıcının seçimine bağlıdır. İstenirse iki gizli katmanda kullanılabilir. Ekranda bulunan aktivasyon fonksiyonlarının da seçilmesi gerekmektedir. Aktivasyon fonksiyonları ilgili grup içerisindeki açılır kutulardan seçilerek belirlenir. Yine ekranda bulunan eğitim sayısı, öğrenme oranı eğitim ve test şeklinin nasıl olacağına dair bilgilerin seçilebildiği alanlarda kullanıcı tarafından değiştirilebilecek alanlardır (Şekil 6.25).



Şekil 6.25. Yapay sinir ağı oluşturma ekran görüntüsü.

Kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra “Eğit” butonuna tıklanarak yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmakta ve modele ait hata grafiği çizdirilmektedir. Örnek bir yapay sinir ağı modeli ve sonuçları Şekil 6.26’deki gibidir.



Şekil 6.26. Eğitim sonucunda oluşan ekran görüntüsü (YSA).

Burada Başarı yüzdesi oluşturulan yapay sinir ağı modelinin test aşamasındaki başarısının yüzdeler olarak karşılığıdır. Doğru tahmin edilen kayıt sayısı modelin doğru tahmin ettiği kayıt sayısını göstermektedir. Hatalı tahmin edilen kayıt sayısı da modelin yanlış tahmin ettiği kayıt sayısını göstermektedir. Bu ekranda kaydet butonuna tıklanarak eğitilen yapay sinir ağı veri tabanına kaydedilebilir. Daha sonra kaydedilen bu yapay sinir ağları Zeki Sistem Eğitimleri menüsü altında bulunan Yapay Sinir Ağları Seçimi tıklanarak görüntülenebilir. Bu ekranda kaydedilen yapay sinir ağları listelenir teşhiste kullanılmak istenen yapay sinir ağı en solunda bulunan kutucuk işaretlenerek aktif edilir. Herhangi bir yapay sinir ağı üzerinde çift tıklanarak yapay sinir ağının detaylı ekran görüntüsü açılabilir (Şekil 6.27).

The screenshot shows the 'Eğitimi Tamamlanmış Yapay Sinir Ağı Seçimi' (Selected Artificial Neural Network after Training) window. It displays a table of trained models with the following columns: Aktif Yap, EĞİTİM\_ORANI, BAŞARI\_YÜZDE, DOĞRU\_TAHMİN, and YANLIŞ\_TAHMİN. The 8th row is selected.

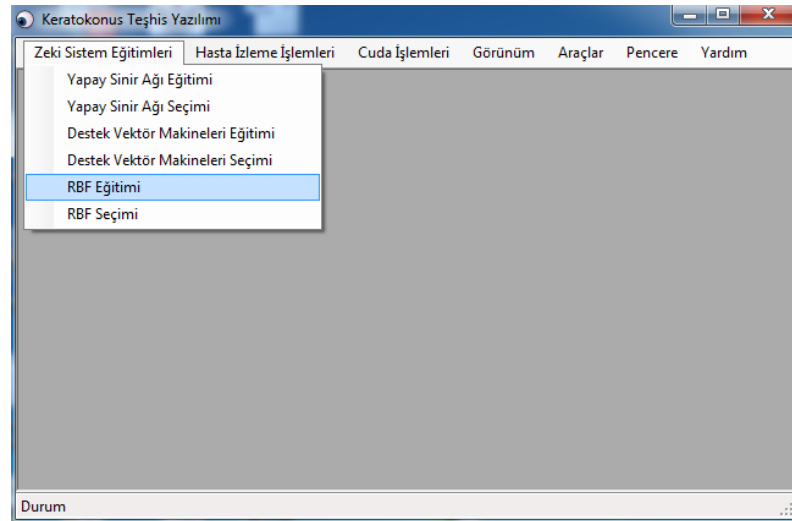
Aktif Yap	EĞİTİM_ORANI	BAŞARI_YÜZDE	DOĞRU_TAHMİN	YANLIŞ_TAHMİN
<input type="checkbox"/>	100	% 96,661	579	20
<input type="checkbox"/>	100	% 91,486	548	51
<input type="checkbox"/>	70	% 94,318	166	10
<input type="checkbox"/>	70	% 94,318	166	10
<input type="checkbox"/>	100	% 96,828	580	19
<input type="checkbox"/>	100	% 96,661	579	20
<input checked="" type="checkbox"/>	80	% 95,69	111	5
<input type="checkbox"/>	100	% 96,16	576	23
<input type="checkbox"/>	100	% 98,831	592	7

Şekil 6.27. Yapay sinir ağı listeleme ve seçim ekranı.

### 6.2.3.2. Radyal Tabanlı Fonksiyon Ağları

Radyal tabanlı fonksiyon ağlarında girdi katmanı sadece modele dış dünyadan veri alınmasını sağlar. Başka bir ifade ile girdi katmanını gizli katmana bağlayan tüm ağırlık değerlerinin “1” olduğu ve çözüm süresince değişmediği varsayılır. Bu özelliği ile öğrenme aşamasında değeri değiştirilecek parametre sayısında önemli bir azalma gerçekleşir ve dolayısıyla öğrenme hızlanır. Bu bakımdan RTFA'nın çok katmanlı YSA'ya göre daha kullanışlı olduğu söylenebilir. Bunun nedeni, eldeki problemin çözümüne uygun RTFA'nın oluşturulması aşamasında ağ mimarisine ilişkin verilecek tek kararın, gizli katmanda bulunacak nöron sayısının belirlenmesi olmasıdır. Gizli katman nöron sayısı RTF ağlarının mimarisinde karar verilmesi gereken en önemli parametrelerden biridir. Aşırı uyum ve yetersiz öğrenme durumlarına düşmemek gerekir.

Uygulamada radyal tabanlı fonksiyonun giriş katmanında 6 adet giriş sütunu olması nedeniyle 6 nöron bulunmaktadır. Gizli katman nöron sayısı 30-150 arasında deneme yanılma yoluyla seçilmiştir. 6 adet çıkış sütunu olması nedeniyle çıkış katmanında 6 nöron bulunmaktadır. En iyi sonuçların elde edilmesi sebebiyle aktivasyon fonksiyonu olarak Gauss kullanılmıştır. Radyal Tabanlı Fonksiyon Ağları ile sınıflandırma uygulamasını başlatmak için Zeki Sistem Eğitimler menüsü altında bulunan RBF Eğitimi seçeneği tıklanır (Şekil 6.28).




Şekil 6.28. Radyal tabanlı fonksiyon ağları giriş ekranı.

Gelen ekranda aktivasyon fonksiyonu ve eğitim test şeklinin nasıl olacağına dair bilgiler seçildikten sonra Eğit butonuna tıklanarak eğitim ve test işlemine geçilebilir (Şekil 6.29).

Başarı Yüzdesi	Doğru Tahmin Edilen Kayıt Sayısı	Hatalı Tahmin Edilen Kayıt Sayısı	Eğitim-Test Süresi
% 95,993	575	24	18,439

Şekil 6.29. Eğitim ve test sonucunda oluşan ekran görüntüsü (RBF).

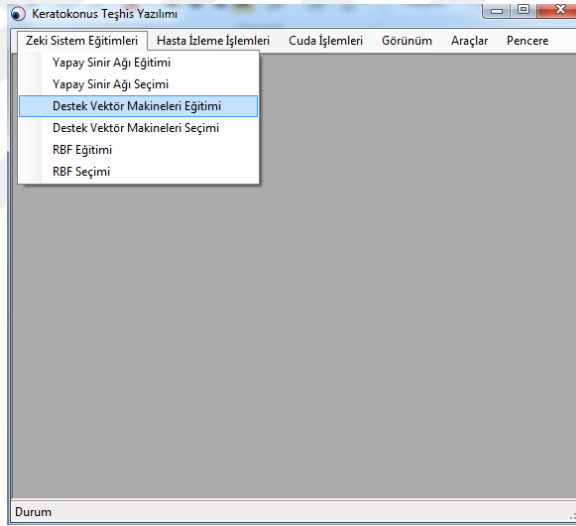
Bu ekranda bulunan kaydet  butonuna tıklanarak eğitilen radyal tabanlı fonksiyon ağı modeli kaydedilebilir. Daha sonra kaydedilen bu ağlar Zeki Sistem Eğitimleri menüsü altında bulunan RBF Seçimi tıklanarak görüntülenebilir. Bu ekranda kaydedilen ağlar listelenir teşhiste kullanılmak istenen radyal tabanlı fonksiyon ağı en solunda bulunan kutucuk işaretlenerek aktif edilir. Herhangi bir ağ üzerinde çift tıklanarak bu modelin detaylı ekran görüntüsü açılabilir (Şekil 6.30).

