

**DALGACIK DÖNÜŐÜMÜ İLE EKG SİNYALLERİNİN İŐLENMESİ VE
ÖZELLİK ÇIKARIMI**

AŐkın PEKÇAKAR

HAZİRAN 2008

**DALGACIK DÖNÜŞÜMÜ İLE EKG SINYALLERİNİN İŞLENMESİ VE
ÖZELLİK ÇIKARIMI**

Aşkın PEKÇAKAR

**DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK EĞİTİMİ ANABİLİM DALINDA
YÜKSEK LİSANS
DERECESİ İÇİN GEREKLİ ÇALIŞMALARI YERİNE GETİRİLEREK
ONAYA SUNULAN TEZ**

HAZİRAN 2008

Fen Bilimleri Enstitüsü'nün Onayı

Prof. Dr. Demet KAYA

Enstitü Müdürü

Bu tezin Yüksek Lisans Derecesinde bir tez olarak gerekli çalışmaları yerine getirdiğini onaylarım.

Yrd. Doç. Dr. Ali ÖZTÜRK

Elektrik Eğitimi Anabilim Dalı Başkanı

Bu tezin Yüksek Lisans Derecesinde bir tez olarak onaylanması, düşüncemize göre, amaç ve kalite olarak tamamen uygundur.

Yrd. Doç. Dr. Pakize ERDOĞMUŞ

Tez Danışmanı

Jüri Üyeleri

1.
2.
3.

ABSTRACT**ECG SIGNALS PROCESSING WITH WAVELET AND FEATURE
EXTRACTION****PEKÇAKAR, AŞKIN**

Master Thesis, Department of Electrical Education

Advisor: Assist. Prof. Dr. Pakize ERDOĞMUŞ

JUNE 2008, 79 pages

In this study, Normal, Rbbb, Lbbb and Pace ECG data was examined. Wavelet and Fourier Transform has been used for Feature Extraction. By this way, classification has been realised the shortest time and the less data size.

Feature Vectors extracted from original signals with Wavelet Transform, have been classified 72 % accuracy. It has also taken successfully results with FFT. But Feature Vector size has been approximately three times bigger than Wavelet Transform and the classification time has increased approximately three times.

Keywords: Wavelet Transform, Signal Analysis, ECG, Feature Extraction

ÖZET

DALGACIK DÖNÜŞÜMÜ İLE EKG SİNYALLERİNİN İŞLENMESİ VE ÖZELLİK ÇIKARIMI

PEKÇAKAR, AŞKIN

Yüksek Lisans, Elektrik Eğitimi Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Pakize ERDOĞMUŞ

HAZİRAN 2008, 79 Sayfa

Bu çalışmada Normal, Rbbb, Lbbb ve Pace olmak üzere dört grup EKG datası incelenmiştir. Dataların dönüşüm işlemlerinde Dalgacık ve Fourier dönüşümleri kullanılmıştır. Bilgilerin daha az veri ile temsil edilmesiyle sınıflandırma işlemleri çok kısa sürede yapılabilir hale gelmiştir.

Dalgacık Dönüşümü ile ham datalardan elde edilmiş özellik vektörleri ortalama %72 gibi bir başarımla sınıflandırılmışlardır. Fourier Dönüşümleri ile de başarılı sonuçlar elde edilmiş, fakat özellik vektörü veri sayısı dalgacık dönüşümüne göre üç kat daha fazla olduğundan hesap yükü fazlalaşmış ve sınıflandırma süresi yaklaşık üç kat artmıştır.

Anahtar Kelimeler: Dalgacık Dönüşümü, Sinyal Analizi, EKG, Özellik Çıkarımı.

Kızlarım Ayşe Sümeyra ve Nalan Elif'e ithaf ediyorum.

TEŞEKKÜR

Tez danışmanlığımı üstlenerek araştırma konusunun seçimi, yürütülmesi ve sunuma hazırlanması sırasında, değerli bilimsel görüş ve önerilerinden yararlandığım Yrd. Doç. Dr. Pakize ERDOĞMUŞ'a teşekkür eder, saygılarımı sunarım.

Tez çalışmam sırasında zaman zaman bilgi ve tavsiyelerinden istifade ettiğim Yrd. Doç. Dr. Recep DEMİRCİ'ye, Yrd. Doç. Dr. İbrahim ŞAHİN'e, Yrd. Doç. Dr. Mehmet ENGİN'e, Yrd. Doç. Dr. Ali ÖZTÜRK'e öğretim görevlileri Erkan Zeki ENGİN'e, İsmail KOYUNCU'ya, Ferzan KATIRCIOĞU'na ve tashih konusunda yardımcı olan Mustafa SÖYLEMEZ'e teşekkür ederim.

Tezin hazırlanması sırasında her konuda fedakarlık gösteren, maddi manevi yardımlarını esirgemeyen eşime, anneme, babama, Abdullah ÜNAL'a teşekkürlerimi sunarım.

İÇİNDEKİLER

ABSTRACT	iii
ÖZET	iv
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	x
TABLolar DİZİNİ	xii
1. GİRİŞ	1
1.1. Konunun Önemi	1
1.2. Çalışmanın Amacı	1
1.3. Dataların Elde Edilmesi	1
1.4. Özellik Çıkarma İşlemi	3
1.5. Sınıflandırma İşlemi	3
2. LİTERATÜR ÖZETİ	4
3. ELEKTROKARDİYOGRAM (EKG)	6
3.1. Kalp İle İlgili Genel Bilgiler	6
3.2. Elektrokardiyogram (EKG)	7
3.3. Derivasyonlar	9
3.3.1. Çevre derivasyonları	9
3.3.2. Göğüs Derivasyonları	10
3.4. EKG Vuruları	11
3.4.1. Normal Vuru (N)	11
3.4.2. Yapay Vuru (P)	12
3.4.3. Sağ Dal Blok Vurusu (RBBB)	13
3.4.4. Sol dal Blok Vurusu (LBBB)	14
4. DÖNÜŞÜM YÖNTEMLERİ	16
4.1. Fourier Dönüşümleri	16
4.2. Dalgacık (wavelet) Dönüşümü	19

4.2.1. Sürekli Dalgacık Dönüşümü (CWT)	20
4.2.3. Ayırık Dalgacık Dönüşümü(DWT)	21
5. YAPAY SİNİR AĞLARI (YSA)	24
5.1. Yapay Sinir Ağlarını Oluşturan Kısımlar	25
5.1.1. Girişler	26
5.1.2. Ağırlıklar	27
5.1.3. Toplama İşlevi.....	27
5.1.4. Etkinlik İşlevi.....	27
5.1.5. Ölçekleme ve Sınırlama	28
5.1.6. Çıkış İşlevi	28
5.1.7. Öğrenme.....	28
5.2. Yapılarına Göre Yapay Sinir Ağları	30
5.2.1. İleri Beslemeli Ağlar	30
5.2.2. Geri Beslemeli Ağlar	31
5.3. Geri Yayılımlı Öğrenme	33
5.4. Geri Yayılım Algoritması	35
6. DÖNÜŞÜM İŞLEMLERİ VE TEST SONUÇLARI	36
6.1. Çalışmada Kullanılan Datalar	36
6.2. Öznitelik Vektörlerinin Çıkarılması	36
6.2.1. Dalgacık Dönüşümü (Wavelet) İle Özellik Çıkarımı	39
6.2.2. Fourier Dönüşümü ile Özellik Çıkarımı	47
6.3. Yapay Sinir Ağları İle Sınıflandırma.....	52
6.3.1. Yapay Sinir Ağının Eğitimi	52
6.3.2. Yapay Sinir Ağları ile Sınıflandırma	58
7. SONUÇ VE GELECEKTE YAPILABİLECEK ÇALIŞMALAR.....	65
KAYNAKÇA.....	67

EKLER.....	71
Ek-A: Dataların Elde Edilmesi İçin Yazılan Program Kodu	71
Ek-B: Dalgacık Dönüşümü Yapan Program(Eğitim için)	71
Ek-C: Dalgacık Dönüşümü Yapan Program(Test için)	72
Ek-D: Fourier Dönüşümü Yapan Program(Eğitim için).....	73
Ek-E: Fourier Dönüşümü Yapan Program(Test için)	74
Ek-F: Yapay Sinir Ağlarında Eğitmeyi Yapan Program(Dalgacık Dönüşümü için) 75	
Ek-G: Yapay Sinir Ağlarında Eğitmeyi Yapan Program(Fourier Dönüşümü için).. 75	
Ek-H: Yapay Sinir Ağlarında Sınıflandırmayı Yapan Program(Dalgacık Dönüşümü için)	76
Ek-I: Yapay Sinir Ağlarında Sınıflandırmayı Yapan Program(Fourier Dönüşümü için)	78

ŞEKİLLER DİZİNİ

Şekil 1.1. Datanın Matlab’da Çizdirilmiş Hali.....	2
Şekil 3.1. Kalbin Yapısı	6
Şekil 3.2. Kalpte Elektrik İletim Yolları ve EKG Dalga Şeklinin Oluşumu	8
Şekil 3.3. EKG Dalgalarının Genlik ve Süre Değerleri	8
Şekil 3.4. Bipolar Çevre Derivasyonları	9
Şekil 3.5. Unipolar Kuvvetlendirilmiş Çevre Derivasyonları.....	10
Şekil 3.6. Göğüs Derivasyonları	11
Şekil 3.7. Normal Vuru	12
Şekil 3.8. Yapay Vuru.....	13
Şekil 3.9. Sağ Dal Blok Vurusu	14
Şekil 3.10. Sol Dal Blok Vurusu.....	15
Şekil 4.1. Fourier Dönüşümü	17
Şekil 4.2. Kısa Zaman Fourier Dönüşümü.....	18
Şekil 4.3. Dalgacık Dönüşümü	19
Şekil 4.4. (a) Daubechies10 (b) Meksikalı Şapkası	21
Şekil 4.5 Dalgacık Ağacı.....	22
Şekil 5.1 Bir Yapay Sinir Hücresi.....	25
Şekil 5.2. Etkinlik Fonksiyonlarından Bazıları	26
Şekil 5.3. Danışmanlı Öğrenme	29
Şekil 5.4. Danışmansız Öğrenme.....	29
Şekil 5.5. Çok Katmanlı İleri Beslemeli YSA	31
Şekil 5.6. Hopfield Ağı	32
Şekil 5.7. Gizli Katmanlı İleri Beslemeli Geri Yayılım Ağ Yapısı	34
Şekil 6.1. Özellik Çıkarımı Yapılacak “Normal” Data	37
Şekil 6.2. Özellik Çıkarımı Yapılacak “Pace” Data.....	37
Şekil 6.3. Özellik Çıkarımı Yapılacak “Rbbb” Data.....	38
Şekil 6.4. Özellik Çıkarımı Yapılacak “Lbbb” Data.....	38
Şekil 6.5. Normal Data 1. Seviye Detay Katsayıları.....	39
Şekil 6.6. Normal Data 2. Seviye Detay Katsayıları.....	40
Şekil 6.7. Normal Data 3. Seviye Detay Katsayıları.....	40
Şekil 6.8. Normal Data 4. Seviye Detay Katsayıları.....	41

Şekil 6.9. Pace Data 1. Seviye Detay Katsayıları	41
Şekil 6.10. Pace Data 2. Seviye Detay Katsayıları	42
Şekil 6.11. Pace Data 3. Seviye Detay Katsayıları	42
Şekil 6.12. Pace Data 4. Seviye Detay Katsayıları	43
Şekil 6.13. Rbbb Data 1. Seviye Detay Katsayıları	43
Şekil 6.14. Rbbb Data 2. Seviye Detay Katsayıları	44
Şekil 6.15. Rbbb Data 3. Seviye Detay Katsayıları	44
Şekil 6.16. Rbbb Data 4. Seviye Detay Katsayıları	45
Şekil 6.17. Lbbb Data 1. Seviye Detay Katsayıları.....	45
Şekil 6.18. Lbbb Data 2. Seviye Detay Katsayıları.....	46
Şekil 6.19. Lbbb Data 3. Seviye Detay Katsayıları.....	46
Şekil 6.20. Lbbb Data 4. Seviye Detay Katsayıları.....	47
Şekil 6.21. Normal Datanın Fourier Dönüşümü	48
Şekil 6.22. Pace Datanın Fourier Dönüşümü	48
Şekil 6.23. Rbbb Datanın Fourier Dönüşümü	49
Şekil 6.24. Lbbb Datanın Fourier Dönüşümü	49
Şekil 6.25. Normal Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü.....	50
Şekil 6.26. Pace Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü	50
Şekil 6.27. Rbbb Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü	51
Şekil 6.28. Lbbb Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü	51
Şekil 6.29. 21 Giriş, 4 Çıkışlı, 1 Gizli Katmana Sahip YSA	52
Şekil 6.30. Scg Eğitim Fonksiyonu ile Öğrenme Eğrisi	53
Şekil 6.31. Lm Eğitim Fonksiyonu ile Öğrenme Eğrisi.....	53
Şekil 6.32. Gdx Eğitim Fonksiyonu ile Öğrenme Eğrisi	54
Şekil 6.33. Bfg Eğitim Fonksiyonu ile Öğrenme Eğrisi	54
Şekil 6.34. 100 Giriş, 4 Çıkışlı, 1 Gizli Katmana Sahip YSA	55
Şekil 6.35. Scg Eğitim Fonksiyonu ile Öğrenme Eğrisi	56
Şekil 6.36. Lm Eğitim Fonksiyonu ile Öğrenme Eğrisi.....	56
Şekil 6.37. Gdx Eğitim Fonksiyonu ile Öğrenme Eğrisi	57
Şekil 6.38. Bfg Eğitim Fonksiyonu ile Öğrenme Eğrisi	57

TABLOLAR DİZİNİ

Tablo 1.1. MIT-BIH Data Örneği	2
Tablo 6.1. Normal Data Girilerek Elde Edilen Sınıflandırma Sonuçları	59
Tablo 6.2. Pace Data Girilerek Elde Edilen Sınıflandırma Sonuçları.....	59
Tablo 6.3. Rbbb Data Girilerek Elde Edilen Sınıflandırma Sonuçları.....	59
Tablo 6.4. Lbbb Data Girilerek Elde Edilen Sınıflandırma Sonuçları.....	59
Tablo 6.5. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	60
Tablo 6.6. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar.....	60
Tablo 6.7. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	60
Tablo 6.8. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	60
Tablo 6.9. Dalgacık Dönüşümü (db-2) Başarım Tablosu	60
Tablo 6.10. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	61
Tablo 6.11. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar.....	61
Tablo 6.12. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	61
Tablo 6.13. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar.....	61
Tablo 6.14. Dalgacık Dönüşümü (db-10) Başarım Tablosu	61
Tablo 6.15. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	63
Tablo 6.16. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar.....	63
Tablo 6.17. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	63
Tablo 6.18. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar	63
Tablo 6.19. FFT Başarım Tablosu	63
Tablo 6.20. Dalgacık Dönüşümü (db-10) Sonuçları ile FFT 'nin Karşılaştırılması .	64

1. GİRİŞ

1.1. Konunun Önemi

Kalp insan için hayati önem taşır. Bundan dolayı kalbin çalışması sırasındaki bozuklukların önceden tespit edilip, tedavi yoluna gidilmesi gerekir. Bu bozukluklar biyolojik işaretlerden anlaşılabilir. İşaret, bilgi taşıyan zamana göre değişen veya değişmeyen büyüklüklerdir. Biyolojik işaretler ise canlı vücudundan elektrotlar ve dönüştürücüler aracılığı ile alınan elektrik kökenli veya elektrik kökenli olmayan işaretlerdir. EKG ve EEG gibi işaretler elektrik kökenlidir. Kan basıncı ve vücut sıcaklığı gibi işaretler elektrik kökenli olmayan işaretlerdir.

1.2. Çalışmanın Amacı

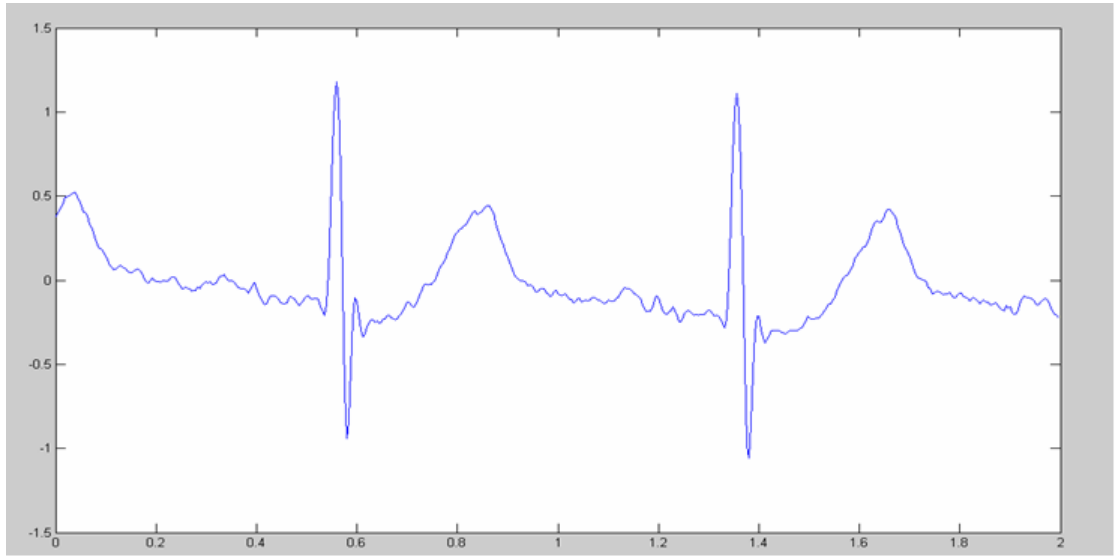
Bu çalışmada amaç EKG Sinyallerindeki anlamlı bilgileri Dalgacık Dönüşümü yöntemi ile daha kısa sürede daha az veri ile elde etmek, elde edilen değişik hastalıklara ait özellik vektörlerini sınıflandırmak, sonuçları Fourier Dönüşümü sonuçları ile karşılaştırmaktır.

