

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**KALP RİTİM BOZUKLUĞU OLAN HASTALARIN
TEDAVİ SÜREÇLERİNİ DESTEKLEMELİK AMAÇLI
MAKİNE ÖĞRENMESİNE DAYALI
BİR SİSTEMİN GELİŞTİRİLMESİ**

Fatih AYDIN
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. Erdem UÇAR
EDİRNE 2011

T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**KALP RİTİM BOZUKLUĞU OLAN HASTALARIN TEDAVİ SÜREÇLERİNİ
DESTEKLEMELİK AMAÇLI MAKİNE ÖĞRENMESİNE DAYALI BİR
SİSTEMİN GELİŞTİRİLMESİ**

Fatih AYDIN

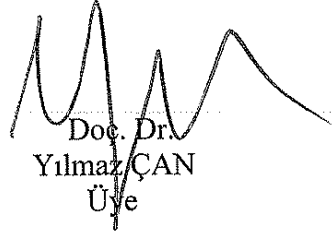
Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Bu tez 22.07.2011 Tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.



Yrd. Doç. Dr.
Erdem UÇAR
Jüri Başkanı (Danışman)



Doç. Dr.
Yılmaz ÇAN
Üye



Yrd. Doç. Dr.
Deniz TAŞKIN
Üye

ÖZET

Yüksek Lisans Tezi, Kalp Ritim Bozukluğu Olan Hastaların Tedavi Süreçlerini Desteklemek Amaçlı Makine Öğrenmesine Dayalı Bir Sistemin Geliştirilmesi, T.C. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu tezde bilgisayar bilimlerinin önemli bir alanı olan yapay zekâ ve bir alt alanı olan makine öğrenmesi teknikleri kullanılarak bir tıp bilişimi uygulaması geliştirilmiştir. Bu uygulamayla aritmi hastalarının tedavi süreçlerine yardımcı olmak amacıyla öğrenmeye dayalı bir sistem gerçekleştirilmiştir. Geliştirilen sistem lokasyondan bağımsız olarak hastanın bazı metabolik parametrelerini alarak bu verileri uzak bir sunucu üzerindeki veritabanında depolamaktadır. Depolanan bu veriler bir uzman tarafından etiketlendikten sonra veri ambarına aktarılmaktadır. Veri ambarındaki bu veriler daha sonra makine öğrenmesi sınıflandırıcısı tarafından eğitim verileri olarak kullanılmaktadır.

Geliştirilen Sistemde, makine öğrenmesi algoritması olarak k-en yakın komşuluk (kNN) algoritması kullanılmıştır. kNN algoritmasının seçilme sebebi düşük bias'a sahip nonlineer bir fonksiyon olmasındandır. Bu nedenle yapılan tahminlerde yüksek oranda doğruluk sağlanmaktadır.

Bu çalışmada bir insan uzman ve makine öğrenmesi yöntemi birlikte kullanılarak indaktif uzman sistem tasarımına gidilmiştir. Böyle bir tasarıma gidilmesindeki temel neden bir insan uzmanın geçmişten gelen tecrübelerinin herhangi bir yöntemle kazanılamamasıdır. Literatürde insan uzmanlığını içinde barındıran bu tür yöntemler en çok önerilen yöntemlerden biridir.

Bu tez 2011 yılında yapılmıştır ve 100 sayfadan oluşmaktadır.

Anahtar Kelimeler: Makine Öğrenmesi, Tembel Öğrenme, Tıp Bilişimi, Tıpta Yapay Zekâ, Aritmi.

ABSTRACT

Master of Science Thesis, Developing a Machine Learning Based System to Assist Treatment Processes of Arrhythmia Patients, T.C. Trakya University, Graduate School Of Natural And Applied Sciences, Department Of Computer Engineering.

In this study, a medical informatics application is developed based on machine learning techniques, itself a subfield of artificial intelligence which has great significance in computer sciences. Using this application, we developed a learning system to aid arrhythmia patients during their treatment processes. Operating independent of location, this system receives some of the metabolic parameters of the patient and stores them on a remote server database. These data are, then, transferred to a data warehouse, upon being inspected and labeled by an expert. The data in the warehouse, in turn, is used by a machine learning classifier as training data.

As for machine learning algorithm, the system developed makes use of the k-nearest neighbor (kNN) algorithm, for it is a nonlinear function with a low bias, thus ensuring high accuracy in prediction.

In this study we created an inductive experts system using an expert and a machine learning method together, the reason being the lack of a method whereby an expert's past experience can be gained. In the literature, methods that include human expertise are among the ones most recommended.

This thesis is done in 2011 and consists of 100 pages.

Key words: Machine Learning, Lazy Learning, Medical Informatics, Artificial Intelligence in Medicine, Arrhythmia.

ÖNSÖZ

“Yapay Zekâ” kavramı modern bilgisayar bilimi kadar eskidir. Alan Mathison Turing’in sofistike bir soru olan “Makineler Düşünebilir mi?” sorusunu ortaya atmasıyla “Yapay Zekâ” kavramı tartışmaya açılmıştır. Bu anlamda “Yapay Zekâ” insan düşüncesinin formelleştirilmesi esasına dayanmaktadır.

Yapay Zekânın bilimsel bir araştırma alanı olmasıyla birlikte çeşitli alanlarda uygulanabilirliği görülmüştür. Bu alanlardan biri de tıp’tır. 1970’lerle birlikte tıp alanında ilk Yapay Zekâ çalışmaları başlamıştır. Bu çalışmalar biyomedikal problemlerle boğuşan tıp araştırmacılarını büyülemektedir.

Tıp alanında, kalp hastalıkları üzerine Yapay Zekâ çalışmaları çok yoğun bir biçimde yapılmaktadır. Bu çalışmaların yapıldığı kalp hastalıklarından biri de Aritmi’dir. Aritmi hastalarının tedavi süreci boyunca lokasyondan bağımsız olarak kontrol edilebilmesi ve metabolik parametrelerin non-invazif yöntemlerle okunup değerlendirilmesi çok önemlidir. Bu anlamda Yapay Zekânın yanı sıra birçok teknolojinin de birlikte kullanılması gerekmektedir. Bu tez çalışması her ne kadar aritmi hastalarının tedavi süreçlerine yardımcı bir sistemin geliştirilmesi üzerine olsada tezin diğer bölümleri Tıpta Yapay Zekâ ve Makine Öğrenmesi konusunda önemli bilgiler vermektedir.

TEŞEKKÜR

Bu tez çalışmasının her aşamasında desteğini esirgemeyen ve beni bu konuya yönlendiren saygıdeğer hocam ve danışmanım Yrd. Doç. Dr. Sayın Erdem UÇAR'a tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

Yapay zekâ dersleri ile ufukumuzu açan ve böyle bir tez hazırlamama katkıda bulunan hocamız Doç. Dr. Sayın Yılmaz KILIÇASLAN'a teşekkürü bir borç bilirim.

Bu konuda ortak çalıştığım değerli arkadaşım Öğretim Görevlisi Adnan Fatih KOCAMAZ'a tüm tez süresince verdiği desteklerden dolayı çok teşekkür ederim.

Trakya Üniversitesi Bilgisayar Mühendisliği Bölümü Araştırma Görevlisi Edip Serdar GÜNER'e WEKA yazılımının ancak tecrübe ile kazanılabilecek bilgilerini benimle paylaştığı için teşekkür ederim.

İÇİNDEKİLER

ÖZET	iii
ABSTRACT.....	iv
ÖNSÖZ	v
TEŞEKKÜR	vi
İÇİNDEKİLER.....	vii
SİMGELER.....	x
ŞEKİLLER TABLOSU.....	xii
ÇİZELGELER TABLOSU	xiii
1. GİRİŞ	1
1.1. Tıp Bilişimi ve Bilgisayar Mühendisliği İlişkisi.....	3
2. DOLAŞIM SİSTEMİ.....	5
2.1. Kalbin Yapısı ve İşleyişi	5
2.1.1. Kalbin odacıkları.....	6
2.1.2. Kalbin kapakları	7
2.1.3. Kalbin çalışması	8
2.2. Kalbin Uyarı ve İleti Sistemi.....	9
2.3. Kan Basıncı.....	11
2.4. Nabız.....	12
3. KALP RİTİM BOZUKLUĞU (ARİTMİ)	13
3.1. Kardiyak Nedenlere Bağlı Aritmiler	14
3.1.1. Sinüs düğümü ve atriyoventriküler düğüm sorunları	14
3.1.1.1. Sinüs bradikardisi	14
3.1.1.2. Sinüs taşikardisi	14
3.1.1.3. Hasta sinüs sendromu	15
3.1.1.4. Atriyoventriküler düğüm hastalıkları.....	15
3.1.2. Supraventriküler aritmiler.....	15
3.1.2.1. Atriyoventriküler nodal reentrant (AVNRT).....	16
3.1.2.2. Atriyoventriküler resprokal reentrant (AVRT).....	16
3.1.2.3. Atriyal taşikardiler	17
3.1.2.4. Atriyum fibrilasyonu	17
3.1.2.5. Atriyal flutter	17
3.1.3. Ventriküler aritmiler	18
3.1.4. Kaçış vuruları ve dal bloklarına bağlı aritmiler	19

3.1.4.1.	Kaçış vurularına bağlı aritmiler	19
3.1.4.2.	Dal bloklarına bağlı aritmiler	19
3.2.	Psikiyatrik Nedenlere Bağlı Aritmiler	19
3.3.	Fiziksel ve Duygusal Strese Bağlı Aritmiler	20
4.	TIPTA YAPAY ZEKÂ DÖNEMİ	21
4.1.	Yapay Zekânın Tarihi	21
4.2.	Tıp Alanında Yapay Zekânın Başlangıcı	23
4.3.	AIM'in Ana Noktaları	23
4.4.	AIM Metotları	24
4.5.	AIM Çalışmalarında Önemli Aşamalar	25
5.	MAKİNE ÖĞRENMESİ.....	29
5.1.	ML Nedir?	30
5.2.	Öğrenme Türleri	31
5.3.	Örneğe Dayalı Öğrenme.....	32
5.4.	kNN Sınıflandırıcı	33
5.5.	k-Kath Çapraz Doğrulama	36
5.6.	Özellik Seçimi	38
5.7.	İstatiksel Değerlendirme Ölçütleri	39
5.7.1.	Kappa istatistiği.....	39
5.7.2.	F-ölçütü	41
5.7.3.	Hata ölçütü: ortalama karesel hatanın karekökü	42
5.8.	Gürültü.....	43
5.9.	Bias-Varyans Ayrışımı.....	44
6.	WEKA.....	53
6.1.	Temel Kavramlar	54
6.1.1.	Verikümesi.....	54
6.1.2.	Sınıflandırıcı	55
6.2.	Explorer	58
6.3.	Experimenter	59
6.4.	KnowledgeFlow	60
6.5.	Simple CLI.....	60
7.	ARİTMİ HASTALARI İÇİN GELİŞTİRİLEN SİSTEM.....	62
7.1.	Teze Konu Olan Projenin Tanıtımı	62
7.2.	Öğrenmeye Dayalı Sistemi Oluşturan Bileşenler	63

7.3.	Veri Toplama.....	64
7.4.	Seçilen Nitelikler	66
8.	GELİŞTİRİLEN UYGULAMALAR	68
8.1.	Uzman Değerlendirmeleri için Geliştirilen Uygulama	68
8.2.	Veri Toplama ve Veri Sınıflandırma için Geliştirilen Uygulama	73
9.	GELİŞTİRİLEN SİSTEMİN DEĞERLENDİRİLMESİ.....	76
10.	SONUÇLAR VE TARTIŞMA	77
11.	KAYNAKLAR	81
EK-A	89
EK-B	95
ÖZGEÇMİŞ	100

SİMGELER

AF	Atriyal Fibrilasyon
AI	Yapay Zekâ
AIM	Tıpta Yapay Zekâ
AT	Atriyal Taşikardi
AV	Atriyoventriküler Düğüm
AVNRT	AV Nodal Reentran Taşikardi
AVRT	AV Resprokal Taşikardi
CV	Çapraz Doğrulama
DNA	Deoxyribonükleik Asit
EKG	Elektrokardiyografi
GPRS	Genel Paket Tabanlı Radyo Servisi
GSM	Mobil İletişim için Küresel Sistem
Hb	Hemoglobin
HbO ₂	Oksihemoglobin
HSS	Hasta Sinüs Sendromu
IBL	Örneğe Dayalı Öğrenme
IML	Tümevarımsal Makine Öğrenmesi
IP	İnternet Protokol
kNN	k-En Yakın Komşuluk Algoritması
KS	Kappa İstatistiği
LL	Tembel Öğrenme
MBL	Hafıza Tabanlı Öğrenme
ML	Makine Öğrenmesi
MSE	Ortalama Karesel Hata
OLAP	Online Analytical Processing
PAT	Paroksizmal Atriyal Taşikardi
RMSE	Ortalama Karesel Hatanın Karekökü
RNA	Ribonükleik Asit
SA	Sinoatriyal Düğüm

SND	Sinüs Nod Disfonksiyonu
SNP	Single Nucleotide Polymorphism
SPO ₂	Periferik Oksijen Saturasyonu
SVT	Supraventriküler Taşikardi
TCP	Transmission Control Protocol
UT	Fayda Teorisi
WWW	World Wide Web

ŞEKİLLER TABLOSU

Şekil 2-1. Kalbin konumu ve komşulukları	5
Şekil 2-2. Kalbin iç kesiti.....	6
Şekil 2-3. Ventrikül kasılmasında kalp kapakları	8
Şekil 2-4. Ventrikül gevşemesinde kalp kapakları.....	8
Şekil 2-5. Kalbin uyarı ve ileti sistemi.....	10
Şekil 2-6. Kalpte elektriksel uyarı akışı	11
Şekil 5-1. k değerinin sınıflandırıcının tahminine etkisi.....	35
Şekil 5-2. Bir verikümesinde gürültü	44
Şekil 5-3. g tahmin edicisinin bias ve varyansı.....	49
Şekil 5-4. kNN sınıflandırıcının k-değerine bağlı olarak MSE, bias ve varyans değişiminin grafiği	52
Şekil 6-1. WEKA grafik arayüzü	54
Şekil 6-2. Bir hastanın örnek ARFF dosyası.....	55
Şekil 6-3. Çalıştırılan komutun çıktısı	57
Şekil 6-4. Explorer penceresinden bir görünüm	58
Şekil 6-5. Experimenter ortamından bir görünüm	59
Şekil 6-6. KnowledgeFlow ortamından genel bir görünüş	60
Şekil 6-7. Simple CLI arayüzünden genel bir görünüm	61
Şekil 7-1. GPRS'in genel bir görünümü	64
Şekil 7-2. Geliştirilen AIM sisteminin genel bir görünümü	65
Şekil 8-1. Geliştirilen uygulamadan bir görünüm.....	69
Şekil 8-2. Hastanın veritabanında kayıtlı bir verisinden görünüm	70
Şekil 8-3. Hastanın tüm verilerine göre sınıflandırıcının değerlendirmesi.....	71
Şekil 8-4. Uzman kişinin verinin eğitim için yeterliliğine karar verme süreci	72
Şekil 8-5. Geliştirilen uygulamanın arayüzü.....	73
Şekil 8-6. Hastadan gelen veriye göre karar verme süreci.....	75

ÇİZELGELER TABLOSU

Çizelge 4-1. AIME-2011'deki konu ve temalar.....	25
Çizelge 4-2. AIME-2007'deki konu ve temalar.....	25
Çizelge 5-1. Landis ve Koch'un Kappa değerleri tablosu.	40
Çizelge 5-2. İki sınıflı bir tahmin için doğruluk tablosu.....	41

1. GİRİŞ

Günümüzde, insan yaşamının her alanında hız kazanan teknolojik değişimler, etkilerini özellikle sağlık sektörünü etkileyecek iki yönde; medikal ve bilişim teknolojileri alanlarında göstermektedirler. Bilişim teknolojilerinde yaşanan gelişmeler sağlık endüstrisini de ileri derecede etkilemiştir. Medikal ve bilişim teknolojilerinin bu noktada kesişmesi sonucu bu etkinin sınırları ve hızı artmaktadır (Saygılı vd., 2001).

Tıp bilişimi(Medical Informatics); bilgi bilimi(Information Science), bilgisayar bilimi(Computer Science) ve sağlık(Healthcare) konularının kesişimi olarak ifade edilebilir. Tıp bilişimi, sağlık ve biyotıp(Biomedical) konularında bilginin elde edilmesi, depolanması, geri alınması ve kullanılması işlemlerini yeniden en uygun hale getirmek için gerekli kaynaklar, cihazlar ve metotlarla ilgilenir. Tıp bilişimi araçları sadece bilgisayarlar değil aynı zamanda klinik prensipleri, formal tıp terminolojileri, bilgi ve iletişim sistemleridir. Bu bağlamda, Tıp Bilişimini tıp alanındaki bilgilerin üretimi, toplanması, değerlendirilmesi, analizi, saklanması, işlenmesi, sunulması ve arşivlenmesi süreçlerinin tamamı ile ilişkilendirmek mümkündür (Kocamaz, 2007).

Tıp Bilişimi (Medical Informatics) tıp alanındaki bilgilerin (data, information, knowledge) etkili ve etkin kullanımı, bu bilgilerin yaygınlaştırılması, analizi, yeni yapılanmalara imkan sağlayacak şekilde yönetilmesi için değişik bilim dalları ile etkileşimli bir şekilde günümüz bilgisayar ve iletişim teknolojisinin en üst düzeyde kullanılmasını amaçlamaktadır. Bilgi teknolojileri ve sağlık bilimlerinin kesiştiği noktada bulunan tıp bilişimi, şimdiye kadar çeşitli tıbbi araştırma ve geliştirme konularında olduğu gibi, kuramsal ve uygulamalı eğitimde de rol almıştır. Tıbbi Bilişim teknikleri, sağlık merkezlerinin veri toplama, işleme ve değerlendirme yetilerini üstel bir şekilde artırarak global ölçekte doğru, ayrıntılı ve güvenilir sonuçlara ulaşılmasını sağlamaktadır (Kocamaz, 2007).

Tıp bilişimi, temel bir araştırma disiplini oluşturmak üzere çeşitli teknik, teori ve yöntemlerin bir araya gelmesiyle oluşmuş bir bilim dalıdır. Bu bilim dalı tıp alanındaki gelişmelerin ortaya çıkardığı bilgi ve verilerin oluşturulması, biçimlendirilmesi, paylaşılması ve sonuçta hastaların bakım ve tedavilerinin belirlenmesi, seçilmesi ve geliştirilmesini hedef alır. Yani tıp bilişimi, hasta hakkında düşünme yöntemleri ve tedavilerin tanımlanma, seçilme ve geliştirilme yolları üzerine ussal bir çalışmadır (<http://www.cozum.info>).

Tıp bilişiminde bilimsel araştırma, *disiplinlerarası* bir nitelik taşıyor ve bilimsel yöntemleri kullanır. Bilimsel araştırmalarda sağlanan bilginin yardımıyla hasta verilerinin elde edilmesi, işlenmesi ve yorumlanması için yöntemler ve sistemler geliştirilir, değerlendirilir. Bütün bunları yaparken de bilgi ve iletişim teknolojilerinin nasıl uygulanacağını ve kullanılacağını araştırır. Bilgi teknolojilerinde gelişmiş yöntemlerin ortaya çıkmasıyla sağlanan ilerleme tıp bilişiminin bir bilim dalı olarak gelişmesinde itici güç olmuştur (<http://www.cozum.info>).

Sağlık verileri arasında karmaşık ve bulanık bir ilişki vardır. Bu nedenle de bilgisayarda işlenebilen bilgi (computational knowledge-bilimsel sorunları analiz etmek ve çözmek için bilgisayar kullanımıyla ilgili bilgi olarak tanımlanabilir) bu alanın olgularını açıklamak için gereklidir. Tıp bilişimi sağlık alanındaki bilgi ile bu bilgiyi kullanan kişiler arasındaki karmaşık ilişkileri anlamak için yeni araçlar ve yöntemler sağlamaktadır (<http://www.cozum.info>).

Tıp bilişimi, klinik karar destek sistemlerinin tasarımından, yazılım ve donanım araçlarının geliştirilmesine kadar geniş bir konseptte çalışma alanı içerir.

Tıp bilişimi birçok alanla işbirliği içindedir. EKG ve çeşitli elektronik tanı aletleri ile ilgili olarak biyomedikal mühendislikle, donanım üretimi açısından elektrik mühendisliğiyle, algoritmalar ve matematiksel yönleri ile bilgisayar mühendisliğiyle, yapay zeka yönüyle bilgisayar bilimi ve bilişsel bilimlerle sürekli iletişim halindedir (Kocamaz, 2007).

Bir bilim olarak tıp bilişimi teoriden pratiğe geniş bir alana hitap eder. Aynı anda hem temel, hem deneysel, hem de uygulamalı bir bilim olma özelliğini taşır. Biryandan model ve bilgi üretirken, diğer yandan bunları çözümler ve deneyler. Ayrıca pratik amaçlarla ortaya çeşitli ürün ve sonuçlar koyar (<http://www.cozum.info>).

Bu tez çalışmasında kalp ritim bozukluğu hastalarının tedavi süreçleri için geliştirilen öğrenmeye dayalı bir sistem anlatılacaktır. Bu amaçla ikinci bölümde dolaşım sistemi, kalbin yapısı ve damarlar hakkında temel bilgiler verilecektir. Üçüncü bölümde kalp ritim bozukluğu hastalığı (aritmi) ile ilgili bilgiler verilecektir. Daha sonra dördüncü bölümde tıpta yapay zekâ uygulamalarının bilimsel araştırma safhaları belirtilecektir. Beşinci bölümde makine öğrenmesi (ML), en yakın k-komşuluk algoritması (kNN) ve sistemin performansını etkileyen/ölçen kriterler açıklanacaktır. Altıncı bölümde WEKA ile ilgili bilgiler verilecektir. Yedinci bölümde sistemi oluşturan bileşenler ve sistemin genel yapısı anlatılacaktır. Sekizinci bölümde ise bu teze konu olan çalışma için geliştirilen yazılım uygulamaları tanıtılacaktır. Dokuzuncu bölümde geliştirilen sistemle ilgili genel bir değerlendirme yapılacaktır. Son bölümde ise sonuçlar ve tartışma bölümü yer alacaktır.

1.1. Tıp Bilişimi ve Bilgisayar Mühendisliği İlişkisi

Bilgisayar Mühendisliği, temelleri 1945'li yıllarda ilk bilgisayarların kullanılmasıyla atılmış olan çok genç bir mühendislik dalı olmasına rağmen, dünyada ve Türkiye'de çok hızlı bir gelişim göstermiştir. Bu mühendislik dalı bilgisayar sistemlerinin yapısı, geliştirilmesi ve bu sistemlerin diğer disiplinler ile etkin kullanımı konularında araştırmalar yapar. Ayrıca bu mühendislik temel bilim dallarının teorik çalışmaları ve deneysel yöntemleri ile diğer mühendislik bilimlerinin tasarım boyutunu da kapsamaktadır. Bu haliyle teorinin pratiğe aktarılmasında ve geniş kitlelerin hizmetine sunulmasında en önemli payı olan disiplinlerden birisidir (Cumhuriyet Üniversitesi, 2001).

Geniş bir açıdan bakıldığında bilgisayar uygulamaları tıp alanında artarak kullanılmaktadır. Tüm laboratuvar tetkiklerinin yapıldığı cihazlarda, tomografi, MRI, US ve Nükleer tıp uygulamalarında bilgisayar teknolojisi uzun bir süredir kullanılmaktadır. Geliştirilmiş robot bilgisayarlar aracılığıyla pek çok ameliyat günümüzde daha hızlı ve düşük hatalar ile yapılabilmektedir. Genetik çalışmalar bilgisayarın tıp alanında en çok kullanılan alanlarıdır. Tıp uygulamaları içinde bilgisayar teknolojileri tıp eğitiminde oldukça yaygındır (Erdem vd., 2005). Bilgisayar mühendisliğinin tıp alanında gerçekleştirdiği çalışmalar aşağıdaki gibi sıralanabilir (Umut, 2011).

- Fizyolojik sistemlerin modellenmesi, simülasyonu ve kontrolü,
- Fizyolojik sinyallerin algılanması, ölçülmesi ve monitörize edilmesi,
- Biyoelektrik verilerin sinyal işleme teknikleri kullanılarak tespiti, sınıflandırılması ve analiz edilmesi,
- Tedavi ve rehabilitasyon prosedürleri ve cihazlarının tasarlanması,
- Vücut fonksiyonlarını yerine getirmede yardımcı olacak, protez vb. ilave parçaların tasarlanması,
- Hastayla ilgili bilgilerin bilgisayarla analizi ile klinik karar vermeye yardımcı olunması,
- Tıbbi görüntüleme: anatomik detayların ve fizyolojik fonksiyonların görüntülenmesi,
- Faydalı amaçlar için yeni biyolojik ürünlerin tasarımı,
- Kliniksel gereçler, cihazlar, sistemler ve prosedürlerin geliştirilmesi ve dizayn edilmesi (Enderle, 2000).

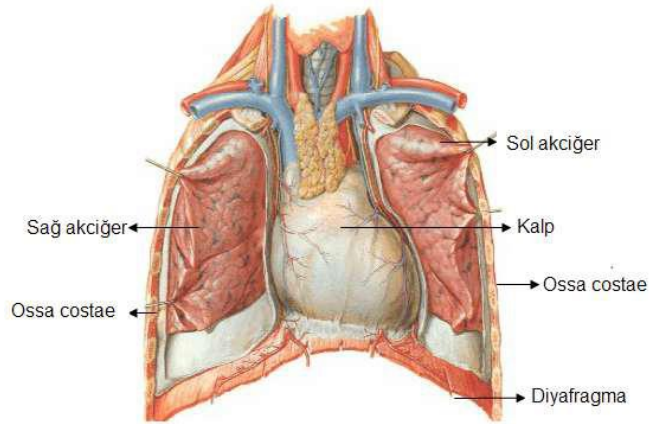
2. DOLAŞIM SİSTEMİ

Dolaşım sistemi, kanın vücuda dağılmasını sağlayan kapalı bir ağ sistemidir. Dolaşım sistemine kardiyovasküler sistem de denir. Bu sistem kalp ve damarlardan (arterler, venler ve kapiller) oluşur.

Dolaşım sistemi kalp tarafından ritmik hareketlerle pompalanan kanın damarlar vasıtasıyla hücrelere ulaşmasını ve hücrelerde kullanılmış olan kanı da damarlar vasıtasıyla toplayarak tekrar kalbe dönmesini sağlar.

2.1. Kalbin Yapısı ve İşleyişi

Kalp, dolaşım sisteminin motor organıdır. Temel işi kanı pompalamak olan kalp, çizgili kastan oluşan bir organdır. Çizgili kastan yapılmış olmasına rağmen isteğimiz dışında çalışır. Güçlü kas dokusuyla sürekli kasılıp gevşeyerek kanın damar içinde hareket etmesini sağlar. Vücudun ihtiyaçlarına bağlı olarak kalp dakikada 5 ile 35 litre arasında kan pompalayabilir.

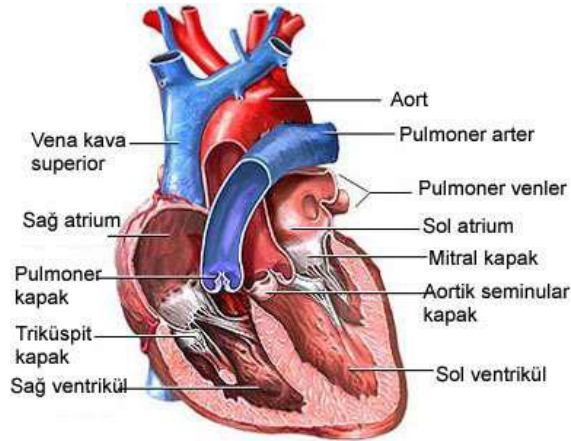


Şekil 2-1. Kalbin konumu ve komşulukları

2.1.1. Kalbin odacıkları

Kalbin sađ ve sol kısımları birbirinden bir duvarla ayrılmaktadır. Kalp ii boş drt odacıktan oluřmuřtur. Bu odacıkları kalbin iini blen eřitli duvarlar oluřturmuřtur. Septum interatriale (atriumlar arası blme), septum interventriculare (ventrikller arası blme) ve septum atrioventriculare (atriumlar ve ventrikller arası blme) ile kalp blmlere ayrılmıř drt odacık oluřmuřtur. Bu odacıklardan kalbin tepe blmndekilere kulakık (atrium), taban blmndekilere ise karıncık (ventrikl) adı verilir. Bu odacıklar řunlardır: Sađ kulakık, Sol kulakık, Sađ karıncık, Sol karıncık.

Sađ atrium ve sađ ventrikln her ikisi birden sađ kalbi oluřturur. Sađ kalpte oksijen bakımından fakir olan venz (kirli) kan bulunmaktadır. Sol atrium ve sol ventrikl ise sol kalbi oluřturur. Sol kalpte oksijen bakımından zengin olan arterial (temiz) kan bulunmaktadır.



řekil 2-2. Kalbin i kesiti

Sağ atrium kalbin basis bölümünün sağında bulunur. Sağ atriuma yukarıdan üst ana toplardamar, aşağıdan alt ana toplardamar açılır. Bu damarlar ile venöz kan kalbe döner.

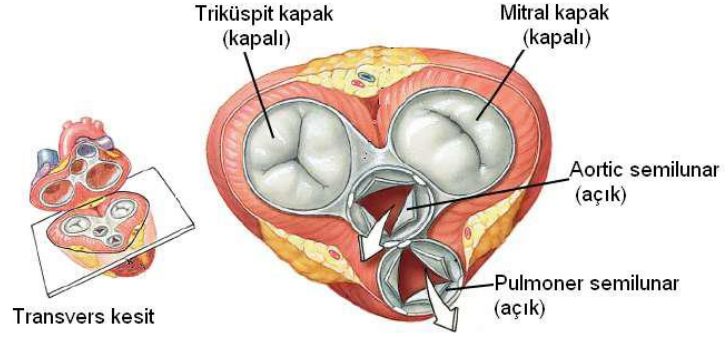
Sağ ventrikül piramid şeklinde bir boşluktur. Bu boşluktan venöz kan akciğerlere pompalanmaktadır. Pompalama görevi nedeniyle duvarı kalındır. Sağ ve sol ventriküller aynı anda kasılır. Kasılan ventrikülden kan akciğer atardamarı yolu ile akciğerlere gönderilir.