	Aktif Yap	EĞİTİM_ORANI	BAŞARI_YÜZDE	DOĞRU_TAHMİN	YANLIŞ_TAHMİN
▶	<input checked="" type="checkbox"/>	100	% 94,824	568	31
	<input type="checkbox"/>	100	% 93,656	561	38
	<input type="checkbox"/>	100	% 95,993	575	24
	<input type="checkbox"/>	100	% 93,989	563	36
	<input type="checkbox"/>	100	% 91,151	545	54
	<input type="checkbox"/>	100	% 91,485	548	51
	<input type="checkbox"/>	100	% 92,320	553	46
	<input type="checkbox"/>	70	% 95,325	173	9
	<input type="checkbox"/>	100	% 95,158	570	29

Şekil 6.30. Radyal tabanlı fonksiyon ağı listeleme ve seçim ekranı.

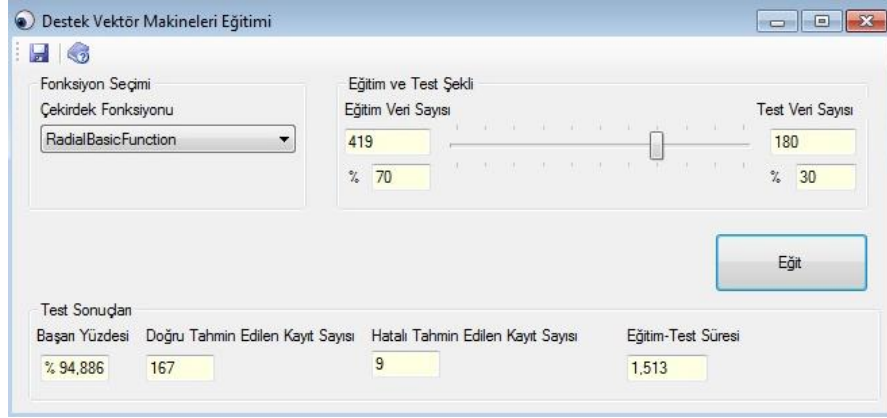
### 6.2.3.3. Destek Vektör Makineleri

DVM sınıflandırıcılar iki sınıf etiketli durumlar düşünülerek eğitilirler. Çünkü DVM iki sınıfı birbirinden doğrudan ayırabilen bir ayırıcı fonksiyonun araştırılması üzerine kurulmuş olan bir yöntemdir. Bu çalışmada da çıkış iki sınıflı olmadığı için çoklu DVM çözümlerinden yararlanılmıştır. Literatürde iki farklı yaklaşım bulunmaktadır. Bunlardan birincisi bire bir diğeri ise bire karşı hepsi mantığıyla işlem yapmaktadır. Uygulamada bire bir DVM çözümü kullanılmıştır. Veri setinin çıkışı 6 sınıflı olduğu için  $k*(k-1)/2$  formülünden 15 farklı DVM eğitilmiştir. Her eğitim aşamasında sadece iki farklı sınıf verisi alınmıştır. Destek Vektör Makineleri ile sınıflandırma uygulamasını başlatmak için Zeki Sistem Eğitimleri menüsü altında bulunan Destek Vektör Makineleri Eğitimi seçeneği tıklanır (Şekil 6.31).




Şekil 6.31. Destek vektör makineleri giriş ekranı.

Gelen ekranda çekirdek fonksiyonu ve eğitim test şeklinin nasıl olacağına dair bilgiler seçildikten sonra Eğit butonuna tıklanarak eğitim ve test işlemine geçilebilir (Şekil 6.32).



Şekil 6.32. Eğitim ve test sonucunda oluşan ekran görüntüsü (DVM).

Bu ekranda bulunan kaydet  butonuna tıklanarak eğitilen destek vektör makinesi modeli kaydedilebilir. Daha sonra kaydedilen bu destek vektör makineleri Zeki Sistem Eğitimleri menüsü altında bulunan Destek Vektör Makinesi Seçimi tıklanarak görüntülenebilir. Bu ekranda kaydedilen destek vektör makineleri listelenir teşhiste kullanılmak istenen destek vektör makineleri en solunda bulunan kutucuk işaretlenerek aktif edilir. Herhangi bir destek vektör makinesi üzerinde çift tıklanarak bu modelin detaylı ekran görüntüsü açılabilir (Şekil 6.33).

	Aktif Yap	EĞİTİM_ORANI	BAŞARI_YÜZDE	DOĞRU_TAHMİN	YANLIŞ_TAHMİN
▶	<input type="checkbox"/>	100	% 99,165	594	5
	<input type="checkbox"/>	70	% 94,886	167	9
	<input type="checkbox"/>	90	% 94,737	54	3
	<input type="checkbox"/>	80	% 93,966	109	7
	<input checked="" type="checkbox"/>	100	% 99,165	594	5
	<input type="checkbox"/>	70	% 94,886	167	9
	<input type="checkbox"/>	80	% 93,966	109	7

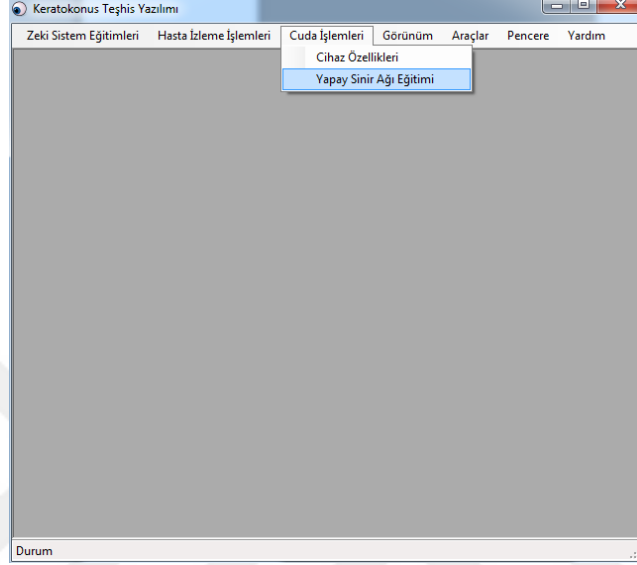
Şekil 6.33. Destek vektör makinesi listeleme ve seçim ekranı.

#### 6.2.4. CUDA Destekli Sınıflandırma Yöntemleri

Kümeleme analizi sonrasında elde edilen gruplara CUDA destekli yapay sinir ağı yöntemi uygulanmıştır.

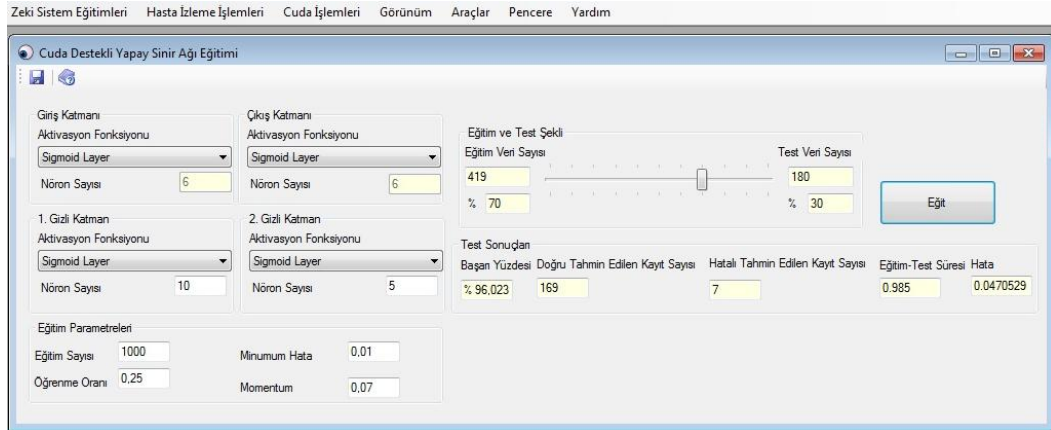
### 6.2.4.1. CUDA Destekli MLP Yapay Sinir Ağı

CUDA destekli MLP yapay sinir ağı eğitimi ekranını açmak için CUDA işlemleri menüsünün altında bulunan Yapay Sinir Ağı Eğitimi seçilir (Şekil 6.34).



Şekil 6.34. CUDA destekli MLP yapay sinir ağı giriş ekranı.

Gelen ekranda kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra “Eğit” butonuna tıklanarak CUDA destekli MLP yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmaktadır. Örnek bir CUDA destekli MLP yapay sinir ağı modeli ve sonuçları Şekil 6.35’deki gibidir.