1.3. Dataların Elde Edilmesi

Çalışmada kullanılan datalar www.physionet.org adlı sitenin MIT-BIH veri tabanından elde edilmiştir. İki nolu standart derivasyon kullanılmıştır.

TABLO 1.1. MIT-BIH data örneđi

zaman (sec)	genlikler (mV)	
	(mV)	(mV)
0.000	0.378	0.388
0.004	0.390	0.393
0.008	0.412	0.385
0.012	0.430	0.407
0.016	0.460	0.410
0.020	0.495	0.398
0.024	0.495	0.400
0.028	0.500	0.420

**Şekil 1.1.** Datanın Matlab'da Çizdirilmiş Hali.

1.4. Özellik Çıkarma İşlemi

Özellik çıkarımında datalar 300 örnekli, 100 örnek kesişmeli olacak şekilde buffer işlemine tabi tutuldu. Hamming penceresinden geçirildi. Matlab komutlarından 'Wawedec' komutu ile dört seviyeli dalgacık katsayıları elde edildi. cD4 adlı 21 ve 36 parametreden oluşan özellik kat sayıları kullanıldı. Detaylar Dalgacık Dönüşümü ve özellik çıkarımı kısımlarında anlatılmıştır.

1.5. Sınıflandırma İşlemi

Bu çalışmada sınıflandırma Normal, Pace , Rbbb , Lbbb vurularının tanınip gruplara ayrılması ile gerçekleştirilmiştir. Sınıflandırma yapay sinir ağları (YSA) kullanılarak yapılmıştır. YSA yapay zeka uygulamalarının bir çeşididir. YSA'nın öğrenme özelliği vardır. Yani girdiler ve çıktılar arasındaki ilişki doğrusal olsun veya olmasın öğrenir ve daha sonra karşılaştığı girdileri ve çıktıları daha önce görmemiş olsa bile tahmin ederek çözüm üretir. Çalışmada 21 girişli, 4 çıkışlı 42 nöronlu 1 gizli katmanı bulunan geri beslemeli yapay sinir ağı ile 36 girişli, 4 çıkışlı 72 nöronlu 1 gizli katmanı bulunan geri beslemeli yapay sinir ağı kullanılmıştır. Daubechies-2 (db-2) ve Daubechies-10 (db-10) olmak üzere iki farklı dalgacık fonksiyonu kullanılmış, bundan dolayı 21 ve 36 olmak üzere iki farklı giriş sayısı elde edilmiştir. %74 gibi bir sınıflandırma başarısı sağlanmıştır. Ayrıca scg, lm, gdx, bfg gibi değişik ağ eğitim fonksiyonları denenmiştir. Bu fonksiyonlarda öğrenmenin hızının arttığı, bazılarında ise öğrenmenin daha iyi olduğu görülmüştür. En yüksek sınıflandırma başarısı db-10 dalgacık fonksiyonu ile özellik çıkarımı sonuçları kullanılarak, YSA'da scg ağ eğitim fonksiyonuyla gerçekleştirilmiştir.

2. LİTERATÜR ÖZETİ

Fujimura ve arkadaşları, çalışmalarında EKG dalga şekillerinin özelliklerinin çıkarılarak sınıflandırılmasında eşit uzunlukta 12 vuru ele almıştır. Bu vuruları 4 farklı sınıfa ayırmış ve %93'e varan bir başarı elde etmiştir [1].

Zhao ve arkadaşları (2003), çalışmalarında her EKG vurusundan dalgacık dönüşü ile kat sayılar elde etmişlerdir. Bu katsayılardan özbağlı modelleme (Autoregressive Modelling) ile geri beslemeli ağlar oluşturmuşlar ve SVM (Support Vector Machine) ile 6 farklı ekg vurusu sınıflandırmışlar ve %99. 68 başarı sağlamışlardır [2].

Wagner ve arkadaşları, çalışmalarında uyku ve uyanıklık durumlarındaki ekg sinyallerinin özelliklerini çıkarmışlardır. Matlab toolbox'ını kullanarak güç spektrumlarını çıkarmışlardır. Bu şekilde uyku ve uyanıklık durumundaki ekg vurularını %99. 87 başarı oranı ile sınıflandırmışlardır [3].

Engin ve arkadaşları, çalışmalarında ekg vurularının özelliklerini üçüncü derece toplam, dalgacık entropisi ve özbağlı modelleme (AR) kullanarak elde etmişlerdir. Sınıflandırma aşamasında istatistiki yöntemler ile yapay sinir ağlarını kullanmışlardır. İstatistiki sınıflandırıcıların bir iterasyon ile kısa sürede sınıflandırma yaptıklarını ortaya koymuşlardır. Fabian ve arkadaşları (2003), çalışmalarında kriter fonksiyonundan bahsetmiştir. Burada amaç frekans dağılımlarında enerji korunumu sağlamaktır. Enerji korunumu ne kadar fazla olursa, özellik vektörleri orijinal veriye o kadar çok benzeyecektir demektir [4].

Jiang ve arkadaşları (2002), çalışmalarında dalgacık dönüşümü katsayıları ile birlikte bağımsız bileşen analizi (ICA) yöntemini kullanarak ekg vurularının özellik

çıkarmasını gerçekleştirmişlerdir. SVM (Support Vector Machine) ile sınıflandırma yapmışlar, %98, 65'lik başarı sağlamışlardır [5].

Mahmoodabadi ve arkadaşları (2005), çalışmalarında dalgacık dönüşümünü kullanarak iki dalgacık filtresi yapmışlardır. Bu dalgacık filtrelerinden ekg sinyaline daha yakın olan dalgacık filtresini daha iyi algılama başarısı göstermiştir. Bu şekilde QRS dedektörü yapılmış %99 başarı sağlanmıştır [12].

Alexakis ve arkadaşları (2003), çalışmalarında şeker hastalığının T dalgasını etkilemesinden dolayı ekg özelliklerini çıkarmışlardır. Yapay sinir ağları kullanarak T dalgası ile şeker hastalarına ait vuruları %85 başarı ile sınıflandırmışlardır [16].

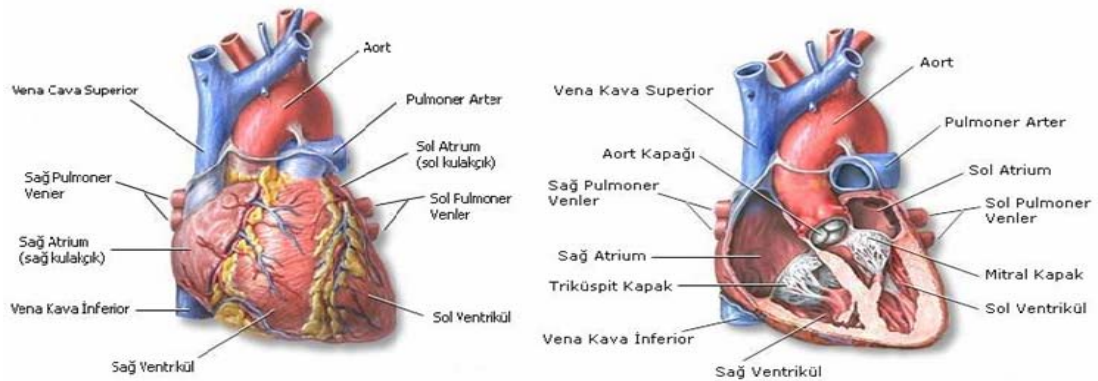
Dokur ve arkadaşları (2000), çalışmalarında ekg vurularından Fourier analizi (DFT) ve Dalgacık Dönüşümü (WT) ile özellik çıkarımı yapmışlardır. Sınıflandırma işleminde ise genetik algoritmalar ve yapay sinir ağları kullanılmıştır.10 farklı sınıflandırma yapılmış ve Fourier Dönüşümünde %92.2 dalgacık dönüşümünde ise %99.4 başarı sağlanmıştır [25].

3. ELEKTROKARDİYOGRAM (EKG)

3.1. Kalp İle İlgili Genel Bilgiler

İnsan kalbi göğüs kafesinin ön kısmında yer alır. Kalp, oksijen ve besleyici maddeler taşıyan kanı, damarlara pompalayan bir kastır. Kalp genel olarak dört parçadan oluşur[1].

Bunlar sağ ve sol kulakçık (atriyum) ile sağ ve sol karıncık (ventrikel)'dir. Kulakçıklar ve karıncıklar ile karıncıklara bağlı damarların arasında kanın tek yönlü akışını sağlayan kapakçıklar vardır. Kalbin çalışması sırasında kan önce kulakçıklar tarafından karıncıklara pompalanır. Karıncıklar dolar ve gelen sinyalle birlikte kasılan karıncıklar kanı damarlara pompalar. Damardan geriye dönen kan sağ taraftan akciğere gönderilir. Kalbin faaliyetini ve ritmini sinüs düğümü kontrol etmektedir. Sinüs düğümü sağ kulakçıktadır. Düzenli olarak küçük elektrik akımları üretmektedir. Üretilen bu uyarılar bir ara istasyonda Av düğümüne, oradan da kalp karıncıklarının bağlantı liflerine iletilmektedir. İletiyi alan karıncıkların kasılıp gevşemesiyle kalp atışları meydana gelir. Normal bir kalp çok düzenli bir şekilde dakikada 50- 80 arası atar. Fazla hareket veya heyecan gibi durumlarda atımlar 100 civarına yükselir. Bu gibi durumlara uyum sağlama işini sinüs düğümü yapmaktadır. Bu gibi durumların dışında meydana gelen yüksek ya da düşük atımlar bir hastalık sonucu gerçekleşir.



Şekil 3. 1. Kalbin Yapısı

3.2 Elektrokardiyogram (EKG)

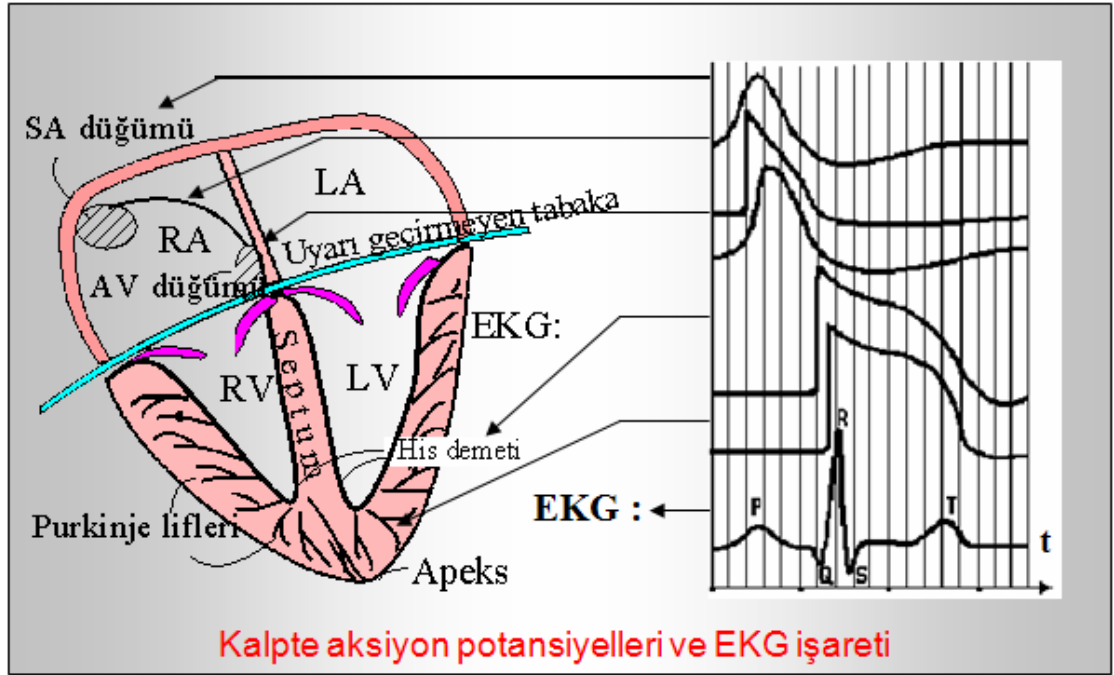
Kalbin sağ kulakçığında yer alan ve sürekli elektrik potansiyeli üreten sinüs düğümü, kalbin bilinen ritmik hareketini gerçekleştirmekle görevlidir. Sinüs düğümü tarafından üretilen bu potansiyeller vücudun iletkenlik özelliğinden yararlanılarak değişik teknikler kullanılarak tespit edilebilir. Bu, çok küçük gerilimler, kollara, bacaklara veya göğse konulan yüzey elektrotlarıyla alınır. Temelde galvanometre ve yükselteç olan elektrokardiyograf cihazı bu çok küçük gerilimleri alır, yükseltir ve kağıt üzerine kaydeder. Bu elektriksel değişimlerin kaydedilip görüntülenmesine Elektrokardiyogram (EKG), elektrokardiyografi sırasında kullanılan cihaza da Elektrokardiyograf denir. Ekg üzerinde kalbin elektrik potansiyellerinin yön, genlik ve zamana göre değişim bilgileri vardır. Ekg dalga şekilleri miyokardiyal iskemi ve enfarktüs, ventriküler hipertrofi ve aktarım problemleri gibi kalp hastalıkları ve anormallikleri durumunda değişir. Elektrokardiyogram ile kalp atışı aritmileri tanımlanır ve biyomedikal sinyal işleme için önemli bir işlemdir[5].

Kalbin elektrik sisteminde, kalbe özel elektriksel aktiviteler ile özel bir iletim sistemi, kalbin ritmik kasılma gevşeme işleminde önemli rol oynar. SA düğümü doğal bir tetikleyicidir ve potansiyel üretir. Bu SA düğümünün potansiyeli tüm kalbe yayılır. Böylece sistemli bir hareketlilik başlar[6].

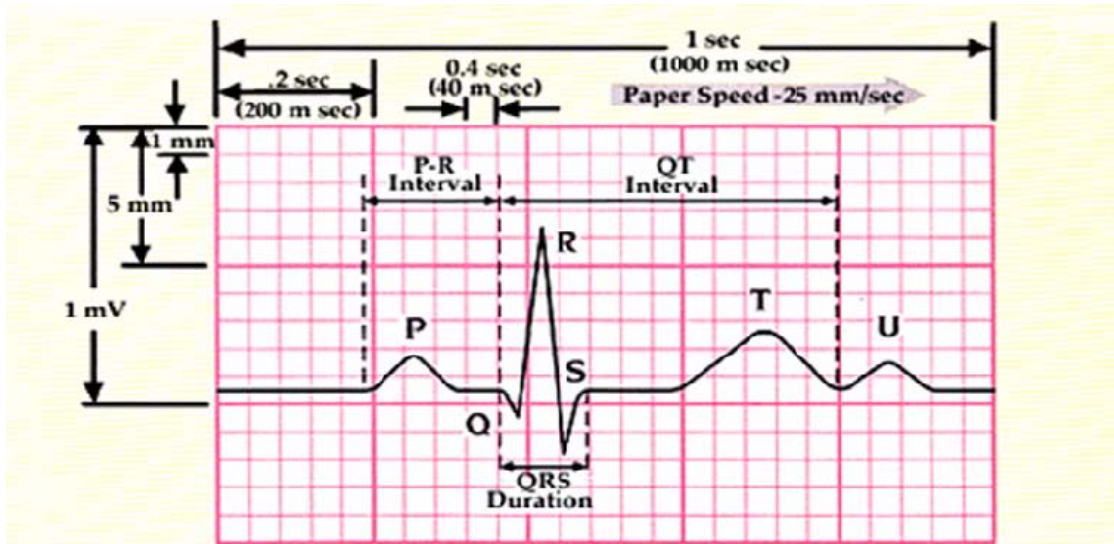
Normal bir EKG sırasıyla P, Q, R, S, T ve U adı verilen dalgalardan oluşur. (Şekil3. 3) Bu oluşum aşağıdaki adımlardan meydana gelir. (Şekil 3. 2)

1. SA düğümü tetiklenir.
2. Kulakçık kas yapısında çok küçük elektrik potansiyeli oluşur. Bu da kulakçıkta kasılma meydana getirir. Bunun sonucunda EKG'deki P dalgası oluşur. Kulakçığın yavaş kasılmasından dolayı P dalgası küçük bir dalgadır. Genliği 0,1-0,2 mV süresi 60-80 ms kadardır.
3. Uyarılma dalgası AV (Kulakçık- karıncık) düğümünde bir iletim duraksaması yapar. Bu da P dalgasından sonra yaklaşık 60-80 ms'lik bir dalga oluşturur. Bu dalgaya PQ dalgası denir. Bu sırada kan kulakçıktan karıncığa geçer.
4. His demeti, demet dalları yani purkinje lifleri kulakçıkların hızlı uyarılmasını sağlarlar.

5. Uyarı dalgası yukarıdan aşağıya doğru yayılır. Kulakçıklarda hızlı kasılmayı sağlar. Bu da EKG' deki QRS dalgasını oluşturur. QRS'nin genliği 1 mv, süresi 80 ms civarındadır. Keskin bir dalgadır.
6. Karıncık kasılmasından sonra 100-120 ms süren ST dalgaları oluşur.
7. Karıncıkların gevşemesi T dalgalarına yol açar. T dalgalarının genliği 0, 1-0, 3 mv süresi 120-160 ms kadardır[6].



Şekil 3.2. Kalpte Elektrik İletim Yolları ve EKG Dalga Şeklinin Oluşumu



Şekil 3.3. EKG Dalgalarının Genlik ve Süre Değerleri [7].

3.3. Derivasyonlar

EKG cihazının elektrotları hastaya değişik şekillerde bağlanır. Bu bağlantılara genel olarak derivasyon denir. Derivasyonlar standart isimlerle anılmaktadır. Çevre derivasyonları ve göğüs derivasyonları olarak iki başlık altında incelenebilir. Toplamda 12 farklı derivasyon vardır. Derivasyonlarda sağ bacak referans elektroduna bağlanır [8].