Sol atrium kalbin arkasında, sol üst yanında yer almaktadır. Bu boşluğa, sağ ve sol akciğerlerden oksijenlenerek dönen kanı getiren dört adet akciğer veni açılır. Buraya gelen kan, sol ventriküle geçer.

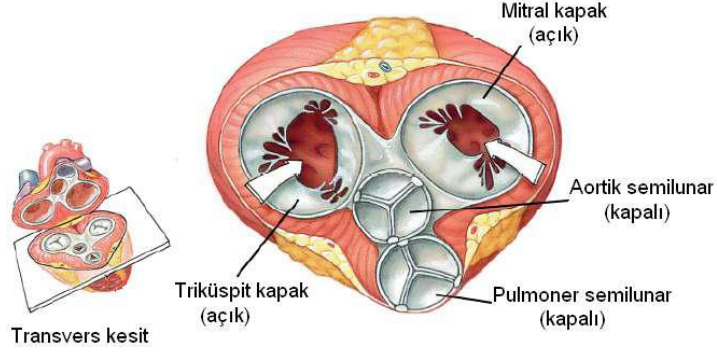
Sol ventrikül kalbin diyaphragmaya bakan yüzünde yer alır. Sol atriumdan gelen arterial kan sol ventriküle pompalanmakta ve buradan da ana atardamarla (aortae) vücudun en ince kapillerine kadar gönderilmektedir. Pompalama görevinden dolayı duvar yapısı diğer boşluklara göre oldukça gelişmiştir.

2.1.2. Kalbin kapakları

Kalpte iki adet atrioventriküler kapak, iki adet de büyük damar kapakları (semilunar kapak) olmak üzere 4 kapakcık bulunmaktadır. Kalp kapakçıklarının amacı kalpte kan akışının yalnızca tek yönde ilerlemesini sağlamak ve kanın geriye dönüşünü engellemektir. Bu kapaklar fibröz yapıda olup kan damarı bulunmaz, beslenmesi difüzyon yolu ile sağlanır (<http://sdb.meb.gov.tr>).



Şekil 2-3. Ventrikül kasılmasında kalp kapakları



Şekil 2-4. Ventrikül gevşemesinde kalp kapakları

2.1.3. Kalbin çalışması

Kalp kası sinirsel impulsa gereksinimi olmayan, kendi uyarılarını kendi oluşturabilen bir kastır. Ancak kalbin çalışması otonom sinir sisteminin denetimi altındadır. Sempatik sinirler kalbin ritmik kasılma ve gevşeme hareketlerini hızlandırırken parasempatik sinirler yavaşlatılmasını sağlar (<http://sdb.meb.gov.tr>).

Kalp, sürekli kasılıp gevşeyerek çalışır. Kalbin kasılmasına “sistol”, gevşemesine “diastol” denir. Kalpte her iki atrium ve her iki ventrikül birlikte kasılır ve gevşer. Atriumlar ve ventriküllerin kasılıp gevşemesi kanın hareketi için itici bir güç oluşturur. Bu kasılıp gevşeme birbirine zıttır. Atriumların her ikisi aynı anda sistol

durumundayken ventriküller diastol durumuna geçer. Kalbin bir sistol ve diastol hareketine bir kalp atışı denir. Kalp atışı yetişkin bir insanda dakikada 60–80 ortalama 70’dir, çocuklarda bu sayı dakikada 90–140 arasındadır (<http://sdb.meb.gov.tr>).

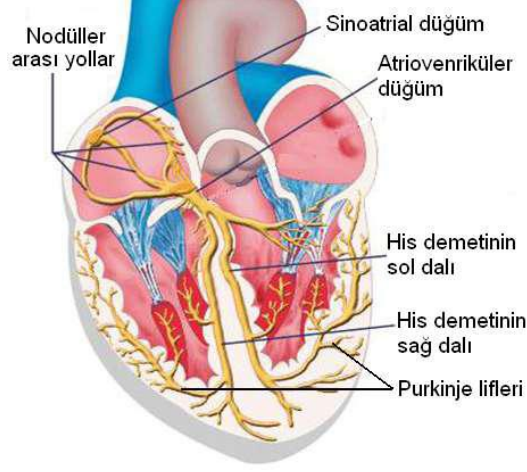
Atriumlar diastolde kanla dolar. Kanla dolduktan 0,1 saniye içinde sistol dönemi başlar. Bu dönemde ventriküller diastol durumundayken basıncın etkisiyle sağ atrium ve sağ ventrikül arasındaki triküspit, sol atrium ve sol ventrikül arasındaki mitral kapakçıklar açılır. Böylece atriumlardaki kan atrio- ventriküler deliklerden ventriküllere geçer ve ventriküller kanla dolar.

Ventriküllerin sistolünde artan basıncın etkisiyle triküsbit ve mitral kapaklar kapanır. Böylece kanın atriumlara geri dönüşü engellenir. Sağ ventriküldeki venöz kan akciğer atardamarı girişindeki seminular kapakçıkların açılmasıyla akciğerlere, sol ventriküldeki arteriyal kan ise aort girişindeki valvula aortun açılmasıyla aorta, oradan da tüm vücut dokularına dağılır.

2.2. Kalbin Uyarı ve İleti Sistemi

Kalbin atrium ve ventriküllerinin kesintisiz bir şekilde sistol ve diastolünü sağlayan özel bir yapısı vardır. Kalbin bu işini düzenli bir şekilde idare eden ve içinde sinir elemanları bulunan özel karakterdeki kas demetine “kalbin uyarı ve ileti sistemi” denir (<http://sdb.meb.gov.tr>).

Bu sistem; özel hücre kümeleri, demetleri ve liflerden oluşur. Uyarı ve ileti sistemi; sinoatrial düğüm (SA), atrioventriküler düğüm (AV), atrioventriküler demet (his demeti) ve purkinje lifleri olmak üzere dört bölümden meydana gelir. Bunlardan ilk ikisi uyarı, diğer ikisi ileti sistemidir.



Şekil 2-5. Kalbin uyarı ve ileti sistemi

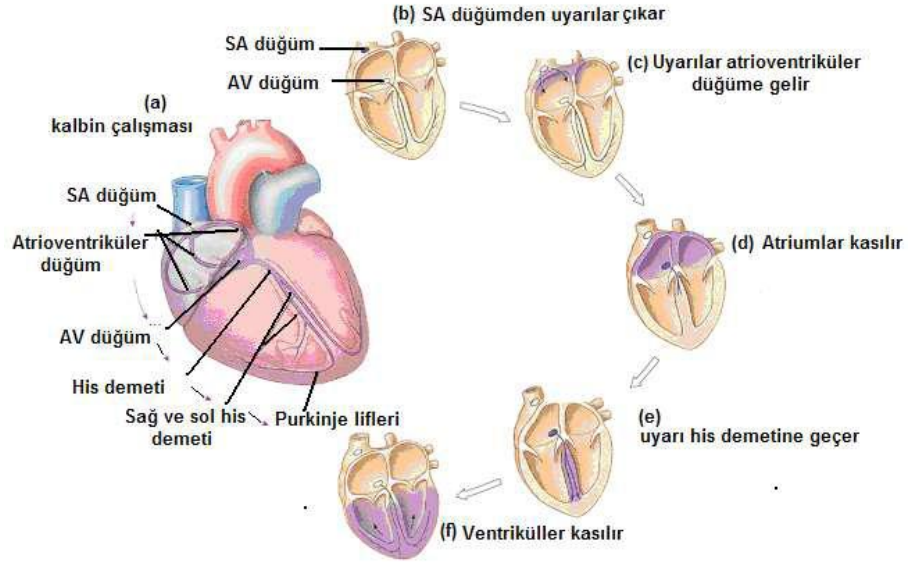
Sinoatrial (SA) düğüm, Sağ atriumun üst yan duvarında üst ana toplardamarın atriuma açıldığı yerin altındadır. Sinoatrial düğüm kalp atımlarını başlatan ve ritmini kontrol eden elektriksel uyarıların başladığı bölgedir. Bu nedenle sinoatrial düğüm “pace maker” (uyarı odağı) olarak tanımlanır. Sinoatrial düğümden çıkan uyarı önce atriumların kasını uyarır ve atrioventriküler düğüme gelir. Kalp sinoatrial düğümün emri altında çalışırken diğer yapılar uyarı çıkarmaz. Sinoatrial düğümün gönderdiği uyarıları iletme görevini yapar. Sinoatrial düğüm çalışmadığı veya sinoatrial düğümden çıkan uyarıların iletilmemesi gibi anormal koşullarda, atrioventriküler düğüm veya diğer yapılar kalbin durmasını engellemek için görevi üstlenip uyarı çıkarmaya başlar.

Atrioventriküler (AV) düğüm, Triküspid kapağın arkasında sağ atriumun arka duvarında bulunur. Sinoatrial düğümde oluşan uyarılar, düğümler arası yollarla atrioventriküler düğüme gelir. Buraya gelen uyarılar 0.1 saniyelik bir gecikmeyle his demetine geçer.

His demeti, atrioventriküler düğüme bağlıdır. Ventriküller arası bölmede sağ ve sol dallara ayrılır. His demetine gelen uyarı his demetinin sağ ve sol dallarında ilerleyerek sağ ve sol ventrikül kasındaki purkinje sistemine ulaşır.

Ventrikül kaslarına dağılan his demetinin daha küçük dallarına purkinje lifleri denir. Purkinje lifleri uyarıyı ventrikül kaslarına ileterek ventriküllerin kasılmasını sağlar.

Bu ileti sistemi sayesinde kalp fonksiyonel bir bütün olarak çalışır. Sinoatrial düğümde oluşan uyarının atrium kasına yayılması sonucunda atrium sistolü, ventrikül kasına yayılması sonucunda ventrikül sistolü olur. Uyarının kalpte yayılması sırasında atrioventriküler düğümdeki 0.1 saniyelik gecikme atriumların ventriküllerden önce kasılmasını sağlar. Böylece, ventriküller diastol hâlindeyken atriumlar sistoldedir. Sinoatrial düğümünden her bir uyarı kalp kasında bir sistolü takip eden bir diastole neden olur. Sinoatrial düğüm dakikada kaç uyarı çıkartıyorsa atriumlar ve ventriküller o kadar sayıda sistol yapar. Bir kalp vuruşu ventriküllerin sistolü olarak tanımlanabilir.



Şekil 2-6. Kalpte elektriksel uyarı akışı

2.3. Kan Basıncı

Dolaşım sisteminin işlevlerinden biri de damarlar içinde belli hızda ve sürekli olarak kan akımını sağlamaktır. Kalbin her sistolünde, arterlere pompalanan kan, damar yüzeyine bir basınç uygular. Kalp ventriküllerinin sistolü sonucu, kanın arter duvarlarına yaptığı basınca sistolik kan basıncı (büyük tansiyon) denir. Kalbin

ventriküllerinin diastolü sırasında arter duvarlarında oluşan direnç basıncına ise diastolik kan basıncı (küçük tansiyon) denir (<http://sdb.meb.gov.tr>).

Kan basıncı, cıva basıncına göre belirlenen standart birim ile tespit edilir. Normal bir yetişkin insanda sistolik basınç ortalama 120 mmHg, diastolik basınç 80 mmHg olarak bilinir. İnsan organizmasının ihtiyacına göre kan akış hızı değişebilir. İhtiyaca göre oluşan bu değişikliği otonom sinir sistemi düzenler, böylece vücudun ortalama kan basıncı düzenlenmiş olur (<http://sdb.meb.gov.tr>).

2.4. Nabız

Sistol esnasında, kalbin sol ventrikülünden aortta atılan kanın oluşturduğu basınç arter duvarında dalgalanmalara yol açar. Basıncın etkisiyle arterlerde oluşan dalgalanmalara nabız denir. Bir dakikalık süre içinde kalbin kasılmasıyla pompalanan kanın arter duvarına yaptığı basıncın sayısına nabız sayısı denir. Nabız sayısı normal insanda 60–80/dk. arasındadır. Nabız, kalbin bir dakikalık süre içerisinde arterlere kaç defa kan pompalandığının ve pompalama işleminin ritmik olup olmadığının göstergesidir. Nabız genellikle el bileğinden alınır (<http://sdb.meb.gov.tr>).

3. KALP RİTİM BOZUKLUĞU (ARİTİMİ)

Kalp hastalıklarının dünya genelinde önde gelen ölüm nedenlerinden olmasından dolayı (Tantimongcolwat vd., 2008; WHO, 2004) kalp hastalıkları üzerine çok sayıda araştırma yapılmaktadır. Bu araştırmalardan bir kısmı da aritmi (kalp ritim bozukluğu) üzerinedir. Aritmi, kalp ritminin düzensizleşmesi, yani anormal kalp ritmi olarak bilinir.

Kalp çarpıntısı polikliniklere ve acil servislere en sık başvuru sebeplerinden birisidir. Yurtdışında yapılmış bir çalışmada (Görenek vd., 2010) dâhiliye ve kardiyoloji polikliniklerine başvuran 500 hastanın %16'sının hekime geliş şikâyetinin kalp çarpıntısı olduğu tespit edilmiştir (Kroenke vd., 1990).

Günümüzde hemen her tür kalp çarpıntısı için tek değil, birden çok tedavi yaklaşımı mümkündür. Tedaviye yönelik yaklaşımların çeşitliliği bir avantaj olarak görülebile de, bu durum bazen hekimleri karar verme konusunda arada bırakabilmektedir (Görenek vd., 2010).

Bir üniversite hastanesinde çarpıntı şikâyeti ile başvuran 190 hastanın incelendiği ve olguların %84'lük kısmında etiyolojik sebeplerin belirlenebildiği bir çalışmada (Weber vd, 1996) çarpıntıların %43'ünde kardiyak sorunlardan, %31'nde psikiyatrik sebeplerden, %10'unda ise diğer nedenlerden kaynaklandığı (örneğin; ilaçlar, tirotoksikoz, kafein, kokain, anemi, vs.) görülmüştür (Görenek vd., 2010).

Aritmilerin oluşmasında genel olarak üç neden öne çıkmaktadır. Bunlar psikiyatrik nedenler, fiziksel ve duygusal strese bağlı nedenler ve son olarak kardiyak nedenlere bağlı aritmilerdir.

3.1. Kardiyak Nedenlere Baęlı Aritmiler

Kardiyak kkenli olguların bir kısmında yapısal ya da elektriksel bir kalp rahatsızlığı vardır. Yapısal kalp rahatsızlıkları arasında kalp kapak hastalıkları, akut miyokart infarktüsü, kardiyomiyopatiler, pacemaker sendromu gibi pek ok durum sayılabilir. Bununla birlikte hibir kardiyak sorunu olmayan bireylerde de zaman zaman supraventriküler hatta ventriküler aritmilerin grlebileceęi hatırdı bulundurulmalıdır (Grenek vd., 2010). Kardiyak kkenli aritmilerin oluřumuna neden olan sorunlardan bazıları ařaęıda maddeler halinde aıklanmaktadır.

3.1.1. Sins dęm ve atriyoventrikler dęm sorunları

3.1.1.1. Sins bradikardisi

Sins bradikardisi, istirahat halindeki bir kalp hızının yavaşlaması yani, dakikada 50 ya da altında olması durumudur. İstirahat halindeki bir kalp hızının dakikada 50 ila 90 arasında olması, normokardi yani normal kalp hızı aralığı olarak kabul edilebilir (Grenek vd., 2010).

3.1.1.2. Sins tařikardisi

Sinoatriyal nodan kaynaklanan ve hızı 100 vuru/dk'yı ařan hızlı ritme sins tařikardisi denir. Egzersiz, anksiyete ve sempatik sinir sisteminin aktivitesini artıran herhangi bir hastalık sins tařikardisine yol aabilir (Bennett, 2008).

3.1.1.3. Hasta sinüs sendromu

Sinüs nod hücrelerinde kardiyak impuls oluşumunun bozulması ile sinüs nod disfonksiyonu (SND) oluşur (Josephson, 2008; Kalman vd, 2001; Dinçkal, 2006). İlk defa 1968 yılında Ferrer tarafından tanımlanan SND, çeşitli nedenlere bağlı olarak sinüs nodunun fizyolojik pacemaker fonksiyonunu yapamamasını ifade etmektedir (Ferrer, 1968; Sweeney, 2004). SND varlığında klinikte fizyolojik ihtiyaca uymayan atriyal hızlarla karakterize hasta sinüs sendromu (HSS) ortaya çıkar (Kalman vd., 2001; Dinçkal, 2006; Bigger vd., 1979). HSS'da, sinüs bradikardisi, sinüs duraklaması, sinoatriyal ileti bozukluğu, atriyal taşiaritmiler, taşikardi bradikardi atakları ve yavaş ventrikül yanıtı atriyal fibrilasyon (AF) görülebilir (Adan vd., 2003; Straus vd., 1977).

3.1.1.4. Atriyoventriküler düğüm hastalıkları

Patolojik olarak AV düğüm hastalıkları konjenital veya kazanılmış hastalıklar olarak sınıflandırılır (Davies, 1976). Herhangi bir hastalık ister akut ister kronik seyir gösterebilir miyokardı etkileyerek AV düğüm seviyesinde AV düğümün kendisi, penetre olan veya dağılan his demeti parçalarına tesir ederek AV blok oluşturabilmektedir (Görenek vd., 2010). Konjenital AV blokta iletim AV nod seviyesinde kesilmiştir. Sonuç olarak yardımcı odak olarak his demetinden uyarı çıkar (40-80 vuru/dk) (Bennett, 2008).

3.1.2. Supraventriküler aritmiler

Anatomik olarak ventrikülün üst kısmından kaynaklandığı varsayılan taşikardik ritim sorunlarına supraventriküler, ventrikülden kaynaklanana da ventriküler taşiaritmiler denir (Lundqvist, 2003).

PAT (Paroksizmal Atriyal Taşikardi) diğer adı SVT (SupraVentrüküler Taşikardi) sık karşılaşılan bir ritim bozukluğudur. Bu çarpıntılar (taşikardiler) genellikle atriyumdan (kulakçık) başlar ya da atriyumları içine aldığı için bu şekilde adlandırılır. Aniden başlayıp aniden sonlanırlar genellikle hızları 150-280/dk'dır. Hayati tehlikeye neden olmazlar fakat oldukça rahatsız eder, hastaların yaşam kalitesini düşürür. Kalp ritim uzmanları (Elektrofizyolog) tarafından çeşitli sınıflamaları vardır. En sık AV düğüm içinde kısa döngü yapan AVNRT (AV Nodal Reentran Taşikardi) çarpıntılar (taşikardiler) gözlenir. Daha nadir olarak AVRT (AV Resprokal Taşikardi) tipi çarpıntılar olabilir (<http://turkaritmi.com>).

3.1.2.1. Atriyoventrüküler nodal reentrant (AVNRT)

En sık karşılaşılan çarpıntı şeklidir. AV nodun içindeki ikili yolun birbirleriyle olan kısa döngüsünden kaynaklanır. Aniden başlar ve aniden sonlanır. Hastaları çoğu çarpıntı başlayınca sol göğüs bölgesinde kuş çırpınıyor gibi hissederler. Boyun bölgesindeki damarlar şişer ve bazı hastalarda göz kararması, halsizlik ve tansiyon düşmesi gözlenir. Kalp hızı genelde 150-280/dk arasında hastadan hastaya değişir fakat her hastada sabit bir hızdadır. Bazen ilk ataktan sonra senelerce tekrar atak olmayabilir, bazen de çok sık ataklar olur (<http://turkaritmi.com>).

3.1.2.2. Atriyoventrüküler resprokal reentrant (AVRT)

AV node ve normalde olmaması gereken aksesuar yol ile kısa döngüsünden kaynaklanır. Akseuar yol normal iletim yönünde elektrik aktivasyonu varsa çarpıntı olmadan normal elktrografi ile tanınabilir (Wolf Parkinson White sendromu). Kalp hızı genelde 150-280/dk arasında hastadan hastaya değişir fakat her hastada sabit bir hızdadır. Bazen ilk ataktan sonra senelerce tekrar atak olmayabilir, bazen de çok sık ataklar olur (<http://turkaritmi.com>).

3.1.2.3. Atriyal taşikardiler

Atriyal taşikardiler (AT), diğer supraventriküler aritmilerle karşılaştırıldığında daha nadir görülen bir dar QRS kompleksli taşikardi türüdür. Atriyal taşikardi denince, atriyal yapılardan kaynaklanan ve devam etmesi için atriyoventriküler noda gerek duymayan, AVNRT ve AVRT gibi taşikardilerden farklı olan taşikardiler anlaşılmaktadır. Bu taşikardiler homojen değildir. Çoğunlukla köken aldıkları atriyal yapı ile ilişkili olarak, armış otomasite, tetiklenmiş aktivite ve reentry gibi mekanizmalar ile oluşur. Taşikardi sırasında atriyal hız 130 ile 250 /dk arasında değişmektedir (Görenek vd., 2010).

3.1.2.4. Atriyum fibrilasyonu

Atriyumdan kaynaklanan önemli bir aritmi türüdür. Atriyuma ait dokularda dakikada 350-600 kez gelişigüzel uyarılar oluşur (Bennett, 2008; Fak, 2011). Sinüs düğümü devre dışıdır. Bu kadar çok ve kaotik uyarı dâhilinde atriyum etkili şekilde kasılamaz. Diğer yandan atriyoventriküler düğüm bu kaotik uyarıları düzensiz olarak ve kısmen ventriküllere iletebilir. Kalp ve nabız atımları tamamen düzensizdir, hızlı veya yavaş olabilir (Fak, 2011).

3.1.2.5. Atriyal flutter

Klinikte atriyal flutter, tipik ve atipik olmak üzere iki şekilde görülmektedir. En sık görülen tipik formda atriyum dakikada 240 ile 350 arasında uyarı çıkarmaktadır. Genellikle atriyum hızı dakikada 300 civarındadır. Aritmi nedeni sağ atriyumdaki bir re entran dairedir. Re entran dairede uyarı sağ atriyum lateral kenarı boyunca inferiyora,

interatriyal septumda ise süperiyora doğru dönmektedir. Sol atriyum sağ atriyumdan kaynaklanan uyarılarla aktive olmaktadır (Bennett, 2008).

Atipik atriyal flutterde atriyum hızı 350-450 /dk'dır. Bu tipte istmus ablasyonu ve hızlı atriyal pacing taşikardiyi sonlandıramaz (Bennett, 2008).

3.1.3. Ventriküler aritmiler

Ventriküler aritmiler önemli morbidite ve mortalite nedenidir. Altta yatan yapısal ve/veya elektriksel kalp hastalığı morbidite ve mortaliteyi etkileyen en önemli faktördür. Ventriküler aritmi sıklığı, tipi ve sempton derecesi morbidite ve mortaliteyi belirleyen diğer faktörlerdir (Zipes vd., 2006).

Genel popülasyonda ani kardiyak ölümün en sık nedeni akut miyokardiyal iskemi/infarktüsüne ikincil gelişen primer ventriküler fibrilasyondur (Myerburg, 2001).

Genel popülasyonda ve göreceli yüksek riskli olgularda (hipertansiyon, diabetes mellitus, hiperkolesterolemi vs. koroner arter hastalığı risk faktörleri bulunan olgular) ani kardiyak ölümün en sık nedeni koroner arter tıkanıklığına bağlı gelişen miyokardiyal iskemi sonucu oluşan primer ventriküler fibrilasyonudur (Zipes vd., 2006; Myerburg, 2001).

Ventriküler erken vuru, beklenen sinus vurusu zamanından önce gelen, geniş QRS kompleksli (≥ 120 msn) ve sıklıkla retrograd P dalgalarının QRS kompleksi içinde gözlendiği vurulardır. Anatomik olarak His demetinin altından kaynaklanırlar. Üç veya daha fazla ventriküler erken vurunun ard arda gelmesi ile oluşan, hızı 100 vuru/dakika olan, geniş QRS kompleksli (≥ 120 msn) taşikardilere ventriküler taşikardi denir. Ventriküler taşikardiler devam sürelerine, morfolojilerine ve oluşturdukları hemodinamik bozukluklara göre sınıflandırılırlar (Görenek vd., 2010).

3.1.4. Kaçış vuruları ve dal bloklarına bağlı aritmiler

3.1.4.1. Kaçış vurularına bağlı aritmiler

Sinüs nodunun uyarı çıkarmasında sorun oluşur ise iletim yollarındaki yardımcı odaklardan kaçış vurusu ortaya çıkar. Kaçış vurusu bozulmuş pacemaker fonksiyonunu göstermektedir. Kaçış ritminin kendisinin tedavi edilmesine gerek yoktur. Eğer tedavi gerekirse yaklaşım temel ritmin hızlandırılmaya çalışılması şeklinde olmalıdır (Bennett, 2008).

Atriyal ve AV kavşak kaçış ritimleri genellikle 40-60/dakika hızında iken ventriküler kaçış ritimlerinde hız 25-40/dakikadır. Kaçış ritimlerinde hızın bu düzeylerin üzerinde olması (atriyumlar ve AV kavşak için >60/dakika, ventriküller için >40/dakika) “hızlanmış kaçış ritmi” olarak adlandırılır (İlerigelen, 2011).

3.1.4.2. Dal bloklarına bağlı aritmiler

His demeti, interventriküler septumda sol ve sağ dal olmak üzere ikiye ayrılmaktadır. Bu iletim yolları sayesinde sol ve sağ ventrikül çok hızlı bir şekilde uyarılmaktadır. Sol ileti dalı, sol anterior ve sol posterior olmak üzere kendi içinde iki fasiküle ayrılmaktadır. Herhangi birinde ortaya çıkan blok, anormal ve uzamış ventriküler QRS komplekslerine neden olmaktadır (Bennett, 2008).

3.2. Psikiyatrik Nedenlere Bağlı Aritmiler

Etyolojik açıdan ele aldığımızda çarpıntı sebepleri arasında psikiyatrik sorunların oldukça önemli bir yer tuttuğunu görmekteyiz. Çarpıntı çok çeşitli psikiyatrik hastalıkların göstergesi olabilir. Bunlar arasında anksiyete bozukluğu, panik atak, somatizasyon veya depresyon sayılabilir (Barsky vd., 1994; Barsky vd., 1994). Bununla

birlikte psikiyatrik sorunları olan hastalar gerçekte bir aritmeleri olmadığı halde çarpıntıdan şikâyet edebilirler. Yani bu grup hastalarda çarpıntı şikâyetinin oluşabilmesi için aritminin varlığı şart değildir (Zeldis vd., 1980).

Çarpıntının anksiyete ya da panik ile ilişkilendirildiği hastalarda çarpıntının mı ruhsal sorunları tetiklediği, yoksa ruhsal sorunlardan dolayı mı çarpıntının oluştuğunun ayırt edilmesi genellikle zor olmaktadır (Görenek vd., 2010).

3.3. Fiziksel ve Duygusal Strese Bağlı Aritmiler

Bazı olgularda uzun süreli supraventriküler ve ventriküler taşikardiler egzersize ya da duygusal strese bağlı sempatik sistem aktivasyonu ve katekolamin fazlalığı sebebi ile ortaya çıkabilmektedir. Egzersiz testlerinin gerçekleştirildiği çeşitli çalışmalarda kısa süreli supraventriküler ve ventriküler aritmilerin uzun süreli aritmilerden daha yaygın olduğu ve kalp hastalarında uzun süreli aritmilerin daha sık geliştiği gözlemlenmiştir. Ancak bu durumun istisnaları da söz konusudur (Görenek vd., 2010).

4. TIPTA YAPAY ZEKÂ DÖNEMİ

4.1. Yapay Zekânın Tarihi

Yapay zekâ (AI) arařtırmalarının tanınmıř isimlerinden D. Lenat ve E. Feigenbaum zekâyı “arama alanı” kavramı aısından řu řekilde tarif etmektedirler (Kocabař, 2011): "Zekâ, karmařık bir problemi özüm arama alanını daraltarak kısa yoldan özebilme kabiliyetidir". Feigenbaum daha sonra zekâyı “bilgi kullanımı” kavramına baėlı olarak tarif etmektedir: "Zekâ, karmařık bir problemi özmek iin gerekli bilgileri toplayıp birleřtirebilme kabiliyetidir."

AI, insan dūřüncesinin formelleřtirilmesi esasına dayanır. inli, Hindistanlı ve Yunanlı filozoflar ilk milenyumda formel tūmdengelimini yapısal metotlarını geliřtirdiler. Onların dūřünceleri Aristo, Öklid, El-Harizmi, Occam ve Duns Scotus gibi filozoflar tarafından ilerleyen dönemler ierisinde gittike geliřtirildi (Berlinski, 2000).

Mayorkalı filozof Ramon Llull (1232–1315) tarafından bilgi üretimine dayalı birkaç mantıksal makine geliřtirildi (Artau vd., 1939). Ramon Llull geliřtirdiėi makineyi řu řekilde tanımlamaktadır: basit mantıksal iřlemlerle birlikte temel ve inkâr edilemez gerekleri birleřtiren mekanik bir varlık (Bonner, 2007). Ramon Llull'ın dıřında miladi 9. yy. da Abbasiler döneminde (750-1256) Musa kardeřlerin, hidrolik prensiplerine göre alıřan otomatik makineler yaptıkları, daha sonra Seluklular döneminde Ebul İzz'in bu alıřmaları devam ettirdiėi bilinmektedir (Kocabař, 2011). Ayrıca otomatik makineler konusu, Rönesans döneminde Leonardo da Vinci'nin (1452-1519) de dikkatini ekmiřti (Kocabař, 2011) ve bu amala mekanik hesap makinesi tasarlamıřtır (Russell vd., 2003).