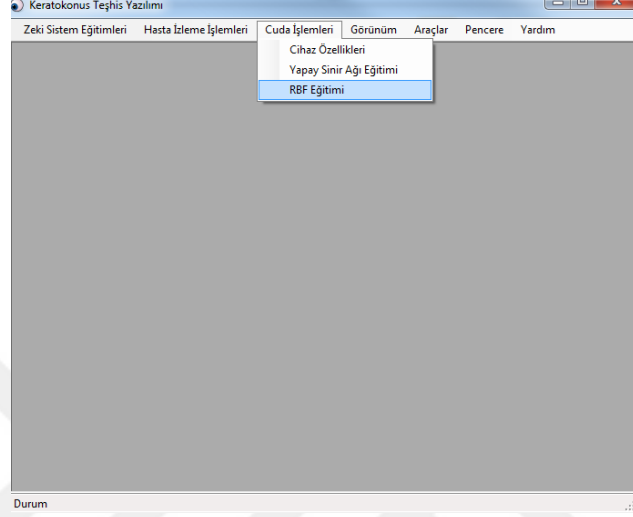


Şekil 6.35. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDAMLP).



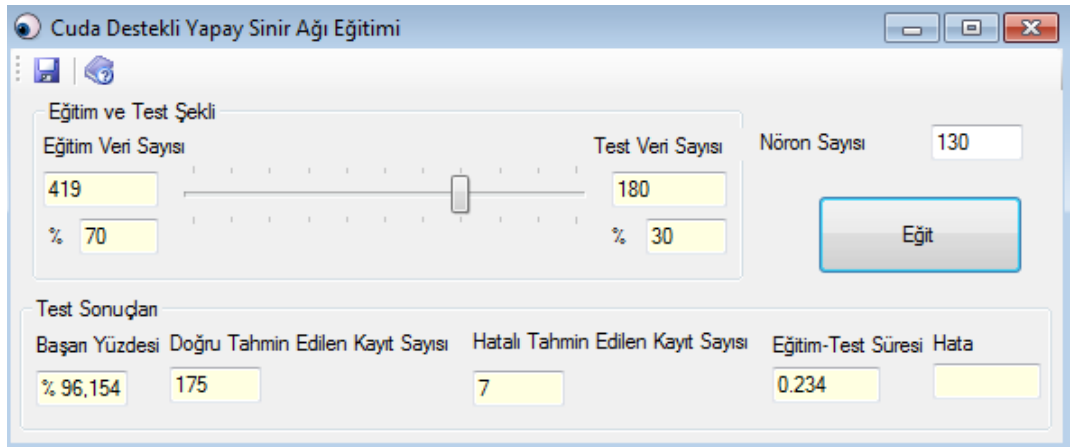
#### 6.2.4.2. CUDA Destekli RBF Yapay Sinir Ağı

CUDA destekli RBF yapay sinir ağı eğitimi ekranını açmak için CUDA işlemleri menüsünün altında bulunan RBF Eğitimi seçilir (Şekil 6.36).



Şekil 6.36. CUDA destekli RBF yapay sinir ağı giriş ekranı.

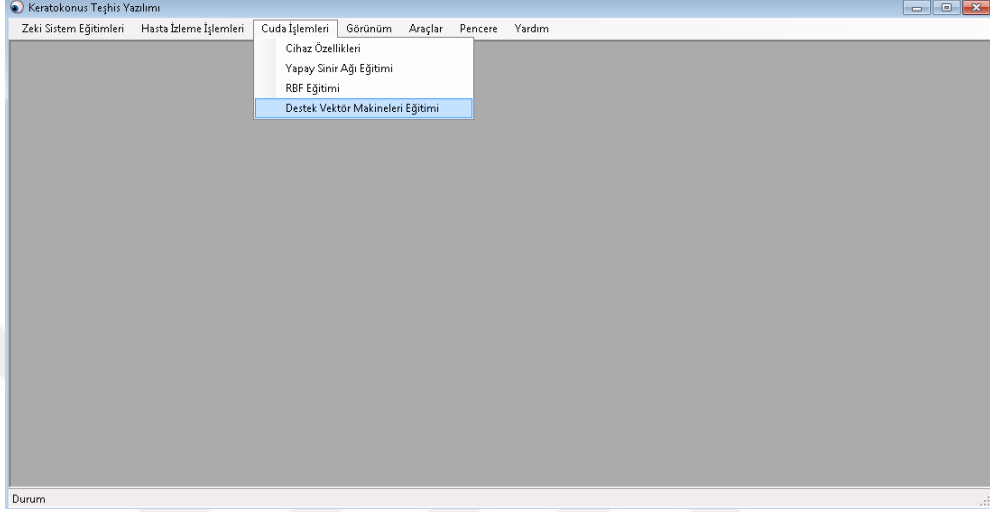
Gelen ekranda kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra “Eğit” butonuna tıklanarak CUDA destekli RBF yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmaktadır. Örnek bir CUDA destekli RBF yapay sinir ağı modeli ve sonuçları Şekil 6.37’deki gibidir.



Şekil 6.37. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDARBF).

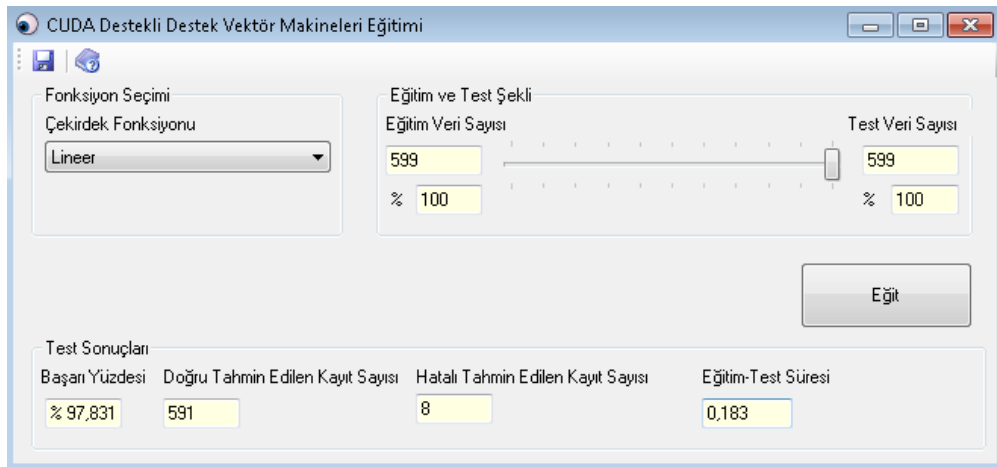
### 6.2.4.3. CUDA Destekli DVM Yapay Sinir Ağı

CUDA destekli DVM yapay sinir ağı eğitimi ekranını açmak için CUDA işlemleri menüsünün altında bulunan Destek Vektör Makineleri Eğitimi seçilir (Şekil 6.38).



Şekil 6.38. CUDA destekli DVM yapay sinir ağı giriş ekranı.

Gelen ekranda kullanıcı tarafından istenilen parametrelerde değişiklikler gerçekleştirildikten sonra "Eğit" butonuna tıklanarak CUDA destekli DVM yapay sinir ağının eğitilmesi işlemine geçilir. Eğitim ve test aşamaları bittikten sonra Test sonuçları başlığı altında bulunan alanlar dolmaktadır. Örnek bir CUDA destekli DVM yapay sinir ağı modeli ve sonuçları Şekil 6.39'daki gibidir.



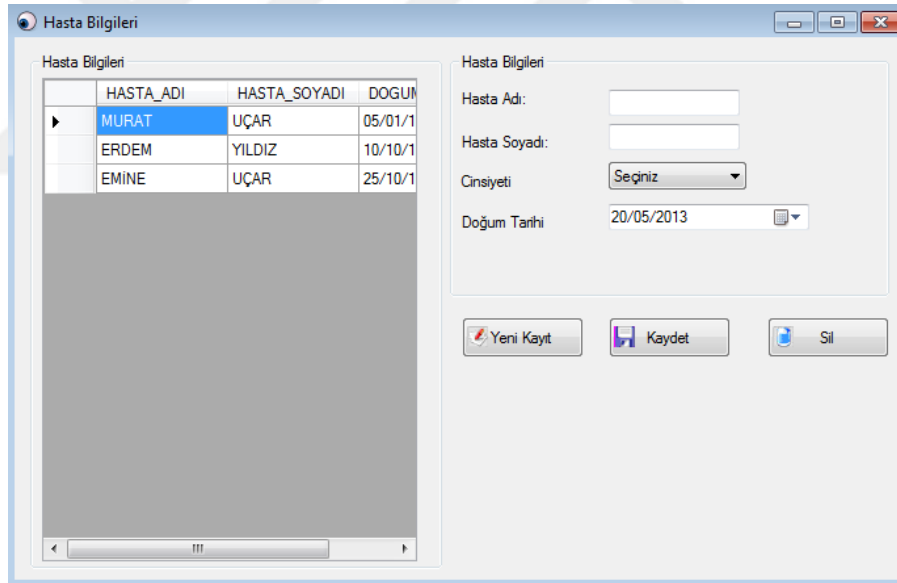
Şekil 6.39. Eğitim ve test sonucunda oluşan ekran görüntüsü (CUDADVM).