3.3.1. Çevre derivasyonları

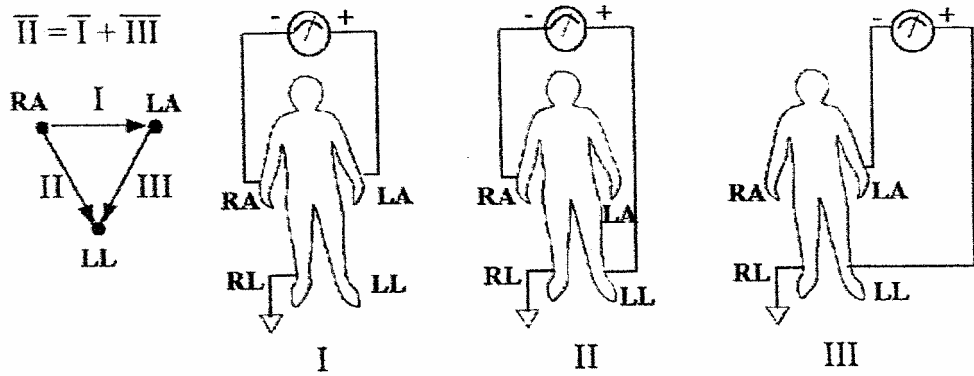
Bipolar ve unipolar olarak ikiye ayrılır.

a) Bipolar Çevre Derivasyonları: Vücudun iki noktası arasındaki potansiyel farkı saptamaya yarar. 3 çeşittir.

D1: EKG cihazının negatif ucuna bağlı elektrot sağ kola, pozitif ucuna bağlı elektrot sol kola bağlanır. (Şekil3. 4. I)

D2: EKG cihazının negatif ucuna bağlı elektrot sağ kola, pozitif ucuna bağlı elektrot sol ayağa bağlanır. (Şekil3. 4. II)

D3: EKG cihazının negatif ucuna bağlı elektrot sol kola, pozitif ucuna bağlı elektrot sol ayağa bağlanır. (Şekil3. 4. III)



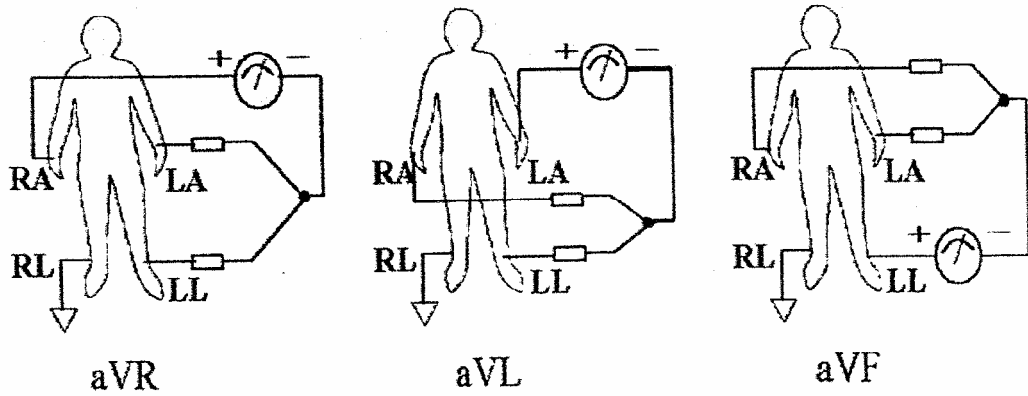
Şekil 3.4. Bipolar Çevre Derivasyonları

b) Unipolar Çevre Derivasyonları: Wilson santral terminali sağ kol, sol kol ve sol bacak birleştirilerek elde edilir. Bu terminalde keşfi yapılacak uzuv terminal dışı bırakılıp EKG cihazı üzerinden aynı uzva bağlanırsa unipolar çevre derivasyonları elde edilir. VR, VL, VF adı verilir. Genelde %50 kuvvetlendirilmiş derivasyon bağlantıları olarak adlandırılır. Bunlara da aVR, aVL, aVF adı verilir. 3 çeşittir.

aVR: Sol kol ile sol ayak santral terminaline bağlanır. Cihazın negatif ucuna bağlı elektrot santrale, pozitif uca bağlı elektrot sağ kola bağlanır. (Şekil3.5 aVR)

aVL: Sağ kol ile sol ayak santral terminaline bağlanır. Cihazın negatif ucuna bağlı elektrot santrale pozitif uca bağlı elektrot sol kola bağlanır. (Şekil 3.5 aVL)

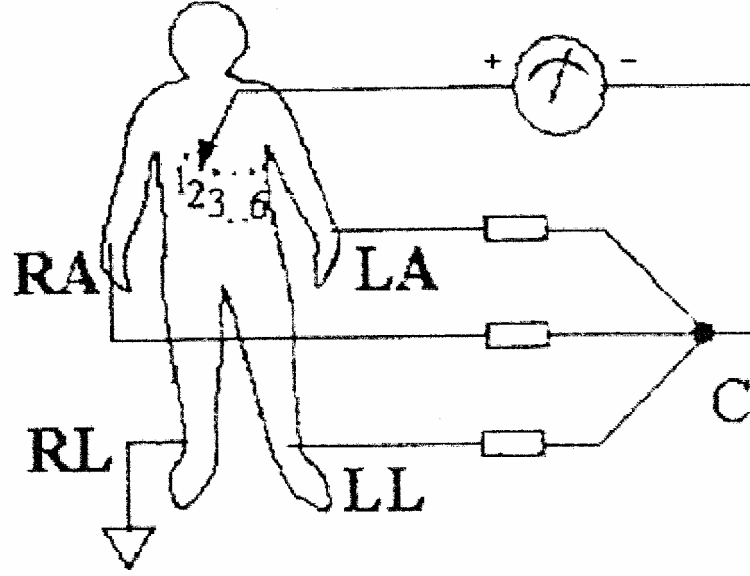
aVF: Sağ kol ile sol kol santral terminaline bağlanır. Cihazın negatif ucuna elektrot santrale, pozitif ucuna bağlı elektrot sol ayağa bağlanır. (Şekil 3. 5 aVF)



Şekil 3. 5. Unipolar Kuvvetlendirilmiş Çevre Derivasyonları

3.3.2. Göğüs Derivasyonları

Göğüs derivasyonları 6 çeşittir ve bunlar V1, V2, V3, V4, V5, V6 olarak adlandırılır. Cihazın negatif ucuna bağlı elektrot santrale bağlanır. Pozitif uca bağlı olan ve araştırıcı elektrot olarak adlandırılan elektrot göğüs üzerinde 6 nokta üzerinde gezdirilir. Her nokta bir derivasyona karşılık gelir. (Şekil 3. 6)



Şekil 3.6. Göğüs Derivasyonları

3.4. EKG Vuruları

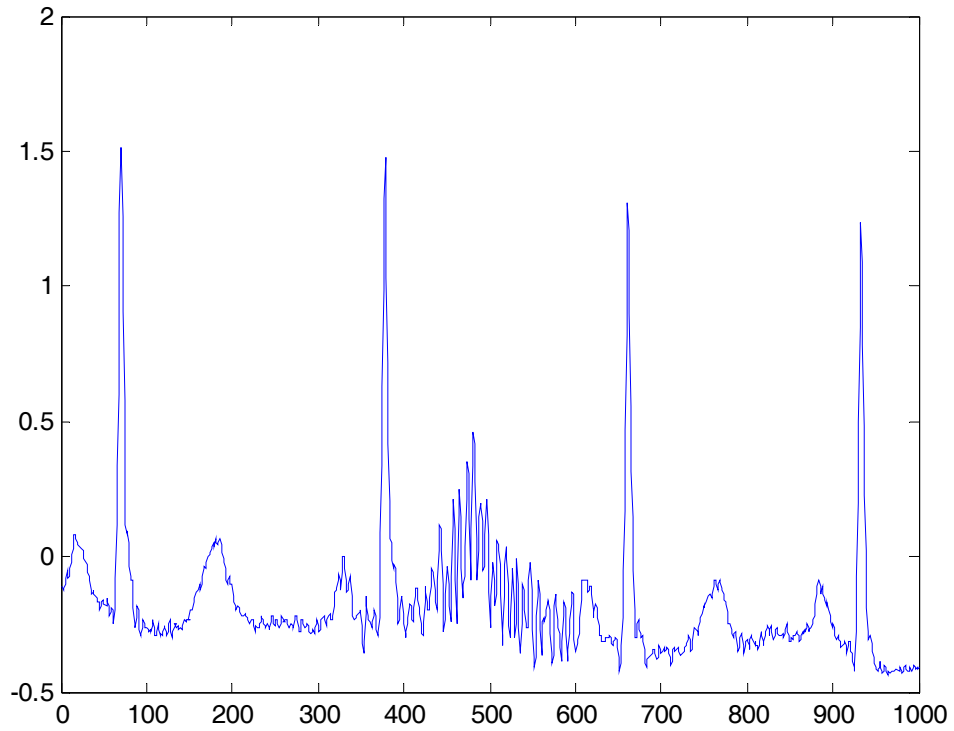
Bu çalışmada biri normal vuru (N, Normal Beat), biri yapay vuru (P, Paced Beat), biri sağ dal blok vurusu (RBBB, Right Bundle Branch Block Beat), biri sol dal blok vurusu (LBBB, Left Bundle Branch Block Beat) olmak üzere 4 farklı EKG vurusu üzerinde çalışılmıştır.

3.4.1. Normal Vuru (N)

Şekil 3.7’de normal vuru görülmektedir. P dalgasının süresi 0.10 saniyedir. Bu süre 0,10- 1,12 sn. arasında ise geniş P dalgasından söz edilir. PR süresi ortalama 0,16 sn’dir. PR süresi genellikle kalp hızının artması ile kısalır, yaş ile uzar. QRS süresi erişkinlerde 0,06- 0,10 sn arasındadır. V1-V6 derivasyonlarında QRS süresi genellikle 0,01- 0,02 sn daha uzundur. P, PR, QRS süreleri normal olarak bipolar

derivasyonlarda ölçülür. QRS süresinin uzaması, karıncıklar içinde iletimin gecikmiş olduğu anlamına gelir. En çok dal bloklarında görülür.

QT süresi, QRS kompleksinin başlangıcından T dalgasının sonuna kadar olan aralıktır. En belirgin olan herhangi bir derivasyondan ölçülebilir. QT süresi kalp hızına bağlı olarak değişiklik gösterir. Bu bakımdan hıza göre gerçek değerinin hesaplanması gerekir. Bu süre normal erişkinlerde 0,35- 0,44 sn arasındadır. T dalgasının normal süresi 0,10- 0,25 sn arasındadır. Ancak QT süresinin içine girmiş olduğu için klinik elektrokardiyografide T dalgasının süresi ayrıca ölçülemez[3].

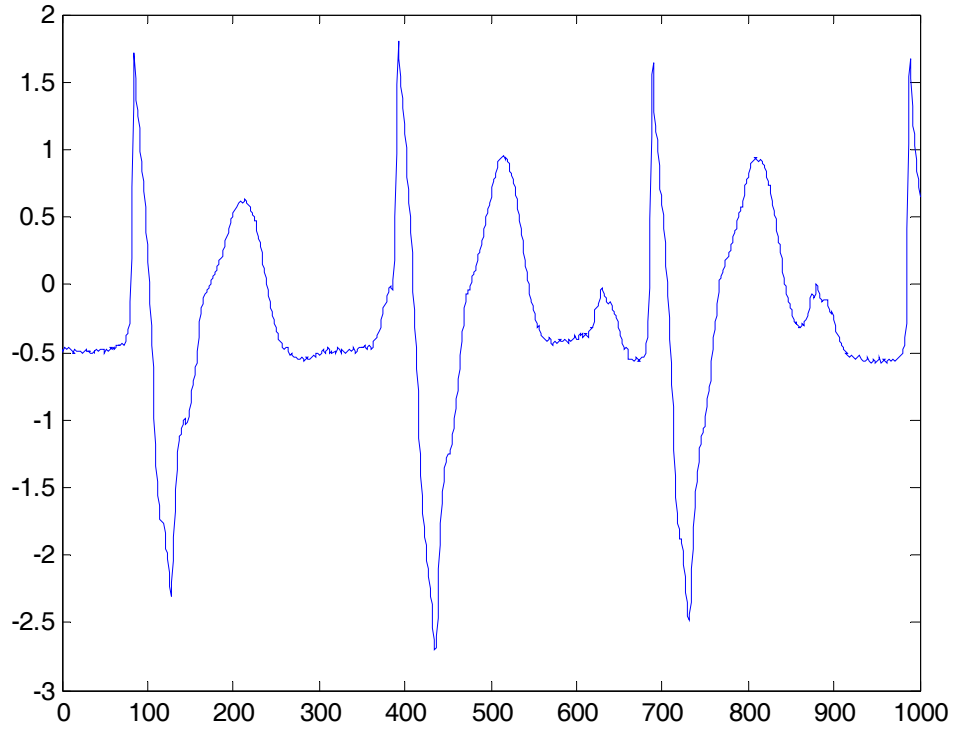


Şekil 3.7. Normal Vuru

3.4.2. Yapay Vuru (P)

Şekil 3.8’de yapay vuru görülmektedir. Kalp kendi üretip oluşturduğu elektrik akımını yine kendi içindeki yollardan kalp kaslarına dağıtarak, kasılmasını sağlar. Kalbin doğal ileti sisteminde bazı aksaklıklar olması durumunda kalp kasının kasılması için muhtaç olduğu elektriği yapay olarak sağlayan ‘Pacemaker’ adı verilen sistemler kullanılmaktadır. Pacemaker, sağ

karıncık, sağ kulakçık ve her ikisinden ardışıl olmak üzere yapay vuruyu kalbe vermektedir. Yapay vuruda P dalgası nadiren görülür. PR aralığı ölçülemez. QRS kompleksi, yapay vurudan sonra görülür. Genellikle 0,12 saniyeden uzundur. Eğer varsa PP aralığı değişen değerler alır. Kalp vuru hızı, RR aralığı, yapay vuru aralığı Pacemaker ayarına göre farklılıklar gösterir[9].

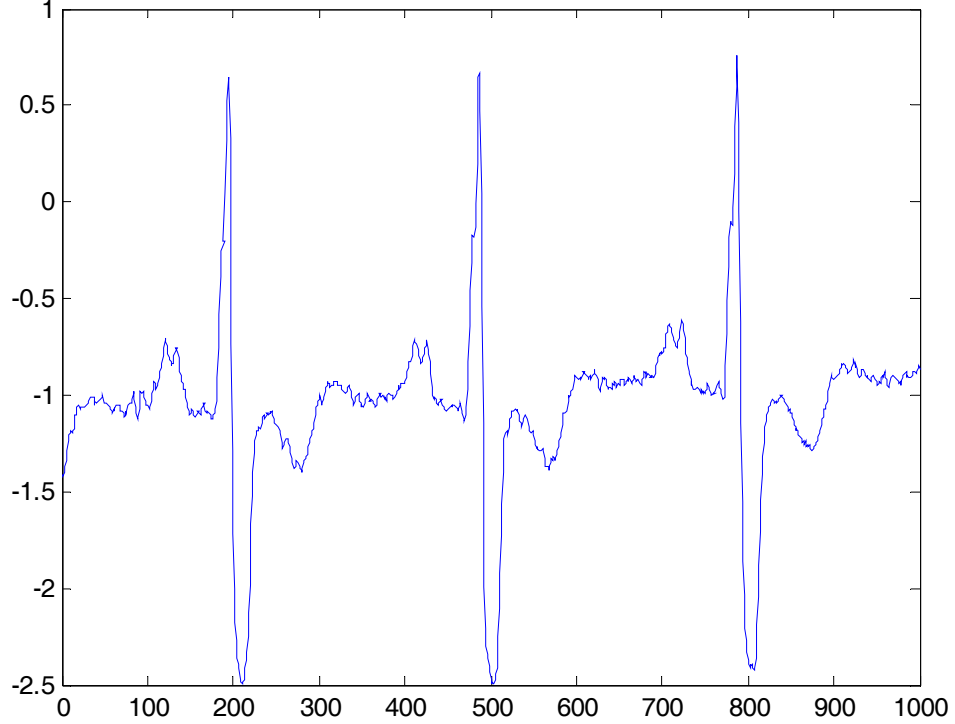


Şekil 3.8. Yapay Vuru

3.4.3. Sağ Dal Blok Vurusu (RBBB)

Şekil 3.9’da sağ dal blok vurusu görülmektedir. Sağ dal bloğunda, sağ karıncık aktivasyonunda gecikme olur. Bütün dal bloklarında olduğu gibi, sağ dal bloğunda da QRS süresi uzamıştır. QRS süresi 0.11 sn ve daha fazladır. Komplikasyonsuz sağ dal bloklarında QRS’ nin başlangıç ve orta kısımları normaldir. Sadece terminal kısımları anormallik gösterir. QRS’ nin terminal kısmında karıncık aktivasyonu

yavaşlamış ve gecikmiştir. Terminalde oluşan bu anormal aktivasyon, V5-6 derivasyonlarında geniş S dalgasının ortaya çıkmasına neden olur. Ayrıca D1-2 derivasyonlarında S dalgası halinde etkisini gösterir[9].