17. yy. geldiėimizde Thomas Hobbes (1588-1679) akıl yürütmenin derin dūřüncelerimizde eklediėimiz ve ıkardıėımız sayısal hesaplamalar gibi olduėunu ileri sürmekteydi. Ayrıca Gottfried Leibniz, Thomas Hobbes ve René Descartes tüm rasyonel dūřüncenin cebir veya geometri kadar sistematik olabilme olasılıėını

keşfettiler (Russell vd., 2003). Daha sonra Fransız matematikçisi Pascal (1623-1662) mekanik bir hesap makinesi geliştirmeye çalıştı. 19. yüzyılda ise İngiliz bilgini Babbage, Fransız mühendis Jacquard'ın dokuma tezgâhları için icat ettiği bir tekniği kullanarak ilk programlanabilir mekanik bilgisayarı geliştirmeye çalışmıştır, fakat o zamanki teknoloji yeteri kadar hassas olmadığı için projesi yarım kalmıştır (Kocabaş, 2011).

Sembolik mantığın, 19. yy. ikinci yarısında Boole ve daha sonra Frege'nin çalışmalarıyla başlayıp 20. yy. Russell ve Whitehead'in çalışmalarıyla gelişmesi de bilgisayar bilimlerinin ve AI'nın gelişmesinde önemli rol oynamıştır. İkinci Dünya Savaşı öncesinde Turing'in hesaplanabilirlik teorisi üzerine yaptığı çalışmalar, zeki sistemler üzerine yapılan önemli teorik çalışmalar olmuştur. 1940'larda sibernetik alanında yapılan çalışmalar ise, insan ve makine arasındaki birçok paralellikleri ortaya çıkarmıştır. Sibernetik, 1940'larda ve 1950'lerde enformasyon teorisi, geri beslemeli kontrol sistemleri ve elektronik bilgisayarlarla ilgili kavramları birleştiren önemli bir araştırma alanı olmuştur (Kocabaş, 2011).

AI, ayrı bir araştırma alanı olarak 1950'lerde bilgisayarların ticari şirketlerde kullanılmasıyla ortaya çıkmıştır. Bu yıllarda Claude Shannon ve Allen Newell'in geliştirdikleri satranç programları ve diğer oyunları oynayan programlar geliştirilmiştir. Gene bu yıllarda otomatik çeviri programları üzerine çalışmalar yapılmıştır (Kocabaş, 2011).

1950'lerin ortaları, AI'nın ayrı bir araştırma alanı olarak ortaya çıktığı dönem olarak kabul edilir. 1956'da IBM tarafından Dartmouth Koleji'nde yapılan bir seminere daha sonra AI'nın öncüleri sayılacak olan Marvin Minsky, Allen Newell ve Herbert Simon katılmış ve aynı toplantıya katılan John McCarthy bu alandaki çalışmalara ilk defa "yapay zekâ" adını vermiştir (Kocabaş, 2011; <http://en.wikipedia.org>).

4.2. Tıp Alanında Yapay Zekânın Başlangıcı

“Yapay Zekâ”nın bilimsel bir hüviyet kazanması ile birlikte AI araştırma alanı kuruldu (<http://en.wikipedia.org>). Bu dönemden 15 yıl sonra yani 1970’lerin başından itibaren Tıpta Yapay Zekâ (AIM) üzerine çalışmalar ortaya çıkmıştır (Vimla vd., 2009). İlk olarak tıp araştırmacıları yaşam biliminde AI metotlarının uygulanabilirliğini keşfettiler. 1960’ların sonu ve 1970’lerin başından itibaren Dendral deneylerde (Kocabaş, 2011; Lindsay vd., 1980; <http://en.wikipedia.org>) AI çalışmaları gözle görülebilir bir şekilde çığır açmıştır. 1970’ler boyunca AI’nın biyomedikal uygulamalarda kullanımına büyük bir ilgi vardı. Stanford Üniversitesinde SUMEX-AIM Computing Resource’un (Freiherr, 1980) kurulmasıyla birlikte kısmen de olsa bu ilgi arttı (Vimla vd., 2009). Tıp alanında kullanılan ilk AI sistemleri şunlardır: Internist-1 (Miller vd., 1982), CASNET (Weiss vd., 1978) ve MYCIN (Shortliffe, 1976).

Tıp dünyasında geliştirilmiş olan çeşitli uygulamalar AI araştırma gruplarını büyülemekteydi. Aynı zamanda ilgi çekici biyomedikal problemlerle boğuşan AIM araştırmacıları için yeni AI metotları da ortaya çıkıyordu. Ayrıca 1978 yılında, bu alanda ki öncü bir dergi (Artificial Intelligence, Elsevier, Amsterdam) sadece AIM araştırmaları ile ilgili makalelere özel bir yayın (Sridharan, 1978) ayırmaktaydı (Vimla vd., 2009).

4.3. AIM’in Ana Noktaları

Edward H. Shortliffe 1991 yılındaki bir sunumunda AIM için üç ana noktaya değinmiştir. Bu noktalardan ilk ikisi bu tez konusu ile doğrudan ilgili olduğundan dolayı bu konu başlığı altında yer almıştır.

Bu noktalardan ilkinde Edward H. Shortliffe, Tıp alanında Yapay Zekânın; biyomedikal bilişimin yanı sıra dünya sağlık planlaması ve politikasından ayrı düşünülmemeyeceğini belirtmiştir. Ayrıca sağlık hizmetleri ve biyomedikal bilimlerde AIM’in etkilerinin gerçekçi beklentileri birçok metodolojik alandan sadece biri olarak

AI'yi kullanmamızı gerektirmektedir. Bu durum, AIM arařtırmacılarının ilkeli yaklařımlardan insan-bilgisayar etkileřimine ya da veritabanı teorisinden sayısal analize ve ileri istatistięe kadar olan alanı ve gerekliyse bilgisayar bilimi ve informatik'in dięer alanlarını da kullanmak için istekli olmalarını gerektiren bir argümanı ifade etmektedir (Vimla vd., 2009).

İkinci olarak geręek dünya ortamında AIM'in pratik etkisi, dięer uygulamalar ve bilgi-tabanlı araçlarla birleřmesine izin veren tümleřik ortamların geliřimine baęlı olmaktadır. Baęımsız konsültasyon sistemleri kavramı 1980'lerin sonlarına doęru aęıęa çıktı (Miller vd., 1990) ve böylece elektronik medikal kayıtlar, sipariř takibi sistemleri, sonuç raporlama sistemleri, reęete sistemleri ya da biyolojik aęıdan genomik/proteomik veri yönetim ve analiz araçları gibi yaygın elektronik sistemlerle AI notasyonları birleřtirilmelidir (Vimla vd., 2009).

4.4. AIM Metotları

AIM, önemli klinik ve biyolojik problemleri ele almak için AI'nin ilham verici biręok metodunu kullanan AI arařtırmacılarıyla birlikte çok canlı bir konudur. AIME 2011'de (<http://www.aimeedicine.info>) ele alınan konu ve temalar Tablo 4-1'de gösterilmektedir. Bilgisayar bilimi ve biyomedikal informatik'in bir alt dalı olarak tanımlanan Tıpta yapay zekâ'yı tanımak ve tanımlamak için iki yılda bir yapılan AIME konferansları ve uluslar arası bir dergi olan Artificial Intelligence in Medicine bu alanda göze çarpan iki büyük unsur olarak karřımıza çıkmaktadır.

AIM arařtırma alanları ięerisinde çok farklı AI metotları yer almaktadır. Özellikle AIME-2007 (<http://www.aimeedicine.eu>) konferansında bu konular üzerinde durulmuřtur. Bu ele alınan konu ve temalar Tablo 4-2'de gösterilmektedir. Bu arada birkaç AIM arařtırma alanında etkileyici ilerlemeler görölmektedir (Vimla vd., 2009). Bu alanlar: bilgi gösterimi (knowledge representation [Stanford Üniversitesi AIM arařtırmalarının bir sonucu olan ve dünyaca yaygın bir ürün olan Prote'ge' (Noy vd., 2003) buna bir örnektir.]), makine öğrenmesi (ML), bilgi keřfi için veri madencilięi ve geęici temsil ve uslamlama.

Çizelge 4-1. AIME-2011'deki konu ve temalar

Bilgi Tabanlı Sistemler
Veri Madenciliği
Yapay Zekâ Uygulamalarında Özel Konular
Olasılık Modelleme ve Uslamlama
Terminolojiler ve Ontolojiler
Geçici Temsil ve Geçici Veri Madenciliği
Terapi Planlama, Zamanlama ve Yönergeye Dayalı Bakım I
Terapi Planlama, Zamanlama ve Yönergeye Dayalı Bakım II
Doğal Dil İşleme I
Doğal Dil İşleme II

Çizelge 4-2. AIME-2007'deki konu ve temalar

Bilgisayar Tabanlı Bilgi Üretimi	Veri ve Bilgi Temsili
Klinik Veri Madenciliği	Bilgi Tabanlı Sağlık Hizmetleri
Olasılık and Bayesian Analizi	Özellik Seçimi/Çıkarımı
Görselleştirme	Sınıflandırma ve Filtreleme
Bilgi Erişimi	Etmen Tabanlı Sistemler
Geçici Veri Madenciliği	Makine Öğrenmesi
Veritabanlarında Bilgi Keşfi	Metin İşleme
Doğal Dil İşleme	Ontolojiler
Karar Destek Sistemleri	Resim İşleme
Örüntü Tanıma	Klinik Yönergeler
İş Akışı	

4.5. AIM Çalışmalarında Önemli Aşamalar

Peter Szolovits AIM çalışmalarının sadece dört önemli aşama ile karşılaştığını belirtmektedir (Vimla vd., 2009). Bunlar aşağıda sırasıyla belirtilmektedir.

- Veri yakalama ve işleme.
- İş akışı için tasarım, destek ve modelleme geliştirme.
- Gizlilik kaygıları olan hastalar için güvenilir metotlar.
- En iyi modelleme teknikleri.

İlk dönem AIM araştırmalarının çoğu gelişmiş bilgisayar programları üzerinden bir insan uzmanın uzmanlığını kullanmaya dönük düşünülmüştür. Bu dönemler içerisinde güncel veri elde etmek çok kolay değildi. Ancak günümüzde bir veri okyanusu içerisinde yüzmekteyiz. Ancak bu duruma paralel olarak uzmanlığın değeri gittikçe düşmektedir (Vimla vd., 2009). Ancak günümüzde sağlık hizmetlerinde toplanan büyük miktarda veri olmasına rağmen, verinin büyük bir kısmı eksik ya da yanlış toplanmaktadır (Vimla vd., 2009). Bu durum hastanın anamnez ya da ilaç tedavisi gibi detaylı kayıtlarında sorunlar teşkil etmesine sebebiyet verir. Aynı zamanda yaygın bir biçimde kabul görmüş terminoloji ve ontolojilerin eksikliği bile iyi kayıt altına alınmış bilginin değişimini ve kullanımını zor kılar (Vimla vd., 2009).

AIM ya da diğer metotlara dayalı olan ya da olmayan sistemler insan pratisyenlerle birlikte yürütülmelidir. Bundan dolayı, bu sistemler pratisyenlerin ne yaptığını, ihtiyaç duydukları bilginin ne olduğu ve sistemdeki bozulmanın sistem tarafından tolere edilebileceğinden daha fazla olduğu zamanı modellemelidirler. Böylelikle birçok hatanın görevden ziyade ihmalden kaynaklandığını görebiliriz. Bu durum, arka planda çalışan sistemlerin sürekli olarak her hasta için verilen hizmetleri izlemesinin garantisini verecektir (Vimla vd., 2009). Bu da bize her açıdan düşünülmüş bir iş akış sürecinin AIM'e dayalı sistemlerin başarımını daha da artırdığını söylemektedir.

Sağlık hizmetlerinin elektronik olarak izlenmesiyle oluşan direnç, sağlık hizmetlerinin toplanması sonucu kitlesel açılımlara olan hassasiyeti arttırmasından kaynaklanır (Institute of Medicine, 1997). Bir anda milyonlarca insanın verilerini etkileyen kurumsal hatalar günlük yaşantımızda sıklıkla karşılaştığımız bir durumdur. Böylece bu kitlesel bilgiler tıbbi ifşadan ziyade kimlik hırsızlığını tehlikesini ortaya çıkarmaktadır. Bu bilgilerin istenmeyen yayılımını güvenlik altına almak için günümüz kullanıcı adı/şifre kombinasyonu tarzındaki yöntemlerden daha iyi güvenli kimlik

doğrulama sistemlerine ihtiyaç vardır. Bunun için kişisel akıllı kartlar, biyometrikler ya da insanların kimliklerini tanımlamaya yardımcı olan mevcut akıllı teknolojiler kullanılabilir (Vimla vd., 2009).

Yüzbinlerce DNA, RNA ve protein seviyelerini eşzamanlı olarak belirleyen mikrodiziler ve yarım milyon SNP veya bir kişinin bütün genetik dizilimini belirleyen yeni ölçüm teknikleri geleneksel teşhis veya tedavi yürütme sistemlerindeki çok sayıdaki bulgu ile basitçe çözülemez. Bu tür bilgileri anlamlandırmak, analizlerini otomatikleştirebilen ileri seviyede AI yöntemlerini gerektirecektir. Toplum olarak çok sayıda veriyi çözümlmek için geleneksel istatistikî ve daha yeni veri madenciliği metotları ve makine öğrenmesi yaklaşımlarını benimsedik. Maalesef, bu teknikler sadece verilerdeki basit ilişkileri keşfeder ve bizim insanoğlu olarak moleküler biyolojiden kompleks multiorganizmalara ve sıtma gibi epidemolojideki çevresel faktörlere kadar bizim anlayışımızı vurgulayan kompleks ilişkiler ağını henüz keşfetme yeteneği yoktur. Yüzyıllık tecrübe ve deneylerle gelişen insan uzmanlığı verileri analiz ederek yeniden keşfedileceği ümidiyle bir kenara atılamaz. Örneğin, Peter Szolovits terabaytlarca kayıtlı yoğun bakım ünitesi izleme verilerini çözümleyebilecek hatta kalp tarafından pompalandığı için kan dolaşımı gibi basit gerçekleri bile keşfedebilecek otomatik bir yöntem bilmediğini söylemektedir. Ayrıca Peter Szolovits, sağlık ve hastalıkta insan organizmasının nasıl işlediğini ölçen hipotezleri formüle eden çok çeşitli mevcut verileri işleyebilen makine öğrenmesi metotları ile insan uzmanlığını birleştiren daha iyi modeller oluşturmanın büyük bir meydan okuma olduğunu düşünmektedir (Vimla vd., 2009).

Michael Berthold'a göre tıpta AI yaklaşımlarının ilerleyen ve devam eden meydan okumaları araştırmadan önce bu araştırma alanındaki bilim türünü kategorize etmek faydalıdır. Bu kategorize etme işi üç safhaya ayrılmaktadır. Bunlar: Toplama, Sistemleştirme ve Biçimlendirme (Vimla vd., 2009).

Toplama, eldeki problemler hakkında veri toplamaktır. Önemli düzenlemeler ve sistemler hakkında ne net bir bilgi var ne de araştırmacılar bu alandaki veri alanları hakkında çok fazla bilgiye sahiptir (Vimla vd., 2009).

Sistemleřtirme, toplanan verinin daha iyi organize edilmesi ve modellerin belli özellikleri tahmin etmek için inşa edilmesidir. Ancak çoęu kez bu modeller belli sistem hakkında net bir bilgiye sahip deęildir. Orijinal verileri oluřturan sistem daha çok kara kutudur (Vimla vd., 2009).

Biçimlendirme de, bu önemli sistemin daha iyi anlaşılması gerçekleştirilir ve teoriler, hedef ve sistematik deneylerle biçimlendirilebilir ve doğrulanabilir (Vimla vd., 2009).

5. MAKİNE ÖĞRENMESİ

AI'nın çalışma alanlarını kronolojik olarak üç ana başlık altında toplayabiliriz. Bunlar: Bilişsel modelleme yaklaşımı, sembolik düşünmeye dayanan kavram öğrenme ve bilgi tabanlı sistemlerdir (Kocabaş, 2011).

Bilişsel modelleme yaklaşımı ile birlikte zihinsel süreçlerin nasıl çalıştığına ilişkin teoriler üretilmeye başlandı (Kılıçaslan, 2011) ve bu konudaki ilk çalışmalar McCulloch ve Pitts tarafından yapıldı (<http://tr.wikipedia.org>). Bu araştırmacıların önerdiği, yapay sinir hücrelerini kullanan hesaplama modeli, önermeler mantığı, fizyoloji ve Turing'in hesaplama kuramına dayanıyordu. Her hangi bir hesaplanabilir fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceğini ve mantıksal "ve/veya" işlemlerinin gerçekleştirilebileceğini gösterdiler. Bu ağ yapılarının uygun şekilde tanımlanmaları halinde öğrenme becerisi kazanabileceğini de ileri sürdüler. Daha sonra Newell ve Simon, insan gibi düşünme yaklaşımına göre üretilmiş ilk program olan Genel Sorun Çözücüyü geliştirmişlerdir. Simon, daha sonra fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturmuştur. Simon'ın bu tanımlaması bilim adamlarının AI'ya yaklaşımlarında iki farklı akımın ortaya çıktığını belirginleştirmesi açısından önemlidir: Sembolik Yapay Zekâ ve Siberetik Yapay Zekâ (<http://tr.wikipedia.org>).

Sembolik Yapay Zekâ ya da diğer bir deyişle sembolik düşünmeye dayanan kavram öğrenme temel olarak mantık kurallarına dayanarak akıllı sistemler üretmeyi amaçlayan bir yaklaşım içerisinde olmuştur. Modern mantığın ve modern felsefenin kurucusu Alman mantıkçısı Gottlob Frege (1848-1925) "Matematik mantığın uygulama alanıdır," görüşünden hareketle matematiğin, mantığın aksiyomatik sistemi üzerine kurulabileceğini düşünmüştür. Bu düşünceden hareket ederek aritmetiğin temelleri konusundaki felsefi çalışmaları için bir mantık sistemi geliştirmiştir (Kocabaş, 2011). Daha sonra, Frege'nin çalışmalarına dayanarak, Russell ve Whitehead 1910-1913

yılları arasında Matematiğin Prensipleri (Principia Mathematica) adını verdikleri eserde matematiği mantığa indirgeyerek formel bir sistem haline getirmeye çalıştılar. Fakat matematiğin formel hale getirilemeyeceğini Gödel 1933'te yayınladığı bir kitabındaki (Über die unentscheidbare Sätze der Principia Mathematica und verwandter Systeme) meşhur teoremiyle gösterdi (Kocabaş, 2011; Kılıçaslan, 2011). A. Robinson, 1967'de "çözülüm teorem ispatlama" (resolution theorem proving) metodunu geliştirdi. Bu metot 1972'de A. Colmaurer tarafından ilk "Mantık Programlama" (Prolog) dilinin geliştirilmesine yol açtı. Bu dil 1975'te D. Warren tarafından "Warren Abstract Machine" (WAM) olarak uygulandı. Kişisel bilgisayarlar üzerinde ilk uygulamalar 1980'lerde ortaya çıktı. Mantığın Frege ve öteki mantıkçılar tarafından 19. yüzyılın sonlarında sembolik hale getirilmesi, dokuzuncu yüzyılda Türk matematikçisi M. Musa bin el-Harezmi tarafından cebirin icat edilmesiyle aritmetik hesapların sembolik hale getirilmesiyle karşılaştırılabilir. Cebirin icadı matematikçiler tarafından matematik tarihinin en büyük soyutlamalarından biri olarak kabul edilmektedir (Kocabaş, 2011).

Bilgi tabanlı sistemler belli bir problemi çözmek için insanın problem çözme becerisini taklit etmeye programlanmış ve bu problemle ilgili daha önceden bilgi ile donatılmış sistemlerdir. Bu tür sistemlere örnek olarak Durum Tabanlı Çıkarılma Sistemleri, Uzman Sistemler verildiği gibi Makine öğrenmesi (ML) de bilgi tabanlı sistemlere bir örnek olup AI araştırma alanlarından bir tanesidir.

5.1. ML Nedir?

"Öğrenme", kesin bir biçimde tanımlanması zor olan süreçler dizisini kapsar. Sözlük anlamı olarak "öğrenme"; öğrenim, talimat ve deneyim yoluyla beceri, anlama ve bilgi kazanımı olarak tanımlanır (Nilsson, 1996; Witten vd., 2005). Bilgi elde edinim süreci insan öğrenmesinin öğrenme süreci boyunca elde etmiş olduğu verileri kullanması ile mümkündür. Çünkü veri, bir nesnenin özellikleri hakkında bize bilgi veren en temel yapıdır. Yaşamımızda ve dünyada veri miktarının arttığı (Witten vd., 2005) ve world wide web (WWW) ile erişiminin gittikçe kolaylaştığı

görülmektedir. Bunun yanı sıra veri, soyutlama seviyesinin en düşük seviyesi; bilgi ise en yüksek seviyesi olarak bilinir.

Veri üzerinde örüntü bulma bilgi keşfi için önemli bir süreçtir. Bilgi keşfi süreci, veri madenciliği olarak da adlandırılır (Glossary of Terms, 1998) diğer bir ifadeyle veri üzerinde ML metotlarının uygulanması veri madenciliği olarak adlandırılır (Alpaydın, 2004). ML, bilgi keşfi sırasında kullanılan tümevarımsal algoritmaların uygulanması sürecini tanımlamak için çok yaygın bir biçimde kullanılan bilimsel bir çalışma alanıdır (Glossary of Terms, 1998). ML genellikle AI ile ilişkili görevleri yerine getiren sistemlerdeki değişim olarak ta adlandırılır. Bu gibi görevler tanıma, teşhis, planlama, robot kontrol, tahmin vb. konuları içerir. “Değişim” ifadesi ile ya zaten çalışan bir sistemin iyileştirilmesi ya da yeni sistemlerin birleştirilmesi anlatılmaktadır (Nilsson, 1996).

ML, örnek veri ya da geçmiş deneyimleri kullanarak bir başarımlı ölçütünü en uygun hale getirmek için bilgisayarları programlar. Bazı parametrelerle tanımlanmış bir modele sahip olduğumuzu düşünürsek “öğrenme”, eğitim verisi yada geçmiş deneyimleri kullanarak bu modelin parametrelerini en uygun hale getirmek için bir bilgisayar programının yürütülmesi olarak tanımlanabilir. Model ise geleceğe yönelik tahminler yapmak için bir tahmin edici ya da veriden bilgi elde etmek için bir tanımlayıcı ya da her ikisi olabilir (Alpaydın, 2004). ML, öğrenci tarafından tutulan herhangi bir ön bilgi ve gözlemlenen veriyi kullanarak olası hipotez uzayı içerisinde en uygun olanına karar vermek için arama gerçekleştirir (Mitchell, 1997).

5.2. Öğrenme Türleri

Öğrenme için uygun geribildirim türü genellikle öğrenme probleminin doğasına karar vermede çok önemli bir faktördür (Kılıçaslan, 2011). ML'nin alanı üç öğrenme türüne ayrılır. Bunlar: Gözetimli Öğrenme, Gözetimsiz Öğrenme ve Takiyeli Öğrenme.

Gözetimli Öğrenme, girişlerle çıkışların birbirleriyle eşleştiği örneklerden bir fonksiyonun öğrenilmesi ya da bir hipotezin bulunmasıdır (Nilsson, 1996; Alpaydın, 2004; Kılıçaslan, 2011). Gözetimli Öğrenmede girişlerle çıkışların birbirleriyle eşleştirilmesi bir gözetici tarafından gerçekleştirilir. Gözetimli öğrenme problemlerinden bir tanesi de sınıflandırma'dır (Alpaydın, 2004).

Gözetimsiz Öğrenmede bir gözetici mevcut değildir (Alpaydın, 2004). Bundan dolayı giriş verilerine karşılık bir çıkış verilmemiştir. Gözetimsiz Öğrenme özel bir çıkış değeri verilmemiş girişler üzerinde bir patern bulmayı hedefler (Nilsson, 1996; Alpaydın, 2004; Kılıçaslan, 2011). Gözetimsiz Öğrenme metotları anlamlı kategoriler biçimindeki verileri sınıflandırmak için yollar bulmaya çalışan taksonomik problemler konusunda uygulaması vardır (Nilsson, 1996).

Bazı uygulamalarda, sistemin çıkışı bir hareketler dizisi şeklindedir. Böyle bir durumda tek bir hareket önemli değildir. Önemli olan hedefe ulaşmak için doğru hareketlerin diziliminin planlanmasıdır. Herhangi bir durumda en iyi hareket diye tanımlayabileceğimiz bir şey yoktur. Bir hareket eğer iyi bir planın parçası ise iyi olarak kabul edilir. Böyle bir durumda, ML programı yapılan planların iyiliğine değer biçebilecek ve geçmişteki iyi hareketlerden bir plan oluşturabilmeyi öğrenebilecektir. Böyle öğrenme türleri takviyeli öğrenme olarak adlandırılır (Alpaydın, 2004).

5.3. Örneğe Dayalı Öğrenme

Öğrenme türlerinden biri olan Gözetimli Öğrenmeyi kullanan birçok tümevarımsal makine öğrenmesi algoritması vardır. Tümevarımsal makine öğrenmesi (IML) algoritmaları çıkışı bilinen etiketlenmiş veriden patern öğrenebilir (Duda vd., 2001). Veri üzerinde patern öğrenme için çok farklı ML yaklaşımı vardır (Bishop, 2006). Bu yaklaşımlardan bir tanesi de Tembel Öğrenmedir. Tembel Öğrenme (LL) durum tabanlı öğrenme ve bir veritabanındaki depolanan paternleri tarayan ve bir sorguya yanıt vermek için gerekli olan yerleştirilmiş bir modeli kuran akıl yürütme yaklaşımlarını tanımlar (Brownlee, 2007).

Örneğe Dayalı Öğrenme ya da hafıza tabanlı öğrenme metotları bazen LL metotları olarak adlandırılır (Mitchell, 1997). Örneğe Dayalı Öğrenme (IBL)de tüm gerçek iş, eğitim kümesi işlendiği zaman değil yeni bir örnek sınıflandırmak istendiği zaman gerçekleştirilir. Bir anlamda IBL ile diğer öğrenme metotları arasındaki fark “öğrenme” işlemi meydana geldiği zamandır. IBL mümkün olduğu kadar gerçek işi ertelediği için “tembel (lazy)”dir (Witten vd., 2005). Bu nedenle IBL yaklaşımının dezavantajı yeni bir örneği sınıflandırma maliyetinin yüksek olabilmesidir (Mitchell, 1997). Diğer öğrenme metotları ise veri gelir gelmez bir genelleme yaptığı için “gayretli (eager)”lerdir (Witten vd., 2005; Kılıçaslan vd., 2009).

IBL’de her bir yeni örnek metrik bir mesafe kullanılarak mevcut olan diğer örneklerle karşılaştırılır ve en yakın mevcut örnek yeni örneğin sınıfına atanır. Bu en yakın komşuluk sınıflandırma metodu olarak adlandırılır (Witten vd., 2005). En yakın komşuluk (kNN) algoritması ML algoritmaları arasında en basit olanlardan biridir.

5.4. kNN Sınıflandırıcı

m-boyutlu bir uzayda bir veri noktası koleksiyonu ve bir sorgu noktası verilsin, bu sorgu noktasına en yakın veri noktasını bulma süreci en yakın komşuluk problemi olarak tanımlanır (Beyer vd., 1999). kNN algoritması ise bir sorgu noktasının bir veri noktası koleksiyonu içerisindeki en yakın k tane veri noktasına olan uzaklığı esasına dayanmaktadır. Uzaklıkların hesaplanmasında Euclidean, Manhattan, Chebyshev, Minkowski v.b. uzaklık formülleri kullanılmaktadır. Minkowski uzaklık formülünde m=1 ve m=2 için hesaplanan uzaklıklar sırasıyla Manhattan ve Euclidean uzaklığını vermektedir.

p ve q veri noktaları olmak üzere d(p, q) iki nokta arasındaki uzaklıktır. Buna göre;

$$d_{Euclidean}(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1)$$

$$d_{\text{Manhattan}}(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (2)$$

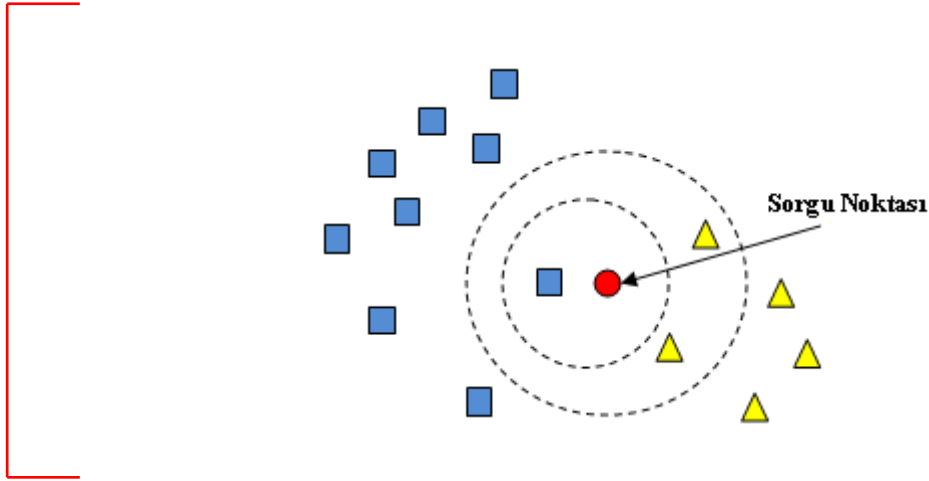
$$d_{\text{Chebyshev}}(p, q) = \max_i |p_i - q_i| \quad (3)$$

$$d_{\text{Minkowski}}(p, q) = \left(\sum_{i=1}^n |p_i - q_i|^m \right)^{1/m} \quad (4)$$

kNN’de sınıflandırma yapılırken veritabanındaki her bir kaydın diğer kayıtlarla olan uzaklığı hesaplanır. Ancak, bir kayıt için diğer kayıtlardan sadece k adedi göz önüne alınır. Bu k adet kayıt, başka bir deyişle veritabanındaki nokta, mesafesi hesaplanan noktaya diğer kayıtlara nazaran en yakın olan kayıtlardır (Silahtaroglu, 2008). Bu yöntem coğrafi bilgi sistemlerinde çok kullanılır; belirlenen bir noktaya en yakın şehir, istasyon vs belirlenmesi aslında kNN algoritmasının temelini oluşturur (Beyer vd., 1999).