## 6.2.5. Zeki Sistemin Hastalık Teşhisinde Kullanımı

Geliştirilen programın hastane ortamında keratokonus hastalığının teşhisinde kullanılması için programa hasta bilgileri bölümü eklenmiştir. Bu bölümde tedavi için başvuran hastaların bilgilerinin takibi ve farklı tarihlerde yapılan muayenelerinde ölçülen değerlerden hastalığın seviyesinin ve ilerleyişinin takibinin yapılması amaçlanmaktadır.

### 6.2.5.1. Hasta Bilgileri Giriş

Hasta İzleme İşlemleri menüsü altında bulunan Hasta Bilgileri Giriş seçeneği tıklanır. Bu ekranda hastalara ait kimlik bilgilerinin kaydedilmesi güncellenmesi ve silinmesi işlemi gerçekleştirilmektedir (Şekil 6.40).



HASTA_ADI	HASTA_SOYADI	DOĞUM
MURAT	UÇAR	05/01/1
ERDEM	YILDIZ	10/10/1
EMINE	UÇAR	25/10/1

Şekil 6.40. Hasta bilgileri giriş ekranı.

### 6.2.5.2. Teşhis İşlemi

Hasta İzleme İşlemleri menüsü altında bulunan Teşhis İşlemi seçeneği tıklanır. Burada öncelikle Hasta Seçim Formu ekranı gelir. Bu ekranda teşhis yapılmak istenen hasta seçilir ve kayıt üzerinde çift tıklanarak Hastalık Teşhis Formu ekranına geçilir (Şekil 6.41).

HASTA_ADI	HASTA_SOYADI	DOGUMTARIHI
MURAT	UÇAR	05/01/1981
ERDEM	YILDIZ	10/10/1983
EMİNE	UÇAR	25/10/1982

Şekil 6.41. Hasta seçim formu ekranı.

Hastalık Teşhis Formu ekranında hastaya ait daha önce ölçülen sonuçlar ve hastalığın tahmin edilen teşhislerine ait veriler bulunmaktadır. Bu ekranda sol tarafta bulunan listeden geçmiş tarihli ölçümler seçilerek ayrıntıları görüntülenebilir. Yapılan yeni ölçümler ilgili alanlara girilerek bu değerlere göre zeki sistem seçim ekranında seçilen yapay sinir ağları ve destek vektör makineleri modelleri kullanılarak hastalığın seviyesi teşhis edilebilir. Önceki sonuçlarla hastanın şu anki durumu karşılaştırılarak hastalığın ilerleyişi ve tedavisi hakkında karar verilebilir (Şekil 6.42).

SIMKFLAT	CENTRALK	CLMIINDEX	HORMS	LOGBC	COMA
45	44.93	1.3	0.706	0	0.360811
41.37	44.4	0.3	0.296	0	0.107781
44.09	49.7	9.24	2.816	0.699	2.2091
45.78	51.6	4.8	1.68	0.0969	1.5789

Şekil 6.42. Hastalık teşhis formu ekranı.

## BÖLÜM 7

### SONUÇLAR VE DEĞERLENDİRME

Keratokonus hastalığının şiddeti ve hangi aşamada olduğu bilgisinin tedavi sonuçlarını etkilemesi nedeniyle hastalığa yaklaşımın ilk adımı hastalığın sınıflandırılmasıdır. Daha önceki çalışmalarda farklı kriterlere göre farklı sınıflandırmalar yapılmıştır. Fakat bu sınıflandırmalar hep tek bir değişkene göre yapılmış ve hastalığın ilerleme durumunun sağlıklı bir şekilde takip edilememesine sebep olmuştur. Bu çalışmayı yapmamızdaki amaç sağlam bir istatistiksel yöntem ile keratokonus hastalarından elde edilen refraksiyon kusurları, görme keskinlikleri, keratometri değerleri, pakimetri değerleri, aberasyonlar ve CLMI büyüklükleri gibi birçok değişken arasından sınıflandırma için en uygun değişkenleri tespit ederek yeni bir sınıflandırma metodu oluşturmaktır. Yapılan çalışma sonucunda sınıflandırmaya etki eden en önemli faktörler SimKflat, CentralK, CLMIIndex, Ho RMS, LogBC ve Coma olarak bulunmuş ve keratokonusun 5 gruba ayrılmasına karar verilmiştir [128]. Yeni sınıflandırma, yapay sinir ağları ve destek vektör makineleri yöntemleri ile test edilerek doğruluğu kanıtlanmıştır [129]. İleride hazırlanan bu program ile hastalık değerlerinin ölçümünde kullanılan cihazların entegre edilerek gerçek zamanlı hastalık teşhisi yapılması düşünülmektedir.

Yapay sinir ağları modellerinde çok katmanlı algılayıcı ve radyal tabanlı fonksiyon ağları mimarisi kullanılmıştır. Çok katmanlı algılayıcı ile yapılan eğitimde 1. gizli katmanda 10 ve çıkış katmanında 6 nörondan oluşan bir ağ yapısı tasarlanmıştır. Momentum ve öğrenme oranı ayarlanabilen bir geri yayılım algoritması kullanılmıştır. Eğitimde öğrenme oranı 0.3 ve momentum 0.7 seçildiğinde en iyi sonuçlar alınmıştır. Eğitim işlemi 3000 adımda tamamlanmıştır. Modelde tüm katmanlar için aktivasyon fonksiyonu sigmoid olarak seçilmiştir. Eğitim ve test aşaması minimum 16,46 maksimum 48,08 saniye sürmüştür Çok katmanlı yapay sinir ağları modelinin hata değeri ve başarı yüzdesi Çizelge 7.1’de verilmiştir.

Çizelge 7.1. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (MLP).

	Öğr. Oranı	Giriş Kat. Nör.	Giriş Kat. Aktiv. Fonk.	1. Gizli Kat. Nör.	1. Gizli Kat. Aktiv. Fonk.	2. Gizli Kat. Nör.	2. Gizli Kat. Aktiv. Fonk.	Çıkış Kat. Nör.	Çıkış Kat. Aktiv. Fonk.	Hata	Başarı Oranı (%)	Eğitim ve Test Süresi
Çok Katmanlı Algılayıcı	0.1	6	Sigmoid	20	Sigmoid	-	---	6	Sigmoid	0.041	97.80	18.44
	0.2	6	Sigmoid	10	Sigmoid	-	---	6	Sigmoid	0.042	97.25	16.46
	0.3	6	Sigmoid	10	Sigmoid	-	---	6	Sigmoid	0.030	98.90	19.23
	0.4	6	Sigmoid	10	Sigmoid	8	Sigmoid	6	Sigmoid	0.084	95.60	25.24
	0.5	6	Sigmoid	10	Sigmoid	8	Sigmoid	6	Sigmoid	0.090	93.95	28.56
	0.3	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.124	91.75	48.08
	0.4	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.088	93.95	47.30
	0.5	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.098	92.30	28.56
	0.2	6	Tanjant	30	Tanjant	-	---	6	Tanjant	2.067	71.12	25.54

Radyal tabanlı fonksiyon ağları ile yapılan eğitimde gizli katmanda sırasıyla 30-150 arasında nöron kullanılmıştır. Bant genişliği [0,2] aralığında kullanılmıştır. Aktivasyon fonksiyonu olarak en iyi sonucu veren gauss fonksiyonu kullanılmıştır. Eğitim işlemi 1000 adımda tamamlanmıştır. Eğitim ve test aşaması minimum 4,46 maksimum 14,32 saniye sürmüştür. Yapılan eğitimler sonucu elde edilen hata değeri ve başarı yüzdesi Çizelge 7.2’de gösterildiği gibidir.

Çizelge 7.2. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (RBF).