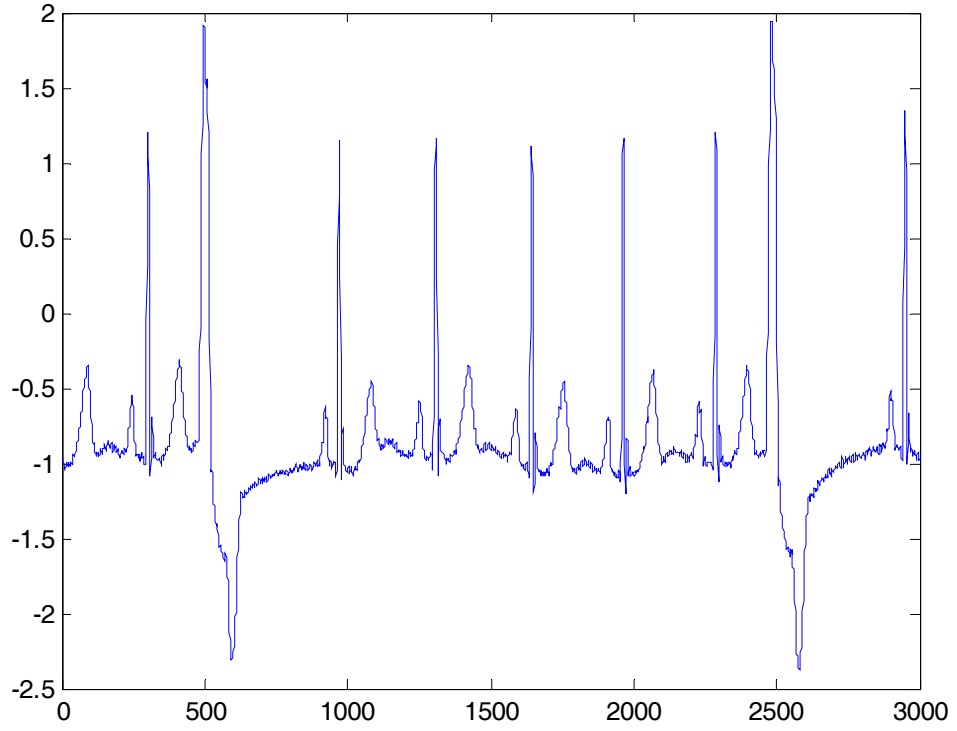


Şekil 3.9. Sağ Dal Blok Vurusu

3.4.4. Sol dal Blok Vurusu (LBBB)

Şekil 3. 10'da sol dal blok vurusu görülmektedir. Sol dal bloğunda sol karıncık aktivasyonu gecikir. Sol karıncık aktive olmaya başladığı zaman da sağ karıncığın bu potansiyellere karşı gelen potansiyelleri artık kalmamıştır. Bu yüzden karşılıksız kalan, dengelenemeyen sol karıncık potansiyelleri büyük sol karıncık vektörleri oluşturur. Bundan dolayı sol dal bloğunda ortaya çıkan QRS vektörlerinin genliği yüksek olabilir. QRS süresi 0.12 sn veya daha uzundur. Karıncık aktivasyonunun gecikmesi, QRS'nin orta ve son kısımlarında olur. QRS dalgasının aksi yönde ST-T dalgası oluşur. Klinikte sol dal bloğunun önemi, hasta hikayesi ve fizik muayene

bulguları ile tespit edilir. Ancak istatistik olarak sol dal bloğu vakalarının büyük çoğunluğunda önemli bir kalp hastalığı vardır.



Şekil 3.10. Sol Dal Blok Vurusu

4. DÖNÜŞÜM YÖNTEMLERİ

Dış ortamdan alınan analog işaretlerden örnekleme ve kuantalama sonucunda elde edilen işaret sayısal işaret olarak adlandırılır. Sayısal işaret aslında bir sayı dizisidir. Bu dizide işaretin taşıdığı bilginin elde edilmesi çoğu zaman mümkün değildir. Bu sebeple işaret, bir dizi işlemde geçirilir. Bu işlemler dönüşüm yöntemi olarak adlandırılır. Sayısal işaret işlemede en çok Fourier Dönüşümü kullanılmaktadır. Son zamanlarda ise Dalgacık Dönüşümü ön plana çıkmıştır. Çünkü durağan sinyallerde ihtiyaca cevap verebilen Fourier Dönüşümleri frekans çözünürlüğüne sahiptir. Zaman çözünürlüğüne sahip değildir. Sinyalin durağan olmadığı durumlarda ise zaman bilgisi de gereklidir. Fourier Dönüşümleri bu zaman bilgisini vermediğinden dolayı Dalgacık Dönüşümü (wavelet) gündeme gelmiştir.

Sayısal sinyal üzerinde gürültü diye adlandırılan istenmeyen bilgiler de vardır. İşaret önce bazı ön işlemlerden geçirilerek gürültülerden arındırılır. Dönüşüm yöntemleri ile işaretin özelliği çıkarılmaya çalışılır. Doğru özelliklerin çıkarılması işaretin sınıflandırılmasında önemlidir.

4.1. Fourier Dönüşümleri

Fourier dönüşümü ile durağan sinyallerde çok iyi frekans çözünürlüğü elde edilir. Fakat zaman çözünürlüğü diye bir şey kalmaz. Yani hangi zaman aralığında hangi frekans birleşmelerinin yer aldığı tespit edilemez[10].

Zamana bağlı işaretin genliğinden yani ham işaretten istenilen bilgiler elde edilmeyebilir. Bu durumda işaret matematiksel bir dönüşüme tabi tutulur. Örneğin ham EKG işareti zamana göre değişen gerilim değerlerini verir. Kalp hastalıklarının teşhisinde sadece zaman bilgisi yeterli olmadığından bu sinyal zaman domeninden

frekans domenine dönüştürülür. Böylece ekg işaretinde zaman domeninde ulaşılamayan bilgilere frekans domeninde ulaşır[9].

Fourier Dönüşümü ile aşağıdaki eşitlikler kullanılarak işaretin frekans bilgisi elde edilir.

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} \cdot dt \quad (4.1)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} \cdot df \quad (4.2)$$

t: Zaman

f: Frekans

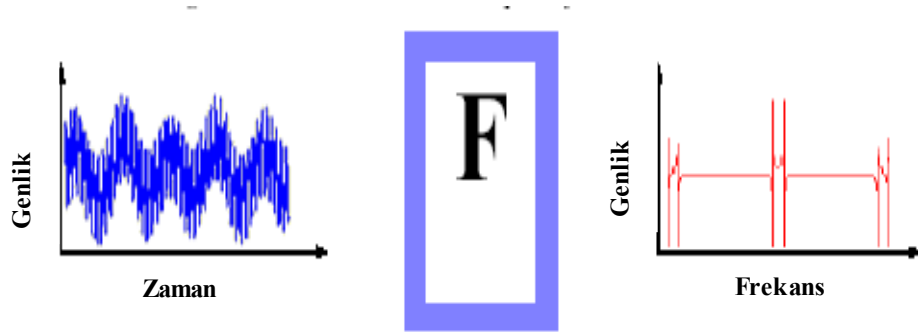
x: Zamana Bağlı İşaret

X: Frekansa Bağlı İşaret

Denklem 4.1: x (t)'nin Fourier dönüşümü

Denklem 4.2 : x (t)'nin Ters Fourier dönüşümü

Dönüşümde integrasyon yapıldığından f frekanslı bileşenin hangi zamanda ortaya çıktığı sonucu değiştirmez. Kısaca özetleyecek olursak, Fourier dönüşümünde işaret sürekli ise işaret ile Fourier çekirdeği olarak adlandırılan $e^{j\omega n}$ çarpılır ve integrali alınır. İşaret kesikli ise işaret ile Fourier çekirdeği $e^{j\omega n}$ çarpılır ve toplam yapılır[21].



Şekil 4.1. Fourier Dönüşümü [13].

Ayrık Fourier Dönüşümü (AFD) ise periyodik olmayan kesikli (ayrık) zaman sinyallerinin frekans birleşenlerini bulmaya yarar. Sinyalin kendisi kesikli olsa bile elde edilen frekans bileşeni sürekli'dir. Bundan dolayı ayrık Fourier Dönüşümü (AFD) alınırken toplam işlemi kullanılır. Dennis Gabor tarafından geliştirilmiştir.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N} \quad (4.3)$$

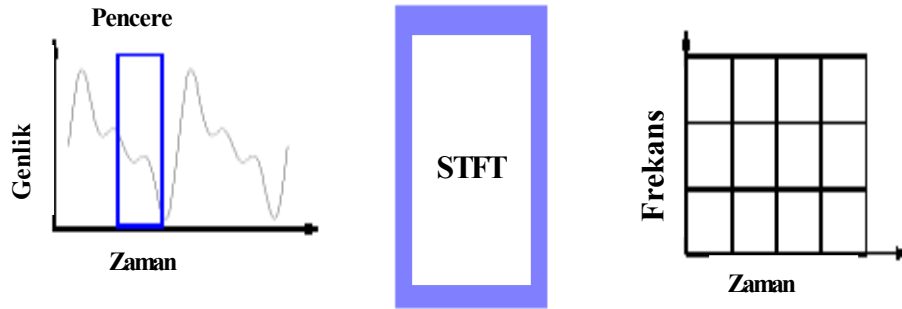
$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j2\pi kn/N} \quad (4.4)$$

$x(n)$: Ayrık Zaman Bileşeni İşareti

$X(k)$: Ayrık Frekans Bileşeni İşareti

N : Pencere Genişliği

N pencere genişliği artarsa frekans bileşeni çözünürlüğü iyi olur, zaman bileşeni çözünürlüğü kötüleşir. Pencere küçük seçilirse bunun tersi olur.



Şekil 4.2. Kısa Zaman Fourier Dönüşümü[13].

Kısa zaman Fourier Dönüşümü (STFT) durağan olmayan sinyallerin analizinde kullanılır. Sabit bir pencere fonksiyonu ile sinyal çarpılır. Daha sonra pencere kaydırılarak Fourier dönüşümü tekrarlanır. İşlemler sırasında kullanılan pencere fonksiyonu aynıdır. Örneğin dikdörtgensel bir pencerenin frekans çözünürlüğü kötüdür. Üçgensel bir pencerede bu çözünürlük daha iyi olur. Daha iyi sonuçlar elde edebilmek için değişik pencereler geliştirilmiştir[11]. Örneğin Hamming penceresinin matematiksel ifadesi şöyledir.

$$w(t) = \begin{cases} [1 + \cos(2\pi t/T)] / 2 & t \in [-T/2, T/2] \\ 0 & \text{Diğer durumlarda} \end{cases} \quad (4.5)$$

Kısa zaman Fourier dönüşümünün matematiksel ifadesi eşitlik 4.6 da gösterilmiştir.

$$KZFD(w, T) = \int_{-\infty}^{\infty} f(t) \cdot w^*(t - T) \cdot e^{-j\omega t} \cdot dt \quad (4.6)$$

$f(t)$ = Dönüşümü yapılacak olan sinyal

$w_{(t)}$ = Kullanılan Pencere Fonksiyonu

T = Öteleme Miktarı

$e^{-j\omega t}$ = Fourier Çekirdeği

KZFD' de hassasiyet pencere seçimine bağlıdır. Bazen pencerenin değişmesi gerekebilir. Bu yöntemde pencere sabit olduğundan pencerenin sürekli değiştirilebildiği Dalgacık (wavelet) dönüşümü yöntemi geliştirilmiştir.

4.2. Dalgacık (Wavelet) Dönüşümü

Dalgacık dönüşümü Fourier dönüşümünün durağan olmayan sinyallerdeki eksiklerini gidermek için geliştirilmiş farklı bir dönüşüm yöntemidir[22,23]. Bu analiz yöntemi, gürültüye karşı daha az hassasiyet göstermekte ve durağan olmayan sinyallere rahatlıkla uygulanabilmektedir. Bundan dolayı sinyal işleme ile uğraşanların ilgisi frekans tabanlı Fourier Dönüşümünden ölçek tabanlı Dalgacık Dönüşümüne doğru kaymıştır.

Dalgacık ifadesi olarak 1909 yılında Alfred Haar tarafından ortaya atılmıştır[10]. Zaman içerisinde Jean Morlet ile Y. Meyer ve arkadaşları eğitim fonksiyonu geliştirmişler ve 1988 yılında Stephane Mallat önemli katkılar sağlamıştır[20]. Daha sonra Ingrid Daubechies, Ronald Coifman gibi uluslar arası araştırmacılar yöntemi geliştirerek bugünkü şekline getirmişlerdir.



Şekil 4.3. Dalgacık Dönüşümü[13].

4.2.1. Sürekli Dalgacık Dönüşümü (CWT)

Sürekli Dalgacık Dönüşümünde dönüşümü yapılacak işaret dalgacık adı verilen bir fonksiyonla çarpılır[12]. Bu dalgacık işaret üzerinde gezdirilirken hem yeri hem de büyüklüğü değiştirilir. Yerinin değişmesine öteleme denir ve ‘T’ işareti ile gösterilir. Dalgacığın büyüklüğünün değişmesine ise ölçekleme denir ve ‘s’ harfi ile gösterilir. Yani pencere fonksiyonları ana dalgacıktan ölçekleme yoluyla elde edilir. Frekans parametresinin yerini ölçek parametresi alır.

$$SDD_x^\psi(T, s) = \Psi_x^\psi = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - T}{s} \right) dt \quad (4.7)$$

SDD ya da Ψ =Sürekli Dalgacık Dönüşümü

$x(t)$: İşaretin Kendisi

ψ = Ana Dalgacık Fonksiyonu (Meksikalı Şapkası, Daubechies gibi)

t: Zaman

T : Zamanda Öteleme ya da Kaydırma Miktarı

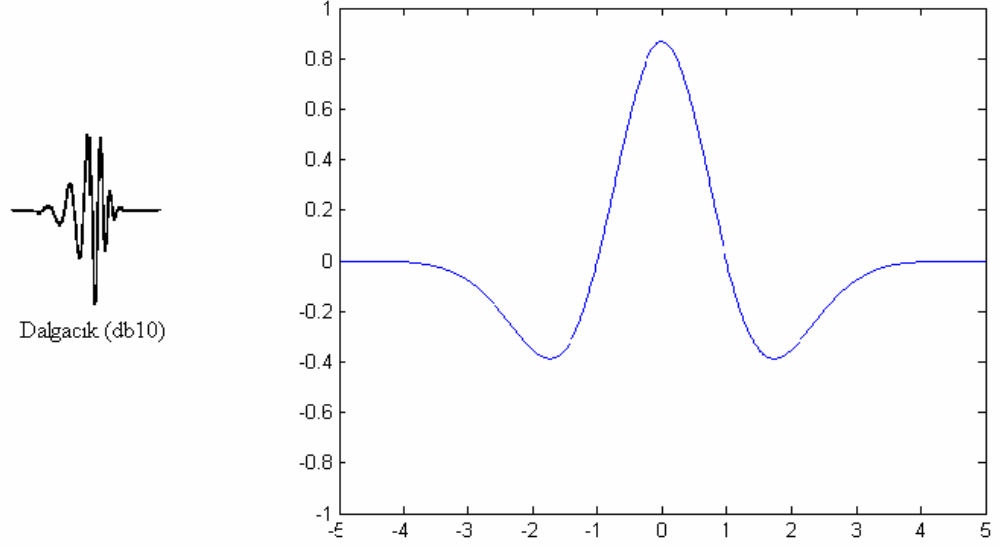
s : Ölçek

$s = \frac{1}{f}$ olarak tanımlanır.

Ölçek değeri büyük ise (düşük frekans) dalgacık genişler ve işaret hakkında genel bir bilgi elde edilir.

Ölçek değeri küçük ise (yüksek frekans) dalgacık daralır ve işaretin ayrıntıları hakkında bilgi elde edilir[9].

Yüksek ölçek katsayılarına yaklaşıklık (approximation), düşük ölçek katsayılarına ise ayrıntı (detail) katsayıları denir. Böylece işaret süzgeçlerden geçirilerek alt bantlarına ayrılır. İşaretin boyutunun değişmemesi için süzgeç çıkışlarının örnek sayısı yarıya indirilir. Örnek olarak dalgacık fonksiyonlarından ‘Daubechies-10’ ile ‘Meksika Şapkası’ şekli şekil 4. 4. a ve b’de görülmektedir.



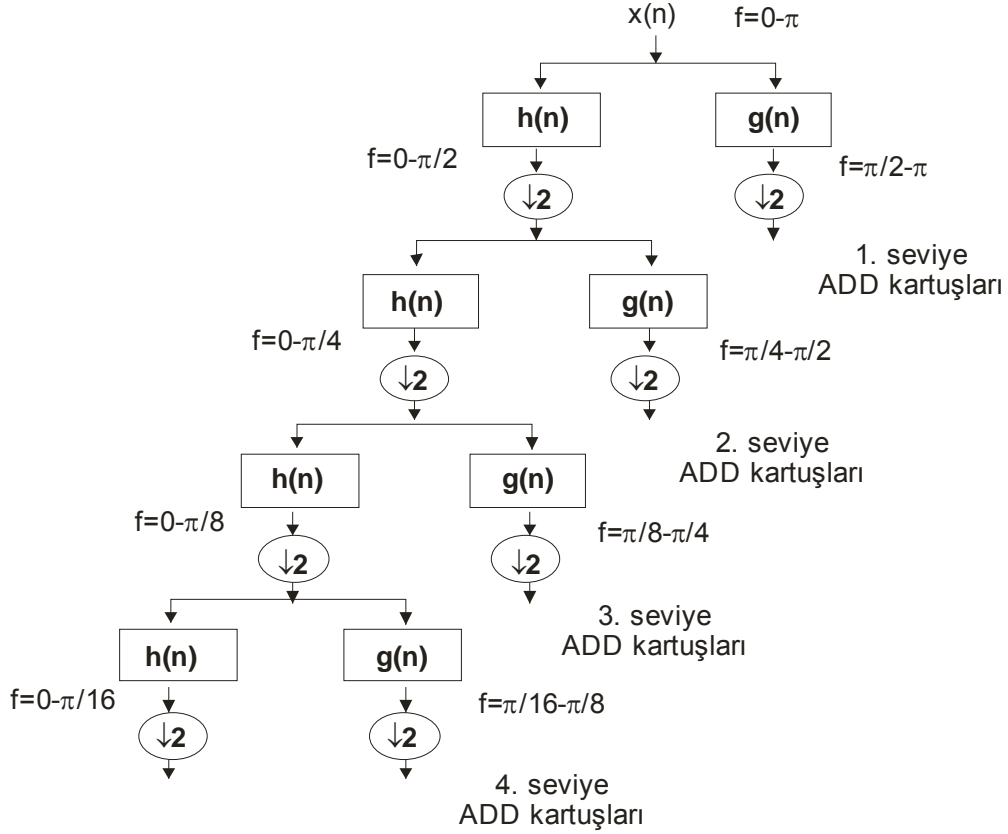
Şekil 4.4. (a) Daubechies10

(b) Meksikalı Şapkası

4.2.3. Ayrık Dalgacık Dönüşümü(DWT)

Sürekli dalgacık dönüşümde hesap yükü çok fazladır. Hesap yükünü azaltmak için Ayrık Dalgacık Dönüşümü (ADD) kullanılır. Ayrık dalgacık dönüşümünde öteleme ve ölçeklerin ikili kuvvetleri alınır. Bu işleme ikili dalgacık (dyadic) dönüşümü denir.