Algoritmada k değeri önceden seçilir; değerinin yüksek olması birbirine benzemeyen noktaların bir araya toplanmasına, çok küçük seçilmesiyle birbirine benzediği, yani aynı sınıfın noktaları oldukları halde, bazı noktaların ayrı sınıflara konmasına ya da o tür noktalar için ayrı sınıfların açılmasına neden olur (Silahtaroglu, 2008). Tipik k değerleri 3, 5 ve 7’dir (Maleq vd., 2002).

kNN algoritması için uygun k değerinin belirlenmesi çok önemlidir; çünkü k değerine göre sorgu noktasının hangi sınıfa ait olacağı değişebilmektedir. Bu durum Şekil 5-1’de gösterilmektedir. k=1 olması durumunda sorgu noktası “kare” sınıfına atanır. Yalnız k değerinin 3 olarak belirlenmesi durumunda ise sorgu noktası “üçgen” sınıfına atanır. k değerinin gittikçe daha büyümesi ise çeşitli sorunlara sebep olacaktır. Bu sorunlardan bir tanesi de sınıflardan herhangi biri için yeterli eğitim verisi bulunmuyorsa yani sınıflar arasındaki eğitim verilerinde bir dengesizlik söz konusu ise bu durumda test örneğinin yanlış bir sınıfa atanması söz konusu olabilir.



Şekil 5-1. k değerinin sınıflandırıcının tahminine etkisi

Eğitim verisi içerisinde bir sorgu noktasının veri noktalarına olan uzaklığı ortalama uzaklıktan çok farklı değilse o zaman en yakın komşuluk yararlı olmayabilir (Beyer vd., 1999). Bu durumun nedeni; sorgu noktasının, eğitim verilerini sınıflandırmak için kullanılan sınıfların ya da etiketlerin dışında bir etikete ya da sınıfa ait olması olasılığıdır. Yani eğitim verilerinin sınıflandırılması esnasında kullanılmayan sınıflar olabilir.

kNN yukarıda anlatıldığı gibi verilen bir gözleme en yakın k komşunun belirlenmesi ve sınıfı bilinmeyen yeni bir gözlem değeri için, bu k gözlem içindeki en fazla tekrar eden sınıfın seçilmesi esasına dayanıyordu. Ancak seçilen bu sınıf, sadece k komşunun göz önüne alınması nedeniyle her zaman uygun olmayabilir. Bu son aşamada k komşu arasında en çok tekrarlanan sınıfı seçme yöntemi yerine ağırlıklı oylama adı verilen bir yöntem uygulanabilir (Özkan, 2008).

Ağırlıklı oylama yöntemi gözlem değerleri için aşağıdaki bağıntıya göre ağırlıklı uzaklıkların hesaplanmasına dayanır:

$$d(i,j) = \frac{1}{d(i,j)} \quad (5)$$

Burada yer alan $d(i,j)$ ifadesi i ve j gözlemleri arasındaki Öklid uzaklığıdır. Her bir sınıf değeri için bu uzaklıkların toplamı hesaplanarak ağırlıklı oylama değeri elde

edilir. En büyük ağırlıklı oylama değerine sahip olan sınıf değeri yeni gözlemin ait olduğu sınıf olarak kabul edilir (Özkan, 2008).

kNN gibi nonlineer metotlar düşük bias'a sahiptir. Buna karşılık yüksek varyansa sahiptir (Manning, 2008). Bir sınıflandırıcının bias'ı, sınıflandırıcının yapmış olduğu tahmin ile verinin gerçek sınıfı arasındaki fark olarak isimlendirilir. Bu farkın çok fazla olması sınıflandırıcının isabetli tahminler yapamadığını göstermektedir. Bir sınıflandırıcının variance'sı ise sınıflandırıcının yaptığı tahminlerdeki değişim olarak ifade edilir. Yani bir sorgu noktası için yapılan tahminlerin birbirlerinden farklı olmasıdır.

5.5. k-Katlı Çapraz Doğrulama

Eğitim verisinin bir sınıflandırıcı tarafından öğrenilmesi için yeterli olup olmadığını belirlemenin farklı yolları vardır. Bu yollardan bir tanesi de k-Katlı Çapraz Doğrulama'dır. k-Katlı Çapraz Doğrulama (CV) verinin yaklaşık olarak k eşit parçaya bölünerek bir sınıflandırıcının doğruluğunu tahmin etmek için bir yöntem sunar. Böylelikle sınıflandırıcı k defa test edilir ve eğitilir. Her zaman sınıflandırıcı veri kümesi üzerinde bir eksik katla eğitilir (Glossary of Terms, 1998). Yani k alt parçadan k-1 i eğitim için kullanılır. Geri kalan kat ise test verisi olarak kullanılır (Alpaydın, 2004). Sınıflandırıcının doğruluk tahmini ise k adet kat için ortalama doğruluktur (Glossary of Terms, 1998).

CV kullanılarak eğitim verileri için veri miktarında sınırlamalar yapılabilir (Witten vd., 2005). Böylelikle gereksiz miktarda verinin eğitim verisi içerisinden ayıklanması sağlanabilmektedir. Bu da kNN gibi hafıza tabanlı öğrenme (MBL) algoritmaları için performans kazancı sağlayacaktır. CV tekniğini kullanarak bias-variance ya da underfitting-overfitting değişimi dengede tutulabilmektedir (Mitchell, 1997).

CV tekniği hesaplamasal olarak çok maliyetlidir. Ancak sınıflandırıcının hata

oranını tahmin etmek istediğimiz zaman çok yararlı bir metottur (Nilsson, 1996). Aynı zamanda sınırlı veriye sahip iki öğrenme algoritmasını birbiriyle kıyaslamak için de CV kullanılır (Mitchell, 1997). Eğer bir sınıflandırıcının genelleme hatasını tahmin etmek için elde yeterince veri varsa bu verinin %50'si eğitim verisi, %25'i doğrulama verisi ve verinin geriye kalan %25'i ise test verisi olarak kullanılabilir. Bunun yanı sıra ML alanında doğrulama işlemi bir sınıflandırıcının parametrelerini ayarlamak için kullanılır (Witten vd., 2005). Bu da sınıflandırıcının genelleme yeteneğini test etmek için kullanılır (Alpaydın, 2004).

CV de önemli bir sorunda eğitim kümesi ile test kümesine dağılmış olan örneklerin sınıfları arasındaki dengesizliktir (Witten vd., 2005). Yani belirli bir sınıfa ait olan örneklerin hepsi ya da büyük çoğunluğunun test kümesine geri kalanların ise eğitim kümesine atanması problemidir. Bu sınıf dengesizliği veri kümesinden de kaynaklanabilir. Yalnız bu durum CV ile ilgili bir problem değildir. Bu sınıf dengesizliğinden dolayı sınıflandırıcı tam olarak bir öğrenme gerçekleştirmez. Bundan dolayı sınıflandırıcının hata oranları yüksek çıkabilir. Bu problemi çözmek için ise her bir sınıfın hem eğitim kümesi hem de test kümesi içerisinde dengeli bir biçimde dağılması gerekmektedir. Bu durum problemin çözümünü garanti etmektedir. Problemin çözümünde uygulanan bu süreç "stratification" olarak adlandırılır. "Stratification" eğitim ve test kümelerindeki dengesiz temsil edilmeye karşı öncü bir koruma sağlar (Witten vd., 2005). WEKA (<http://www.cs.waikato.ac.nz>), CV tekniği sırasında "stratification" sürecini uygular. Bu da güvenilir sonuçlar verir (Bouckaert vd., 2008).

Değişmeyen belirli bir veri örneği üzerinde bir öğrenme tekniğinin hata oranını tahmin etmenin standart yolu stratified 10-Katlı CV kullanmaktır. Bu durumda veri tüm veri kümesi üzerinde yaklaşık olarak aynı oranlarda temsil edilen sınıflara ait rastgele 10 parçaya bölünür. Herbir parça sırayla döndürülür. Ve parçaların onda dokuzu eğitim verisi olarak kullanılır. Daha sonra hata oranı dışarıda tutulan parça üzerinde hesaplanır. Böylece öğrenme süreci farklı eğitim kümeleri üzerinde toplam 10 defa yürütülür. Sonuç olarak, 10 hata tahmini bütün bir hata tahminini bulmak için ortalaması alınır (Witten vd., 2005).

k değerinin niçin 10 alındığına gelince: farklı öğrenme teknikleri kullanılarak yapılan çeşitli veri kümeleri üzerindeki yaygın testlerde hata oranının en doğru tahminini yapmak için doğru sayının 10 civarında olduğu görülmektedir. Aynı zamanda teorik kanıtlar da bu durumu desteklemektedir (Witten vd., 2005).

5.6. Özellik Seçimi

ML algoritmalarının karar alma süreçlerini önemli ölçüde etkileyen faktörlerden biri de niteliklerin uygun olup olmadığıdır. İyi bir nitelik altkümesi seçildiği zaman temel olarak iki yaklaşım vardır. Birinci yaklaşım verinin genel karakteristiğine dayalı bağımsız bir değerlendirme yapmaktır; diğer yaklaşım ise öğrenme işlevi için kullanılan ML algoritmasını kullanarak nitelik altkümesini değerlendirmektir (Witten vd., 2005).

İlk yaklaşım kullanılarak öğrenme aşamasına başlanmadan önce öğrenmeye dayalı sistemin etkin kararlar alması için iyi bir öğrenme gerçekleştirmeye yardımcı olabilecek nitelikler belirlendi.

İkinci yaklaşımı kullanarak aynı zamanda niteliklerin seçiminde fayda teorisi (Parmigiani, 2009) yaklaşımı da kullanılmış oldu. Fayda Teorisi (UT) bir durum hakkında kararsız kaldığımız zaman rasyonel kararlar verme üzerine dayanır (Alpaydın, 2004). Bunu şu şekilde ifade edersek: daha önce gözlemlediğimiz nitelik x olsun ve S durumu ayrıntılı durumlardan oluşsun; $S_k, k = 1, \dots, n$. Buna göre x niteliğine ait olduğu bilinen S_k durumunun olasılığı $P(S_k|x)$ olarak hesaplanır. x niteliğini S_k durumuna atayan karar hareketi α_i ve fayda fonksiyonumuzu da U_{ik} olarak tanımlayalım. Böylece beklenen fayda:

$$EU(\alpha_i|x) = \sum_k U_{ik} P(S_k|x) \quad (6)$$

Beklenen faydayı maksimize eden hareket α_i olduğunu varsayarsak x niteliği

için u_i hareketinin beklenen faydası (7) eşitliğindeki gibi gösterilir.

$$EU(x) = \max_i \sum_k U_{ik} P_k | x \quad (7)$$

Nitelik kümesine eklenen yeni bir niteliğimiz y olsun. Buna göre beklenen fayda (8) eşitliğindeki gibi gösterilir.

$$EU(x, y) = \max_i \sum_k U_{ik} P_k | x, y \quad (8)$$

Buna göre eğer $EU(x, y) > EU(x)$ ise y niteliğinin dikkate değer bir nitelik olduğu söylenebilir. Yani nitelik kümesine eklenen yeni niteliğin beklenen faydası önceki nitelik kümesinin beklenen faydasından büyük ise o zaman yeni niteliğin yararlı bir nitelik olduğu söylenebilir.

5.7. İstatiksel Değerlendirme Ölçütleri

Bir sınıflandırıcının öğrenme sürecinin sonucunda ne kadar başarılı olduğuna karar verilmesi için çeşitli ölçütlere gereksinim vardır. Bu tez çalışmasına konu olan öğrenme sisteminin başarımını ölçmek için kullanılan ölçütler şunlardır: Kappa İstatistiği, F-Ölçütü değeri ve Ortalama Kareysel Hatanın Karekökü'dür. Bu ölçütler aşağıda açıklanmaktadır.

5.7.1. Kappa istatistiği

Kappa İstatistiği (KS) bir veri kümesindeki tahmin edilen ve gözlenen sınıflandırmalar arasındaki uyumu ölçmek için kullanılan bir ölçüttür (Witten vd., 2005). KS ilk Cohen (1960) (Cohen, 1960) tarafından ortaya konuldu (Witten vd., 2005). Cohen'e göre KS, iki gözlemci arasındaki uyum derecesini

ölçmede kullanılan bir ölçüttür (Cohen, 1960). KS o zamandan beri birkaç disiplinde kullanıldı (Kılıçaslan vd., 2009). ML alanı da bu disiplinlerden bir tanesidir. KS, ML alanında bir sınıflandırıcının doğruluğunu arttırmaya yardımcı bir ölçüt olarak kullanılmaktadır (Kılıçaslan vd., 2009). KS şu şekilde tanımlanmaktadır:

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \quad (9)$$

$\Pr(a)$ sınıflandırıcının doğruluğunu göstermektedir ve $\Pr(e)$ aynı veri kümesi üzerinde rastgele bir biçimde tahmin yapan sınıflandırıcının elde etmiş olduğu beklenen doğruluktur (Kılıçaslan vd., 2009). Yani yapılan tahminin ortaya çıkma olasılığıdır. Cohen'in KS'sine getirilen en büyük eleştiri bazen sezi ile beklenenden değişik sonuçlar vermesidir (Gwet, 2010). Yani iki sınıftan oluşmuş ve eşit büyüklükteki iki veri kümesi üzerinde yapılan öğrenme işlemi sonucunda sınıflandırıcının yapmış olduğu doğruluk oranları eşit olmasına rağmen KS değeri iki veri kümesi için de farklı çıkabilmektedir.

KS değeri -1 ve 1 arasında değer almaktadır. -1 değeri tümüyle bir uyumsuzluğu ya da ters yönde bir ilişki olduğunu göstermektedir. 1 ise mükemmel uyumu göstermektedir (Kılıçaslan vd., 2009). KS 0.4 ya da üzeri bir değere sahip ise şansın ötesinde kabul edilebilir bir uyumluluktan söz edilebilir (Landis vd., 1977). Landis ve Koch (1977) tarafından sunulan Kappa tablosu aşağıda gösterilmektedir.

Çizelge 5-1. Landis ve Koch'un Kappa değerleri tablosu.

Kappa	Yorum
< 0	Uyuşma yok
0.0 – 0.20	Önemsiz uyuşma
0.21 – 0.40	Orta derecede uyuşma
0.41 – 0.60	Kabul edilebilir uyuşma
0.61 – 0.80	Önemli derecede uyuşma
0.81 – 1.00	Mükemmel uyuşma

5.7.2. F-ölçütü

F-Ölçütü (F-Measure) değeri precision ve recall değerlerinin harmonik ortalaması olarak ifade edilir (Kılıçaslan vd., 2009). Pozitif ve negatif olarak sınıflandırılan örnekler için precision, pozitif olarak tahmin edilen örnekler içerisinde doğru olarak sınıflandırılan pozitiflerin oranını verir. Recall ise gerçek sınıfı pozitif olan örnekler içerisinde doğru olarak sınıflandırılan pozitif örneklerin oranını verir. Tablo 5-2’de iki sınıflı bir veri kümesinin doğruluk tablosu gösterilmektedir.

Çizelge 5-2. İki sınıflı bir tahmin için doğruluk tablosu

		Tahmini Sınıf	
		Evet	Hayır
Gerçek Sınıf	Evet	Doğru pozitif (TP)	Yanlış pozitif (FP)
	Hayır	Yanlış negatif (FN)	Doğru negatif (TN)

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F-Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

F-ölçütü özellikle eğitim verilerinin hazırlanış sürecinde sınıflandırıcının performansını arttırmak için önemli bir ölçüt olmaktadır. Bu açıdan Yılmaz Kılıçaslan’ın (2008) sınıflandırıcıların performans analizinde f-ölçütü ve veri büyüklüğü arasındaki ilişkiyi ortaya koyduğu çalışma önemlidir (Kılıçaslan vd., 2009). Bu teze konu olan çalışmada da f-ölçütü ve veri büyüklüğü arasında çizilen öğrenme eğrisinden

bir mutluluk grafiđi (Russell vd., 2003) elde edilmeye alıřılmıştır. Bylelikle gereksiz veriler ayıklanarak sınıflandırıcının performansı artırılmaya alıřılmıştır. Bunun yanı sıra geliştirilen uygulama da F-ölütü sınıfların tanı için yeterli olup olmadığını tespit etmek amacıyla kullanılmıştır. Kabul edilebilir F-ölütü deđeri genellikle minimum 0.5 olarak alınır. Bu tez alıřmasında da bu deđer 0.5 olarak alınmıştır.

5.7.3. Hata ölütü: ortalama karesel hatanın karekökü

Ortalama Karesel Hatanın Karekökü (RMSE) Ortalama Karesel Hata deđerinin karekökünün alınmasıyla hesaplanır. Ortalama Karesel Hata (MSE) belli bařlı ve ok yaygın olarak kullanılan bir ölüttür. Birok matematiksel teknik (lineer regresyon gibi) matematiksel olarak işlenmesi en kolay ölüt olduđu için MSE'yi kullanır. ML'de kullanım alanı ise sınıflandırıcıların performans ölçümünü yapmaktır (Witten vd., 2005).

Bir tahmin edicinin hata oranı geliřigüzel bir tahmin yapmasından ya da dođru bir tahmin yapabilmesini sađlayacak bilgiyi yakalayamamasından dolayı meydana gelir (Lehmann vd., 1998). MSE ve RMSE deđerleri sifıra ne kadar yakınsa hata oranı o kadar azdır. Ayrıca her ğrenme için MSE ya da RMSE'nin kabul edilebilir hata deđerleri farklıdır.

MSE ve RMSE'nin hesaplanmaları ařađıda gösterilmektedir. Buna göre p, yapılan tahmini deđerleri; a ise gerek deđerleri ifade etmektedir.

$$MSE = \frac{\phi_1 - a_1)^2 + \dots + \phi_n - a_n)^2}{n} \quad (11)$$

$$RMSE = \sqrt{\frac{\phi_1 - a_1)^2 + \dots + \phi_n - a_n)^2}{n}} \quad (12)$$

5.8. Gürültü

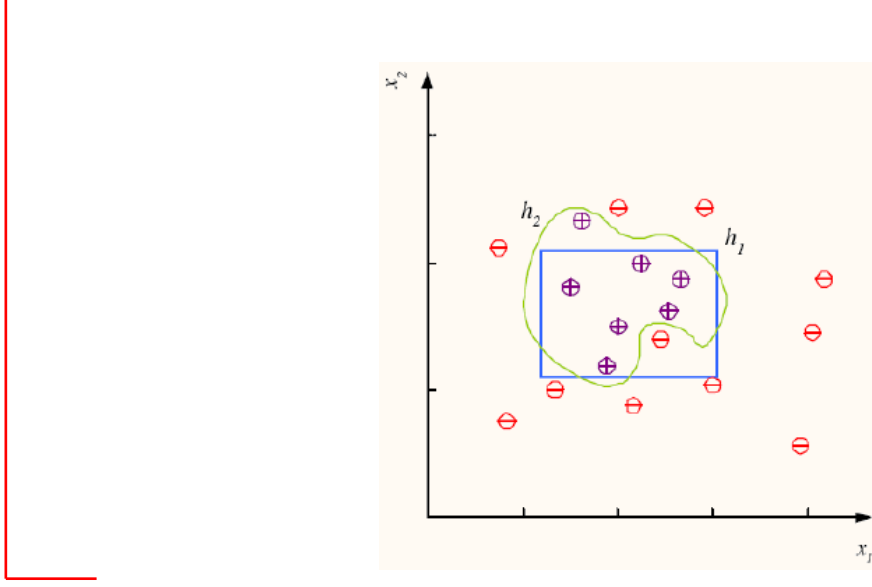
Gürültü, veride hiç istenmeyen bir anormalliktir. Verideki gürültüden dolayı, bir test verisinin sınıfını öğrenmek zor olabilir ve basit bir hipotez sınıfı için bile sıfır hata mümkün olmaz. Aynı zamanda, iki sınıflı bir veri kümesi için pozitif ve negatif örnekler arasında bile basit bir sınır yoktur (Alpaydın, 2004). Bu durum Şekil 5-2'de gösterilmektedir. Gürültünün birkaç nedeni vardır. Bunlar:

- Veri kümesinin hazırlanışından kaynaklanan sorunlar olabilir.
- Verikümesindeki bir örneğin sınıfını etkileyen ve göz önüne alınmayan ilave nitelikler olabilir. Bu gibi nitelikler gizli olabilir ve bu ihmal edilen niteliklerin etkisi gelişmiş güzel bir eleman gibi modellenir ve bu durumda veri içerisinde gürültü dâhil edilmiş olur (Alpaydın, 2004).

Şekil 5-2'deki durum incelendiğinde veri kümesi içerisinde gürültü olmasaydı pozitif ve negatif örnekleri birbirinden ayırmak için h_1 hipotezi yeterli olacaktı. Yalnız gürültüden dolayı öğrenme algoritması h_1 hipotezi yerine h_2 hipotezini oluşturdu. Bu durumda h_1 hipotezi gibi basit bir modelden h_2 hipotezi gibi kompleks bir modele geçildi. Kompleks bir model kullanılarak veriye mükemmel bir uygunluk sağlanır (Alpaydın, 2004; Schölkopf vd., 2002). Basit bir model kullanımında ise bazı hatalar oluşabilir (Alpaydın, 2004). Yalnız basit bir modeli kullanmanın bazı avantajları mevcuttur. Bunları şu şekilde özetleyebiliriz:

- Hesaplamasal komplekslik daha düşük olduğu için kullanımı daha kolaydır (Alpaydın, 2004).
- Uzay kompleksliği daha düşük olduğu için eğitmek daha kolaydır. Yani bir dikdörtgen ile tanımlanan bir uzayın köşelerini bulmak daha kolaydır. Böylelikle bu köşeler ile kontrol yapmak ve sistemi eğitmek daha kolay olur (Alpaydın, 2004).
- Basit bir modeli açıklamak kolaydır. Bundan dolayı eğitim kümesi içinde yer alan ham veriden enformasyon çıkartabiliriz (Alpaydın, 2004).

- Basit bir model daha az varyansa sahip olduğundan dolayı kompleks bir modele göre daha iyi genelleme gücüne sahiptir. Bu prensip Occam'ın usturası (<http://en.wikipedia.org>) olarak bilinir (Alpaydın, 2004).



Şekil 5-2. Bir verikümesinde gürültü

5.9. Bias-Varyans Ayrışımı

Bias-Varyans ayrışımı ML algoritmalarını anlamada anahtar bir araçtır. Son yıllarda deneysel çalışmalarda Bias-Varyans ayrışımının kullanımı gittikçe artmaktadır. Bias ve Varyans kavramları çok basit tahmin edicilerin çok karmaşık olanlara ne kadar üstün olabildiğini ve model topluluklarının basit modellere ne kadar üstün olabildiklerini açıklamaya yardım eder. Bias-Varyans ayrışımı diğer istatistiksel hata fonksiyonlarının yanı sıra karesel kayıplar için de türetilmiştir (Domingos, 2000). Bu türetme Geman, Bienenstock ve Doursat (Geman vd., 1992) tarafından yapılan bir çalışmada gösterildi. Bu gösterimde öğrenme algoritmasının ortalama hatası bias'ın karesi ile varyans'ın toplamı olarak ifade edilmektedir. Bu gösterimin dışında başka yazarlar bias-varyans değişimini farklı biçimde göstermektedirler. Bu gösterimlerden bir tanesi 0-1 kaybı için gösterilmiştir. Bu gösterimlerin önemli bir kısmı önemli

eksiklikler içermektedir (Domingos, 2000; James, 2003). Örneğin Kong and Dietterich'in (Kong vd., 1995) 0-1 kaybı için yaptığı ayrışım varyans'ın negatif çıkmasına neden olmakta ve gürültüden kaynaklanan sorunları dikkate almamaktadır (Domingos, 2000). Kayıp fonksiyonu (loss function) olarak karesel hatayı kullanmak çok uygundur. Çünkü karesel hata, onu kullanmayı çok cazip kılan bias-varyans ayrışımı gibi iyi bilinen matematiksel özelliklere sahiptir (James, 2003).

ML alanında bias ve varyans kavramlarını kısaca açıklayacak olursak ML alanında bias gerçek değer ile tahmini değer arasındaki fark; Varyans ise yapılan tahminlerin ortalama tahminden ne kadar saptığı olarak açıklanır. Şimdi ML alanında MSE'nin bias-varyans ayrışımı olarak ifadesini verelim. Yalnız bu işlemlerden önce istatistiksel olarak "beklenen değer" ifadesini açıklayalım. Çünkü matematiksel olarak MSE'nin ayrıştırılmasında "beklenen değer" oldukça sık kullanılmaktadır.

S bir deneydeki örneklem uzayı olarak varsayalım. Deneyin sonuçları yani S nin örneklem noktaları sayı olmayabilir. Bununla beraber çoğu zaman her sonuca belirli bir sayı bağlanır. Örneğin bir çift zardaki sayılar toplamı, briçteki asların sayıları v.b. Böyle sayılara raslantı değişkenleri ya da rassal değişken denir. Buna göre X, sonlu görüntü kümeli örneklem uzayındaki $X(S) = (x_1, x_2, \dots, x_n)$ gibi bir rassal değişken olsun. X(S) olasılık uzayında, x_i 'nin olasılığı $f(x_i)$ biçiminde yazılan $P(X=x_i)$ olarak tanımlanarak yapılır. X(S)'deki bu f fonksiyonu, yani $f(x_i) = P(X=x_i)$ olarak tanımlanan fonksiyona X'in dağılım ya da olasılık fonksiyonu denir (Lipschutz, 1990). Buna göre f dağılımı için;

$$(i) \quad f(x_i) \geq 0$$

$$(ii) \quad \sum_{i=1}^n f(x_i) = 1 \quad \text{Koşullarını sağlar.}$$

Şimdi X yukarıdaki dağılımı olan bir rassal değişken ise X'in $E(X)$, E , μ_x ya da μ biçiminde gösterilen ortalama ya da beklenen değeri

$$E(X) = x_1 f(x_1) + x_2 f(x_2) + \dots + x_n f(x_n) = \sum_{i=1}^n x_i f(x_i) \quad (13)$$

Olarak tanımlanır. Kısacası $E(X)$ bir rassal değişkenin alabileceği bütün değerlerin, olasılıklarıyla çarpılması ve bu işlemin bütün değerler üzerinden toplanmasıyla elde edilen değerdir. Ayrıca X 'in olası değerlerinin ağırlıklı ortalaması da denilmektedir. Beklenen değer çeşitli özellikleri vardır. Bunlar aşağıda gösterilmektedir.

- Sabit sayıların beklenen değeri kendisine eşittir, yani a bir sabit ise $E(a) = a$ 'dır.
- a ve b sabit sayılar ve X bir değişken olmak üzere $E(aX) = aE(X)$ ve $E(aX + b) = aE(X) + b$ olur.
- a sabit bir sayı, X bir değişken ve $Y=U(X)$ olarak tanımlanmış bir başka değişken (X değişkeninin bir fonksiyonu) ise $E(aY) = E(aU(X)) = aE(U(X))$ dir.

Diğer önemli bir konu da beklenen değer koşullu olma durumudur. Bu durumda koşullu dağılımlardan elde edilen beklenen değere koşullu beklenen değer denir. X değişkeninin A gibi olaya göre koşullu beklenen değeri $E(X|A)$ biçiminde gösterilir ve aşağıdaki gibi tanımlanır:

$$E(X | A) = \sum_{x \in A} x P(x | A) \quad (14)$$

(14) eşitliğinde tanımlanan $P(x | A)$ koşullu olasılığı şu şekilde tanımlanmaktadır.

$$P(x | A) = \frac{P(x)}{P(A)} \quad (15)$$

Bu koşullu olasılık bize, A olayı bilindiği takdirde x 'in olma olasılığını verir ve x 'in olasılığının hesaplanmasında ek bilgi verir.

Şimdi bir tahmin edicinin MSE'sinin bias-varyans ayrışımını inceleyelim. Buna göre $T = \{(x_1, a_1), (x_2, a_2), \dots, (x_N, a_N)\}$ şeklinde bir eğitim kümesi verilmiş olsun. Bu eğitim kümesinin tahmin edicisi $p=g(x)$ olsun ve bu tahmin edici $f=f(x)$ şeklinde tanımlanan gerçek fonksiyona yaklaştırmaya çalışsın. Eğitim kümesi T için gürültü olduğu varsayılarak çıkışlar $a = f + \varepsilon$ şeklinde farzedilmekte olup $E[\varepsilon] = 0$ 'dır. Çünkü beklenen

değerin birinci özelliğine göre sabit değerlerin beklenen değeri sıfırdır. Böylece bu tahmin edicinin MSE'si aşağıdaki gibi tanımlanır.

$$MSE = \frac{1}{N} \sum_{i=1}^N (q_i - p_i)^2 \quad (16)$$

Buna göre tahmin edicinin tahmin başarımını değerlendirmek için bir $f(x)$ 'den oluşan bir test örneğini test ederek $g(x)$ için beklenen MSE değerini elde edebiliriz. Beklenen MSE değeri aşağıdaki gibi tanımlanmaktadır.

$$E[MSE] = E\left[\frac{1}{N} \sum_{i=1}^N (q_i - p_i)^2\right] = \frac{1}{N} \sum_{i=1}^N E[(q_i - p_i)^2] \quad (17)$$

Şimdi (17) eşitliğinde tanımlanan eşitliğin sağ tarafına eşitliği bozmayacak şekilde eklemeler yapalım. Buna göre:

$$\begin{aligned} E[(q_i - p_i)^2] &= E[(q_i - f_i + f_i - p_i)^2] \\ &= E[(q_i - f_i)^2] + E[(f_i - p_i)^2] + 2E[(q_i - p_i)(f_i - f_i)] \\ &= E[\varepsilon^2] + E[(f_i - p_i)^2] + 2\left(E[f_i a_i] - E[f_i^2] - E[f_i a_i] + E[f_i f_i]\right) \quad (18) \\ &= E[\varepsilon^2] + E[(f_i - p_i)^2] + 0 \\ &= E[\varepsilon^2] + E[(f_i - p_i)^2] \end{aligned}$$

Yukarıdaki eşitlikte son terim sonuç olarak sıfıra eşittir. Bunu göstermek için her bir terimin sonucunu hesaplayalım. Buna göre:

- $E[f_i a_i] = E[f_i] \cdot E[a_i]$ dir ve $E[a_i] = f_i$ olduğundan dolayı sonuç $E[f_i^2]$ olarak bulunur. Ayrıca $f(x)$ fonksiyonu kararlı olduğu için $E[f_i a_i] = f_i^2$ şeklinde hesaplanır.
- $f(x)$ fonksiyonu kararlı olduğu için $E[f_i^2] = f_i^2$ şeklinde hesaplanır.