	Giriş Katmanı Nöron	Ara Katman Aktivasyon Fonksiyonu	Ara Katman Nöron Sayısı	Çıkış Katmanı Nöron	Hata	Başarı Oranı (%)	Eğitim ve Test Süresi
Radyal Tabanlı Fonksiyon	6	Gauss	30	6	0.090	91.151	4.461
	6	Gauss	40	6	0.085	91.485	4.572
	6	Gauss	60	6	0.063	93.656	5.856
	6	Gauss	80	6	0.060	93.989	6.895
	6	Gauss	100	6	0.051	94.824	7.348
	6	Gauss	110	6	0.048	95.158	8.652
	6	Gauss	120	6	0.049	95.325	9.854
	6	Gauss	130	6	0.040	95.993	10.593
	6	Gauss	140	6	0.055	94.490	12.242
	6	Gauss	150	6	0.076	92.320	14.328

Destek vektör makineleri ile yapılan eğitimde veri setinin çıkışı 6 sınıflı olduğu için 15 DVM eğitilmiştir. Her bir eğitim aşamasında sadece iki farklı sınıf verisi alınmıştır. Çekirdek fonksiyonu lineer seçildiğinde en yüksek başarı elde edilmiştir. Eğitim ve test aşamaları birlikte minimum 1,38 maksimum 1,51 saniye sürmüştür. Yapılan eğitimler sonucu elde edilen başarı yüzdesi ve parametreler Çizelge 7.3'de gösterildiği gibidir.

Çizelge 7.3. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (DVM).

	Giriş Katmanı Değişken Sayısı	Çekirdek Fonksiyonu	Çoklu DVM Çözümü	Başarı Oranı (%)	Eğitim ve Test Süresi
Destek Vektör Makineleri	6	Polinom	Bire Karşı Bir	96.591	1.388
	6	Sigmoid	Bire Karşı Bir	64.773	1.513
	6	Radyal Tabanlı	Bire Karşı Bir	94.886	1.435
	6	Lineer	Bire Karşı Bir	97.727	1.404

CUDA destekli çok katmanlı algılayıcı ile yapılan eğitimde 1. gizli katmanda 10 ve çıkış katmanında 6 nörondan oluşan bir ağ yapısı tasarlanmıştır. Momentum ve öğrenme oranı ayarlanabilen bir geri yayılım algoritması kullanılmıştır. Eğitimde öğrenme oranı 0.3 ve momentum 0.7 seçildiğinde en iyi sonuçlar alınmıştır. Eğitim işlemi 3000 adımda tamamlanmıştır. Modelde tüm katmanlar için aktivasyon fonksiyonu sigmoid olarak seçilmiştir. Eğitim ve test aşaması minimum 2,39 maksimum 6,55 saniye sürmüştür. Çizelge 7.4'de değiştirilen parametrelere göre CUDA destekli çok katmanlı yapay sinir ağları modelinin hata değeri ve başarı yüzdesi verilmiştir.

Çizelge 7.4. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDAMLP).

	Öğr. Oranı	Giriş Kat. Nör.	Giriş Kat. Aktiv. Fonk.	1. Gizli Kat. Nör.	1. Gizli Kat. Aktiv. Fonk.	2. Gizli Kat. Nör.	2. Gizli Kat. Aktiv. Fonk.	Çıkış Kat. Nör.	Çıkış Kat. Aktiv. Fonk.	Hata	Başarı Oranı (%)	Eğitim ve Test Süresi
CUDA Destekli Çok Katmanlı Algılayıcı	0.1	6	Sigmoid	20	Sigmoid	-	---	6	Sigmoid	0.016	97.25	4.39
	0.2	6	Sigmoid	10	Sigmoid	-	---	6	Sigmoid	0.017	97.80	2.40
	0.3	6	Sigmoid	10	Sigmoid	-	---	6	Sigmoid	0.013	98.40	2.39
	0.4	6	Sigmoid	10	Sigmoid	8	Sigmoid	6	Sigmoid	0.018	97.25	3.36
	0.5	6	Sigmoid	10	Sigmoid	8	Sigmoid	6	Sigmoid	0.029	96.15	3.11
	0.3	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.026	96.70	6.03
	0.4	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.028	96.59	6.54
	0.5	6	Sigmoid	30	Sigmoid	-	---	6	Sigmoid	0.029	96.12	6.55

CUDA destekli Radyal tabanlı fonksiyon ağı ile yapılan eğitimde gizli katmanda sırasıyla 30-150 arasında nöron kullanılmıştır. Bant genişliği [0,2] aralığında kullanılmıştır. Aktivasyon fonksiyonu olarak en iyi sonucu veren gauss fonksiyonu kullanılmıştır. Eğitim işlemi 1000 adımda tamamlanmıştır. Eğitim ve test aşaması minimum 0,09 maksimum 0,31 saniye sürmüştür. Yapılan eğitimler sonucu elde edilen hata değeri ve başarı yüzdesi Çizelge 7.5’de gösterildiği gibidir.

Çizelge 7.5. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDARBF).

	Giriş Katmanı Nöron	Ara Katman Aktivasyon Fonksiyonu	Ara Katman Nöron Sayısı	Çıkış Katmanı Nöron	Hata	Başarı Oranı (%)	Eğitim ve Test Süresi
CUDA Destekli Radyal Tabanlı Fonksiyon	6	Gauss	30	6	0.093	90.659	0.095
	6	Gauss	40	6	0.087	91.208	0.112
	6	Gauss	60	6	0.076	92.307	0.139
	6	Gauss	80	6	0.065	93.406	0.167
	6	Gauss	100	6	0.061	93.956	0.184
	6	Gauss	110	6	0.054	94.505	0.198
	6	Gauss	120	6	0.049	95.055	0.214
	6	Gauss	130	6	0.038	96.154	0.238
	6	Gauss	140	6	0.043	95.604	0.265
	6	Gauss	150	6	0.054	94.505	0.312

CUDA destekli Destek vektör makineleri ile yapılan eğitimde veri setinin çıkışı 6 sınıflı olduğu için 15 DVM eğitilmiştir. Her bir eğitim aşamasında sadece iki farklı sınıf verisi alınmıştır. Çekirdek fonksiyonu lineer seçildiğinde en yüksek başarı elde edilmiştir. Eğitim ve test aşamaları birlikte minimum 0,13 maksimum 0,18 saniye sürmüştür. Yapılan eğitimler sonucu elde edilen başarı yüzdesi ve parametreler Çizelge 7.6’da gösterildiği gibidir.

Çizelge 7.6. Değiştirilen parametrelere göre hata değeri ve başarı yüzdesi (CUDADVM).

	Giriş Katmanı Değişken Sayısı	Çekirdek Fonksiyonu	Çoklu DVM Çözümü	Başarı Oranı (%)	Eğitim ve Test Süresi
CUDA Destekli Destek Vektör Makineleri	6	Polinom	Bire Karşı Bir	96.724	0.136
	6	Sigmoid	Bire Karşı Bir	64.773	0.157
	6	Radyal Tabanlı	Bire Karşı Bir	93.527	0.142
	6	Lineer	Bire Karşı Bir	97.831	0.183



Doğruluk oranlarına bakıldığında çok katmanlı yapay sinir ağları modeli %98,90'lık bir değerle en iyi sonucu vermiştir. Çok katmanlı yapay sinir ağlarının ardından %97,727'lük bir sonuçla destek vektör makineleri modeli gelmektedir. %95,993'lük bir sonuçla radyal tabanlı fonksiyon yapay sinir ağı modeli son sırada yer almaktadır. Fakat eğitim ve test aşamalarının süresine bakıldığında destek vektör makineleri radyal tabanlı fonksiyon ağları modeline göre daha iyi bir performans göstermiştir. Süre olarak çok katmanlı yapay sinir ağı modeli son sırada yer almıştır. Öte yandan CUDA destekli yapay sinir ağı modelleri ile diğer yapay sinir ağı modelleri karşılaştırıldığında doğruluk oranlarının neredeyse aynı olduğu görülmüş fakat CUDA destekli modellerin diğer yapay sinir ağı modelinden yaklaşık 10 kat daha hızlı işlem yaptığı gözlemlenmiştir (Çizelge 7.7).

Çizelge 7.7. Yöntemlerin doğruluk oranı ve zamana göre karşılaştırılması.

YÖNTEM	DOĞRULUK ORANI	SÜRE
MLP	%98.90	19.23 sn.
RBF	%95.99	10.59 sn.
DVM	%97.72	1.40 sn.
MLP (MATLAB)	%98.38	7 sn.
CUDAMLP	%98.40	2.39 sn.
CUDARBF	%96.15	0.23 sn.
CUDADVM	%97.83	0.18 sn.