Mallat ikili dalgacık dönüşümünü sayısal süzgeçlerle gerçekleştirmiştir. Böylece dalgacık dönüşümü işlemi hızlanmıştır. İşaret kademeli yüksek ve düşük frekanslara ayrılır. Yüksek frekans kısmından ayrıntılar (detail) elde edilir[13]. (Şekil 4.5)



Şekil 4.5 Dalgacık Ağacı[9].

$h(n)$: Alçak Geçiren Filtre

$g(n)$: Yüksek Geçiren Filtre

Ayrık dalgacık dönüşümünde işarettaki yüksek frekans değişimlerini analiz ederken yüksek geçiren filtreler, alçak frekans değişimlerini analiz ederken alçak geçiren filtreler seçilir. Çözünürlük filtreleme ile değiştirilir[14].

Ayrık dalgacık dönüşümü matematiksel olarak iki aşamada tanımlanabilir[15].

Birinci aşamada fark denklemini sağlayan ölçekleme fonksiyonunu tanımlanır. (Eşitlik 4. 8)

$$\Phi(x) = \sqrt{2} \sum_k h(k) \Phi(2x - k) \quad (4.8)$$

İkinci aşamada ana dalgacık eşitlik 4.8' deki ölçekleme fonksiyonu ile ilişkilendirilir. (Eşitlik 4. 9)

$$\psi(x) = \sqrt{2} \sum_k g(k) \Phi(2x - k) \quad (4.9)$$

Daha sonra $h(k)$ ve $g(k)$ fonksiyonları kullanılarak dönüşüm katsayıları bulunabilir. J seviyesinde bir çözümleme şöyle ifade edilebilir.

$$f(x) = \sum_k C_{o,k} \cdot \Phi_{o,k}(x)$$

$$f(x) = \sum_k \left(C_{j+1,k} \cdot \Phi_{j+1,k}(k) + \sum_{j=0}^j d_{j+1,k} \Psi_{j+1,k}(x) \right) \quad (4.10)$$

Burada $C_{o,k}$ katsayıları bellidir. $j+1$ seviyesindeki $C_{j+1,n}$ ve $d_{j+1,n}$ katsayıları ile j seviyesindeki $C_{j,n}$ katsayıları arasında şöyle bir bağıntı vardır.

$$c_{j+1,n} = \sum_k c_{j,k} \cdot h(k-2n) \quad (4.11)$$

$$d_{j+1,n} = \sum_k d_{j,k} \cdot g(k-2n) \quad (4.12)$$

Bu eşitliklerin anlamı şudur: $j+1$ çözünürlüğündeki $c_{j+1,n}$ ve $d_{j+1,n}$ katsayılarının elde edilmesi için, j çözünürlüğündeki $c_{j,n}$ ve $d_{j,n}$ katsayılarının ve $\tilde{h}(n)$ ve $\tilde{g}(n)$ fonksiyonları ile evriştirilip örnek sayısının iki ile seyreltilmesi gerekmektedir. Burada $\tilde{h}(n)$ ve $\tilde{g}(n)$ fonksiyonları sırasıyla h alçak geçirgen ve g yüksek geçirgen süzgeçleridir ve şöyle tanımlanmışlardır.

$$\tilde{h}(n) = h(-n) \quad (4.13)$$

$$\tilde{g}(n) = g(-n) \quad (4.14)$$

Bu işlemin sonucunda her j seviyesindeki çözünürlük için alçak çözünürlüğe sahip yaklaşıklık (${}^c j, n$) katsayıları ve yüksek çözünürlüğe sahip detay (${}^d j, n$) katsayıları elde edilir[15].

5. YAPAY SİNİR AĞLARI (YSA)

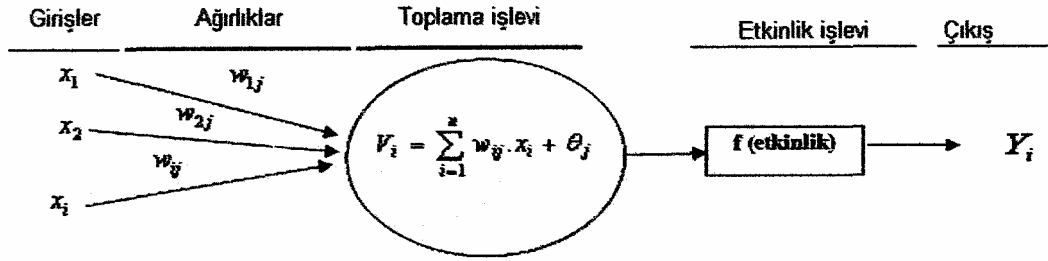
Yapay sinir ağları (YSA) biyolojik beyin sinir sisteminin çalışma şekli örnek alınarak tasarlanmış hesaplama yaklaşımıdır. Biyolojik sistemlerde öğrenme nöronlar arasındaki sinaptik bağlantıların ayarlanması ile olur. Bu ayarlama işlemi zaman içerisinde insanların yaptıkları, yaşadıkları ve tecrübeleri ile gerçekleşir.

Bu durum YSA için de geçerlidir. Öğrenme eğitme yoluyla örnekler kullanılarak olur[16]. Eğitme işleminde ise girdi-çıkı verileri işlenir. Öğrenme algoritması bu verileri kullanarak yakınsama sağlanana kadar ağırlıkları tekrar tekrar ayarlar. Bu işlem birimi, girişlerden ya da kendinden önceki nöronlardan aldığı sinyalleri birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkarır. Bu işlem birimleri kısaca nöronlara karşılık gelir ve bir bağ içinde birbirlerine bağlanırlar. Bu ağlar öğrenmenin yanında ezberleme ve bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptirler. Birçok yapay sinir ağı tipi bulunmakla birlikte bazılarının kullanımı diğerinden daha yaygındır. En çok kullanılan YSA “Geri Yayılımlı Yapay Sinir Ağı” olarak bilinendir. Çalışma içerisinde de bu mimari kullanılmıştır. Bu tip yapay sinir ağı tahmin ve sınıflandırma işlemlerinde çok iyi sonuçlar vermektedir [17]. Bir diğeri Kohonen Özörgütlemeli Haritadır. Bu tip sinir ağları karışık bilgi kümeleri arasında ilişki bulma konusunda başarılı sonuçlar vermektedir.

YSA’ da bilgi matematiksel işlev ile temsil edilir. İşlev elemanları arasındaki bağlantının ağırlıkları bu işlevin değişkenleri olarak görev yaparlar. Bu arada ağırlıklar saklanan bilginin ne olduğuna karar verir. Ağ giriş ile tüm ihtimalleri birleştirerek en iyi ve en uygun örneği çıkış olarak seçer. Yapay sinir ağları biyolojik işaretlerin tanınması ve sınıflandırılması, ses tanıma, parmak izi tanıma, otomatik araç kontrolü, denetim ve modelleme gibi bir çok alanda kullanılmaktadır[18].

5.1. Yapay Sinir Ağlarını Oluşturan Kısımlar

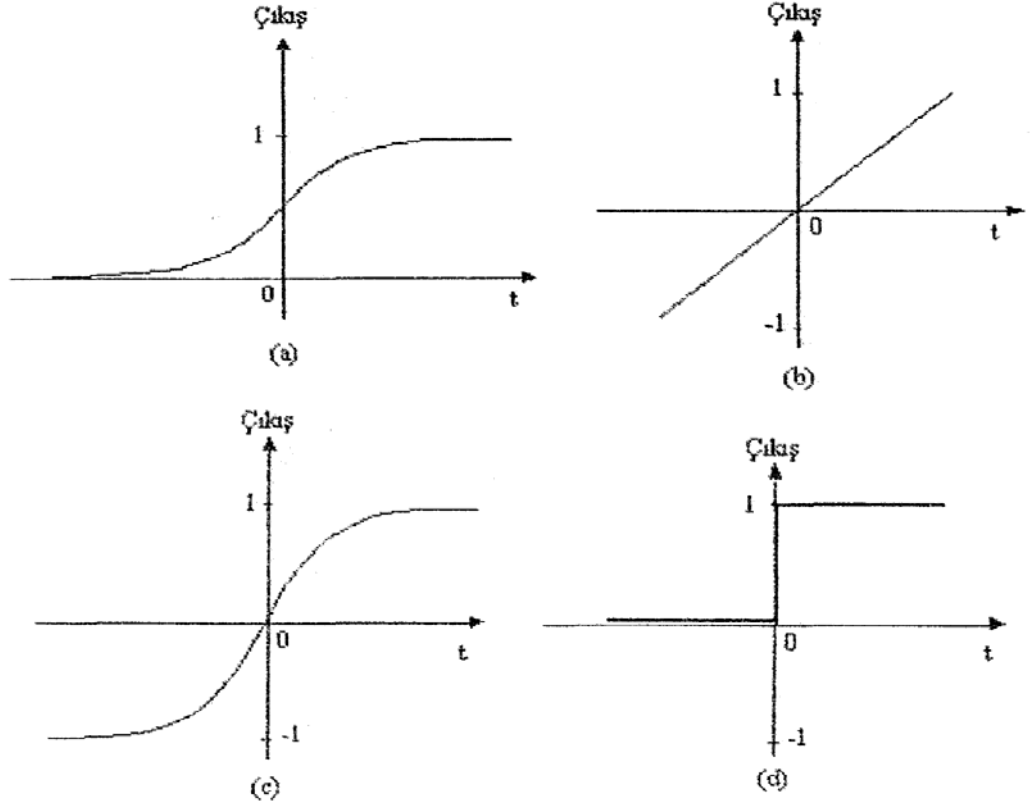
Yapay bir sinir ağı girişler, ağırlıklar, toplama işlevi, etkinlik işlevi ve çıkış olarak 5 kısımdan oluşur. (Şekil 5.1)



Şekil 5.1 Bir Yapay Sinir Hücresi [18].

x ile gösterilen girişlerin her biri w ile gösterilen kendi ağırlıkları ile çarpılır. θ_j ile gösterilen eşik değeri ile toplanır. Genel toplam f ile gösterilen etkinlik işlemine tabi tutulur ve Y ile gösterilen çıkış elde edilir.

En çok kullanılan etkinlik fonksiyonları tek kutuplu sigmoid, doğrusal, hiperbolik tanjant ve keskin sınırlayıcı fonksiyonlardır. (Şekil 5.2)



Şekil 5.2. Etkinlik Fonksiyonlarından Bazıları[11].

- Tek Kutuplu Sigmoid
- Doğrusal
- Hiperbolik Tanjant
- Keskin Sınırlayıcı

5.1.1. Girişler

x_1, x_2, \dots, x_n ile gösterilen girişler bilgiyi sinire getirir. Giriş dışarıdan gelebileceği gibi bir önceki sinirden de gelebilir.

5.1.2. Ağırlıklar

$\omega_1, \omega_2, \dots, \omega_n$ ile gösterilen ağırlıklar girişleri etkileyen katsayılarıdır. Herbir girişin kendine ait bir ağırlığı vardır. Giriş sinire güçlü bağlıysa ağırlık büyük, zayıf bağlıysa ağırlık küçük olur.

5.1.3. Toplama İşlevi

V ile gösterilir. Her bir giriş kendi ağırlığıyla çarpılarak toplanır, θ eşik değeri ilave edilir. Toplamın sonucu, f ile gösterilen etkinlik işlevine gönderilir.

5.1.4. Etkinlik İşlevi

f ile gösterilir. Toplama işlevinin sonucu etkinlik işlevinden geçirilir, çıkışa iletilir. Şekil 5.2'deki çeşitli etkinlik fonksiyon şekilleri verilmiştir. Etkinlik işlevinin amacı zaman söz konusu olunca toplama işlevinin çıkışının değişmesine izin vermektedir. Keskin sınırlayıcı fonksiyonlu sinir, etkinlik işlevinin eşik seviyesinin üzerinde çıkış üretir.

(Eşitlik 5.1)

$$y_i = \begin{cases} 1 & \text{eger } \omega_1.x_1 + \omega_2.x_2 + \dots + \omega_n.x_n \geq T \\ 0 & \text{eger } \omega_1.x_1 + \omega_2.x_2 + \dots + \omega_n.x_n < T \end{cases} \quad T:\text{Eşik seviyesi} \quad (5.1)$$

5.1.5. Ölçeleme ve Sınırlama

Sinirlerde etkinlik işlevinin sonucu işlevinin sonucu ölçek veya sınır işlemlerinden geçirilebilir. Ölçeklendirme, etkinliğin bir ölçek değeriyle çarpılmasıdır. Sınırlandırma ise ölçeklenmiş sonuçların en az veya en çok değerini ifade eder. Ölçeleme ve sınırlama yapmak zorunlu değildir.

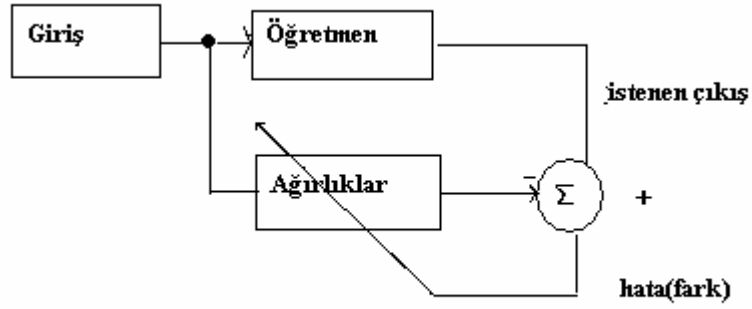
5.1.6. Çıkış İşlevi

Etkinlik işlevinin sonucunun dışarıya ya da varsa bir sonraki sinire gönderildiği kısımdır. Her sinirin bir çıkışı vardır.

5.1.7. Öğrenme

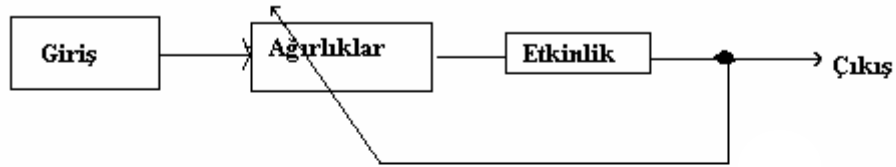
Öğrenme Hebbian öğrenme kuralı denilen basit bir modele dayanır. Bu kural temelde “Eğer iki düğüm aynı zamanda etkin ise aralarındaki bağ gücü artar” kuramına dayanmaktadır. Öğrenmenin amacı her bir sinir girişindeki değişken bağlantı ağırlıklarını belirlemektir. İstenen bazı sonuçları elde etmek için, giriş bağlantılarının ağırlıklarını değiştirme işlemi uyma işlevi olarak adlandırılabilir.

Danışmanlı ve danışmansız olmak üzere iki tip öğrenme türü vardır. Danışmanlı öğrenmede ağ önce eğitilir. Eğitime, sinir ağına giriş ve çıkış bilgilerini vermek demektir. Bu bilgiler eğitim kümesi olarak adlandırılır. Çıkış, istenen çıkışla karşılaştırılır. Ağırlıklar değiştirilerek hata azaltılır. (Şekil 5.3)



Şekil 5.3. Danışmanlı Öğrenme

Danışmansız öğrenmede sistemin doğru çıkış hakkında bilgisi yoktur, girişlere göre kendi kendini örnekler. Girişlere göre ağırlık değerleri ayarlanır. Danışmansız öğrenmeye Hebbian, Grossberg, Kohonen'in Özörgütlemeli Harita Ağı örnek verilebilir. Örneğin Kohonen'in Özörgütlemeli Harita Ağı Eğitim Fonksiyonunda sinirler öğrenmek için kendilerini güncellemede yarışır. En büyük çıkış ile işlenen sinir kazanır ve komşularına bağlantılarını güncellemeleri için izin verir. (Şekil 5. 4)



Şekil 5.4. Danışmansız Öğrenme

Önemli öğrenme kurallarında birkaçı aşağıda verilmiştir.

Hebb Kuralı : “Eğer bir sinir başka bir sinirden giriş alırsa ve her ikisi de aynı işaretli ise sinir arasındaki ağırlığı artırır.” Şeklinde dir.

Hopfield Kuralı: “Eğer istenilen çıkış ve girişin her ikisi de aktif veya her ikisi de durgun ise, ağırlıkları öğrenme oranı kadar artırır, aksi halde ağırlıkları öğrenme oranı kadar azaltır.” şeklindedir.

Delta Kuralı: Bu kural bir sinirin çıkışı ile istenilen çıkış değeri arasındaki farkı azaltmak için ağırlıkları sürekli olarak değiştirme fikrine dayanır.