- Eğitim verisinde gürültü olduğu varsayıldığından dolayı gerçek değerler $a = f + \varepsilon$ şeklinde hesaplanmaktadır. Ayrıca $E[\varepsilon] = 0$ 'dır. Buna göre $E[p_i a_i]$ şu şekilde yazılabilir: $E[p_i(f_i + \varepsilon)] = E[p_i f_i + p_i \varepsilon] = E[p_i f_i] + 0 = E[p_i f_i]$ 'dir.

Bu durumda son terim aşağıdaki gibi sıfır olarak hesaplanır:

$$\begin{aligned} 2 \left(E[p_i a_i] - E[f_i^2] - E[p_i a_i] + E[p_i f_i] \right) &= 2 \left(E[f_i^2] - E[f_i^2] - E[p_i f_i] + E[p_i f_i] \right) \\ &= 2 \cdot 0 \\ &= 0 \end{aligned} \quad (19)$$

Böylelikle MSE, verideki gürültüden kaynaklanan varyans ve gerçek fonksiyon ile tahmin edilen değerler arasındaki MSE olarak ayrıştırılabilir. Şimdi (18) eşitliğindeki sonucu oluşturan ifadelerden son terimi alalım: $E[(f_i - p_i)^2]$. (18) eşitliğinde yaptığımız ekleme işlemlerinin bir benzerini bu ifade için de uygulayalım. Buna göre:

$$\begin{aligned} E[(f_i - p_i)^2] &= E[(f_i - E[p_i] + E[p_i] - p_i)^2] \\ &= E[(f_i - E[p_i])^2] + E[(E[p_i] - p_i)^2] + 2E[(f_i - E[p_i])(E[p_i] - p_i)] \\ &= bias^2 + Var(f_i) + 2 \left(E[f_i E[p_i]] - E[f_i] E[p_i] - E[E[p_i]^2] + E[E[p_i]] E[p_i] \right) \\ &= bias^2 + Var(f_i) + 0 \\ &= bias^2 + Var(f_i) \end{aligned} \quad (20)$$

Yukarıdaki eşitlikte son terim yine sıfır olarak hesaplanmaktadır. Bunu göstermek için son terimin her bir teriminin sonucunu hesaplayalım. Buna göre:

- $f(x)$ kararlı bir fonksiyon ve $E[E[z]] = z$ olduğu için $E[f_i E[p_i]] = f_i E[p_i]$ olarak hesaplanır.
- $E[E[p_i]^2] = E[p_i]^2$ olarak hesaplanır.

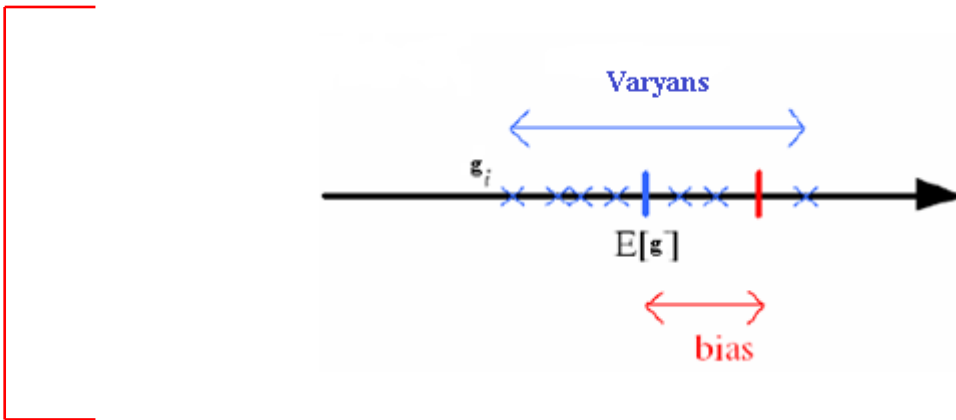
- $E[p_i f_i] = f_i E[p_i]$ olarak hesaplanır.
- $E[p_i E[p_i]] = E[p_i]^2$ olarak hesaplanır.

Bu durumda son terim aşağıdaki gibi sıfır olarak hesaplanır:

$$\begin{aligned}
 2 \left(E[p_i] E[p_i] - E[p_i p_i] - E \left[E[p_i]^2 + E[p_i] E[p_i] \right] \right) &= 2 \left(f_i E[p_i] - f_i E[p_i] - E[p_i]^2 + E[p_i]^2 \right) \\
 &= 2 \left(0 \right) \\
 &= 0
 \end{aligned} \tag{21}$$

Sonuç olarak beklenen MSE'nin ayrıştırılmasıyla aşağıdaki yapı oluşur. Ayrıca $g(x)$ tahmin edicisinin yaptığı tahminler, bias ve varyans Şekil 5-3'te gösterilmektedir.

$$E \left[\left(g_i - p_i \right)^2 \right] = \text{Var} \left(\text{Gürültü} \right) + \text{bias}^2 + \text{Var} \left(g_i \right) \tag{22}$$



Şekil 5-3. g tahmin edicisinin bias ve varyansı

(22) eşitliğindeki son duruma göre Gürültünün varyansı minimize edilemez (James, 2003; Vijayakumar, 2011). Çünkü Gürültü tahmin ediciden bağımsızdır. MSE'yi minimize etmek için bias ve varyansı azaltmak gerekmektedir. Ancak bunu yapmak önemli değildir. Çünkü giriş verilerini önemsemeyip bir şekilde çıkışı tahmin etmeye çalışmak kesinlikle yapılan tahminlerin varyansını minimize edecektir. Ancak yapılan tahminlerin bias'ı son derece fazla olacaktır. Genellikle uygun bias-varyans

değişimini bulmak zordur. Ancak CV gibi yollarla uygun çözümler bulunabilir (Vijayakumar, 2011).

Şimdi yukarıda tanımladığımız bias-varyans ayrışımını biraz daha ayrıntılı inceleyip tahmin edicinin bias ve varyansını eğitim kümesinin değişimine göre yorumlayalım. Böylelikle eğitim kümesi ile tahmin edici arasındaki ilişki iyi bir biçimde analiz edilebilir. Buna göre bilinmeyen ortak bir olasılık dağılımından oluşan bir $X=\{x^t, r^t\}, t=1, \dots, N$ örneği olsun. Bu örneği kullanarak $g(x)$ tahminlerimizi oluşturalım. Buna göre beklenen karesel hata aşağıdaki gibi hesaplanır:

$$E\left[\left(r - g\left(\bar{x}\right)\right)^2 \mid x\right] = E\left[\left(r - E\left[r \mid x\right]\right)^2 \mid x\right] + E\left[\left(E\left[r \mid x\right] - g\left(\bar{x}\right)\right)^2\right] \quad (23)$$

Gürültü *Karesel Hata*

(23) eşitliğinde sağ taraftaki ilk terim verilen x 'in çıkışının varyansıdır. Yani r 'nin varyansıdır. Bu durum eğitim verisinde gürültü olarak adlandırılır ve bu durum $g(x)$ 'ten bağımsızdır. Bundan dolayı hatanın bu kısmı kullandığımız tahmin edici ne olursa olsun azaltılamaz (Alpaydın, 2004). İkinci kısım ise $g(x)$ 'in $f(x)$ 'ten ne kadar saptığını gösteren kısımdır. Hatanın bu kısmı eğitim kümesi ve tahmin ediciye bağımlıdır (Alpaydın, 2004). Bu durumda bir örnek için tahmin edici yani $g(x)$ çok iyi uygunlukta olabilir. Ancak diğer bazı örnekler için ise $g(x)$ çok kötü bir uygunluk gösterebilir. Bu durumda $g(x)$ tahmin edicisinin ne kadar iyi tahminler yaptığını değerlendirmek için olası veri kümelerinin ortalamasını almak gerekmektedir (Alpaydın, 2004). Buna göre aynı olasılık yoğunluk fonksiyonu ile oluşan N elemanlı bir X örneğinin ortalama hatası ya da hatanın beklenen değerinin ayrışımı aşağıdaki gibidir:

$$E_x\left[\left(r - g\left(\bar{x}\right)\right)^2 \mid x\right] = \underbrace{E_x\left[\left(r - E_x\left[r \mid x\right]\right)^2\right]}_{bias} + \underbrace{E_x\left[\left(E_x\left[r \mid x\right] - g\left(\bar{x}\right)\right)^2\right]}_{var\ yans} \quad (24)$$

Buna göre değişen örnekler için bias ve varyansı tahmin etmeye çalışalım. Bunun için $X_i=\{x_i^t, r_i^t\}, i=1, \dots, M$ gibi bir veri kümesi oluşturalım. Bu veri kümesinin gerçek fonksiyonu $f(x)$ olsun ve değişen her bir örneğimiz için tahmin fonksiyonumuz $g_i(x)$

olsun. Buna göre yapılan ortalama tahmin fonksiyonu ya da beklenen tahmin fonksiyonu $E[g(x)]$ aşağıdaki gibi hesaplanır:

$$E[\hat{g}(x)] = \frac{1}{M} \sum_{i=1}^M g_i(x) \quad (25)$$

Bu durumda bias ve varyans aşağıdaki gibi hesaplanır:

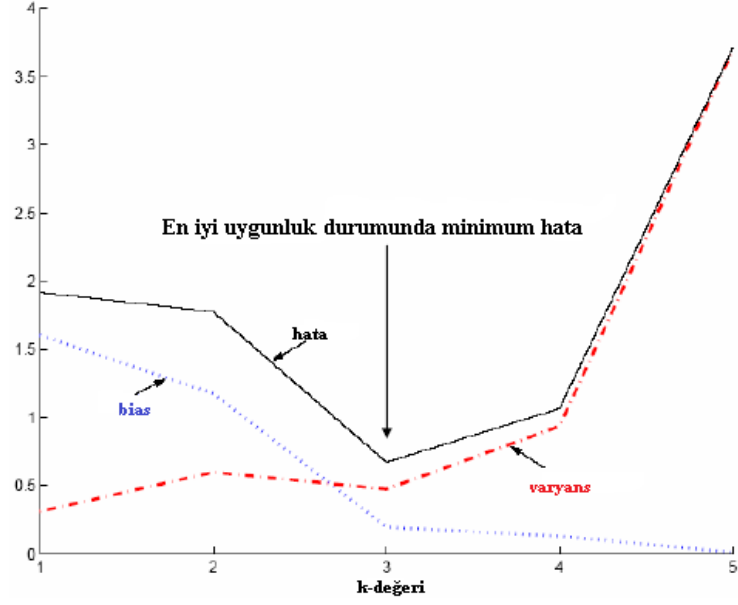
$$bias(\hat{g})^2 = \frac{1}{N} \sum_t \left[\bar{g}(x^t) - f(x^t) \right]^2 \quad (26)$$

$$var\ yans(\hat{g}) = \frac{1}{NM} \sum_t \sum_i \left[g_i(x^t) - \bar{g}(x^t) \right]^2 \quad (27)$$

Şeklinde hesaplanır. Buna göre bias ve varyans hesaplamalarına göre aşağıdaki yorumları yapabiliriz:

- Tahmin fonksiyonu gerçek fonksiyondan daha basit ise bu durumda bias artar. Eğer tahmin fonksiyonu gerçek fonksiyondan daha kompleks ise bu durumda da tahmin edicinin varyansı artar.
- Tahmin fonksiyonunun bias'ı arttığı zaman varyans'ı düşer, varyans'ı arttığı zaman da bias'ı düşer. Bu durum bias-varyans ikilemi olarak isimlendirilir ve bias-varyans ikilemi herhangi bir makine öğrenmesi sistemi için sözkonusudur (Geman vd., 1992). Bundan dolayı uygun bias-varyans değişimini bulmak zordur (Vijayakumar, 2011). Bunun için eğitim kümesinin değişimine bağlı olarak MSE grafiği oluşturulur ve bir mutluluk grafiği (Russell vd., 2003; Kılıçaslan vd., 2009) elde edilmeye çalışılarak uygun bias-varyans değişimi yakalanabilir. MSE, bias ve varyansın birbirlerine göre durumunu gösteren grafik Şekil 5-4'te gösterilmektedir.
- Eğer tahmin edici de bias varsa, model sınıf çözümü içermez. Buna underfitting denir. Eğer varyans varsa, model sınıfın genelleme gücü uygunluğun üzerinde olur ve verideki gürültüleri de öğrenir. Buna da overfitting denir. Bu durumda

sınıfı belli olan bir test verisi fazla genellemeden dolayı farklı bir sınıfa atanabilir.



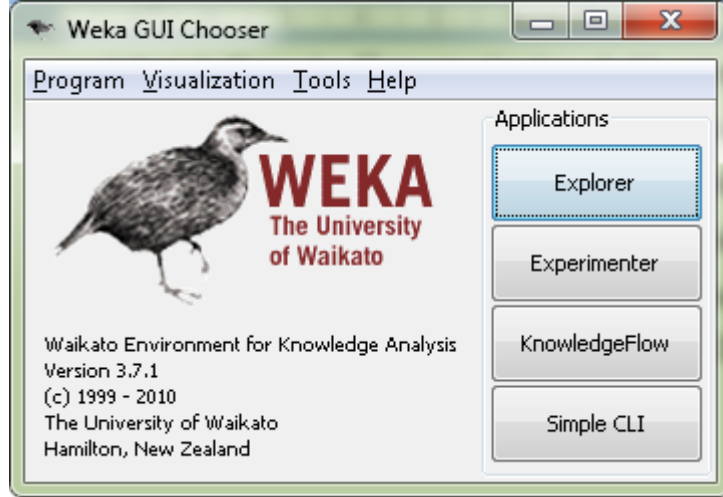
Şekil 5-4. kNN sınıflandırıcının k-değerine bağlı olarak MSE, bias ve varyans değişiminin grafiği

6. WEKA

WEKA, verikümelerine kolaylıkla uygulanabilen öğrenme algoritmalarını içeren bir araçtır. WEKA, Yeni Zellanda'da Waikato Üniversitesinde geliştirildi ve ismi "*Waikato Environment for Knowledge Analysis*" sözcüğünden gelmektedir. WEKA ismi aynı zamanda sadece Yeni Zellanda da bulunan ve nesli tükenmiş uçamayan bir kuşun da adından esinlenilmiştir. WEKA, Java programlama dili ile geliştirilmiş ve Genel Açık Lisans ile dağıtımı yapılmaktadır. WEKA; Linux, Windows ve Macintosh işletim sistemleri altında test edilmiş ve herhangi bir platformda çalışabilmektedir. WEKA ile verikümelerini işleyebilir, sınıflandırıcı sonuçlarını analiz edebilir ve herhangi bir kod yazmadan WEKA'nın performansından faydalanabiliriz (Witten vd., 2005).

WEKA tüm veri madenciliği problemleri için metotlar içerir. Bu metotlar şunlardır: Regresyon, sınıflandırma, kümeleme, birliktelik kuralları oluşturma ve nitelik seçimi. WEKA'daki tüm algoritmalar basit ilişkisel tablo şeklinde olan ARFF dosya formatını giriş olarak alabilmektedir.

WEKA'yı kullanarak ilk olarak bir verikümesine öğrenme metotlarını uygulayabilir ve veri hakkında daha fazla şey öğrenmek için onun çıkışını analiz edebiliriz. İkinci olarak yeni gelen örnekler üzerinde tahminler yapmak için öğrenilmiş modelleri kullanmak. Üçüncü olarakta farklı öğrenme algoritmalarını uygulamak ve onların yapmış oldukları tahminleri birbiriyle karşılaştırmak için kullanılmaktadır. Öğrenme metotları sınıflandırıcı olarakta isimlendirilmektedir. Etkileşimli WEKA arayüzünden istenilen bir sınıflandırıcı kolay bir biçimde seçilebilmektedir. WEKA içerisinde dört temel uygulamaya sahiptir. Bu uygulamalar şunlardır: Explorer, Experimenter, KnowledgeFlow ve Simple CLI. Şekil 6-1'de Wekanın arayüzü görülmektedir.



Şekil 6-1. WEKA grafik arayüzü

6.1. Temel Kavramlar

6.1.1. Verikümesi

Veri elemanlarının bir topluluğu olan verikümesi makine öğrenmesinin çok temel bir kavramıdır. Verikümesi iki boyutlu bir tablo ya da veritabanı tablosuna eşittir. WEKA’da verikümesi `weka.core.Instances` sınıfı ile uygulanır. Bir verikümesi `weka.core.Instance` sınıfının her bir örneklerin bir koleksiyonudur. Her bir örnek bir çok nitelikten oluşur. Bu değerler nominal (değerler listesi biçiminde), nümerik (reel ya da tamsayı) ya da string (karakter katarı) olabilirler. Bu türlere ilaveten date ve relational veri tipleri de mevcuttur. Bir örnek sınıfın harici temsili bir ARFF dosyasıdır (Bouckaert vd., 2008). ARFF dosyasının yapısı üç kısımdan oluşmaktadır. Bunlar: relation, attribute ve data’dır. Şekil 6-2’de bir hastanın örnek ARFF dosyasından bir kesit gösterilmektedir.

```

%Verikümesi ile ilişkili uygun bir isim veriliyor
@relation Arrhythmia

%Verikümesinin nitelikleri tanımlanıyor
@attribute HS1 integer
@attribute HS2 integer
@attribute HS3 integer
@attribute HS4 integer
@attribute HS5 integer
@attribute HS6 integer
@attribute HS7 integer
@attribute HS8 integer
@attribute HS9 integer
.
.
.
.
.
@attribute HS117 integer
@attribute HS118 integer
@attribute HS119 integer
@attribute HS120 integer
@attribute HS121 integer
@attribute HS122 integer
@attribute HS123 integer
@attribute SignalQualify integer
@attribute SPO2 integer
@attribute HeartBeatRate integer
@attribute class {normal, critical, very_critical}

%Veri kümesinde yer alacak veriler tanımlanıyor.
@data
0, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 2
38, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 3
5, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164
, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31,
146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0
, 50, 31, 146, 164, 43, 8, 35, 125, 238, 255, 0, 50, 31, 146, 164, 43, 8, 35, 125, 23
8, 255, 0, 50, 31, 146, 164, normal

```

Şekil 6-2. Bir hastanın örnek ARFF dosyası

6.1.2. Sınıflandırıcı

WEKA’da herhangi bir öğrenme algoritması soyut `weka.classifiers.Classifier` sınıfından türetilmektedir. Bunların az bir kısmı temel bir sınıflandırıcı için gereklidir. Burada temel sınıflandırıcı, bir eğitim verikümesinden bir sınıflandırıcı modeli oluşturan bir rutindir. Diğer rutinler ise, öngörülemez bir test verikümesi üzerinde oluşturulmuş bir modeli değerlendiren bir rutin ya da tüm sınıflar için bir olasılık dağılımı oluşturan bir rutindir (Bouckaert vd., 2008).

Geliştirilen sistem içerisinde kullanılan kNN sınıflandırıcı `weka.classifiers` sınıfından türetilmektedir. WEKA’da sınıflandırıcılar WEKA’nın çekirdeğinde yer

almaktadırlar. Sınıflandırıcılar için birçok ortak seçenek vardır. Bu seçeneklerden en önemlileri ve geliştirilen yazılım uygulamalarında da kullanılan seçenekler aşağıda kısaca açıklanmaktadır.

- -t ARFF formatından hazırlanmış eğitim dosyasını belirtir.
- -T ARFF formatında hazırlanmış test dosyasını belirtir. Eğer bu parametre kullanılmamış ise o zaman eğitim verileri üzerinde CV işlemi gerçekleştirilir. CV için k değeri varsayılan olarak 10 olarak alınır.
- -x Eğer -T parametresi belirtilmemişse CV için k değerini set eder.
- -c Bu parametre bir tabanlı indeksli sınıf değişkenini set eder. Yani hangi niteliğin sınıf niteliği olacağı set edilir. Varsayılan olarak en son nitelik sınıf niteliğidir.
- -d Bu parametreyle birlikte eğitim işleminden sonra oluşan model kaydedilir. Yalnız CV ile oluşturulan çoklu modellerde kaydetme işlemi olmaz.
- -l Önceden kaydedilmiş modeli yükler.
- -p# Bir test dosyası uygulanmış ise bu parametre sonuç sınıfı için yapılan tahminleri gösterir.
- -i Bu parametre ile doğruluk tablosundan oluşturulan precision, recall, true ve false pozitif oranlar gibi performans değerleri çıkış olarak verilir.
- -v Eğitim verisi için istatistiklerin yer almadığı bir çıktı verir.
- -o Sınıflandırıcı tanımlarının yer almadığı ama istatistiksel sonuçların yer aldığı bir çıktı verir.

Geliştirilen yazılım uygulamalarında WEKA içerisinde yer alan kNN sınıflandırıcı çağrılmaktadır. Bu açıdan WEKA buna olanak sağlamaktadır. WEKA'nın çağrılması ile ilgili kodlar EK-A'da yer almaktadır. Yalnız komut satırından gönderilen parametrelerin incelenmesi açısından aşağıdaki komut verilmektedir.

```
java -cp weka.jar weka.classifiers.lazy.IBk -t training.arff -x 10 -v -o -i
```

WEKA aracı Java programlama dili ile geliştirildiğinden dolayı ister herhangi bir program üzerinden istenirse de komut penceresinden yukarıdaki komut yazılabilir. Öncelikle Java Çalışma Ortamı Derleyicisi çağrılır ve -cp parametresi ile orta seviye koda dönüştürülen weka.jar dosyası yani WEKA aracı çalıştırılır. Daha sonra

weka.classifiers.lazy.IBk sınıfı WEKA içerisinde çağrılır. Daha önce weka.classifiers sınıfının WEKA'nın çekirdeğinde olduğunu söylemiştik. Bu sınıf içerisinde lazy algoritmaları içeren bir sınıf ve bu sınıftan türetilen IBk isminde de başka bir sınıf yer almaktadır. WEKA'da kNN sınıflandırıcı IBk olarak çağrılmaktadır. Bu aşamadan sonra yazılan `-t training.arff` ifadesi ile `training.arff` dosyası eğitim verisi olarak seçilir. `-x 10` ifadesi ile de CV için k değeri 10 olarak belirlenir. `-v`, `-o` ve `-i` parametreleri yukarıda açıklanmaktadır. Buna göre bu kod çalıştırıldığı zaman 10 elemanlı bir örnek verikümesi için Şekil'deki çıktıyı verir.

```

=== Stratified cross-validation ===

Correctly Classified Instances      10          100    %
Incorrectly Classified Instances    0           0     %
Kappa statistic                     1
Mean absolute error                 0.0159
Root mean squared error             0.0168
Relative absolute error             14.2857 %
Root relative squared error         14.2857 %
Coverage of cases (0.95 level)     100     %
Mean rel. region size (0.95 level) 33.3333 %
Total Number of Instances          10

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1       0       1           1       1           ?       normal
          0       0       0           0       0           ?       critical
          0       0       0           0       0           ?       very_critical
Weighted Avg.  1       0       1           1       1           0

=== Confusion Matrix ===

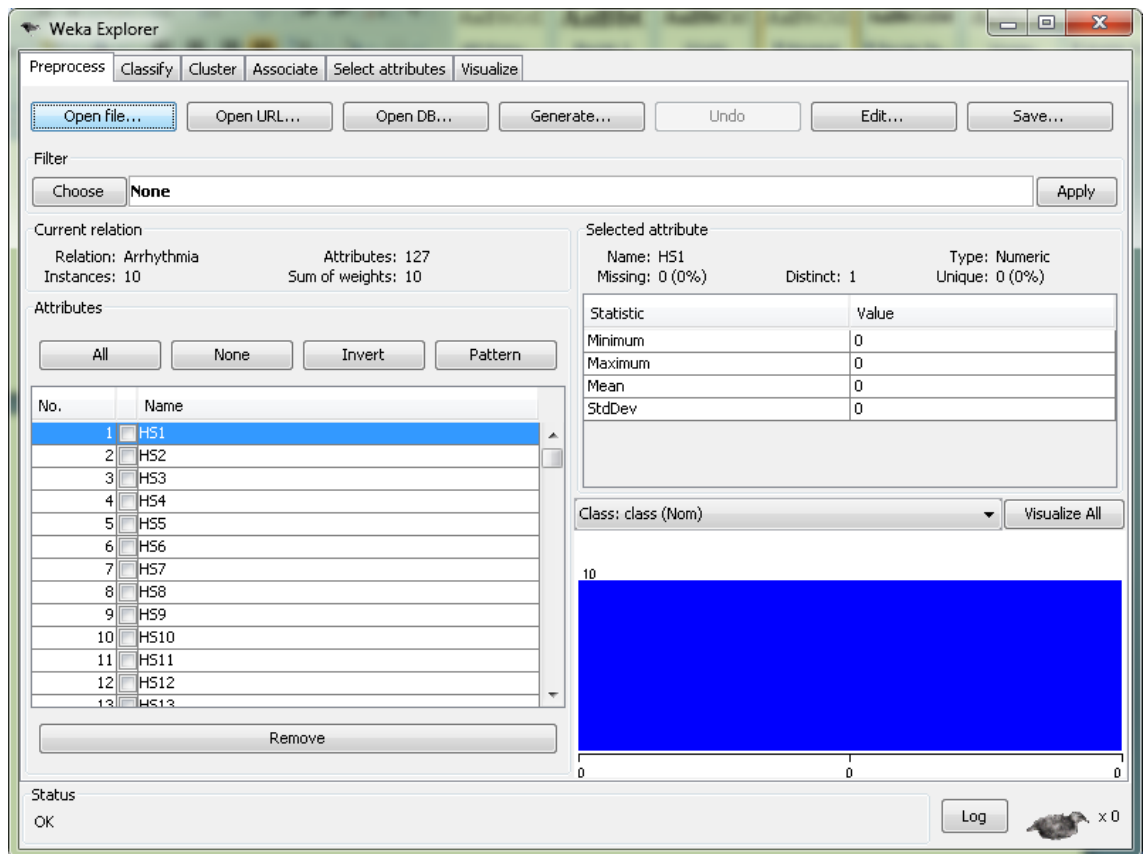
  a  b  c  <-- classified as
10  0  0 | a = normal
 0  0  0 | b = critical
 0  0  0 | c = very_critical

```

Şekil 6-3. Çalıştırılan komutun çıktısı

6.2. Explorer

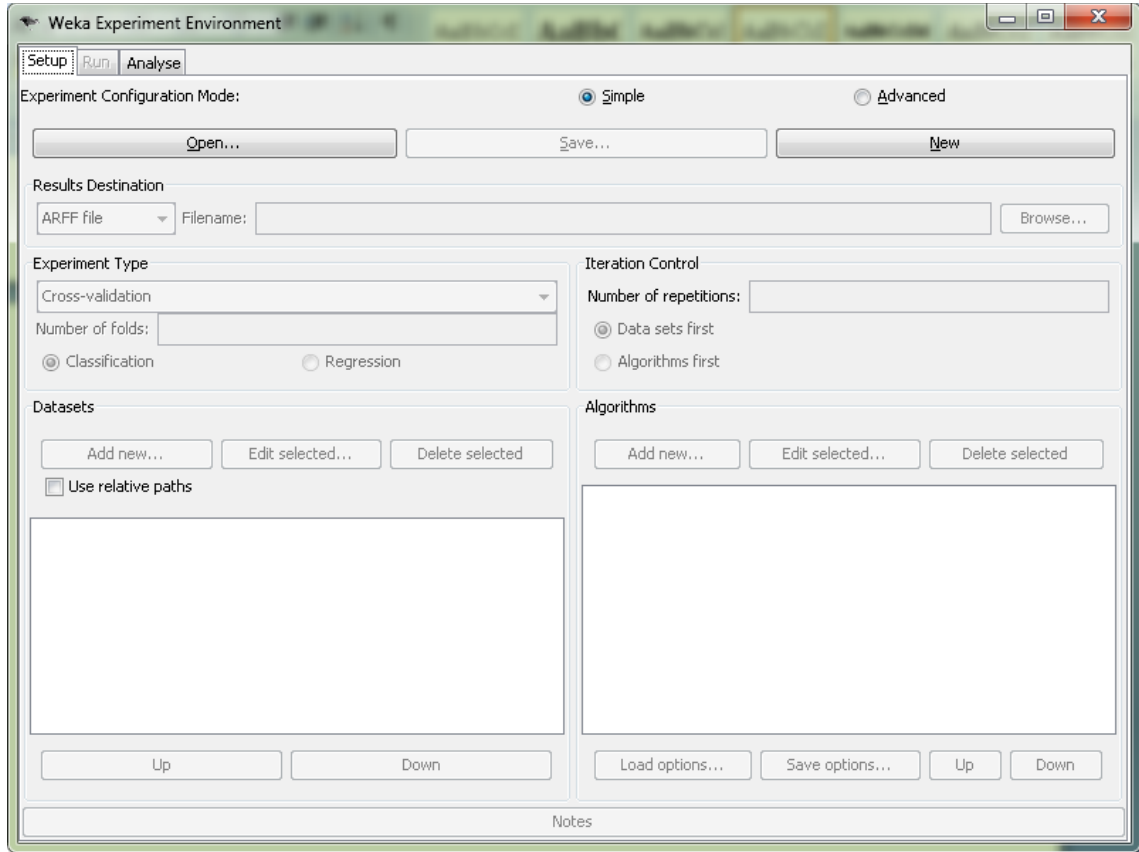
WEKA'nın ana grafik arayüzü Explorer olarak bilinir (Bouckaert vd., 2008). Şekil 6-4'te Explorer penceresi gösterilmektedir. Bu pencere altı menüden oluşmaktadır. Preprocess menüsü ile eğitim verisini içeren ARFF dosyası yüklenir ve verikümesi ile ilgili ön işlemler yapılır. Classify menüsü ile bir sınıflandırıcı seçilir ve bu sınıflandırıcı eğitim kümesi yardımıyla eğitime çalışılır. Cluster menüsü ile birbirine yakın örnekler bir arada gruplanır. Yani kümeleme işlemi gerçekleştirilir. Associate menüsü ile veri içerisinde birliktelik kuralları elde edilir. Select attributes menüsü ile verikümesi içerisindeki bir çok ilişkili nitelik seçilir. Visualize menüsü ile öğrenme işlemi sonunda oluşan analiz ve verikümesinin analizi 2D olarak gösterilir.