## KAYNAKLAR

1. Nottingham, J., "Practical observations on conical cornea", *Churchill*, London, 1-19 (1984).
2. Auffarth, G. U., Wang, L. and Völcker, H. E., "Keratoconus evaluation using the Orbscan topography system", *J Cataract Refract Surg.*, 26: 222-8 (2000).
3. Prisant, O., Legais, J. M. and Renard, G., "Superior keratoconus", *Cornea*, 16: 693-4.(1997).
4. Krachmer, J. H., Feder, R. S. and Belin, M.W., "Keratoconus and related noninflammatory corneal thinning disorders", *Surv Ophthalmol.*, 28: 293-322 (1984).
5. Rahmen, W. and Anwar, S., "An unusual case of keratoconus", *J Pediatr Ophthalmol Strabismus.*, 43: 373-5 (2006).
6. Rabinowitz, Y. S., "Keratoconus", *Surv Ophthalmol.*, 42: 297-319 (1998).
7. Rabinowitz, Y.S, Yang, H., Rasheed, K. and Li, X., "Longitudinal analysis of the fellow eyes in unilateral keratoconus", *Invest Ophthalmol Vis Sci.*, 44: 1311 (2003).
8. Wilson, S. E. and Klyce, S.D., "Screening for corneal topographic abnormalities before refractive surgery", *Ophthalmology*, 101: 147-152 (1994).
9. Sharma, N., Singhvi, A., Sinha, R. and Vajpayee, R. B., "Reasons for not performing LASIK in refractive surgery candidates", *J Refract Surg.*, 21 (5): 496-8 (2005).
10. Wang, J. C., Hufnagel, T. J. and Buxton, D. F., "Bilateral keratectasia after unilateral laser in situ keratomileusis: a retrospective diagnosis of ectatic corneal disorder", *J Cataract Refract Surg.*, 29: 2015-2018 (2003).
11. Raiskup-Wolf, F., Hoyer, A., Spoerl, E. and Pillunat, L. E., "Collagen cross-linking with riboflavin and ultra violet-A light in keratoconus: Long term results", *J Cataract Refract Surg.*, 34: 796-801 (2008).
12. Rabinowitz, Y. S., Garbus, J. and McDonnell, P. J., "Computer assisted corneal topography in family members of patients with keratoconus", *Arch Ophthalmol.*, 108: 365-371 (1990).

13. Rabinowitz, Y.S. and McDonnell, P. J., "Computer assisted corneal topography in keratoconus", *Refract Corneal Surg.*, 5: 400-408 (1989).
14. Maeda, N., Klyce, S. D., Smolek, M. K. and Thompson H. W., "Automated keratoconus screening with corneal topography analysis", *Invest Ophthalmol Vis Sci.*, 35: 2749-2757 (1994).
15. Li, X., Yang, H. and Rabinowitz, Y. S., "Keratoconus: Classification scheme based on videokeratography and clinical signs", *J Cataract Refract Surg.*, 35(9): 1597-1603 (2009).
16. Smolek, M. K. and Klyce, S. D., "Current keratoconus detection methods compared with a neural network approach", *Invest Ophthalmol Vis Sci.*, 38: 2290-2299 (1997).
17. Rabinowitz, Y. S. and Rasheed, K., "KISA% index: a quantitative videokeratography algorithm embodying minimal topographic criteria for diagnosing keratoconus", *J Cataract Refract Surg.*, 25: 1327-1335 (1999).
18. Maeda, N., Fujikado, T., Kuroda, T., Mihashi, T., Hirohara, Y., Nishida, K. and Tano, Y., "Wavefront aberrations measured with Hartmann-Shack sensor in patients with keratoconus", *Ophthalmology*, 109(11): 1996-2003 (2002).
19. Alio, J. L. and Shabayek, M. H., "Corneal higher order aberrations: a method to grade keratoconus", *J Refract Surg.*, 22: 539-545 (2006).
20. Lagana, M. A., Cox, I. G. and Potvin, R. J., "The effect of keratoconus on the wavefront aberration of the human eye", *Invest Ophthalmol Vis Sci.*, 41(4): 3610 (2000).
21. Barbero, S., Marcos, S., Merayo-Loves, J. and Moreno-Barriuso, E., "Validation of the estimation of corneal aberrations from videokeratography in keratoconus", *J Refract Surg.*, 18: 263-270 (2002).
22. Gobbe, M. and Guillon, M., "Corneal wavefront aberration measurement to detect keratoconus patients", *Contact Lens Anterior Eye*, 28: 57-66 (2005).
23. Schwiegerling, J. and Greivenkamp, J. E., "Keratoconus detection based on videokeratoscopic height data", *Optom Vis Sci.*, 73: 721-728 (1996).
24. Twa, M. D., Parthasarathy, S., Roberts, C., Mahmoud, A. M., Raasch, T. W. and Bullimore, M. A., "Automated decision tree classification of corneal shape", *Optom Vis Sci.*, 82(12): 1038-1046 (2005).
25. Mahmoud, A. M., Roberts, C. J., Lembach, R. G., Twa, M. D., Herderick, E.E. and McMahan, T. T., "CLMI: The cone location and magnitude index", *Cornea*, 27: 480-487 (2008).

26. Jafri, B., Li, X., Yang, H. and Rabinowitz, Y. S., "Higher order wavefront aberrations and topography in early and suspected keratoconus", *J Refract Surg.*, 23: 774-781 (2007).
27. Cullen, J. F. and Butler, H. G., "Mongolism(Down Syndrome) and keratoconus", *Br J Ophthalmol.*, 47: 321-330 (1963).
28. Grünauer-Kloevekorn, C. and Duncker, G. I., "Keratoconus: epidemiology, risk factors and diagnosis", *Klin Monbl Augenheilkd.*, 223(6): 493-502 (2006).
29. Copeman, P. W. M., "Eczema and keratoconus", *BMJ.*, 2: 977-979 (1965).
30. Davies, P. D., Lobascher, D., Menon, J. A., Rahi, A. H. and Ruben, M., "Immunological studies in keratoconus", *Trans Ophthalmol Soc.*, 96: 173-178 (1976).
31. Bawazeer, A. M., Hodge, W. G. and Lorimer, B., "Atopy and keratoconus: a multivariate analysis", *Br J Ophthalmol.*, 84: 834-840 (2000).
32. Gritz, D. C., and McDonnell, P. J., "Keratoconus and ocular massage". *Am J Ophthalmol.*, 106: 757-765 (1988).
33. Gölemez, E., "Normal bireylerde ve keratokonus olgularında Pentacam HR ve Orbscan II ölçümlerinin karşılaştırılması", Tıpta Uzmanlık Tezi, *Eskişehir Osmangazi Üniversitesi Tıp Fakültesi*, Eskişehir, (2011).
34. Uçşul E., "Keratokonüs hastalarında oküler aberasyonların wavefront analizör ile değerlendirilmesi", Tıpta Uzmanlık Tezi, *Dokuz Eylül Üniversitesi, Tıp Fakültesi*, İzmir, (2011).
35. Çuçen B., "Keratokonüs tedavisinde kullanılan Crosslinking cihazının deneysel olarak üretilmesi, güvenilirliğinin ve etkinliğinin değerlendirilmesi", Tıpta Uzmanlık Tezi, *Kocaeli Üniversitesi, Tıp Fakültesi*, Kocaeli, (2011).
36. Arbelaez, M. C., Versaci, F., Vestri, G., Barboni, P. and Savini, G., "Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data", *Ophthalmology*, 119: 2231-2238 (2012).
37. Mesplie, S. L., Mortemosque, B., Touboul, D., Malet, F., Praud, D., Mesplie N. and Colin J., "Scalability and severity of keratoconus in children", *American Journal of Ophthalmology*, 154: 56-62 (2012).
38. Uçakhan, Ö. Ö., Çetinkor, V., Özkan, M. and Kanpolat, A., "Evaluation of Scheimpflug imaging parameters in subclinical keratoconus, keratoconus and normal eyes", *Journal of Cataract & Refractive Surgery*, 37: 1116-1124 (2011).