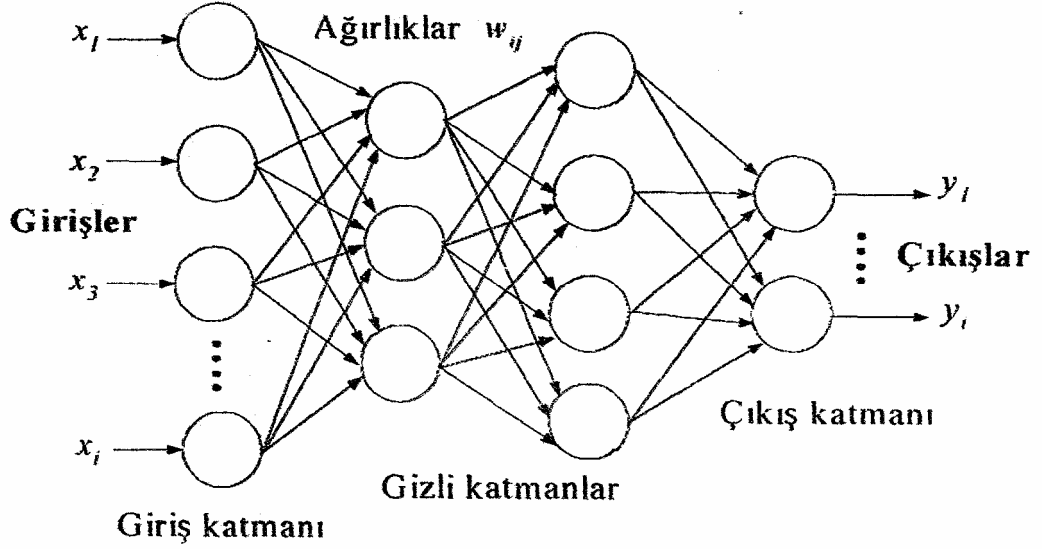
Kohonen Öğrenme Kuralı: Bu yöntemde sinirler öğrenmek amacıyla ağırlıklarını güncellemek için yarışır. Çıkışı büyük olan sinir kazanır. Komşularına ağırlıklarını güncellemeleri için izin verir. Neocognitron, Yarışmalı öğrenme, Adaptif rezonans teorisi gibi danışmansız yapay sinir ağı çeşitleri vardır.

5.2. Yapılarına Göre Yapay Sinir Ağları

Yapay sinir ağlarının düğüm ve bağlantıları çok farklı şekillerde olabilir. Ağlar bu düğüm ve bağlantılara göre farklı isimlerle anılırlar. Yapay sinir ağları aşağıdaki işlemlerin akış yönlerine göre ileri beslemeli (feedforward) ve geri beslemeli (feedback) olmak üzere genel olarak ikiye ayrılır.

5.2.1. İleri Beslemeli Ağlar

Ağ içindeki düğümler, katmanlar halinde yerleştirilir. İlk katmandaki girişe verilen bilgi ağ içinde ileriye doğru yayılır. Her katmandaki düğümlere sadece önceki katmandaki düğümlerden girişlere izin verilir. Bir düğüm kendisinden sonraki herhangi bir düğüme bağlanabilirken kendisine asla bağlanamaz. Son katmadaki işaretler ağın çıkışıdır.



Şekil 5.5. Çok Katmanlı İleri Beslemeli YSA[18].

Burada gizli katman sinirlerinin her biri kendinden önceki düğüm ya da sinirlerden aldığı girişleri ağırlıkları ile çarpar, toplamını alır ve kendinden sonraki varsa gizli katman nöronlarına, yoksa çıkış nöronlarının her birine iletir. Bu ağlar geri yayılım algoritmaları ile uygulanabilir hale gelmiştir.

Çıkışta türevi alınabilir etkinlik işlevi kullandığımız için, eğimli iniş öğrenme yönteminin gerçekleştirebileceği, türevi alınabilir hata işlevi tanımlanabilir.

$$\Delta w_{ij} = \eta \delta_j \cdot x_{ij} \quad (5.2)$$

Eşitlik (5.2)'de tanımlanan bu kural, daima “ hataların kareleri toplamını” en küçükleyecek ağırlık kümelerini bulmaya yöneliktir.

5.2.2. Geri Beslemeli Ağlar

Geri beslemeli ya da tekrarlanan ağlarda, en azından bir düğümün geriye yayıldığı bir dönüş bağlantısı vardır. Bu bağlar tamamen veya kısmen geri besleme yollarına sahiptirler. Tasarım ve davranışları oldukça karmaşıktır. Bu tür ağ mimarileri genellikle danışmansız öğrenme kurallarının uygulandığı ağlarda kullanılmaktadır. Hopfield ağı geri beslemeli ağa sahip bir yapay sinir ağıdır[19]. Şekil 5.6'da geri beslemeli Hopfield ağı görülmektedir. Bu tür ağlarda bir sinirin

çıkışı diğer her bir sinirin girişine bağlıdır. i . sinirin dış girdisi x_i , çıkışı y_i , j sinirinin çıkışı ile i . siniri arasındaki bağlantının ağırlığı ise w_{ij} ile gösterilmektedir. Ağın çalışması eşitlik (5.3) ile tanımlanır.

$$\frac{d v_i}{d t} = -v_i + \sum_{i=1}^n w_{ij} \cdot y_i + x_i \quad (5.3)$$

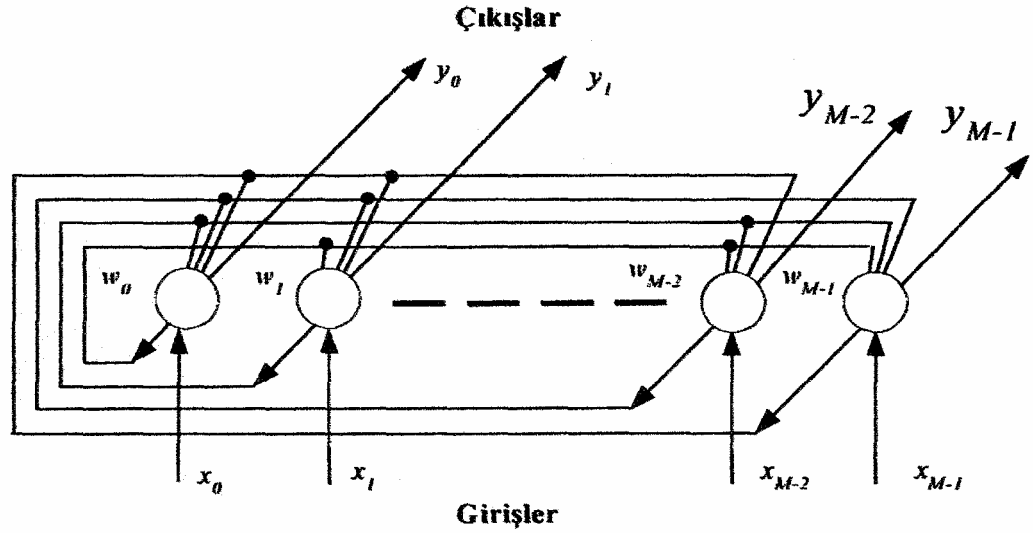
Buradan $y_i = g(v_i)$

$$g(x) = \frac{1}{2} \left[\tan\left(\frac{x}{a_o}\right) + 1 \right] \quad (5.4)$$

ile verilir. $a_o = 0$ olması durumu keskin sınırlayıcı etkinlik işlevine karşılık gelir. a_o değeri yeterince küçük ise, ağırlıkların simetrik, yani i ve j 'ler için $w_{ij} = w_{ji}$ olması durumunda Hopfield Ağı;

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \cdot y_i \cdot y_j - \sum_{i=1}^n y_i \cdot x_i \quad (5.5)$$

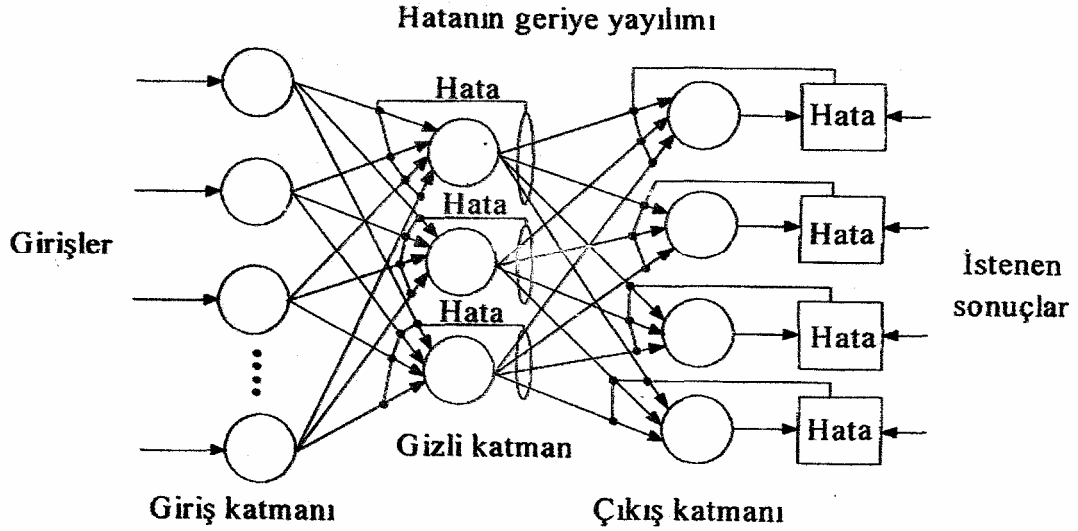
şeklinde bir sistem enerji işlevini en aza indirgeyecek şekilde davranır ve bu enerji işlevinin bir yerel minimuma karşılık gelen kararlı duruma ulaşır.



Şekil. 5.6. Hopfield Ağı[18].

5.3. Geri Yayılımlı Öğrenme

Geri yayılım eğitim fonksiyonu, bir öğrenme eğitim fonksiyonudur. Yapay sinir ağı yapılarından geri beslemeli ağlarla karıştırılmamalıdır. Geri yayılım ağı ileri beslemeli ağlarda kullanılmaktadır. Bu çalışmada da Matlab komutlarından olan “newff” komutu kullanılarak eğitim işlemi gerçekleştirilmiştir. Geri yayımlı öğrenen ağlar hiyerarşik yapıdadır. Giriş, çıkış ve en az bir gizli katman olarak üzere üç katmandan oluşur. Gizli katman ve gizli katmandaki düğüm sayısı değiştirilebilir. Düğüm sayısının artması ağın hatırlama yeteneğini arttırmakla birlikte öğrenme işleminin süresini uzatır. Düğüm sayısının azaltılması eğitim süresini kısaltmakta fakat hatırlatma yeteneğini azaltmaktadır. Giriş katmanındaki her bir düğüm gizli katmandaki her düğüme, gizli katman birden fazla ise bu katmandaki her bir düğüm kendisinden sonra gelen katmandaki her düğüme ve gizli katman çıkışındaki her düğüm çıkış katmanındaki her düğüme bağlıdır. Bir katmandaki hiçbir düğüm kendi katmanındaki diğer bir düğüme bağlı değildir. Her katmanın çıkış değeri bir sonraki katmanın giriş değeridir. Bu şekilde giriş değerlerinin ağın girişinden çıkışına doğru ilerlemesine ileri besleme denir. (Şekil 5.7)



Şekil 5.7. Gizli Katmanlı İleri Beslemeli Geri Yayılım Ağ Yapısı[18].

Geri yayılım ağında hatalar ileri besleme aktarım işlevinin türevi tarafından, ileri besleme mekanizması içinde kullanılan aynı bağlantılar aracılığıyla geri doğru yayılır.

Geri yayılım öğrenme yöntemi, türevi alınabilir etkinlik işlevlerini çok katmanlı herhangi bir ağa uygulayabilir. Delta kuralı gibi, bu da sistem hatasını azaltma esasına dayanan bir nevi optimizasyon işlemidir. Ağırlıklar geriye doğru ayarlandığı için geri yayılım adını almıştır. Öğrenme fazında giriş örnekleri ağa belli bir sırada verilir. Her örneğin çıktısı hesaplanana kadar ileri doğru katman katman hesap yapılır. Alınan çıkış, istenilen çıkışla karşılaştırılır. Aradaki fark “hata” olarak bulunur. Bu hatalar katman katman sinaptik ayarlamaların yapıldığı geri besleme bağlantılarında “girdi” olarak kullanılır. Şekil 5.7’de ileri doğru olan bağlantılar hem öğrenme, hem de işlemler için kullanılırken, geriye doğru olan bağlantılar sadece “öğrenme fazı” için kullanılır. Geri yayımlı öğrenme kullanıldığında, sonraki katmanların hataları kullanılarak gizli katmanın ağırlıkları ayarlanır. Böylece çıkış katmanında hesaplanan hatalar son gizli katman ile çıkış katmanı arasındaki ağırlıklar ayarlanır. Aynı şekilde bu işlemler ilk gizli katmana kadar tekrarlanır. Bu yolla hatalar katman katman ilgili katmanın ağırlık düzeltmeleri yapılarak geriye doğru yayılır. “Toplam hata” en aza indirilinceye kadar bu işlemler tekrarlanır.

Aşağıda geri yayılım algoritması anlatılacaktır. Anlatımı basitleştirmek için tek gizli katmandan oluşan çok katmanlı ileri beslemeli ağ ele alınacaktır. Anlatımdan sonra yine bu konu ile ilgili basit bir örnek yapılacak, son olarak da Geri Yayılım Kuralı Algoritması verilecektir.

5.4. Geri Yayılım Algoritması

Önemli olduğu için Q tane katmanlı ileri beslemeli ağ için geri yayılım algoritması aşağıda anlatılmıştır.

Algoritmada kullanılan değişkenler şu anlamlara gelmektedir.

$q=1, 2, \dots, Q$ katman numarası

$H_i^q = q.$ katmandaki i biriminin girdisi

$y_i^q = q.$ katmandaki i biriminin çıktısı

$w_{ij}^q = (q-1).$ katmandaki i birimini, $q.$ katmandaki j birimine bağlayan ağırlıktır.

1. w' ye reel değerli küçük rastgele sayılar başlangıç değeri olarak atanır.
2. Rastgele bir $\{x^p, t^p\}$ (giriş, hedef) eğitim çifti seçilir ve q katmanındaki her bir j birimi için ileri yönde "çıkış" değerleri hesaplanır. Böylece çıkış,

$$y_i^q = f\left(\sum_i y_i^{q-1} w_{ij}^q\right)$$

olur. 1 nolu katmanın girdileri 0 üssüyle etiketlenmiştir, yani;

$$y_i^0 = x_i$$

3. son Q katmanında hesaplanan i birimleri için y_i^0 ve karşılık gelen y_i^p değerleri Delta miktarlarının hesaplanmasında kullanılmalıdır. Çıkış birimleri için hata terimleri hesaplanır.

$$\delta_i^0 = (v_i^0 - y_i^p) f'(H_i^0)$$

4. $q=Q, Q-1, \dots, 2$ katmanlarındaki tüm i birimleri için geriye yayılımla deltalar yani gizli katman birimleri için hata terimleri hesaplanır.

$$\delta_i^{q-1} = f'(H_i^{q-1}) \sum_i \delta_i^q w_{ij}^q$$

5. Bütün ağırlıklar w_{ij} ler kullanılarak güncellenir.

$$w_{ij}^{yeni} = w_{ij}^{eski} + \Delta w_{ij}^q$$

$$\Delta w_{ij}^q = \eta \delta_i^q y_i^{q-1}$$

6. 2. adıma dönülüp fark kabul edilebilir bir düzeye gelene kadar her bir p modeli için işlemler tekrarlanır[18].