Şekil 6-4. Explorer penceresinden bir görünüm

6.3. Experimenter

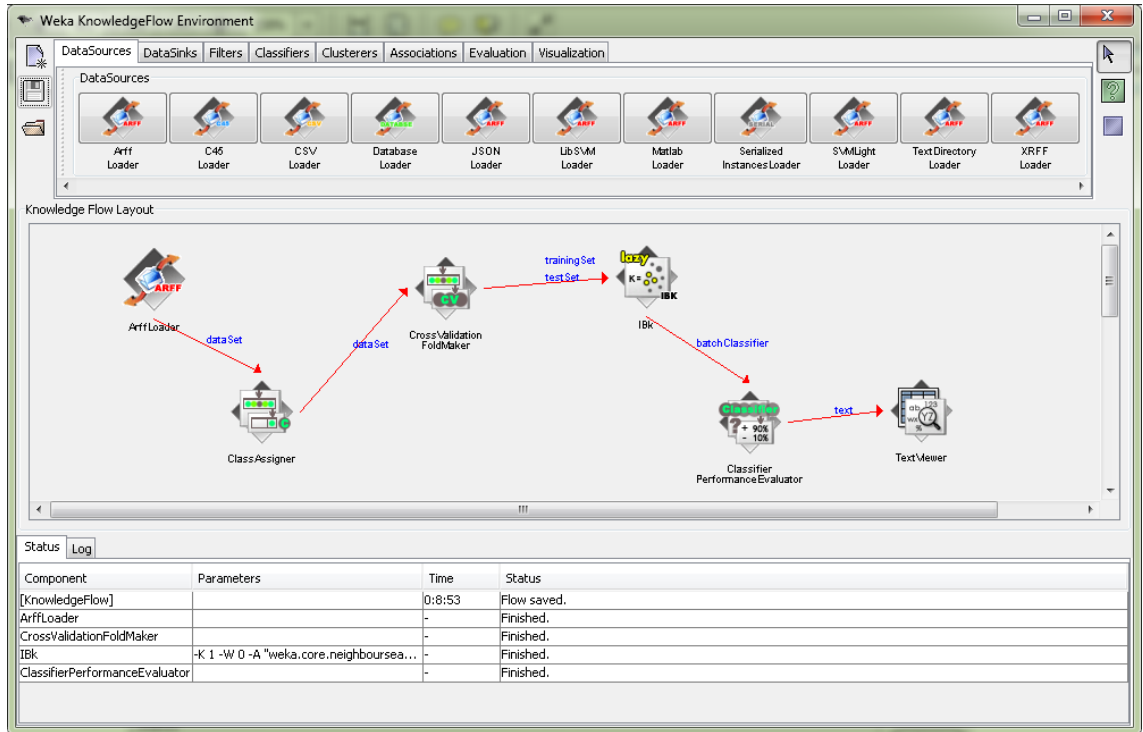
Explorer ve KnowledgeFlow ortamları ML planlarının verilen bir verikümesi üzerinde ne kadar iyi karar verdiği konusunda kullanıcılara yardım eder. Ancak ciddi araştırma çalışmalarında çeşitli parametre ayarlamaları, farklı verikümelemleri üzerinde çalışan birkaç öğrenme algoritmasının sınanması gibi bazı deneyler gerçekleştirilmektedir. Experimenter kullanıcılara geniş ölçekte deneyler gerçekleştirilebilmelerine olanak sağlar (Bouckaert vd., 2008). Şekil 6-5'te Experimenter ortamından bir görünüm gösterilmektedir.



Şekil 6-5. Experimenter ortamından bir görünüm

6.4. KnowledgeFlow

KnowledgeFlow ortamı Explorer'a alternatif olarak geliştirilmiş bir arayüzdür. Explorer ile yapabildiklerimizi araç çubuğu üzerinde yer alan bileşenlerle gerçekleştirilir. Gerçekleştirilen işlemler Explorer'dan farklı olarak görsellik taşımaktadır. Şekil 6-6'da KnowledgeFlow arayüzüne ait bir görünüm gösterilmektedir.



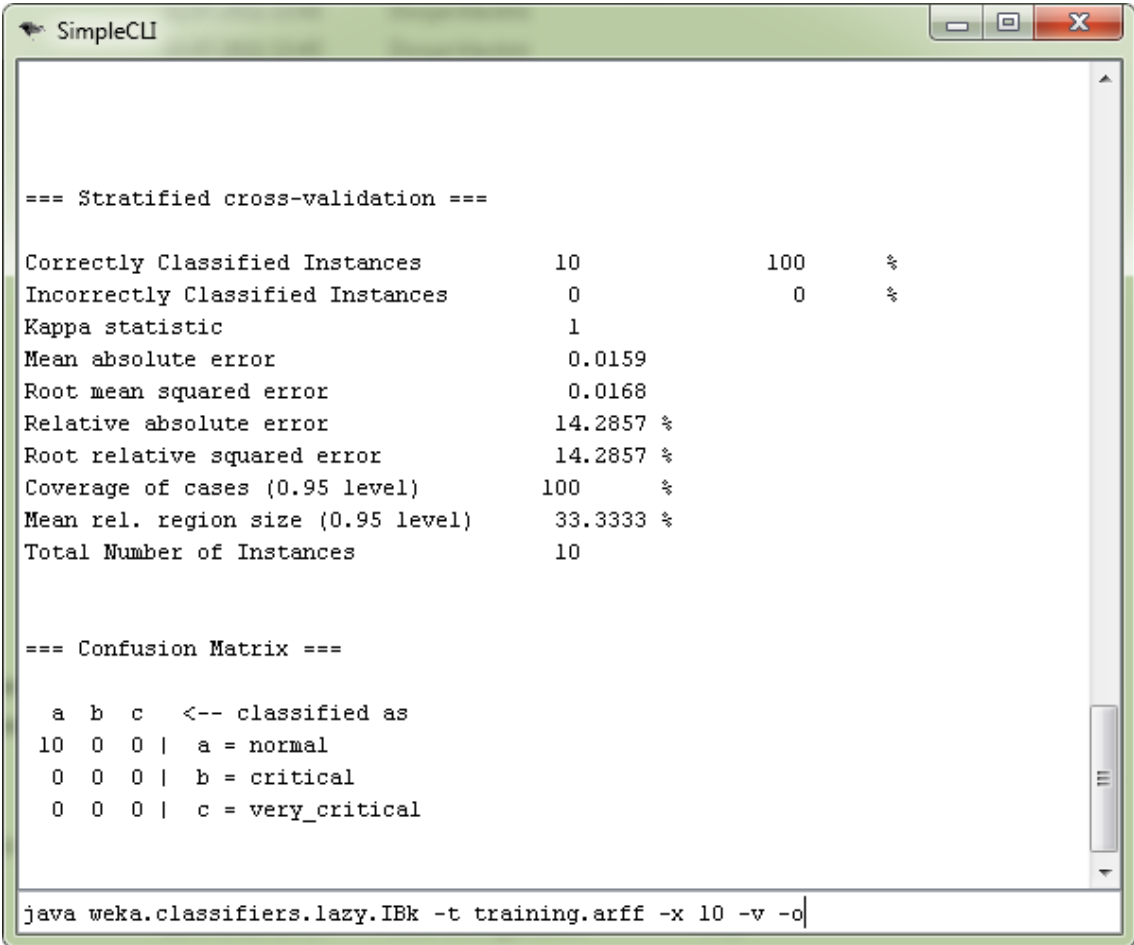
Şekil 6-6. KnowledgeFlow ortamından genel bir görünüş

6.5. Simple CLI

Simple CLI arayüzü ile birlikte WEKA komut satırı kullanılarak verikümleri üzerinde deneyler gerçekleştirilir. Yukarıda yazmış olduğumuz komutu Simple CLI ortamını kullanarak yazacak olursak Şekil'deki görüntüyü elde ederiz. Yalnız bunun

için komutta ufak bir değişikliğe gitmek zorundayız. Bu durumda komut satırına yazacağımız komut aşağıdaki gibi olur.

```
java weka.classifiers.lazy.IBk -t training.arff -x 10 -v -o
```



```

SimpleCLI

=== Stratified cross-validation ===

Correctly Classified Instances      10          100   %
Incorrectly Classified Instances    0           0    %
Kappa statistic                     1
Mean absolute error                 0.0159
Root mean squared error             0.0168
Relative absolute error             14.2857 %
Root relative squared error         14.2857 %
Coverage of cases (0.95 level)     100        %
Mean rel. region size (0.95 level) 33.3333 %
Total Number of Instances          10

=== Confusion Matrix ===

  a  b  c  <-- classified as
10  0  0 | a = normal
 0  0  0 | b = critical
 0  0  0 | c = very_critical

java weka.classifiers.lazy.IBk -t training.arff -x 10 -v -o

```

Şekil 6-7. Simple CLI arayüzünden genel bir görünüm

7. ARİTMİ HASTALARI İÇİN GELİŞTİRİLEN SİSTEM

7.1. Teze Konu Olan Projenin Tanıtımı

Aritmi birçok nedene bağlı olarak ortaya çıkan kaotik bir hastalıktır. Bu hastalığın oluşmasına kalbin kaotik bir biçimde kasılması neden olmaktadır. Bu düzensiz kasılmalar bazen vücut tarafından tolere edilebilmesine karşın bazen de metabolizmayı olumsuz bir biçimde etkileyebilmektedir. Bu durum aritmi hastalarının sürekli bir ölüm riski taşımaları anlamına gelmektedir. Hastanın bu ölüm riski karşısında hastalığın teşhisi ve tedavisi için gözetim altında tutulması gerekmektedir. Bazen yanlış konulan teşhis ve bunun sonucunda tedaviden dolayı hastanın aritmi şikâyetleri devam edebilmektedir. Bu durum da hastanın diğer günlük faaliyetlerini de aksatmadan gözetim altında tutulması gerekebilmektedir. Bundan dolayı hastanın bir sağlık kurumunun dışında herhangi bir ortamda da gözetim altında tutulması ve hastanın durumu hakkında bilgi alınması gerekmektedir. Bunun doğal sonucu olarak hasta lokasyondan bağımsız olarak izlenebilecektir.

Aritmi hastalarının teşhis sürecinde non-invazif bir yöntem olan EKG çok önemli bir görev görmektedir. EKG ile kalpten kaynaklanan aritmi şikâyetleri tespit edilebilmekte ve kalbin hangi bölgesinin görevini yerine getiremediği bulunabilmektedir. Yalnız bir sağlık kurumu dışında çok hassas (12-derivasyonlu) bir EKG cihazını hastanın bulunduğu konumda bulundurabilmesi çok zordur. Aynı zamanda hastanın mobil olduğu düşünülürse bu durum imkânsız olur. Bu ve buna benzer nedenlerden dolayı ölçüm cihazları için çeşitli yöntemler geliştirilmiştir (Koivistoinen vd., 2004; Alihanka vd., 1981; Brink vd., 2006; Shirouzu v d., 2001). Bu çalışmada EKG gibi kalp sinyallerini işleyen bir cihaz yerine metabolizmanın çalışmasının bir sonucu olarak ortaya çıkan nabız, kan basıncındaki dalgalanmalar, kandaki oksijen miktarı vb. parametreleri okuyan bir cihaz kullanılmış ve bu cihaz mobilize hale getirilerek geliştirilen sisteme entegre edilmiştir.

Geliştirilen cihazın hastadan aldığı bilgiler GPRS (Bettstetter vd., 1999; Brasche vd., 1997) teknolojisi yardımıyla uzaktaki Server'ın işlemsel veritabanına kaydedilir. Daha sonra veritabanındaki veriler işlenerek veri ambarına taşınır. Bir veri ambarı karar verme sürecine yardım etmek için kurulan büyük bir tümleşik veri topluluğudur (Hurtado vd., 1999). Veri ambarları işlemsel veritabanlarından oluşturulurken tüm verilerin iş akış sürecine yönelik, bütünsel ve belirli bir zaman dilimine ait olacak şekilde düzenlenmesi gerekmektedir (Guerrero vd., 2004). Bu aşamadan sonra OLAP (Codd, 1993) ya da veri madenciliği (Witten vd., 2005) gibi karar vermeye yardımcı olacak veri analizi ve sorgulamalar gerçekleştirilir. Bu tez çalışmasında OLAP yerine veri madenciliği ya da daha açık bir ifadeyle ML metotlarından kNN sınıflandırıcı kullanılmıştır.

7.2. Öğrenmeye Dayalı Sistemi Oluşturan Bileşenler

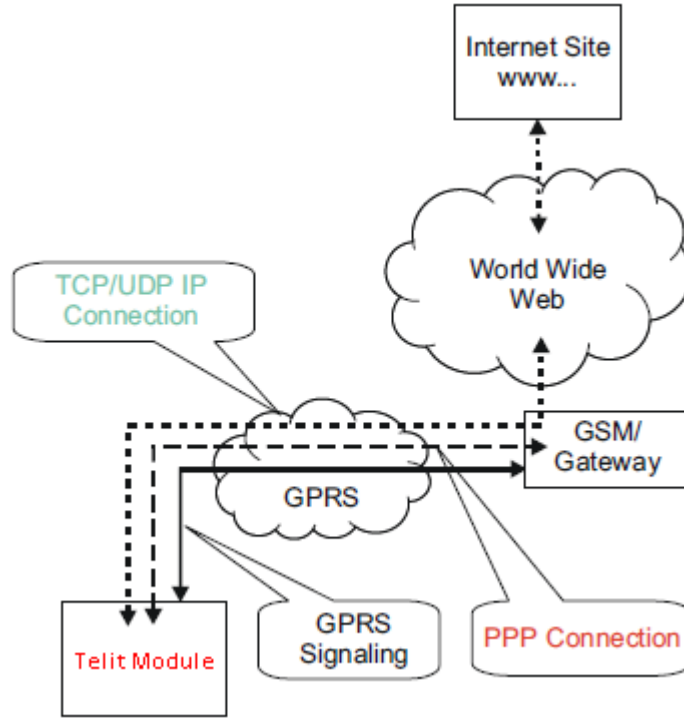
Bu tez çalışması için geliştirilen projenin bileşenleri aşağıda sunulmaktadır.

- PIC 16F877
- Digital Pulse Oximeter Module ChipOx (<http://www.envitec.com>)
- Telit GM862-GPS module (<http://www.telit.com>)
- Workstation
- Weka Software Developer version 3.7.1

Bu bileşenlerden ilk üçü hastanın üzerinde taşıdığı cihazı oluşturan bileşenlerdir. Diğer bileşenler ise Server üzerinde toplanan verilerin öğrenildiği ve değerlendirildiği kısmı oluşturmaktadır. Bu kısım, geliştirilen web tabanlı bir uygulama ile beraber çalışmaktadır. Bu makalede daha çok ikinci kısım üzerine odaklanıldı.

7.3. Veri Toplama

Hastalardan elde edilen veriler GPRS teknolojisi kullanılarak Server üzerindeki veri tabanına kaydedilmektedir. Böylelikle hastanın bulunduğu ortamda herhangi bir kablolu iletişim teknolojisinin bulunmamasının bir önemi olmamaktadır. Ayrıca veriler ilgili GSM/Gateway üzerinden WWW'e aktarılmakta ve TCP/IP paketleri hâlinde Server'a iletilmektedir. Bu sisteme ait alt yapı Şekil 7-1'de gösterilmektedir.

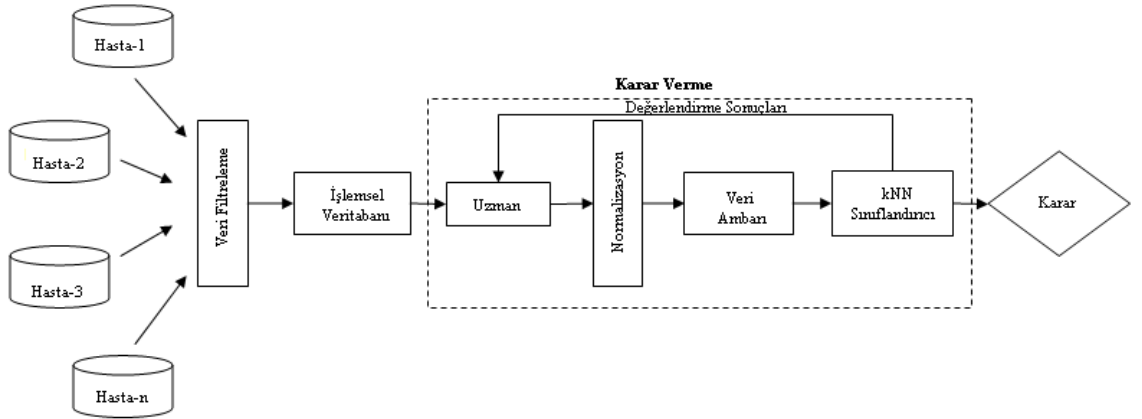


Şekil 7-1. GPRS'in genel bir görünümü

GPRS ağlarında TCP performansı karardır (Meyer, 1999; Othman vd., 2007). Bu nedenle veri toplama için kullanılan sistem performans, esneklik ve maliyet açılarından etkindir.

Toplanan veri, veritabanına kaydedilmeden önce veri filtreleme işlemi gerçekleştirilir. Bu işlem veri üzerindeki bozuklukları tespit etmeyi ve pulse oksimetre (Nellcor, 1988) cihazı ile hastadan alınan verilerin sinyal kalitesinin düşük olması sonucu bu verileri iptal etmeyi içermektedir. Veri filtreleme işleminden sonra veriler işlemsel veritabanına kaydedilir. Daha sonra uzman tarafından belirlenen veriler normalize edilerek veri ambarına taşınır. Veri ambarındaki veriler daha sonra öğrenmeye dayalı sistem için eğitim verileri olarak kullanılır.

AIM ya da diğer metotlara dayalı geliştirilen sistemler insan uzmanlarla birlikte yürütülmesi gerektiğini (Vimla vd., 2009) daha önce de vurgulamıştık. Bunun nedeni olarakta hiçbir sistemin bir uzmanın yıllar içerisinde kazandığı tecrübeyi kazanamamasıdır. Bu nedenle geliştirilen AIM sistemi içerisinde toplanan verileri etiketleme işlemini bir insan uzman yapmaktadır. Böylelikle geliştirilen AIM sistemi içerisinde bir uzman katılımıyla gerçekleştirilen öğrenme ile tümevarımsal uzman sistem tasarımına da gidilmiş oldu. Bu sistem Şekil 7-2’de gösterilmektedir.



Şekil 7-2. Geliştirilen AIM sisteminin genel bir görünümü

7.4. Seçilen Nitelikler

UT'ye göre yapılan özellik seçiminde seçilen nitelikler şunlardır: Pletismogram, Periferik oksijen saturasyonu, nabız ve sinyal kalitesi. Bu nitelikler aşağıda açıklanmaktadır.

Pletismogram: Kalp atışı ile birlikte kan akışının değişimi atardamarlarda iletilir ve bir pulse dalgası olarak ölçülebilir. Ölçülen bu puls dalgasının yoğunluk değişimi pletismogram olarak adlandırılır (Sakane vd., 2003). Yani bu yoğunluk değişiminin nedeni atardamardaki kan miktarındaki dalgalanmalardır.

Pletismogram testi kullanılarak kol ve bacaklardaki kan pıhtıları kontrol edilebilir ya da akciğerde ne kadar hava tutulduğu ölçülebilir.

Hastalarda kullanılan pulse oksimetre cihazının pletismogram değer aralığı: 0 LSB ile 255 LSB'dir (<http://www.envitec.com>).

Periferik Oksijen Saturasyonu (SPO₂): Kandaki oksijen saturasyonu ya da erimiş oksijen seviyesi olarak ölçülen SPO₂ aşağıdaki formüle göre pulse oksimetre ile hesaplanır. HbO₂: oksihemoglobin, Hb: hemoglobin

$$SPO_2 = \frac{HbO_2}{HbO_2 + Hb} \quad (27)$$

Hastalarda kullanılan pulse oksimetre cihazının ölçülen SPO₂ değer aralığı 45% ile 100%'dür. Bu değerler için doğruluk aralığı ise kullanılan pulse oksimetre cihazı tarafından 70% < SPO₂ < 100% olarak belirlenmektedir (<http://www.envitec.com>).

Nabız: Nabız, kalbin ne kadar yeterlilikte çalıştığının bir göstergesidir. Ayrıca güçlü ve ya zayıf nabız kalp ve damar problemlerinin teşhisine yardımcı olabilir. Çünkü bu durum kalbin kanı nasıl pompaladığını gösterir.

Kullanılan pulse oksimetre için deęer aralıęı 0‰ ile 250‰'dir. Bu deęer 10‰ altında ise nabız dūşüktür. Eęer bu deęer 15‰ ise nabız yeterlidir (<http://www.envitec.com>).

Sinyal Kalitesi: Sinyal kalitesi deęer aralıęı 0% - 100%'dür ve sinyal kalitesi 90%'nin üzerindeyse sinyal kalitesinin iyi olduęu söylenebilir (<http://www.envitec.com>). Pulse oksimetre ile yapılan ölçümler sırasında 90% sinyal kalitesinin üzerindeki veriler veritabanına kaydedilmektedir. Veritabanına kaydedilen veriler veri ambarına taşınmadan önce sinyal kalitesi nitelięi üzerinde min-max normalleştirme yöntemi uygulandı. Min-max normalleştirilme yöntemi uygulanarak sinyal kalitesi nitelięinin dięer nitelikler üzerindeki baskısı azaltılmış oldu. Böylelikle kullanılan öğrenme algoritması, sinyal kalitesi dışındaki niteliklerin öğrenme sürecindeki etkinlięini daha iyi kullanabilmiştir.

Min-max normalleştirme yöntemi ile veriler 0 ile 1 arasındaki deęerlere dönüştürülür. Bu dönüştürülme işleminin (28) eşitlięinde gösterilmektedir. Bu eşitlięe göre; X^* dönüştürülmüş deęerleri, X gözlem deęerlerini, X_{\min} en küçük gözlem deęerini ve X_{\max} en büyük gözlem deęerini ifade etmektedir.

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (28)$$

Sinyal kalitesi nitelięi için uygulanan min-max normalleştirilmesi yönteminde X_{\min} deęeri 90 ve X_{\max} deęeri ise 100 olarak alındı.

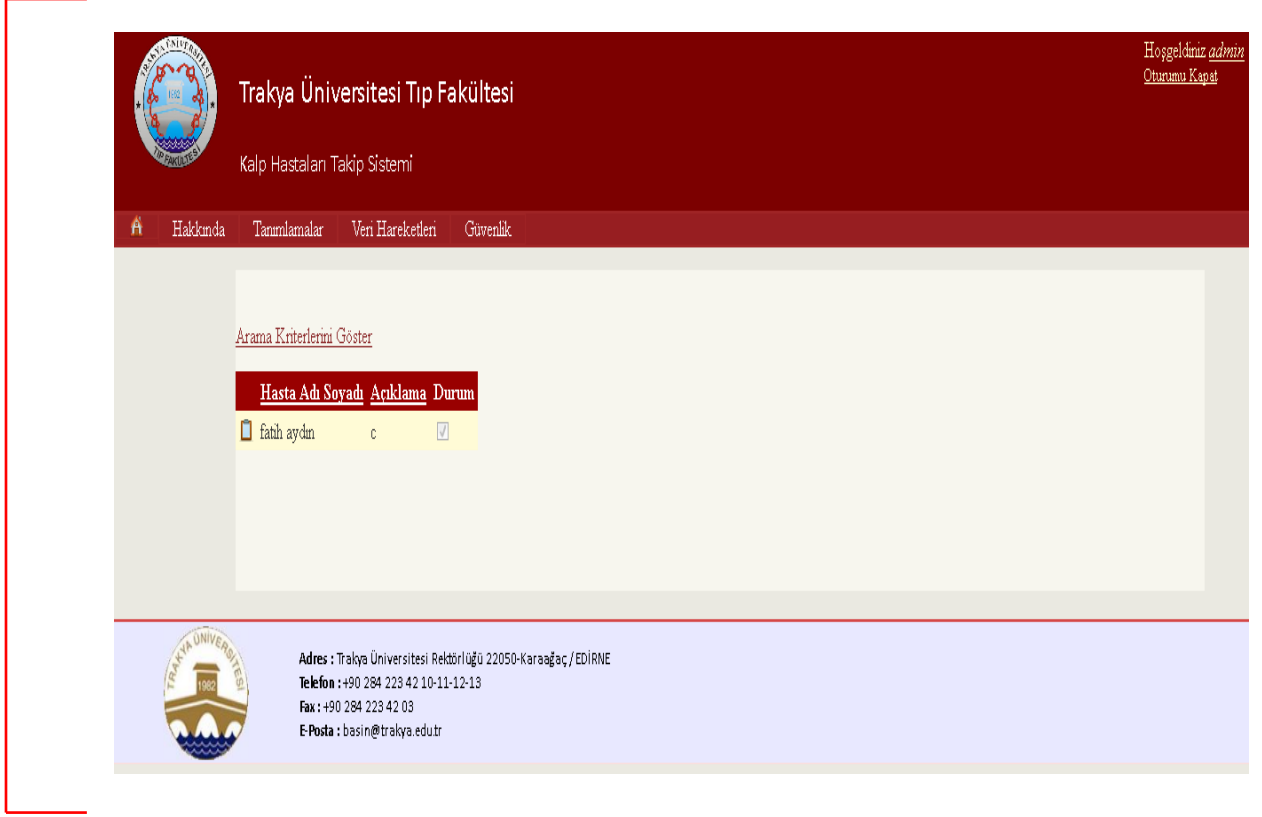
8. GELİŞTİRİLEN UYGULAMALAR

Bu tez çalışmasının amacını gerçekleştirmek için biri web tabanlı diğeri desktop tabanlı olmak üzere iki yazılım geliştirildi. Bu bölümde bu yazılımların geliştirilme nedenleri ve görevleri tanıtılacak.

Geliştirilen yazılımlar Visual Studio 2010 aracı kullanılarak .NET platformunda C# programlama dili ile geliştirildiler. Web tabanlı uygulama .NET 2.0, desktop tabanlı uygulama ise .NET 4.0 sürümünü kullanmaktadır. Uygulamalar veritabanına MS SQL Server 2005 sunucusunu kullanarak bağlanmaktadır. Aynı zamanda geliştirilen uygulamalar aynı sunucu üzerinde hizmet vermektedirler.

8.1. Uzman Değerlendirmeleri için Geliştirilen Uygulama

Geliştirilen web tabanlı uygulama, veritabanında depolanan aritmi hastalarının verilerini inceleyen bir uzmanın verileri analiz edip bu verileri etiketleyebilmesini sağlayabilen bir arayüze sahiptir. Bu arayüz Şekil 8-2'de gösterilmektedir. Bu işlemde önce tüm aritmi hastaları kullandıkları cihazın ID'si ile eşleştirilerek veritabanına kaydedilmektedir. Ayrıca her hasta ile ilgilenen bir uzman doktor bulunmaktadır ve bu eşleştirme de yapılabilmektedir.

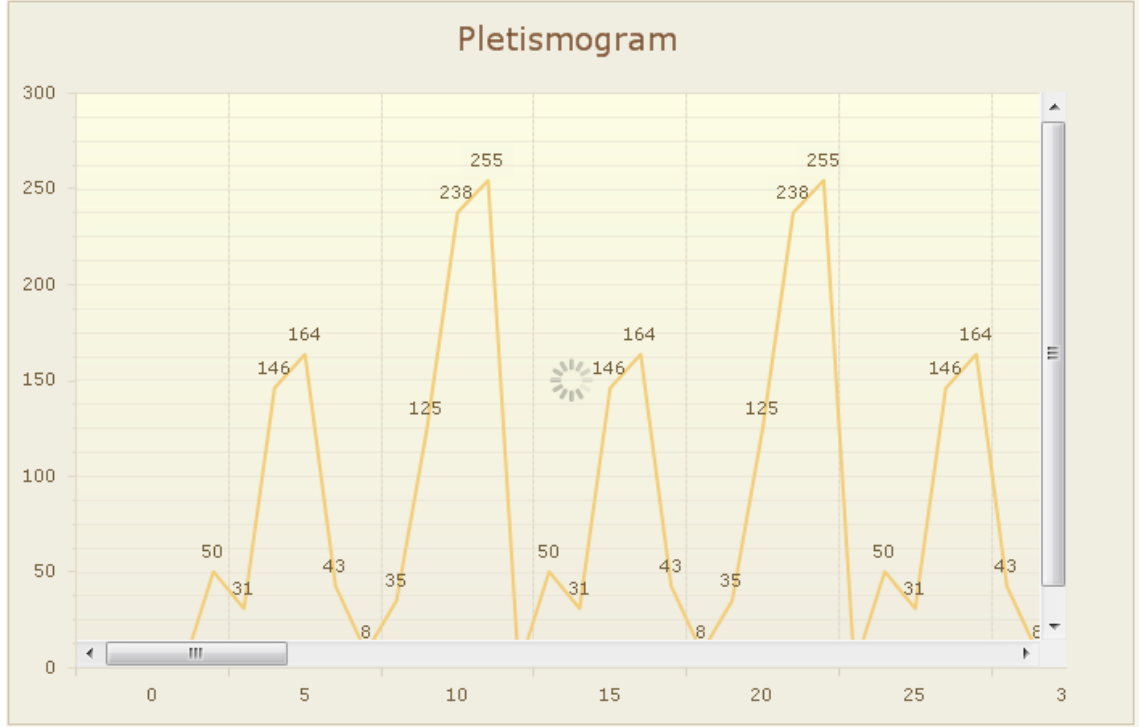


Şekil 8-1. Geliştirilen uygulamadan bir görünüm

Geliştirilen sistem, veritabanında depolanan verilerin öğrenme için yeterli olup olmadığı ile ilgili istatistikî bilgileri uzman kişiye sunmaktadır. Bununla ilgili arayüz Şekil 8-3'te gösterilmektedir. Bu bilgilerin hazırlanması ile ilgili yazılan kodlar EK-A'de yer almaktadır. Uzman kişi bu bilgilere göre verilerin yeterli geldiğine karar verirse, hasta için depolanan veriler veri ambarına taşınır ve eğitim verileri hazırlanmış olur. Eğer veriler yeterli değilse uzman kişi yeni gelen verileri etiketlemeye devam eder ve bu döngü tekrarlanır.