39. Timuçin, Ö. B., Karadağ, M. F., Cinal, A., Asker, M., Asker, S., and Timuçin D., “Assessment of corneal endothelial cell density in patients with keratoconus not using contact lenses”, *Contact Lens and Anterior Eye*, 36(2): 80-85 (2013).
40. Vega-Estrada A., Alio, J. L., Brenner, L. F., Javaloy, J., Puche, A. B. P., Barraquer, R. I., Teus, M. A., Murta, J., Henriques, J. and Uceda-Montanes, A., “Outcome analysis of intracorneal ring segments for the treatment of keratoconus based on visual, refractive, and aberrometric impairment”, *American Journal of Ophthalmology*, 155(3): 575-584 (2013).
41. Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L., “Graphics processing unit (GPU) programming strategies and trends in GPU computing”, *Journal of Parallel and Distributed Computing*, 73(1): 4–13 (2013).
42. Li, Q., Salman, R., Test, E., Strack, R. and Kecman, V., “Parallel multitask cross validation for Support Vector Machine using GPU”, *Journal of Parallel and Distributed Computing*, 73(3): 293–302 (2013).
43. Bressan, S., Cuzzocrea, A., Karras, P., Lu, X. and Nobari, S. H., “An effective and efficient parallel approach for random graph generation over GPUs”, *Journal of Parallel and Distributed Computing*, 73(3): 303–316 (2013).
44. Upadhyaya, S. R., “Parallel approaches to machine learning- A comprehensive survey”, *Journal of Parallel and Distributed Computing*, 73(3): 284–292 (2013).
45. Kamiya, K., Ishii, R., Shimizu, K., Igarashi, A., “Evaluation of corneal elevation, pachymetry and keratometry in keratoconic eyes with respect to the stage of Amsler-Krumeich classification”, *Br J Ophthalmol*, 98(4): 459–463 (2014).
46. Kanellopoulos, J. A., and Asimellis G., “OCT corneal epithelial topographic asymmetry as a sensitive diagnostic tool for early and advancing keratoconus”, *Clinical Ophthalmology*, 8: 2277–2287 (2014).
47. Oruçoğlu, F., and Toker, E., “Comparative analysis of anterior segment parameters in normal and keratoconus eyes generated by Scheimpflug tomography”, *Journal of Ophthalmology*, DOI: 10.1155/2015/925414 (2015).
48. Belin, M. W., Duncan, J., Ambrosio, Jr. R., and Gomes, J. AP., “A new tomographic method of staging/classifying keratoconus: The ABCD grading system”, *Int J Kerat Ect Cor Dis*, 4(3): 85–93 (2015).

49. Gomes, J. A., Tan, D., Rapuano, C. J., Belin, M. W., Ambrósio, Jr R, Guell, J. L., et al. “Global consensus on keratoconus and ectatic diseases”, *Cornea*, 34(4): 359–369 (2015).
50. Toprak, I., Yaylali, V., and Yildirim, C., “A combination of topographic and pachymetric parameters in keratoconus diagnosis”, *Cont Lens Anterior Eye*, 38(5): 357–362 (2015).
51. Du, X. L., Chen, M., and Xie L.X., “Correlation of basic indicators with stages of keratoconus assessed by Pentacam tomography”, *Int J Ophthalmol*, 8(6): 1136–1140 (2015).
52. Cavas-Martinez, F., De la Cruz Sánchez, E., Nieto Martinez, J., and Fernández Cañavate, F. J., “Corneal topography in keratoconus: state of the art”, *Eye and Vision*, 3(5) DOI 10.1186/s40662-016-0036-8 (2016).
53. Hung, S., Yen, D., C. And Wang, H., “Applying data mining to Telecom Churn Management”, *Expert Systems with Applications*, 31:515-524 (2006).
54. Babadağ, K., “Zeki veri madenciliği: Ham veriden altın bilgiye ulaşma yöntemleri”, *Industrial Application Software*, 85-87 (2006).
55. Jacobs, P., ”Data mining: What general managers need to know”, *Harvard Management Update*, 4 (10): 8 (1999).
56. Alatas, B. and Akın, E., “Veri madenciliğinde yeni yaklaşımlar”, *Ya/Em-2004-Yöneylem Araştırması/Endüstri Mühendisliği XXIV. Ulusal Kongresi*, Gaziantep-Adana, (2004).
57. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, R. and Uthurusamy, R., “Advances in knowledge discovery and data mining”, *Aaa/Mit Pres*, 17(3): 37-54 (1996).
58. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. and Zanasi, A., “Discovering data mining: From concept to implementation”, *Prentice Hall*, Upper Saddle River, 517 (1998).
59. Lee, S. J. and Siau, K., “A review of data mining techniques”, *Industrial Management & Data Systems*, 101(1): 41-46 (2001).
60. Han, J. and Kamber, M., “Data mining: Concepts and techniques”, *Morgan Kaufmann Publishers*, San Francisco, (2006).

61. Dasu, T. and Johnson, T., “Exploratory data mining and data cleaning”, **Wiley**, NY USA, (2003).
62. Maimon, O. and Rokach, L., “Data mining and knowledge discovery handbook”, **Springer**, London, (2007).
63. Sumathi,S. and Sivanandam, S.N., “Introduction to data mining and its applications”, **Wiley**, NY USA, (2006).
64. Berti, L., “Measuring and modelling data quality for quality-awareness in data mining”, (in: “Quality Measures in Data Mining” Ed: Hamilton, H.J), **Springer**, London, (2007).
65. Stepaniuk, J., “Rough - Granular Computing in knowledge discovery and data mining”, **Springer**, London, (2008).
66. Atılgan, E., “Karayollarında meydana gelen trafik kazalarının karar ağaçları ve birliktelik analizi ile incelenmesi”, Yüksek Lisans Tezi, **Hacettepe Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, (2011).
67. Özkan, Y., “Veri madenciliği yöntemleri”, **Papatya Yayıncılık**, İstanbul, (2008).
68. Koç, S., “İllerin sosyo-ekonomik özelliklere göre sınıflandırılması” **Çukurova Üniversitesi 5.Ulusal Ekonometri ve İstatistik Sempozyumu**, Çukurova Üniversitesi, Adana. (2001).
69. Mc Grew, J. C. and Monroe, C. B., “Statistical problem solving in geography”, **Ww C. Brown Publishers**, Oxford-England, (1993).
70. Hair, J. F., Anderson, R. E., Tatham R.L. and Black, W. C., “Multivariate data analysis”, **Prentice Hall**, New Jersey, 95-99 (1998).
71. Kalaycı, Ş., “SPSS uygulamalı çok değişkenli istatistik teknikleri”, 2. Baskı, **Asil Yayın Dağıtım**, Ankara, 321-322 (2006).
72. Albayrak, A. S., “Türkiye’de illerin sosyoekonomik gelişmişlik düzeylerinin çok değişkenli, istatistik yöntemlerle incelenmesi”, **PROMAT Basım Yayın**, İstanbul, 86-109 (2005).
73. Büyüköztürk, Ş. “Sosyal bilimler için veri analizi el kitabı”, 3. Baskı, **Pegem Yayıncılık**, Ankara, 119 (2003).
74. Çakmak, Z., “Kümeleme analizinde geçerlilik problemi ve kümeleme sonuçlarının değerlendirilmesi”, **Dumlupınar Üniversitesi Sosyal Bilimler Dergisi**, 1(3): 187-205 (1999).
75. Tatlıdil, H., “Uygulamalı çok değişkenli istatistik”, **H.Ü. Fen Fak. İstatistik Bölümü**, 329-343 (1992).

76. Internet: Hartigan, J.A. ‘Clustering algorithms’, Wiley New York, <http://statsoft.com/textbook/esc.html> (1975).
77. Atbaş, Günay, A. C., “Kümeleme analizinde küme sayısının belirlenmesi üzerine bir çalışma”, Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, (2008).
78. Vatansever, M., “Görsel veri madenciliği tekniklerinin kümeleme analizlerinde kullanımı ve uygulanması”, Yayınlanmamış Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimler Enstitüsü*, İstanbul, (2008).
79. Martinez, W.L. and Martinez A. R., “Computational statistics handbook with MATLAB”, *Boca Raton : CRC Press*, USA, (2002).
80. Orhunbilge, N., “Örnekleme yöntemleri ve hipotez testleri”, *Avciol Basım Yayın*, İstanbul, (2000).
81. Johnson, A.R. and Wichern, D.W., “Applied multivariate statistical analysis”, *Prentice-Hall International Editions*, New Jersey, 554 (1988).
82. Anderberg, M. R., “Cluster analysis for applications”, *Academic Press*, New York, 553–555 (1973).
83. Sharma, S., “Applied multivariate techniques”, *John Wiley and Sons Inc*, New York, (1996).
84. Özdamar, K. “Paket programlar ile istatistiksel veri analizi 2”, *Kaan Kitabevi*, Eskişehir, (2004).
85. Turanlı, M., Özden H. Ü., and Türedi, S., “Avrupa Birliği’ne aday ve üye ülkelerin ekonomik benzerliklerinin kümeleme analizi ile incelenmesi” *İstanbul Ticaret Üniversitesi Sosyal Bilimler Dergisi*, 5(9): 99 (2006).
86. Tan, P., Steinbach, M. and Kumar, V., “Introduction to data mining”, *Addison Wesley*, USA, (2006).
87. Bilen, Ö., “ÖSS sınav sonuçlarının okul bazında veri madenciliği ile incelenmesi”, Yayınlanmamış Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul. (2004),
88. Green, E.P., “Analysing multivariate data”, *Philadelphia*, 427 (1989).
89. Hubert, L., “Approximate evaluation techniques for the single-link and complete-link hierarcihal clustering procedures”, *Journal of the American Statistical Association*, 69: 698-704 (1974).