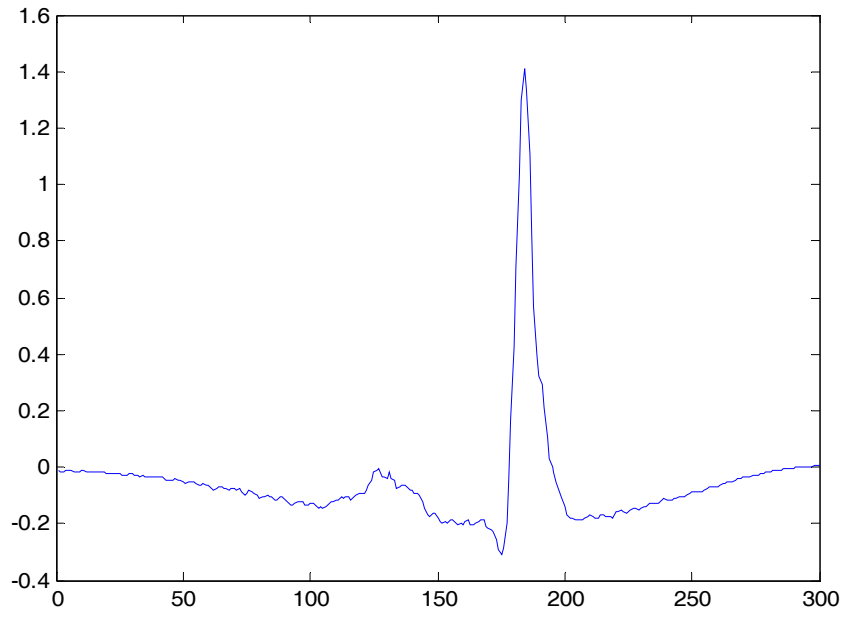
6. DÖNÜŞÜM İŞLEMLERİ VE TEST SONUÇLARI

6.1. Çalışmada Kullanılan Datalar

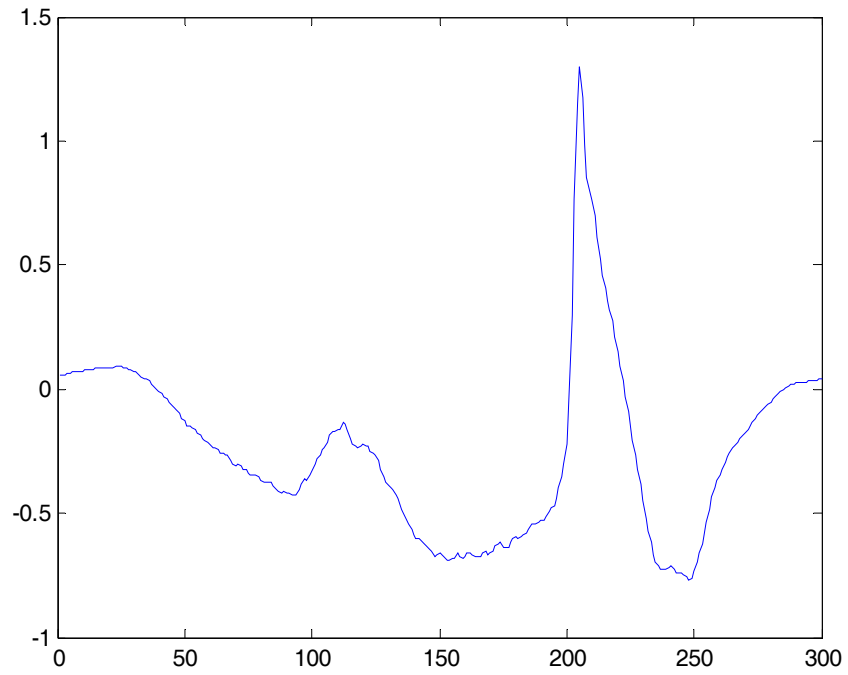
Bu çalışmada kullanılan EKG dataları MIT (Massachusetts Institute of Technology) veri tabanından elde edilmiştir. Normal, Pace, Rbbb, Lbbb olmak üzere dört farklı data grubu vardır. Ve her biri 634000 veriden oluşmuştur. Öznitelik çıkarmada bu dataların 150000'den 200000'e kadar olan kısmı alınmıştır. Sınıflandırmada ise 350000'den 400000'e kadar olan kısmı alınmıştır. "Normal" adlı data sağlıklı data olup, diğerleri hastalıklı datalardır. Dataların periyodu 0.004 sn. olup frekansı 250 Hz. 'dir. (Tablo1.1)

6.2. Özellik Vektörlerinin Çıkarılması

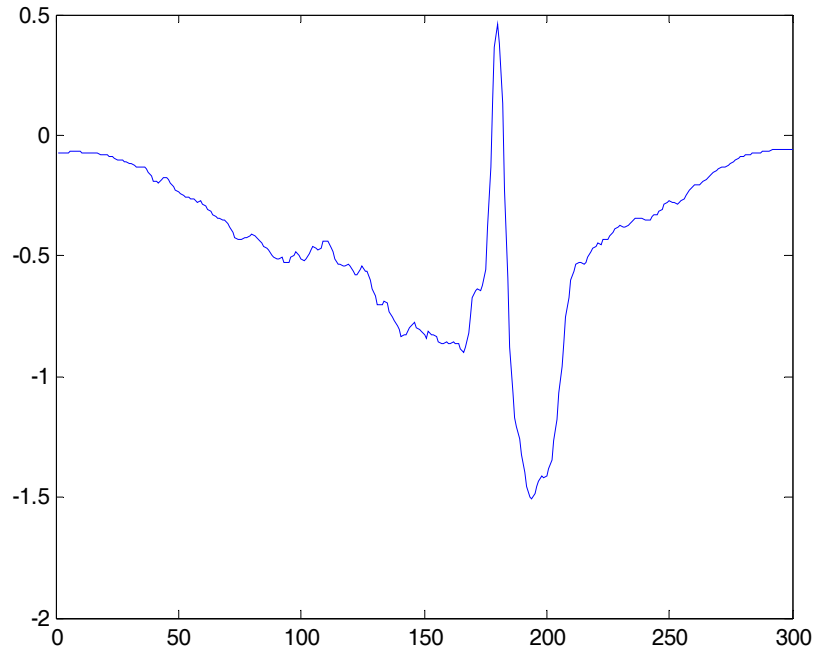
Seçilen özelliklerin mümkün olduğu kadar basit işlemlerle oluşturulması, eğitim ve sınıflandırma zamanlarının kısılmasına , böylece algoritmanın hızlı çalışmasına yol açacaktır[24,25]. 50000 verili bu ham datalar önce Matlab "Buffer" komutu ile 300 verili 250 vuruya ayrılır. Herbir vuru arasında bilgi kaybı olmaması için de bu noktalar kesiştirilir. Bu kesiştirme çalışmada 100 veri şeklinde yapıldı. Aşağıdaki şekillerde özellik çıkarımı yapılacak olan dört farklı data tipine ait vurulardan birer tane örnek görülmektedir.



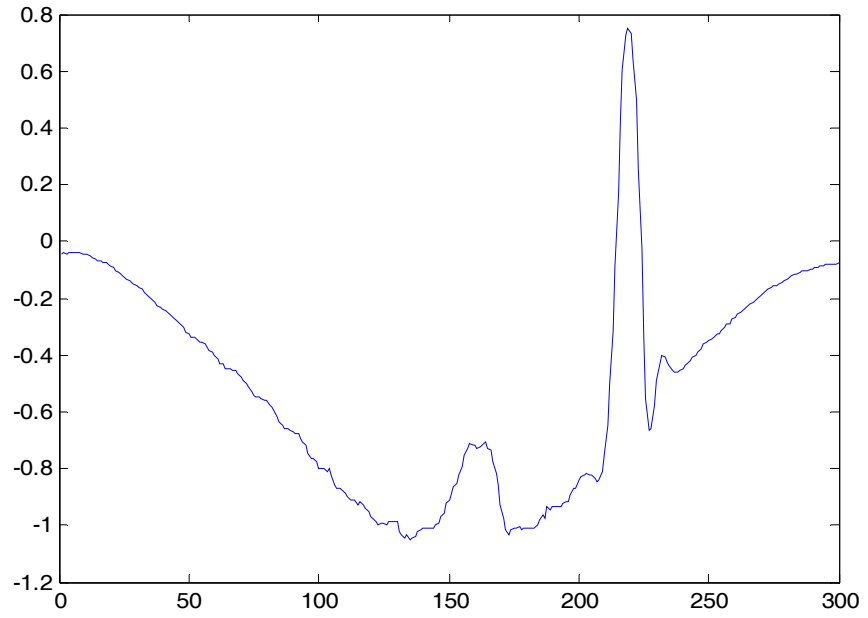
Şekil 6.1. Özellik Çıkarımı Yapılacak “Normal” Data



Şekil 6.2. Özellik Çıkarımı Yapılacak “Pace” Data



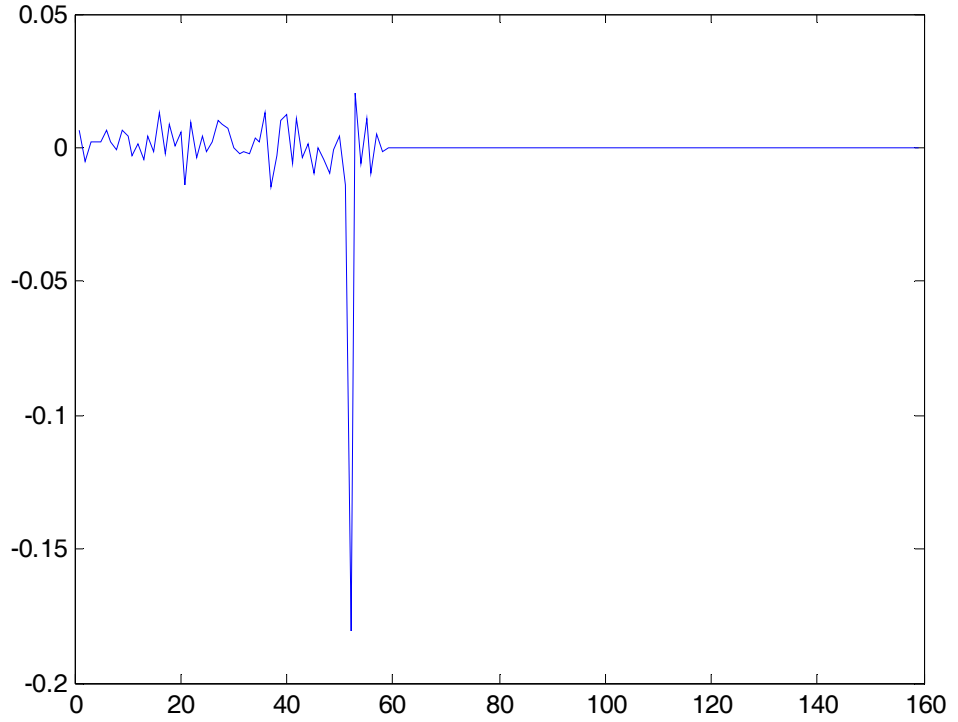
Şekil 6.3. Özellik Çıkarımı Yapılacak “Rbbb” Data



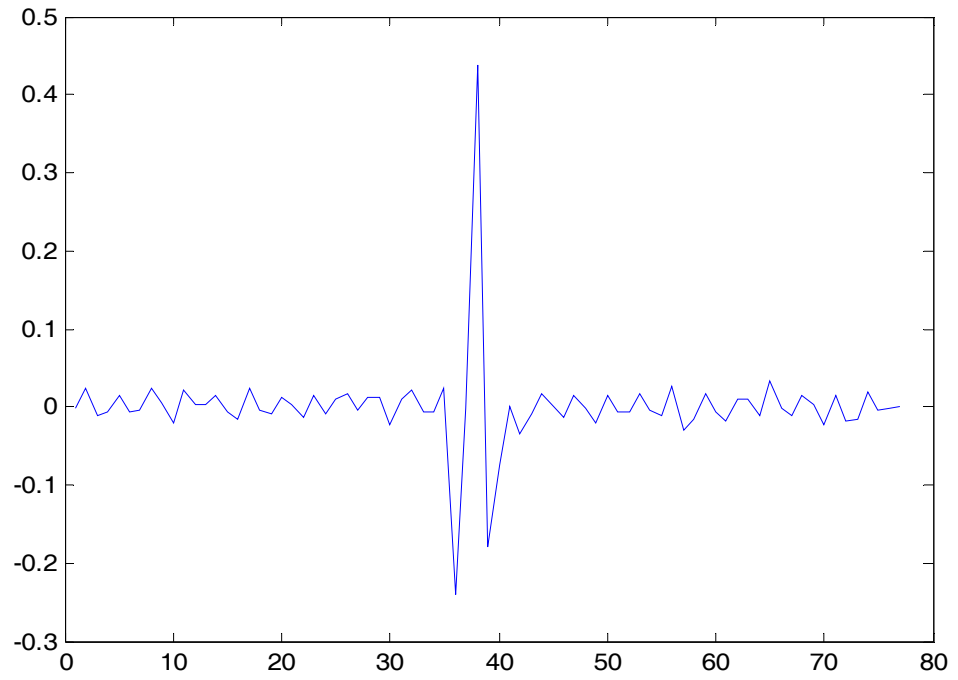
Şekil 6.4. Özellik Çıkarımı Yapılacak “Lbbb” Data

6.2.1. Dalgacık Dönüşümü (Wavelet) İle Özellik Çıkarımı

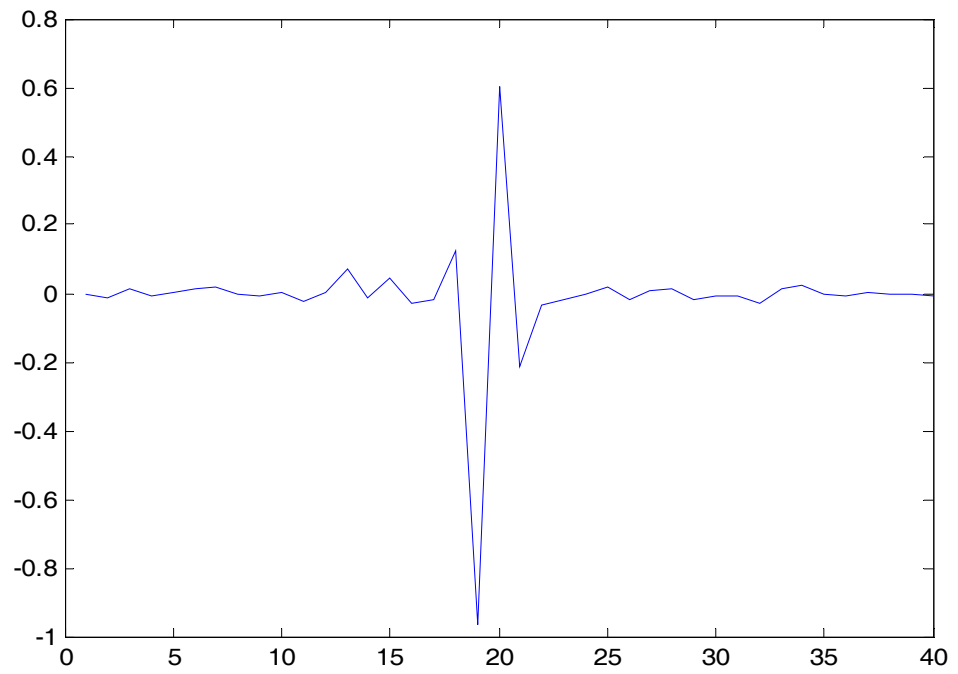
Parçalara ayrılmış ham datalar Matlab “Wavedec” komutu ile dalgacık dönüşümüne tabi tutulur. cD1, cD2, cD3, cD4 olmak üzere dört tane detay katsayısı elde edilir. Dalgacık olarak Daubechies-2 (db-2) ile Daubechies-10 (db-10) kullanılmıştır. db-10 adlı dalgacığın daha iyi sonuç verdiği görülmüştür. QRS kompleksinin R tepesinin frekansı 4. seviye detay katsayısına karşılık geldiğinden özellik çıkarımında cD4 olarak adlandırılan değerler kullanılmıştır.



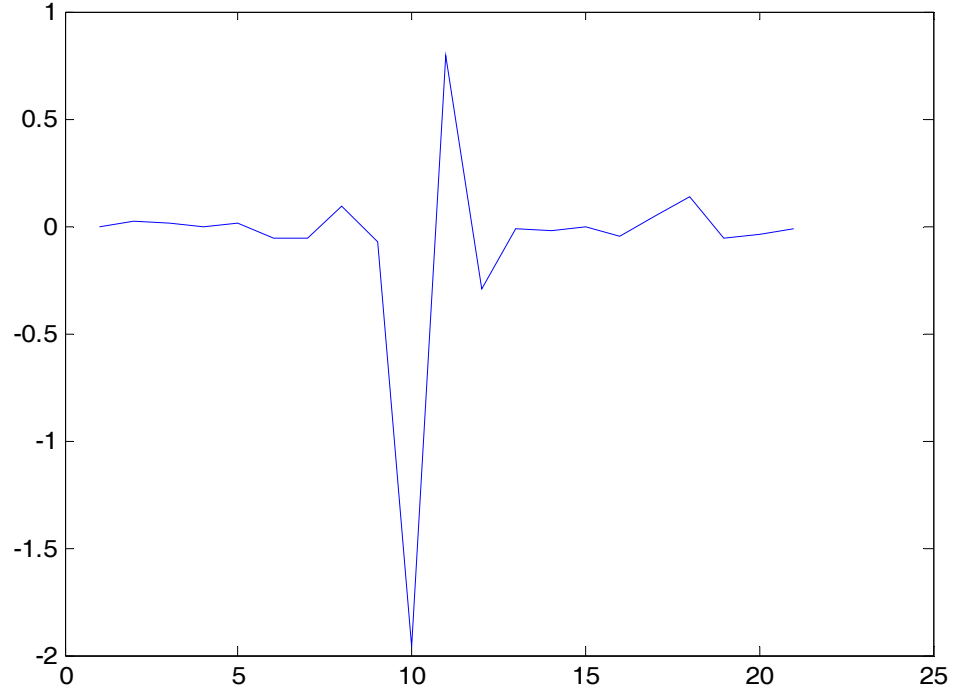
Şekil 6.5. Normal Data 1. Seviye Detay Katsayı Değerleri



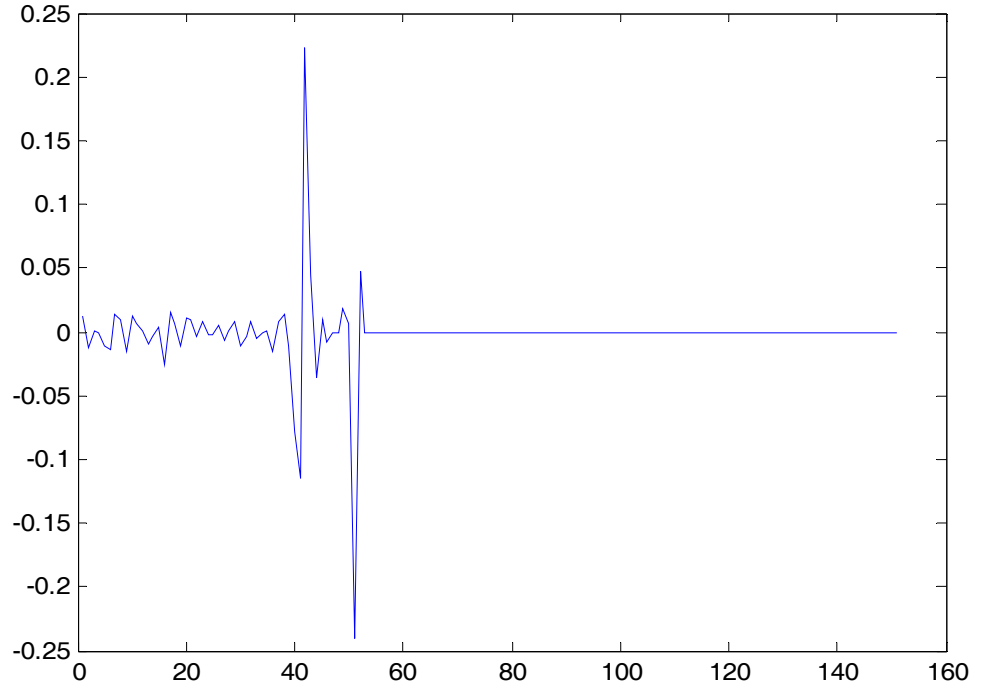
Şekil 6.6. Normal Data 2. Seviye Detay Katsayı Değerleri