Bu uygulamanın web tabanlı geliştirilmesinin avantajı uzman kişinin nerede olursa olsun internetin olduğu her noktadan hasta bilgilerine erişim sağlayabilmesidir.

Sinyal Kalitesi: 31
SPO2: 146
Nabız: 164



Şekil 8-2. Hastanın veritabanında kayıtlı bir verisinden görünüm

Arama Kriterlerini Göster

Hastanın Adı Soyadı : fatih aydın










Sistem Değerlendirmesi**Yeterli tanınm konulamadığı sınıflar**

Kritik

Çok Kritik

Sistemin Öğrenme Durumu

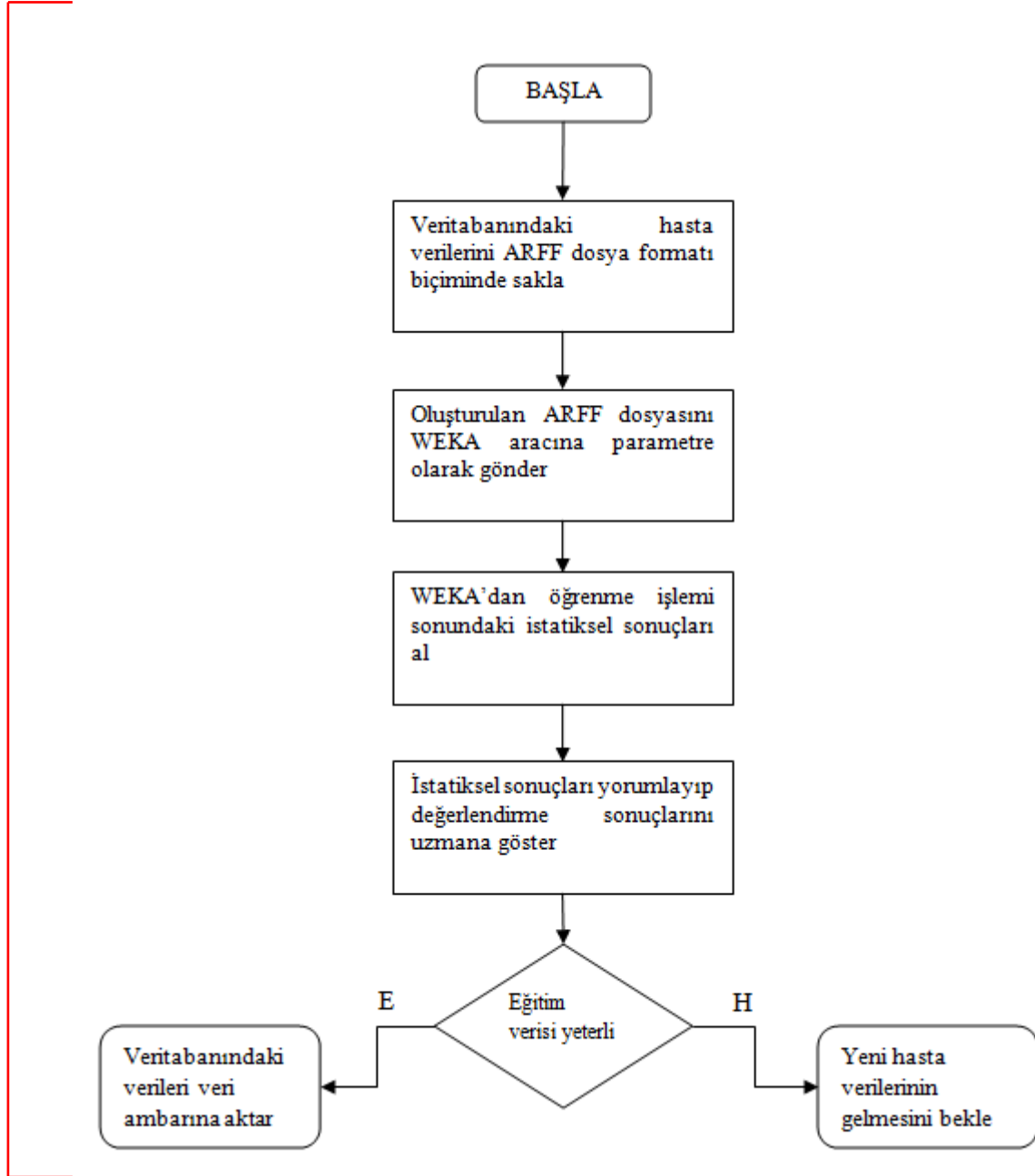
Sistem mükemmel derecede öğrenme gerçekleştirmektedir. Kappa değeri : 1

	<u>Cihaz No</u>	<u>Doktor Adı Soyadı</u>	<u>Kayıt Zamanı</u>	<u>Değerlendirme</u>
	× 1		02.11.2010 19:04:02	Normal
	× 1		02.11.2010 19:05:00	Normal
	× 1		02.01.2011 01:58:47	Normal
	× 2		02.01.2011 01:58:55	Normal
	× 2		02.01.2011 01:58:59	Normal
	× 1		02.01.2011 01:59:03	Normal
	× 1		02.01.2011 01:59:07	Normal
	× 1		02.01.2011 01:59:10	Normal
	× 1		02.01.2011 01:59:13	Normal

Şekil 8-3. Hastanın tüm verilerine göre sınıflandırıcının değerlendirilmesi

Şekil 8-4'te EK-A'da verilen kodların akış şeması ile birlikte uzman kişinin karar verme sürecindeki rolü gösterilmektedir. Buna göre öncelikle veritabanında depolanan hasta verilerinin WEKA'da tanımlı olan ARFF dosya formatına uygun bir biçimde formatlanması gerçekleştirilir. Daha sonra oluşturulan ARFF dosyası, konsol ortamda çağrılan WEKA'ya eğitim verisi olarak gönderilir. Bu bilginin yanı sıra CV değeri, hangi sınıflandırıcının kullanılacağı vb. bilgiler de WEKA'ya parametre olarak gönderilir. Bu öğrenme işlemi tamamlandıktan sonra WEKA'dan istatistiksel sonuçlar alınır. Alınan bu istatistiksel sonuçlar değerlendirildikten sonra uzman kişiye bir değerlendirme sonucu gösterilir. Uzman kişi bu değerlendirme sonucuna göre hastanın

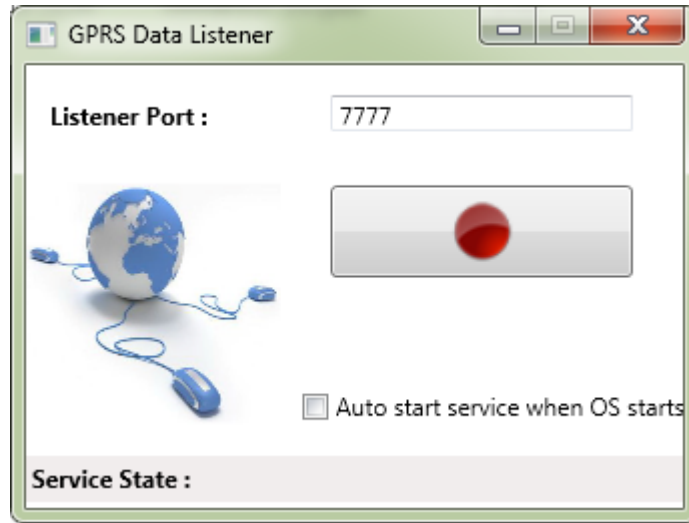
mevcut verilerinin öğrenme için yeterli gelip gelmediğine karar verir. Eğer yeterli gelmediğine kanaat getirirse o zaman hastanın yeni verilerinin gelmesini bekleyecektir. Eğer yeterli geldiğine kanaat getirirse o zaman da mevcut veriler veritabanından veri ambarına taşınır. Yalnız veritabanındaki veriler silinmez. Bu işlem her hasta için aynı şekilde devam eder.



Şekil 8-4. Uzman kişinin verinin eğitim için yeterliliğine karar verme süreci

8.2. Veri Toplama ve Veri Sınıflandırma için Geliştirilen Uygulama

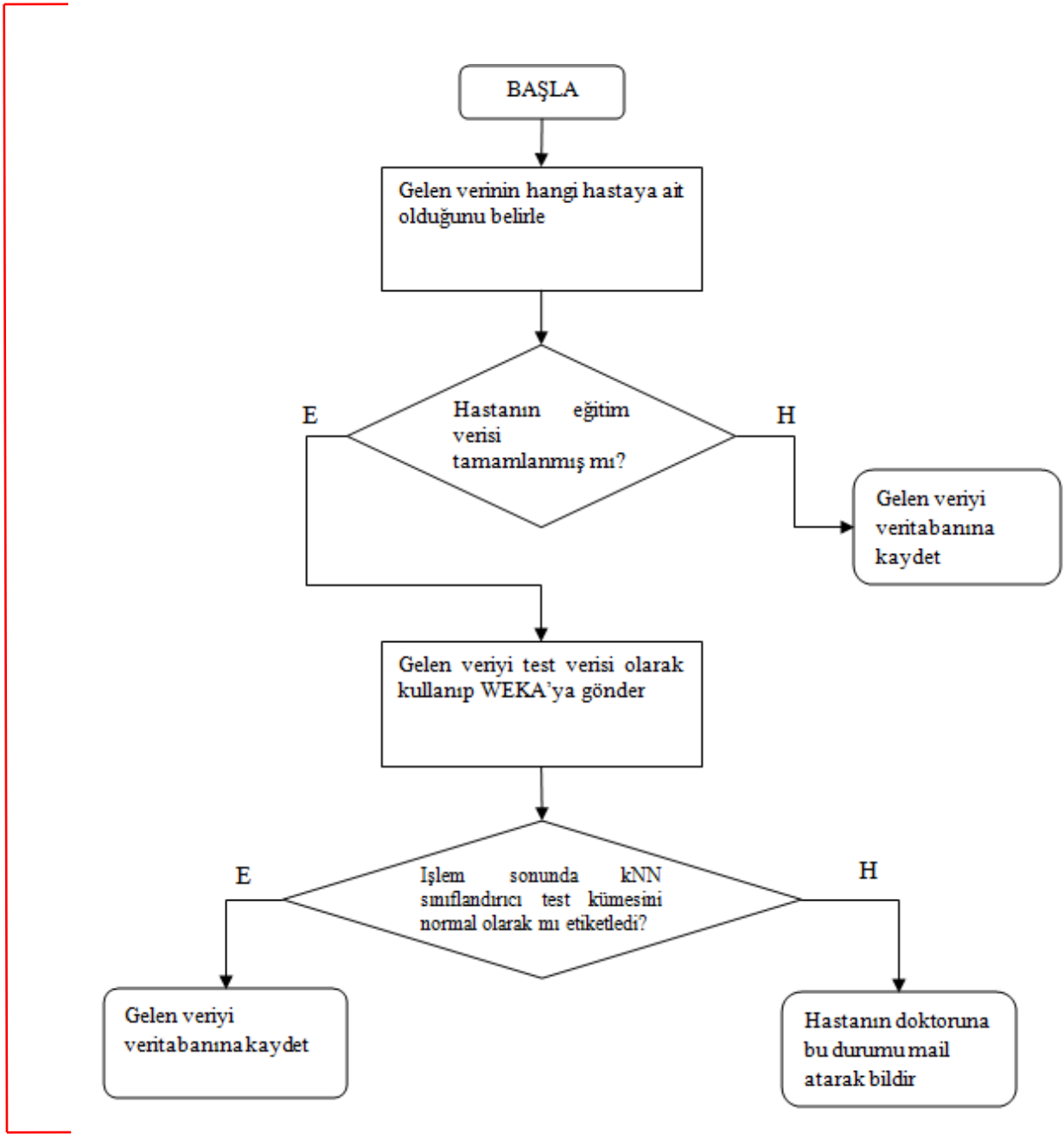
Geliştirilen bu uygulama internet üzerinden gelen aritmi hastalarının bilgilerini alıp veritabanına kaydeder. Eğer hastanın bilgileri öğrenme için yeterli ise ve uzman kişi bunu onaylamışsa gelen veri test verisi olarak kullanılır. Daha sonra gelen bu test verisi kNN sınıflandırıcı tarafından sınıflandırılır. Eğer kritik bir durum varsa bu bilgi hastanın doktoruna mail yoluyla bildirilir. Hastanın toplanan verileri eğitim verisi olmak için yeterli değilse gelen veriler toplanmaya devam eder. Şekil 8-5'te geliştirilen uygulamanın arayüzü görülmektedir. İnternet kanalıyla gelen verilerin veritabanına kaydedilmesi ile ilgili kodlar EK-B'de yer almaktadır.



Şekil 8-5. Geliştirilen uygulamanın arayüzü

Şekil 8-6'da verilen akış şemasında GPRS teknolojisi ile gönderilen ve internet üzerinden gelen hastada verisinin değerlendirilme süreci gösterilmektedir. Buna göre öncelikle gelen verinin hangi hastaya ait olduğu belirlenir. Bu tespit işlemi hastaya verilen cihazın ID'si ile hastanın veritabanında eşleştirilmesi ile yapılır. Gelen verinin

ilk verisi cihazın ID'si olmaktadır. Hastanın belirlenmesinden sonra bu hastanın eğitim verilerinin hazırlanma aşamasının tamamlanıp tamamlanmadığının belirlenmesine gelmektedir. Eğer bu aşamada hasta verisi uzman kişi tarafından eğitim verisi olarak yeterli görülmüşse o zaman veritabanında hastanın eğitim verisinin hazır olduğuna dair niteliğin değeri set edilmiş olur. Bu da bize eğitim verisinin hazır olup olmadığı ile ilgili bilgiyi vermektedir. Eğer eğitim verisi hazır değilse hastadan gelen veriler veritabanına kaydedilir ve birer eğitim verisi adayı olarak kalırlar. Bu kaydetme işlemi ile ilgili kodlar EK-B'de yer almaktadır. Eğer eğitim verisi hazır ise bu durumda da gelen veri test verisi olarak kullanılır. Bu test verisi, veriambarımdan ARFF olarak hazırlanan eğitim verisi ve diğer bilgiler WEKA'ya parametre olarak gönderilir. Bu işlemden sonra WEKA test verisinin sınıfına dair yapmış olduğu tahmin sonucunu döndürür. Bu tahmin sonucuna göre eğer test verisinin sınıfı "normal" değilse o zaman sistemde tanımlanan hastanın doktoruna hasta ile ilgili veriler mail yolu ile gönderilir ve bilgiler veritabanına kaydedilir. Eğer test verisinin sınıfı "normal" olarak etiketlenmişse o zamanda gelen veri sadece veritabanına kaydedilmiş olur.



Şekil 8-6. Hastadan gelen veriye göre karar verme süreci

9. GELİŞTİRİLEN SİSTEMİN DEĞERLENDİRİLMESİ

Veri üzerinde örüntü keşfi için geleneksel istatistik, veri madenciliği, makine öğrenmesi v.b yaklaşımlar kullanılır. Bu tekniklerin ortak özelliği veri üzerinde göreceli olarak basit ilişkileri keşfetmektir. Çünkü bu tekniklerin kompleks yapılar arasındaki ilişkileri keşfetmek için henüz yeterli olmadığı görülmektedir. Bu yüzden bu tür tekniklerle birlikte insan uzmanlığını da içeren teknikler kullanmak gerekmektedir. Bu tez çalışmasında da ML ile insan uzmanlığı entegrasyonuna dayanan bir modelleme geliştirilmiştir. Böylelikle indaktif uzman sistem tasarımı gerçekleştirilmiştir.

Bilimsel bir araştırmanın sınıflandırılması temel olarak üç aşamadan oluşur. Bunlar: veri toplama, sistematizasyon ve formelleştirmedir. Bu tez çalışmasının temel yaklaşımında bu üç aşama üzerine kurulmaktadır. İlk aşama ile birlikte problemin ele alınmasında gerekli verinin toplanması sağlandı. İkinci safha ile birlikte toplanan verinin düzenlenmesi (filtreleme, normalizasyon, etiketleme v.b.) ve modelin seçilmesi\kurulması sağlandı. Son aşama ile de deneylerin analitik bir biçimde analiz edilmesi için gerekli olan çok yönlü metotların seçimi, doğrulanması ve biçimlendirilmesi gerçekleştirildi.

Geliştirilen sistemin yeteneği ve performansı aynı zamanda öğrenme sürecinde kullanılan sınıflandırıcının performansı ile doğru orantılıdır. Bu nedenle sınıflandırıcının performans analizi ve bu analizde kullanılacak metotlar önemli olmaktadır. Bu da bilimsel araştırmanın son aşamasını oluşturmaktadır. kNN gibi tembel sınıflandırıcılar için eğitim verisinin büyüklüğü dezavantaj olmaktadır. Bu amaçla en uygun veri büyüklüğü seçilmelidir. Bu sorun bize bias-variance dengesini akla getirmektedir. Bu dengenin sağlanması ile sınıflandırıcının uygun verilerle etkin bir öğrenme gerçekleştirmesi sağlanır. Bu amaçla f-measure, RMSE ve kappa ölçütleri kullanıldı. Öncelikli olarak f-measure ve RMSE ölçütleri birlikte değerlendirilmeye alınarak sınıflandırıcının performansı incelenmeye çalışıldı. Daha sonra kappa ölçütü kullanılarak sınıflandırıcının ne kadar iyi bir öğrenme gerçekleştirdiği ölçülmeye çalışıldı.

10. SONUÇLAR VE TARTIŞMA

Tıp Bilişimi, uygulama alanı hayal gücü ile sınırlı bir alandır. Bu anlamda tıp bilişimi çalışmalarını birden çok alanı kapsayan bir çalışma olarak düşünebiliriz. J.H. Thrall 2000’li yıllarda tıp bilimlerine üç ayrı alanın etki edeceğini belirtmiştir. Bu alanlar: Moleküler biyoloji ve moleküler genetik, Nörobiyoloji ve beyin-algı-davranış ilişkisini inceleyen dallar ve son olarak elektronik-bilgisayar bilimleridir. Ayrıca J.H. Thrall tıp bilimlerinin teknik alanlarla ve davranışsal bilimlerle ilişki içinde olması gerektiğini de vurgulamaktadır (Thrall, 1998). Nitekim modern bilim pek çok alanda disiplinlerarası olmaya yönelmiştir; problemlerin etkili ve kapsamlı çözümüne ancak ekip çalışması ile varılabileceği gerçeği pek çok bilim çevresinde kabul görmektedir. Bu anlamda tıp bilimi ile makine öğrenmesi gibi iki popüler konu bir araya getirilerek aritmi hastalarının tedavi süreçlerine yardımcı bir çözüm metodu geliştirilmiştir. Bu konuda yapılan çalışmalar ileriki dönemlerde daha iyi sistemlerin geliştirilmesine öncülük ederek aritmi hastalarının tedavilerinde yardımcı olacaktır. Bunun göstergesi de yapılan tedavi sonrasında tedavinin sonucunun uzaktan dahi izlenebilir olması ile birlikte ne kadar başarılı olduğunun takibinin yapılabilmesi şeklinde olabilir.

Bu tez çalışması için geliştirilen sistemin bileşenleri elektronik, bilgisayar ve tıp konularını içermektedir. Elektronik kısım, hastanın metabolik parametrelerini alarak uzaktaki sunucuya iletme görevini üstlenmektedir. Bu kısımda kullanılan cihazlar sağlık kuruluşlarında da oldukça sık kullanılmaktadır. Bu açıdan ölçüm değerlerini değerlendirme doktorlar açısından bir sorun oluşturmamaktadır. Hasta verilerinin iletiminde ise projenin ilk aşamalarında dial-up bağlantı kullanılmıştır. Daha sonra hasta açısından kullanımı esnek olmayacağından, maliyetinin yüksek olmasından ve bağlantının yavaş (56 kbps) olmasından dolayı bu bağlantı türü değiştirilmiştir. Bu bağlantı türünün yerine GPRS teknolojisi kullanılmıştır. Böylelikle yukarıda sıraladığımız maddeler birer avantaja dönüştürülmüştür. Bilgisayar bilimleri alanında ise toplanan verinin öğrenilmesi için ML algoritmalarından kNN algoritması kullanılmıştır. Daniel Tunkelang’ın (Tunkelang, 2002) “Making the Nearest Neighbor

Meaningful” ve Kevin Beyer’in “When Is ‘Nearest Neighbor’ Meaningful?” isimli çalışmalarında kNN algoritmasının kullanımının hangi durumlarda uygun olacağı anlatılmaktadır. Bu iki çalışmanın ortak sonucu olarak kNN algoritması, sınıf sayısı çok fazla olmayan ve sınıfları birbirinden oldukça ayırık olan veri kümelerinde oldukça iyi başarı sağlamaktadır. Yapılan çalışmada da sınıf sayısı üç olarak belirlenmiştir. Ayrıca veriler incelendiğinde sınıfların birbirinden ayırık oldukları görülmektedir. Bu durum kNN’in başarımını arttırmaktadır. Bunun dışında kNN algoritması veri kümeleri üzerinde genellikle düşük bias’a sahiptir. Bu durum da kNN’in diğer sınıflandırıcılara göre biraz daha fazla tercih edilmesine sebep olmaktadır.

Geliştirilen uygulama da öğrenme yöntemi olarak Gözetimli öğrenme kullanılmıştır. Gözetimli öğrenmede eğitim verilerinin etiketlenmiş olması gerekmektedir. Verilerin etiketlenmesi işlemini bir uzman gerçekleştirmektedir. Öğrenme işlemi ise ML algoritması tarafından gerçekleştirilmektedir. Böyle bir yöntemi Peter Szolovits büyük bir meydan okuma olarak nitelendirmektedir (Vimla vd., 2009). Bu tür sistemler uzman kişinin geçmişten gelen deneyiminden de faydalanarak veri üzerinde öğrenme gerçekleştirirler. ML, veri kümesi üzerinde genel bir hipotez oluşturmaya dayanan yöntemler içerir. Bu yöntemlerin bir uzmanın deneyimiyle birleştirilmesi ile indaktif bir uzman sistem tasarımına gidilmiştir. Böyle bir sistem, bir uzmanın yer almadığı bir sistemden daha fazla tercih edilmekte ve literatürde en çok vurgulanan yöntemler arasında olmaktadır. Bu durum hibrit sistemlerin daha çok tercih edilmesine sebep olmaktadır.

Geliştirilen sistem için gerekli olan cihaz ve yazılımlar farklı zamanlarda birçok defa test edilmiştir. Yapılan uzun süreli testlerde sistem bileşenlerinin birbirleriyle sorunsuz bir biçimde çalıştığı görülmüştür.

Geliştirilen sistemde elde edilen sonuçlar maddeler halinde aşağıda belirtilmektedir. Buna göre:

- **Hasta verilerinin toplanmasında dial-up bağlantı yerine GPRS teknolojisinin kullanılması sisteme esneklik, maliyet ve zaman açısından olumlu etkiler sağlamıştır.**

- Gelen verinin filtrelenmesi ve sorunsuz verilerin veritabanına kaydedilmesi eğitim verisi içerisindeki gürültülü veri sayısını azaltmaktadır. Bu da gürültüden kaynaklanan hata oranını düşürmektedir.
- Özellik seçimi için ilk olarak aritmi hastalığının belirtisi olabilecek niteliklerin seçilmesi daha sonra öğrenme işlemi sonrasında doğruluk oranını arttıran niteliklerin seçilmesi aritmi hastaları üzerinde geliştirilen sistemin uygulanabilirliğini arttırmaktadır.
- “Sinyal kalitesi” niteliği üzerinde min-max normalizasyon işleminin yapılması bu niteliğin diğer niteliklerin etkisinin önüne geçmesini engelemekte bu da sınıflandırıcının tahmin etmedeki doğruluğunu arttırmaktadır.
- Geliştirilen sistemde eğitim verilerinin etiketlenmesi ve toplanan verinin eğitim verisi için yeterli olup olmadığına karar verilmesi işlemlerini bir insan uzman gerçekleştirmektedir. Bu durum ML’ye dayanan sistemin bir insan uzmanın tecrübesi ile birleştirilmesinin mümkün olabileceğini göstermektedir.
- WEKA gibi bir veri madenciliği yazılımı kullanılarak hesaplamasal hatalardan kaçınılmış olması ve matematiksel işlemler açısından iş yükünün geliştirilen uygulamaların üzerinden alınması sistemin performansını arttırmaktadır.
- Veri kümesi için belirlenen sınıf sayısının üç olması kNN sınıflandırıcının tahmin etmedeki doğruluk oranını arttırmaktadır. Bu durum k değerinin esnekliğini bir nebze de olsa azaltmaktadır.
- Veri toplama işlemi asenkron olarak yapılmaktadır. Bu da aynı anda birden fazla hastanın sisteme bağlanabilmesine olanak sağlamaktadır.

Bu çalışma projesinden elde edilen sonuçlardan sonra sistemin daha fazla geliştirilmesi düşünülmektedir. Buna göre sistem bileşenlerinin birbirlerinden ayrık olması yerine tümleşik bir sistemin geliştirilmesi gerekmektedir. Buna göre bilginin iletildiği ve depolandığı bilgisayar sunucusunun kaldırılıp yerine gömülü sistemlerin kullanılması uygun olmaktadır. Çünkü gömülü sistemler özelleştirilmiş sistemler olmalarının yanı sıra bir bilgisayarın yapabildiği birçok görevi de icra

edebilmektedirler. Buna göre bu teze konu olan çalışma gelecekte de farklı teknoloji ve tekniklerin kullanılmasıyla birlikte geliştirilmeye devam edilecektir.

11. KAYNAKLAR

Adan V, Crown LA. Diagnosis and treatment of sick sinus syndrome. *Am Fam Physician* 2003;67:1725-32.

Alihanka J., Vaahtoranta K., and Saarikivi I.; "A new method for longterm monitoring of the ballistocardiogram, heart rate, and respiration,"; *Am. J. of Physiol.*, vol. 240, pp. R384–R392, 1981.

Alpaydın, E., *Introduction to Machine Learning*, The MIT Press, 2004, Printed and bound in the United States of America. ISBN 0-262-01211-1.

Barsky AJ, Cleary PD, Coeytaux RR, Ruskin JN. Psychiatric disorders in medical outpatients complaining of palpitations. *J Gen Intern Med* 1994;9:306-13.

Barsky AJ, Cleary PD, Sarnie MK, Ruskin JN. Panic disorder, palpitations, and the awareness of cardiac activity. *J Nerv Ment Dis* 1994;182:63-71.

Bennett D.H., *Kardiyak Aritmiler*, Nobel Kitabevi, 2008, Adana, ISBN: 978-605-397-015-6.

Berlinski, David (2000), *The Advent of the Algorithm*, Harcourt Books, ISBN 0-15-601391-6.

Bettstetter C., Vögel H.J., and Eberspacher J., "GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols and Air Interface", *IEEE Communications Surveys*, Vol. 2, No. 3, 1999, pp. 2–14.

Beyer K., Goldstein J., Ramakrishnan R., and Shaft U.: "When Is 'Nearest Neighbor' Meaningful?", *Proceedings of the 7th International Conference on Database Theory*, Jerusalem, Israel, 1999, pages 217-235.

Bigger JT Jr, Reiffel JA. Sick sinus syndrome. *Annu Rev Med* 1979; 30:91-118.

Bishop Christopher M., *Pattern Recognition and Machine Learning*, Springer Science+Business Media, 2006, Printed in Singapore. ISBN: 0-387-31073-8

Blomström Lundqvist et al. ACC/AHA/ESC Guidelines for the Management of Supraventricular Tachyarrhythmias. *JACC* 2003;42:1493-531.

Bonner, Anthony, *The Art and Logic of Ramón Llull: A User's Guide*, Brill, 2007.

Bouckaert R.R.; Frank E.; Hall M.; Kirkby R.; Reutemann P.; Seewald A.; Seuse D., *WEKA Manual for 3-6-0*, December 18, 2008,

<http://prdownloads.sourceforge.net/weka/WekaManual-3.6.0.pdf?download>. [Erişim Tarihi: 24 Haziran 2011]

Brasche G., and Walke B., “Concepts, Services and Protocols of the New GSM Phase 2+ General Packet Radio Service”, IEEE Communications Magazine, Vol. 35, No. 8, 1997, pp. 94–104.

Brink M, Müller CH, Schierz C.; Contact-free measurement of heart rate, respiration rate, and body movements during sleep; Behavior Research Methods 2006 Aug;38(3):511-21.

Brownlee J., eds. (2007), Lazy and Competitive Learning, Technical Report 070508A, Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Australia.

Cfr. Carreras Artau, Tomás y Joaquín. Historia de la filosofía española. Filosofía cristiana de los siglos XIII al XV. Madrid, 1939, Volume I.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press, 2008, Introduction to Information Retrieval. ISBN: 0521865719.

Codd E.; (1993) Providing OLAP (On-Line Analytical Processing) to Users-Analysts: An IT Mandate. Technical report, E.F. Codd and Associates.

Cohen, Jacob (1960), A coefficient of agreement for nominal scales, Educational and Psychological Measurement Vol.20, No.1, pp. 37–46.

Cumhuriyet Üniversitesi Bilgisayar Mühendisliği,
<http://www.cumhuriyet.edu.tr/bolum.php?bolumkodu=331200000&fakultekodu=33&birimtipi=3>. [Erişim Tarihi: 20 Temmuz 2001]

Davies MJ. Pathology of chronic A-V Block. Acta Cardiol 1976;21:19-30.

Değişen Dünya, Tıp ve Teknoloji.
<http://www.cozum.info/haber/page?SYF=Detay&hb=1197>. [Erişim Tarihi: 25 Temmuz 2011]

Dendral. <http://en.wikipedia.org/wiki/Dendral>. [Erişim Tarihi: 08 Haziran 2011].