90. Chatfield, C. and Collins, A. J., "Introduction to multivariate analysis", *Chapman and Hall*, (1980).
91. Everitt, B., "Cluster analysis". *Heinmann*, London, 122 (1974).
92. Dibb, S., "Market segmentation: Strategies for success", *Marketing Intelligence&Planning*, 16(7): 394–406. (1998).
93. Blashfield, R. K. and Aldenderfer, M.S., "The literature on cluster analysis", *Multivariate Behavioral Research*, 13: 271-295 (1978).
94. Han, J. and Kamber, M., "Data mining concepts and techniques", *Morgan Kaufmann Publishers*, San Francisco, (2001).
95. Bolshakova, N. and Francisco A. "Machao CVE cluster validation for gene expression data", *Bioinformatics*, 19(18): 2494-2495 (2003).
96. Rousseeuw, P. J., "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*, 20: 53-65 (1987).
97. Calinski, R.B. and Harabasz, J., "A dendrite method for cluster analysis", *Communications in Statistics*, 3:1-27 (1974).
98. Krzanowski, W.J. and Lai, Y.T., "A criterion for determining the number of groups in a data set using sum of squares clustering", *Biometrics*, 44: 23-44 (1985).
99. Haykin S., "Neural networks: A comprehensive foundation", *Prentice Hall*, USA, (1994).
100. Giudici P., "Applied data mining: Statistical methods for business and industry", *John Wiley & Sons*, England, (2003).
101. Öztemel, E., "Yapay sinir ağları", *Papatya Yayınları*, İstanbul, 232 (2006).
102. Internet: Stergiou, C., Siganos, D., "Neural networks", [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html#Feed-forward%20networks](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Feed-forward%20networks) (1996).
103. Güven, A., "Göze ait elektrofizyolojik sinyaller kullanılarak yapay sinir ağları destekli bazı göz hastalıklarının teşhisi", Doktora Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, 92 (2005).
104. Internet: Kakıcı, A., "Yapay sinir ağlarına giriş", <http://www.ahmetkakici.com/yazilim/yapay-sinir-aglarina-giris/> (2009).

105. Kuzucuoğlu, A.E., “İklımlendirme deney tesisatının Neuro-Fuzzy yöntemiyle kontrolü”, Doktora Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul. 169 (2000).
106. Curram, S.P. and Minger, J., “Neural networks, decision tree induction and discriminant analysis: An empirical comparison”, *Journal of the Operational Research Society*, 45(4): 440-450 (1994).
107. Şenol, C., “Standart ve hibrid yapılar kullanarak yapay sinir ağları ile imza tanıma”, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, (2004).
108. Montazer, G. A., Sabzevari R. and Khatir H.G., “Improvement of learning algorithms for RBF neural networks in a helicopter sound identification system”, *In Proceedings of Neurocomputing*, 167-173 (2007).
109. Jiang, Z.G., Fu, H.G. and Li, L.J., “Support vector machine for mechanical faults classification”, *Journal of Zhejiang University Science*, 6(5): 433-439 (2005).
110. Smola, A.J. and Schölkopf, B., “A tutorial on support vector regression”, *Technical Report NC2-TR-1998-030*, 200-222 (1998).
111. Burges, C.J.C., “A tutorial on support vector machines for pattern recognition”, *Data Mining and Knowledge Discovery*, 2: 121-167 (1998).
112. Dash, P.K., Samantary, S.R. and Ganapati, P., “Fault classification and section identification of an advanced series-compensated transmission line using support vector machine”, *IEEE Transactions on Power Delivery*, 22: 67-73 (2007).
113. Marwala, T., Mahola, U. and Chakraverty, S., “Fault classification in cylinders using multi-layer perceptrons, support vector machines and gaussian mixture models”, *The Intl. Symposium on Neural Networks and Soft Computing*, 14(2): 307-316 (2007).
114. Pöyhönen, S., “Support vector machine based classification in condition monitoring of induction motors”, Doktora Tezi, *Helsinki University of Technology*, 63 (2004).
115. Vapnik, V., “The nature of statistical learning theory”, *New York: Springer Verlag*, New York, 187 (1995).
116. Zhang, J.H.R., “A new algorithm of improving fault location based on SVM”, *Eighth IEE International Conference on Developments in Power System Protection*, 1: 204-207 (2004).

117. Vojtech, F. and Hlavac, V., “Statistical pattern recognition toolbox for MATLAB (SPRTOOL)”, *User’s Guide*, (2007).
118. Jason, S. and Edward, K., “CUDA by example, an introduction to general-purpose GPU programming”, (2010).
119. Internet: M. Harris, D. Göddeke, General-purpose computation on graphics hardware, <http://gpgpu.org>. (2013).
120. Owens, J., Houston, M., Luebke, D., Green, S., Stone, J. and Phillips, J., “GPU computing”, *Proceedings of the IEEE*, 96(5): 879–899 (2008).
121. Brodtkorb, A.R., Dyken, C., Hagen, T.R., Hjelmervik, J.M. and Storaasli, O., “State-of-the-art in heterogeneous computing”, *Scientific Programming*, 18(1): 1–33 (2010).
122. Asanovic, K., Bodik, R., Catanzaro, B., Gebis, J., Husbands, P., Keutzer, K., Patterson, D., Plishker, W., Shalf, J., Williams, S. and Yelick, K., “The landscape of parallel computing research: a view from Berkeley”, *Technical report, EECS Department*, University of California, Berkeley, (2006).
123. Taylor, G., “Energy efficient circuit design and the future of power delivery”, Conference presentation, *Electrical Performance of Electronic Packaging and Systems*, (2009).
124. Levenberg K. “A method for the solution of certain nonlinear problems in least squares”, *Quart. Appl. Math.*, 2: 164-168 (1944).
125. Marquardt D. W. “An algorithm for least-squares estimation of nonlinear parameters”, *J. Soc. Ind. Appl. Math.*, 11: 431-441 (1963).
126. Sağıroğlu, S., Beşdok, E., and Erler M., Control chart pattern recognition using artificial neural network. *Turkish Journal of Electrical Engineering*, 8(2): 137-147 (2000).
127. Mackay, D. J. C., “Bayesian interpolation,” *Neural Computation*, 4: 415-447 (1992).
128. Uçar, M., Çakmak, H.B., and Şen, B., “A statistical approach to classification of keratoconus”, *International Journal of Ophthalmology*, 9(9) (2016).
129. Uçar, M., and Şen, B., “A novel classification approach for detecting keratoconus disease with intelligent systems”, *8<sup>th</sup> International Conference on Electrical and Electronics Engineering, ELECO 2013*, Bursa, (2013).

## ÖZGEÇMİŞ

Murat UÇAR 1982 yılında İzmir’de doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı. Halide Gencer Anadolu Ticaret Meslek Lisesi Bilgi İşlem Bölümü’nden mezun olduktan sonra 2000 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Bölümü’nde öğrenime başlayıp 2004 yılında iyi derece ile mezun oldu. 2004 yılında Kastamonu ili Pınarbaşı ilçesi Şehit Ayhan Sütçü Pansiyonlu İlköğretim Okulu’nda öğretmen olarak göreve başladı. 2005 yılında Milli Eğitim Bakanlığı Eğitim Teknolojileri Genel Müdürlüğü’nde yazılım geliştirme uzmanı olarak başlamış olduğu göreve MEB Bilgi İşlem Dairesi Başkanlığı’nda devam etmektedir. 2007 yılında ZKÜ Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda başlamış olduğu yüksek lisans programını, 2009 yılında tamamladı.

### **ADRES BİLGİLERİ**

Adres : Milli Eğitim Bakanlığı  
Bilgi İşlem Dairesi Başkanlığı  
Çankaya / ANKARA

Tel : (505) 751 9909

E-posta : [mucar@meb.gov.tr](mailto:mucar@meb.gov.tr)