Dinçkal MH. Sinus nod disfonksiyonu: hasta sinus sendromu. Türkiye Klinikleri. 2006;2(24):7-14.

Domingos P., 2000, A Unified Bias-Variance Decomposition and its Applications, In Proc. 17th International Conf. on Machine Learning, pages 231-238.

Duda RO, Hart PE, Stork DG. Pattern classification, Second edition, New York: Wiley-Interscience; 2001.

Enderle J, Blanchard S, Bronzino J, 2000, Introduction to Biomedical Engineering, Academic Press, San Diego, A.B.D.

ENVITEC, 2008, Digital Pulse Oximeter Module ChipOx User Manual [online], <http://www.envitec.com/download/media/61.pdf/47-07-00330002-h.pdf;jsessionid=676D89A877E2D3FA24BFFBA31FFA1659>. [Eriřim Tarihi: 5 Temmuz 2011]

Erdem A, Arıöz U, Erdem A. Y, Kaymakođlu B, 2005, Tıp Alanında Simülasyon Uygulamaları, 2. Ulusal Tıp Biliřimi Kongresi/Medical Informatics '05 Turkey, p:165

Fak A.S., “Çarpıntı ve Aritmi (Ritim Bozukluđu) Nedir?”, http://www.doktorsitesi.com/yazi/2915/Carpinti-ve-Aritmi-%28Ritim-BozukluGu-%29-Nedir-?a_id=1 [Eriřim Tarihi: 21 Mayıs 2011].

Ferrer MI. The sick sinus syndrome in atrial disease. JAMA.1968;206:645-6.

Freiherr G. The seeds of artificial intelligence: SUMEX-AIM (1980). U.S. G.P.O; DHEW publication no.(NIH) 80-2071. Washington, D.C.; U.S. Dept. of Health, Education, and Welfare, Public Health Service, National Institutes of Health; 1980.

Geman, S., Bienenstock, E., Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural Computation, 4(1), 1-58.

Glossary of Terms, Machine Learning 30, 2-3 (February 1998), 271-274.

Görenek B. vd., Aritmiler Nedenleri, Güncel Tanı ve Tedavi Yöntemleri, Nobel Kitabevi (Ed.), 2010, İstanbul, ISBN: 978-975-420-743-9.

Gwet, K. (2010). Handbook of Inter-Rater Reliability (2.Ed.) ISBN 978-0-9708062-2-2.

Guerrero E. B., Collet C., Adiba M.; THE WHES APPROACH TO DATA WAREHOUSE EVOLUTION .e Gnosis[online], Vol.2Art.2004.

History of artificial intelligence. http://en.wikipedia.org/wiki/History_of_artificial_intelligence. [Eriřim Tarihi: 08 Haziran 2011].

Hurtado C. A., Mendelzon A. O., and Vaisman A. A.; Maintaining Data Cubes under Dimension Updates; In XVth International Conference on Data Engineering (ICDE 99), Sydney, Australia, pages 346–355. IEEE Computer Society, 1999.

Witten I.H., Frank E., 2005, Data mining : practical machine learning tools and techniques – 2nd ed. p. cm. – Morgan Kaufmann series in data management systems. ISBN: 0-12-088407-0.

Institute of Medicine. For the record: protecting electronic health information. Washington, DC: National Academy Press; 1997.

İlerigelen B.; Mutlu H., “EKG Kursu Kitapçığı”, http://www.ctf.edu.tr/stek/EKG_Kurs_Kitap.pdf [Erişim Tarihi: 05 Haziran 2011].

James G. M., 2003, Variance and Bias for General Loss Functions, Machine Learning Volume 51, Number 2, 115-135, DOI: 10.1023/A:1022899518027.

Josephson M., Sinus node function, In: Josephson M. Ed. Clinical Cardiac Electrophysiology. 4th ed. Philadelphia: Lippincott Williams and Wilkins;2008.p.6992.

Kalman JM, Sparks PB. Sinus node dysfunction. In: Kusumoto FM, Goldschlager NF. Eds. Cardiac Pacing for the Clinician. 1st ed. Philadelphia: Lippincott Williams and Wilkins;2001.p.213-228.

Kılıçaslan Y., Artificial Intelligence, Ders Notları, <http://yilmazklicaslan.trakya.edu.tr/teaching/yz2008.ppt>. [Erişim Tarihi: 17 Haziran 2011]

Kılıçaslan Y., Learning from Observations, Ders Notları, <http://yilmazklicaslan.trakya.edu.tr/teaching/learning.ppt>. [Erişim Tarihi: 23 Haziran 2011]

Yılmaz Kılıçaslan, Edip Serdar Güner and Savaş Yıldırım; Learning-based pronoun resolution for Turkish with a comparative evaluation; Computer Speech & Language Volume 23, Issue 3, July 2009, Pages 311-331.

Kocabaş Ş., Yapay Zekaya Giriş, Ders Notları, www.sakirkocabas.com/files/yzgir_1n.rtf, [Erişim Tarihi: 08 Haziran 2011].

Kocamaz A.F., 2007, Tıp Bilişimi ve Uygulamaları, Seminer (Msc.), Trakya Üniversitesi.

Koivistoinen, T. Junnila, S. Varri, A. Koobi, T.; A new method for measuring the ballistocardiogram using EMFi sensors in a normal chair; Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE, on page(s): 2026 – 2029, 1-5 Sept. 2004.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. Proceedings of the Twelfth International Conference on Machine Learning (pp. 313-321). Tahoe City, CA: Morgan Kaufmann.

Kroenke K., Arrington ME, Mangelsdorff AD. The prevalence of symptoms in medical outpatients and the adequacy of therapy. Arch Intern Med 1990; 150:1685-9.

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33, 159–174.

Lehmann, E. L.; Casella, George (1998). Theory of Point Estimation (2nd ed.). New York: Springer. ISBN 0-387-98502-6.

Lindsay RK, Buchanan BG, Feigenbaum EA, Lederberg J. Applications of artificial intelligence for organic chemistry: the DENDRAL Project. New York: McGraw-Hill; 1980.

Lipschutz S., Olasılık Teori ve Problemleri, Nobel Yayın Dağıtım, 1990, ISBN O-07-037982-3.

Maleq K. vd., K-Nearest Neighbor Classification on Spatial Data Streams Using P-Trees, 6. Pasifik Asya Knowledge discovery and Data Mining Conference PAKKDD'02, Taiwan, 2002, s. 517-518.

M.E.B. Sağlık İşleri Dairesi Başkanlığı, 2010, Dolaşım Sistemi, <http://sdb.meb.gov.tr/dersler/modul/Alanlar Ortak Modülleri/Anatomi Dersi Modülleri/6. MODÜL DOLAŞIM SİSTEMİ.pdf> [Erişim Tarihi: 11 Mayıs 2011].

Meyer M., "TCP performance over GPRS", WCNC 1999 - IEEE Wireless Communications and Networking Conference, no. 1, September 1999, pp. 1248-1252

Miller RA, Maserie F. The demise of the Greek oracle model for medical diagnosis systems. Methods of Information in Medicine 1990;29:1—2.

Miller RA, Pople HE, Myers JD. Internist-1: an experimental computer-based diagnostic consultant for general internal medicine. New England Journal of Medicine 1982;307(8): 468—76.

Mitchell, T.M., Machine Learning, McGraw-Hill Science/Engineering/Math, 1 March 1997, ISBN 0070428077, 432 pages.

Myerburg RJ. Sudden cardiac death: exploring the limits of our knowledge. J Cardiovasc Electrophysiol 2001;12:369-81.

Nellcor N-200 pulse oximetry note number 6. C-LOCK ECG synchronization principles of operation. Calif, Nellcor: Hayward, 1988.

Nilsson, N.J., Introduction to Machine Learning An Early Draft of a Proposed Textbook, Robotics Laboratory, Department of Computer Science, Stanford University, 4 December 1996.

Noy NF, Crubezy M, Ferguson RW, Knublauch H, Tu SW, Vendetti J, et al. Prote'ge'-2000: an open-source ontology development and knowledge-acquisition environment. In: Musen MA, Friedman CP, Teich JM, editors. Proceedings of the 27th annual symposium of the American Medical Informatics Association AMIA 2003: biomedical and health informatics: from foundations to applications. Bethesda: American Medical Informatics Association; 2003. p. 953

Occam's razor, http://en.wikipedia.org/wiki/Occam's_razor. [Erişim Tarihi: 28 Haziran 2011]

Othman, A.K.; Zakaria, M.; Ab Hamid, K., "TCP performance measurement in different GPRS network scenarios", Applied Electromagnetics, 2007. APACE 2007. Asia-Pacific Conference on 4-6 Dec. 2007, pp. 1 5.

Özkan Y., Veri Madenciliği Yöntemleri, 2008, Papatya Yayıncılık Eğitim, İstanbul, ISBN 978-975-6797-82-2.

Parmigiani, G.; Inoue, L., "Decision Theory: Principles and Approaches", WILEY Series, 2009. ISBN 978 0 471 49657 1.

Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.

Sakane, A., Tsuji, T., Tanaka, Y., Saeki, N. & Kawamoto, M. Estimating arterial wall impedance using a plethysmogram. IECON03 29th Annual Conference of the IEEE Industrial Electronics Society IEEE Cat No03CH37468 1, 580-585 (2003).

Saygılı S., Erdal M., 2001, Yeni Teknolojilerin Sağlık Sektöründeki Etkileri ve İnternet Kullanımı, 7. Türkiye'de İnternet Konferansı, 1-3 Kasım 2001 İstanbul.

Schölkopf, B., Smola, A.J., 2002. Learning with Kernels. MIT Press, Cambridge, MA.

Shirouzu, S.; Sugano, H.; Hara, K.; A new method for unconstrained heart rate monitoring; Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE, on page(s): 3960 - 3961 vol.4, 2001.

Shortliffe EH. Computer-based medical consultations: MYCIN. New York: Elsevier; 1976.

Silahtaroglu G., Kavram ve Algoritmalarıyla Temel Veri Madenciliği, 2008, Papatya Yayıncılık Eğitim, İstanbul, ISBN 978-975-6797-81-5.

Sridharan NS. Guest editorial. Artificial Intelligence 1978:11 (1—2);1—4.

Straus HC, Prytowsky EN, Scheinman MM. Sinoatrial and atrial electrogenesis. Prog Cardiovasc Dis 1977;19:385-404.

Supraventriküler Taşikardi, <http://turkaritmi.com/hastalar.php?id=42&tum=tamam> [Erişim Tarihi: 21 Mayıs 2011].

Sweeney MO. Sinus node dysfunction. In: Zipes DP, Jalife J, eds. Cardiac Electrophysiology from Cell to Bedside. 4th ed. Philadelphia:Saundes;2004. P.879-883.

Tanawut Tantimongcolwat, Thanakorn Naenna, Chartchalerm Isarankura-Na-Ayudhya, Mark J. Embrechts and Virapong Prachayasittikul; Identification of ischemic heart disease via machine learning analysis on magnetocardiograms; Computers in Biology and Medicine Volume 38, Issue 7, July 2008, Pages 817-825.

TELIT, 2011, Telit GM862 GSM Datasheet, <http://www.telit.com/module/infopool/download.php?id=165>. [Erişim Tarihi: 5 Temmuz 2011]

Thrall JH, Directions in radiology for the next millennium, AJR Am J Roentgenol. 1998 Dec;171(6):1459-62.

Tunkelang D., Making the Nearest Neighbor Meaningful, 2002.

Umut İ., Dijital Sinyal İşleme Yazılımı Geliştirilmesi ve Geliştirilen Bu Yazılım İle Elektroensefalografî Kayıtları Kullanılarak Tıkayıcı Uyku Apnesi Bulunan Bireylerin Apnesiz Bireylerden Ayırt Edilmesi, 2011, Doktora Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı.

Vijayakumar S., The Bias-Variance Tradeoff, <http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf>. [Erişim Tarihi: 01 Temmuz 2011]

Vimla L. Patel, Edward H. Shortliffe, Mario Stefanelli, Peter Szolovits, Michael R. Berthold, Riccardo Bellazzi, Ameen Abu-Hanna; The coming of age of artificial intelligence in medicine; Artificial Intelligence in Medicine; May 2009 (Vol. 46, Issue 1, Pages 5-17).

Weber BE, Kapoor WN. Evaluation and outcomes of patients with palpitations. Am J Med 1996; 100:138-48.

Weiss SM, Kulikowski CA, Amarel S, Safir A. A model-based method for computer-aided medical decision making. Artificial Intelligence 1978; 11:145—72.

WEKA, Available: <http://www.cs.waikato.ac.nz/ml/weka/>. [Erişim Tarihi: 24 Haziran 2011]

World Health Organization, (2004) The world health report 2004, http://www.who.int/whr/2004/annex/topic/en/annex_2_en.pdf [Erişim Tarihi: 2 Şubat 2011].

Yapay Zekâ. http://tr.wikipedia.org/wiki/Yapay_zekâ. [Erişim Tarihi: 17 Haziran 2011]

Zeldis SM, Levine BJ, Michelson EL, et al. Cardiovascular complaints. Correlation with cardiac arrhythmias on 24-hour electrocardiographic monitoring. Chest 1980;78:456-61.

Zipes DP, Camm AJ, Borggrefe M, Buxton AE, Chaitman B, Fromer M, et al. ACC/AHA/ESC 2006 Guidelines for Management of Patients With Ventricular Arrhythmias and the Prevention of Sudden Cardiac Death. *Circulation* 2006;114:385-484.

11th Conference on Artificial Intelligence in Medicine (AIME 07), <http://www.aimedicine.eu/AIME07>. [Eriřim Tarihi: 14 Haziran 2011]

13th Conference on Artificial Intelligence in Medicine (AIME 11), <http://www.aimedicine.info/AIME11>. [Eriřim Tarihi: 14 Haziran 2011]

EK-A

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;
using System.Text;
using System.Diagnostics;
using System.Data;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
using HospitalApplication.Model;

namespace HospitalApplication.Needs
{
    public class HeartSignalLearning
    {
        private class Accuracy
        {
            private string f_measure;
            private string roc_area;

            public Accuracy()
            {
                f_measure = "";
                roc_area = "";
            }

            public string F_Measure
            {
                get { return f_measure; }
                set { f_measure = value; }
            }

            public string RocArea
            {
                get { return roc_area; }
                set { roc_area = value; }
            }
        }

        private class Classes
        {
            private Accuracy normal;
            private Accuracy critical;
            private Accuracy very_critical;

            public Classes()
            {
                normal = new Accuracy();
                critical = new Accuracy();
                very_critical = new Accuracy();
            }
        }
    }
}
```

```

public Accuracy Normal
{
    get { return normal; }
    set { normal = value; }
}

public Accuracy Critical
{
    get { return critical; }
    set { critical = value; }
}

public Accuracy VeryCritical
{
    get { return very_critical; }
    set { very_critical = value; }
}
}

private class MLStatistics
{
    private Classes c1;
    private string kappa;
    private string output;

    public MLStatistics()
    {
        output = "";
        kappa = "";
        c1 = new Classes();
    }

    public MLStatistics(string text)
    {
        output = text;
        c1 = new Classes();
        Evaluate();
    }

    public string Kappa
    {
        get { return kappa; }
        set { kappa = value; }
    }

    public Classes Classes
    {
        get { return c1; }
        set { c1 = value; }
    }

    public string Output
    {
        get { return output; }
        set
        {
            output = value;
            Evaluate();
        }
    }
}

```

```

    }

    private void Search(string value, ref string fmeasure, ref string
roc)
    {
        string ou = "";
        StringBuilder pattern = new StringBuilder(@"[?01]\.?\d?\d?\d?\d?
*?[?01]\.?\d?\d?\d?\d? *");
        pattern.Append(value);

        ou = (new
Regex(pattern.ToString())).Match(output).Value.Replace(value, "").Trim();
        fmeasure = (new Regex(@"^[01]\.?\d?\d?\d?\d?")).Match(ou).Value;
        roc = (new Regex(@"
*?[?01]\.?\d?\d?\d?\d?$")).Match(ou).Value.Trim();
    }

    private void Evaluate()
    {
        string fmeasure = "", roc = "";

        kappa = (new Regex(@"Kappa statistic *-
?[01]\.?\d?\d?\d?\d?")).Match(output).Value.Replace("Kappa statistic",
 "").Trim();

        Search("normal", ref fmeasure, ref roc);
        cl.Normal.F_Measure = fmeasure;
        cl.Normal.RocArea = roc;

        Search("critical", ref fmeasure, ref roc);
        cl.Critical.F_Measure = fmeasure;
        cl.Critical.RocArea = roc;

        Search("very_critical", ref fmeasure, ref roc);
        cl.VeryCritical.F_Measure = fmeasure;
        cl.VeryCritical.RocArea = roc;
    }
}

private string patient_id;
private string Path;

public HeartSignalLearning()
{
}

public HeartSignalLearning(string p_id, string path)
{
    this.patient_id = p_id;
    this.Path = path;
}

public string PatientID
{
    get { return this.patient_id; }
    set { this.patient_id = value; }
}

private void CreateFile()
{

```



```

StringBuilder str = new StringBuilder();
StringBuilder s = new StringBuilder();

using (TextWriter w = new StreamWriter(Path + "\\training.arff",
false))
{
    w.WriteLine("@relation heart_signal");
    w.WriteLine("");
    for (int i = 1; i <= 123; i++)
    {
        str.Append("@attribute HS");
        str.Append(i.ToString());
        str.Append(" integer");
        w.WriteLine(str.ToString());
        str.Clear();
    }
    str.Clear();
    str = null;

    w.WriteLine("@attribute SignalQualify integer");
    w.WriteLine("@attribute SPO2 integer");
    w.WriteLine("@attribute HeartBeatRate integer");
    w.WriteLine("@attribute class {normal, critical,
very_critical}");
    w.WriteLine("");
    w.WriteLine("@data");

    using (SqlDataReader sdr = (new Database()).ExecuteReader("select
data, patient_data_evaluate from tbl_movement where patient_data_evaluate is not
null and patient_data_evaluate <> '---' and patient_id = " +
this.patient_id.ToString()))
    {
        object obj = null;
        while (sdr.Read())
        {
            s.Append(sdr["data"].ToString().Replace(' ', ','));
            obj = sdr["patient_data_evaluate"];
            if(obj.Equals("Normal"))
                s.Append("normal");
            else if (obj.Equals("Kritik"))
                s.Append("critical");
            else
                s.Append("very_critical");
            w.WriteLine(s.ToString());
            s.Clear();
        }
        s.Clear();
        s = null;
    }

    w.Flush();
    w.Close();
}

public string EvalStatement()
{
    CreateFile();

    ProcessStartInfo weka = new ProcessStartInfo();

```

```

weka.CreateNoWindow = true;
weka.UseShellExecute = false;
weka.RedirectStandardOutput = true;
weka.WindowStyle = ProcessWindowStyle.Hidden;
weka.FileName = "java";
weka.WorkingDirectory = Path;
weka.Arguments = "-cp weka.jar weka.classifiers.lazy.IBk -t
training.arff -x 10 -v -o -i";

using (Process p = Process.Start(weka))
{
    string result = GetEvaluate(p.StandardOutput.ReadToEnd());
    p.WaitForExit();
    return result;
}

private string GetEvaluate(string p)
{
    if (p == "")
        return "Yeterli eğitim verisi bulunmamaktadır.";

    StringBuilder msg = new StringBuilder();
    MLStatistics mls = new MLStatistics(p);

    double Kappa = Convert.ToDouble(mls.Kappa);
    string ROCArea_Normal = mls.Classes.Normal.RocArea;
    string FMeasure_Normal = mls.Classes.Normal.F_Measure;
    string ROCArea_Kritik = mls.Classes.Critical.RocArea;
    string FMeasure_Kritik = mls.Classes.Critical.F_Measure;
    string ROCArea_CokKritik = mls.Classes.VeryCritical.RocArea;
    string FMeasure_CokKritik = mls.Classes.VeryCritical.F_Measure;

    msg.Append("<table style=\"border: 1px; border-color: #9F2425;
border-style:solid;\"><tr><td colspan=\"2\" align=\"center\" style=\"background-
color: #7C0000; color:white;\">Sistem Değerlendirmesi</td></tr>");
    msg.Append("<tr><td valign=\"middle\"><b>Yeterli tanının konulmadığı
sınıflar</b></td><td>");

    if (ROCArea_Normal != "?")
    {
        if (Convert.ToDouble(ROCArea_Normal) < 0.5 &&
Convert.ToDouble(FMeasure_Normal) < 0.5)
            msg.Append("Normal <br/>");
    }
    else if(Convert.ToDouble(FMeasure_Normal) < 0.5)
        msg.Append("Normal <br/>");

    if (ROCArea_Kritik != "?")
    {
        if (Convert.ToDouble(ROCArea_Kritik) < 0.5 &&
Convert.ToDouble(FMeasure_Kritik) < 0.5)
            msg.Append("Kritik <br/>");
    }
    else
        msg.Append("Kritik <br/>");

    if (ROCArea_CokKritik != "?")
    {

```

```

        if (Convert.ToDouble(ROCArea_CokKritik) < 0.5 &&
Convert.ToDouble(FMeasure_CokKritik) < 0.5)
            msg.Append("Çok Kritik <br/>");
    }
    else
        msg.Append("Çok Kritik <br/>");

    msg.Append("</td></tr><tr><td colspan=\"2\" style=\"background-color:
#7C0000;\"></td></tr>");

    msg.Append("<tr><td colspan=\"2\"><b>Sistemin Öğrenme
Durumu</b></td></tr><tr><td colspan=\"2\">");
    if (Kappa < 0)
        msg.Append("Eğitim verileri arasında uyumsuzluk bulunmaktadır.
Bu nedenle sistem öğrenememektedir. Kappa değeri : " + Kappa.ToString());
    else if (Kappa == 0)
        msg.Append("Eğitim verileri öğrenme gerçekleştirememektedir. Bu
nedenle sistem öğrenememektedir. Kappa değeri : " + Kappa.ToString());
    else if (Kappa > 0 && Kappa <= 0.2)
        msg.Append("Sistem çok önemsiz derecede öğrenme
gerçekleştirmektedir. Kappa değeri : " + Kappa.ToString());
    else if (Kappa > 0.2 && Kappa <= 0.4)
        msg.Append("Sistem önemsiz derecede öğrenme gerçekleştirmektedir.
Kappa değeri : " + Kappa.ToString());
    else if (Kappa > 0.4 && Kappa <= 0.6)
        msg.Append("Sistem kabul edilebilir derecede öğrenme
gerçekleştirmektedir. Kappa değeri : " + Kappa.ToString());
    else if (Kappa > 0.6 && Kappa <= 0.8)
        msg.Append("Sistem önemli derecede öğrenme gerçekleştirmektedir.
Kappa değeri : " + Kappa.ToString());
    else
        msg.Append("Sistem mükemmel derecede öğrenme
gerçekleştirmektedir. Kappa değeri : " + Kappa.ToString());

    msg.Append("</td></tr></table>");

    return msg.ToString();
}
}
}
}
}

```

EK-B

```

using System;
using System.Windows;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.Runtime.Remoting.Messaging;
using System.Data;

namespace GPRSDataListener
{
    public partial class MainWindow : Window
    {
        private Socket server = null;
        private ManualResetEvent connectDone = null;
        private AsyncCallback ReceiveCallback = null;
        private delegate void InvokeDelegate(string str);
        private delegate void DataHandler(Queue<byte> data);
        private Dictionary<Socket, Queue<byte>> workerSocketList;

        // Asenkron çağrılar arasında değerleri geçirmek için bir durum nesnesi
        oluşturma
        private class StateObject
        {
            public Socket workSocket = null;
            public byte[] dataBuffer = new byte[256];
        }

        private void StartListenerService()
        {
            connectDone = new ManualResetEvent(true);
            workerSocketList = new Dictionary<Socket, Queue<byte>>();

            try
            {
                // Socket oluştur
                server = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);

                // Socketi herhangi bir ip adresi ve porta bağla
                server.Bind(new IPEndPoint(IPAddress.Any,
Convert.ToInt32(textBox1.Text)));

                // Maximum bağlantı sayısı ile dinlemeye başla
                server.Listen(Properties.Settings.Default.MaxClientCount);

                // Bağlantı isteklerini kabul et
                server.BeginAccept(new AsyncCallback(OnConnection), server);

                // Bir sinyal alana kadar güncel threadı beklet
            }
        }
    }
}

```

```

        connectDone.WaitOne();
    }
    catch (SocketException se)
    {
        Log.Write(se.ErrorCode, se);
    }
}

// Clienttan gelen bağlantıyı sockette bağlama
private void OnConnection(IAAsyncResult asyn)
{
    try
    {
        // Bağlantıyı bitir(tamamla)
        Socket worker = server.EndAccept(asyn);

        workerSocketList.Add(worker, new Queue<byte>());

        // Bağlantı kurulduktan sonra uygulama threadına signal gönder
        connectDone.Set();

        WaitForData(worker);

        // Bağlantı kabul etmeye devam et
        server.BeginAccept(new AsyncCallback(OnConnection), null);
    }
    catch (ObjectDisposedException oe)
    {
        Log.Write(oe);
        //System.Diagnostics.Debugger.Log(0, "1", "\n OnClientConnection:
Socket has been closed\n");
    }
    catch (SocketException se)
    {
        Log.Write(se.ErrorCode, se);
    }
}

private void WaitForData(Socket worker)
{
    try
    {
        if (ReceiveCallback == null)
        {
            // Bağlı bulunan client tarafından herhangi bir veri gönderme
işlemi yoksa callback fonksiyonu tanımla
            ReceiveCallback = new AsyncCallback(OnDataReceive);
        }

        // Durum nesnesi oluşturma
        StateObject state = new StateObject();
        state.workSocket = worker;

        // Clienttan veri alma işlemine başla
        worker.BeginReceive(state.dataBuffer, 0, state.dataBuffer.Length,
SocketFlags.None, ReceiveCallback, state);
    }
    catch (SocketException se)
    {
        Log.Write(se.ErrorCode, se);
    }
}

```

```

    }
}

// Clienttan verinin alındığı callback fonksiyonu
private void OnDataReceive(IAsyncResult asyn)
{
    try
    {
        // Asenkron durum nesnesinden StateObject ve client elde ediliyor
        StateObject state = (StateObject)asyn.AsyncState;
        Socket client = state.workSocket;

        // Client tarafından stream'e yazılan karakterlerin sayısı elde
        // ediliyor. Aynı zamanda clienttan veri okunuyor
        int bytesRead = client.EndReceive(asyn);
        byte last=10;
        lock (((ICollection)workerSocketList).SyncRoot)
        {
            if (workerSocketList.ContainsKey(client))
            {
                int i = 0;
                Queue<byte> q = workerSocketList[client];
                foreach (byte b in state.dataBuffer)
                {
                    last = b;
                    if (bytesRead < i)
                        break;
                    q.Enqueue(b);
                    ++i;
                }
                workerSocketList[client] = q;
            }
        }

        // Alınan byte sıfırsa bağlantı kopmuş olmalı
        if (bytesRead == 0 || last == 0)
        {
            DataHandler handle = new DataHandler(ProcessData);
            handle.BeginInvoke(workerSocketList[client], new
AsyncCallback(AfterProcessData), null);
            workerSocketList.Remove(client);
            client.Close();
            return;
        }

        // Socket üzerindeki veri için beklemeye devam ediliyor. Yani
        // kalan veri için bekleme yapılıyor
        WaitForData(state.workSocket);
    }
    catch (ObjectDisposedException oe)
    {
        Log.Write(oe);
    }
    catch (SocketException se)
    {
        Log.Write(se.ErrorCode, se);
    }
    catch (Exception ex)
    {
        Log.Write(ex);
    }
}

```

```

    }
}

private void ProcessData(Queue<byte> data)
{
    StringBuilder s = new StringBuilder();
    byte[] b = data.ToArray();

    for (int i = 1; i < b.Length; i++)
        s.Append(b[i] + " ");

    DataTable dt = Database.ExecuteQuery("SELECT id, doctor_id FROM
tbl_patient WHERE device_id = " + b[0].ToString(), QueryType.Dataset);
    if (dt.Rows.Count > 0)
    {
        StringBuilder values = new StringBuilder("INSERT INTO
tbl_movement (patient_id, device_id, doctor_id, data, creation_date) VALUES(");
        values.Append(dt.Rows[0]["id"]);
        values.Append(",");
        values.Append(b[0]);
        values.Append(",");
        values.Append(dt.Rows[0]["doctor_id"] == DBNull.Value ? "NULL" :
dt.Rows[0]["doctor_id"]);
        values.Append(",");
        values.Append(s.ToString());
        values.Append(",");
        values.Append(DateTime.Now.ToString("G",
System.Globalization.DateTimeFormatInfo.InvariantInfo));
        values.Append(")");
        Database.ExecuteQuery(values.ToString(),
QueryType.ExecuteNonQuery);

        values.Clear();
        values = null;

        s.Clear();
        s = null;
    }
}

private void AfterProcessData(IAsyncResult result)
{
    AsyncResult async = (AsyncResult)result;
    DataHandler hand = (DataHandler)async.AsyncDelegate;
    hand.EndInvoke(result);
}

private void StopListenerService()
{
    foreach (Socket s in workerSocketList.Keys)
    {
        if (s != null)
            s.Close();
    }

    if (server != null)
        server.Close();

    connectDone.Set();
    connectDone = null;
}

```

```
        workerSocketList.Clear();  
        workerSocketList = null;  
    }  
}
```


ÖZGEÇMİŞ

22.11.1982 yılında Malatya'da doğdum. İlkokul, ortaokul ve lise eğitimimi Malatya'da tamamladım. 1999 yılında Kubilay Lisesinden mezun olduktan sonra 2000-2004 yılları arasında Fırat Üniversitesi Bilgisayar Mühendisliği bölümünde öğrenim gördüm. 2006-2008 yılları arasında GOSB Teknoparkta RFID teknolojisi alanında faaliyet gösteren STS Technology bünyesinde yazılım uzmanı olarak çalıştım. 2008 yılında Kırklareli Üniversitesi Teknik Bilimler MYO Bilgisayar Programcılığı programında Öğretim Görevlisi olarak çalışmaya başladım. Halen Kırklareli Üniversitesi Teknik Bilimler MYO'da Öğretim Görevlisi olarak çalışmaktayım.