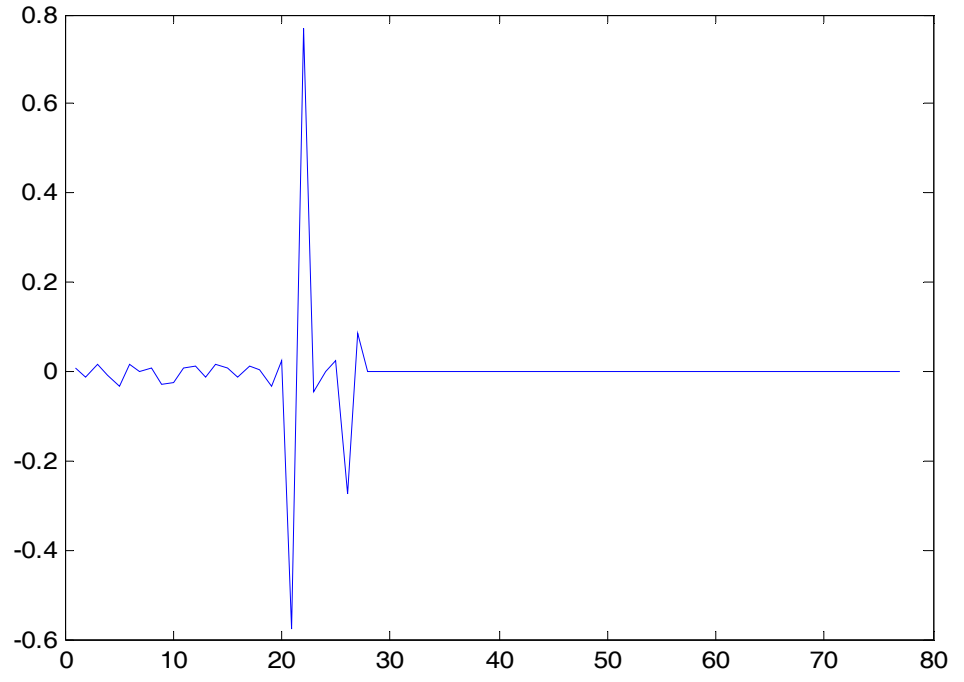
Şekil 6.7. Normal Data 3. Seviye Detay Katsayı Değerleri



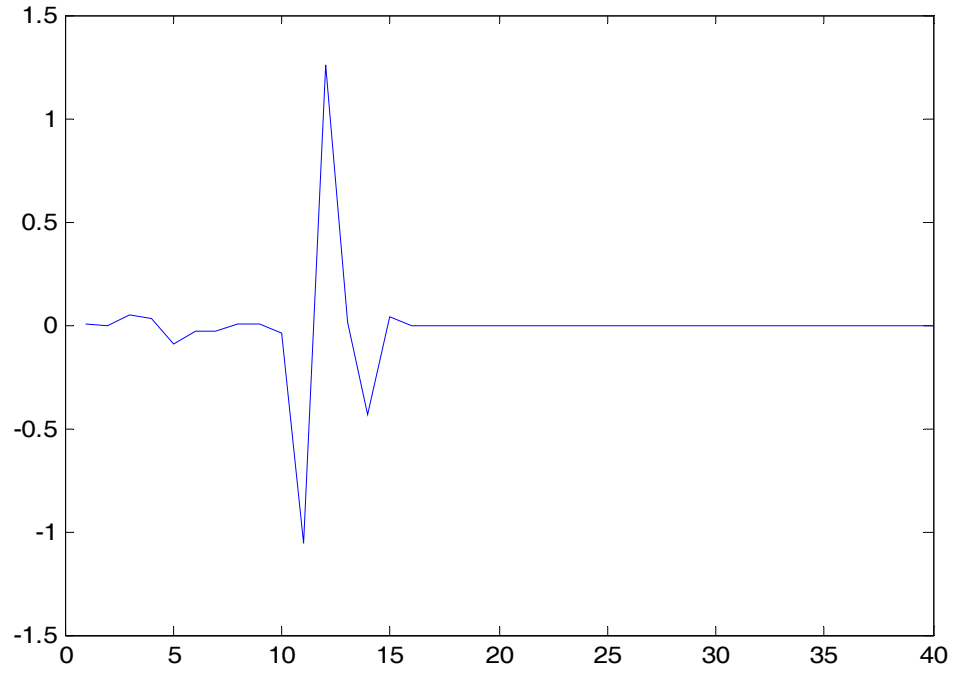
Şekil 6.8. Normal Data 4. Seviye Detay Katsayı Değerleri



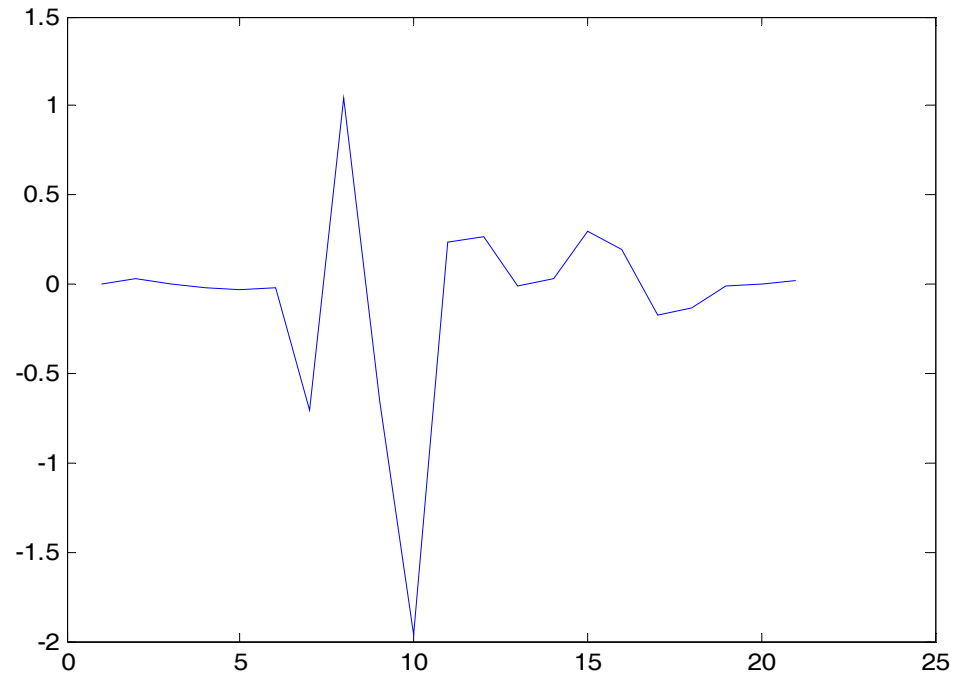
Şekil 6.9. Pace Data 1. Seviye Detay Katsayı Değerleri



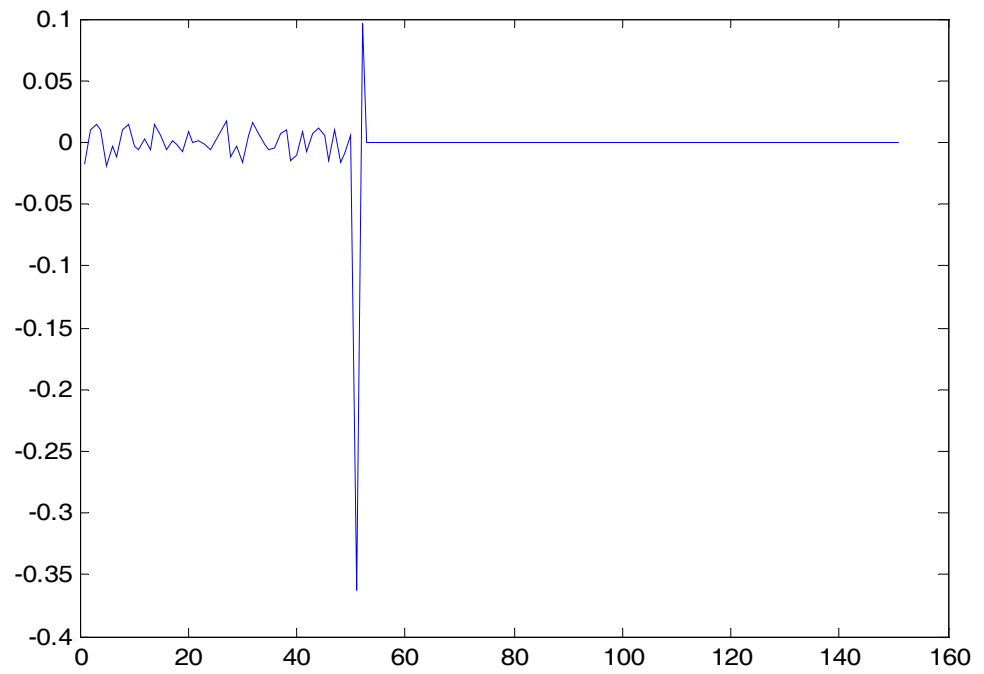
Şekil 6.10. Pace Data 2. Seviye Detay Katsayı Değerleri



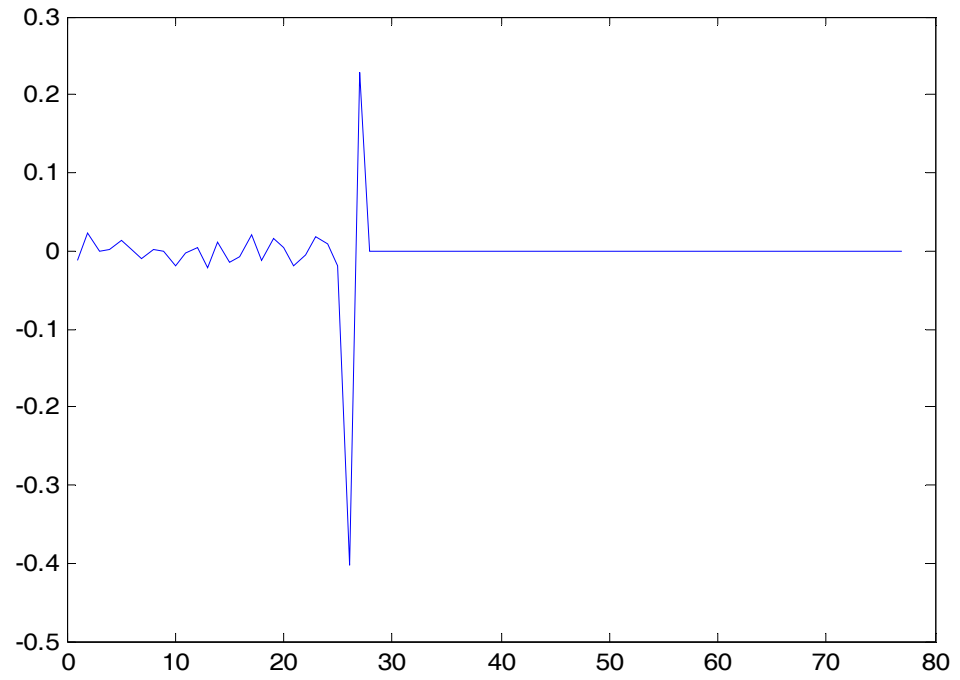
Şekil 6.11. Pace Data 3. Seviye Detay Katsayı Değerleri



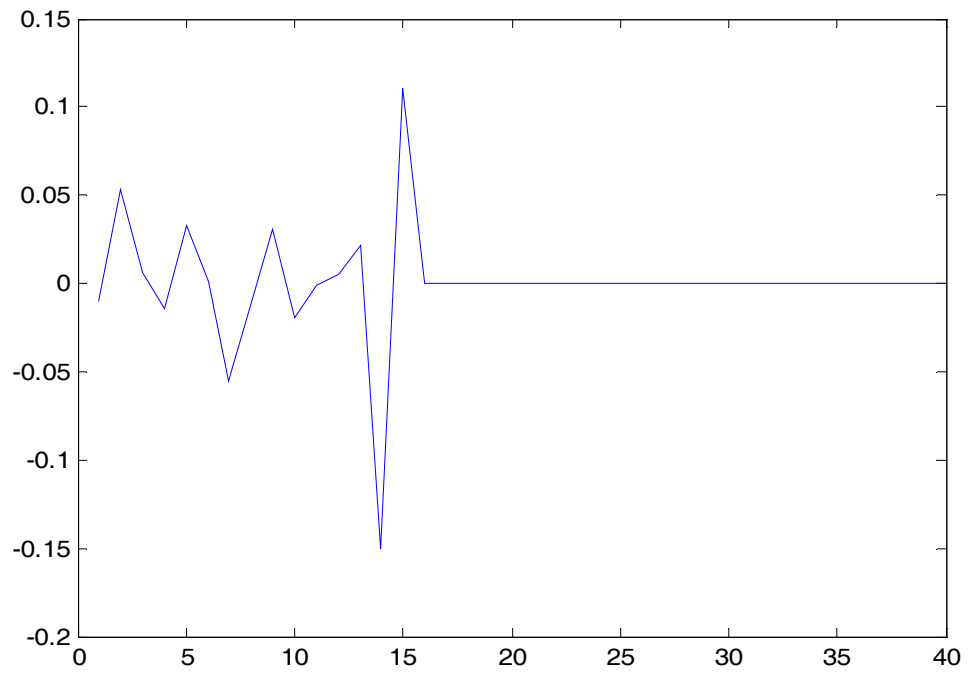
Şekil 6.12. Pace Data 4. Seviye Detay Katsayı Değerleri



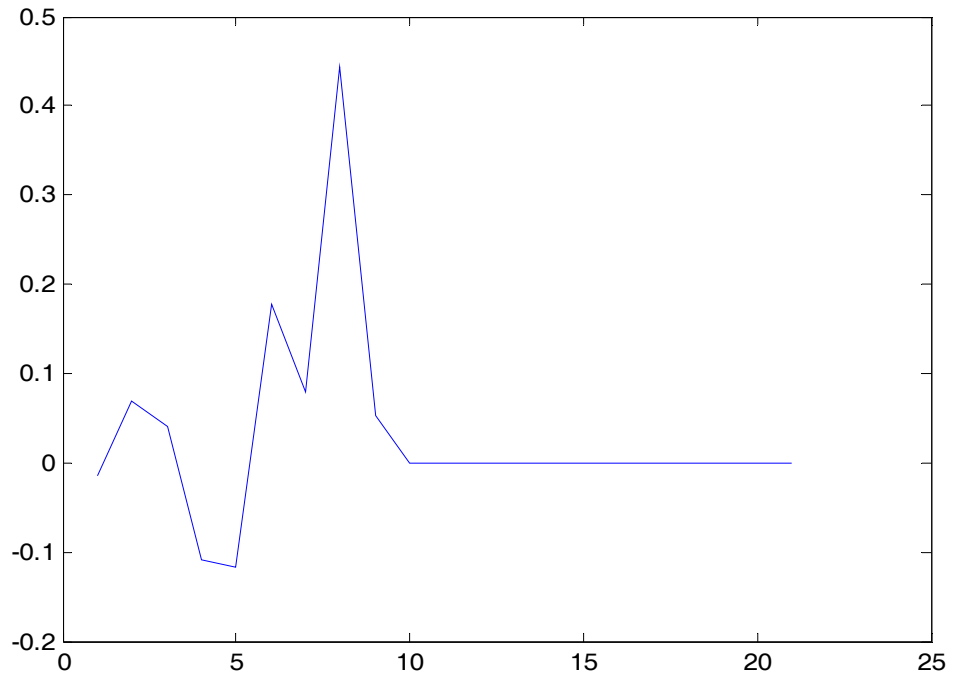
Şekil 6.13. Rbbb Data 1. Seviye Detay Katsayı Değerleri



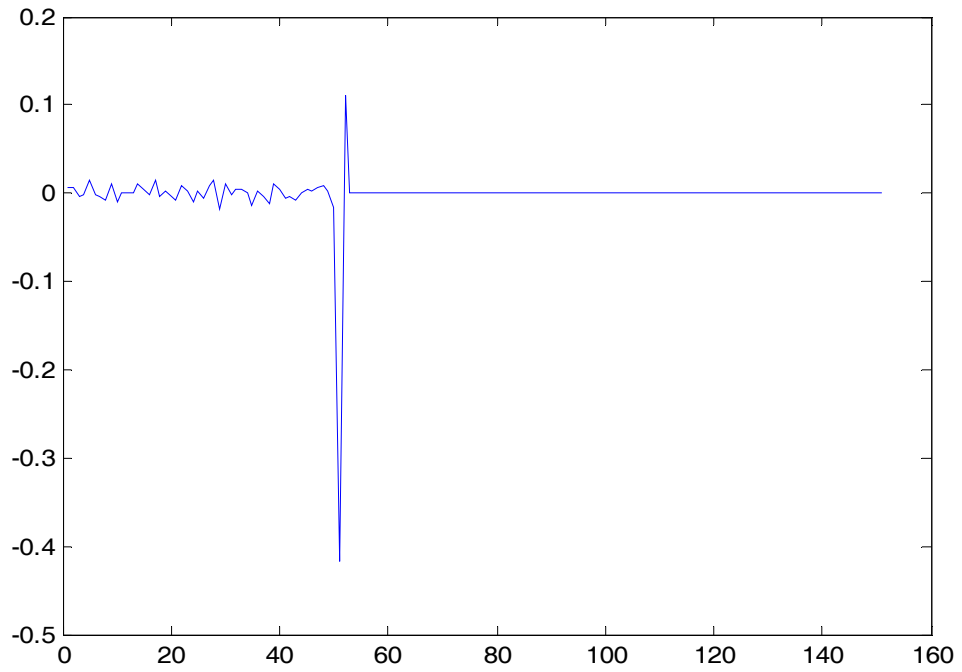
Şekil 6.14. Rbbb Data 2. Seviye Detay Katsayı Değerleri



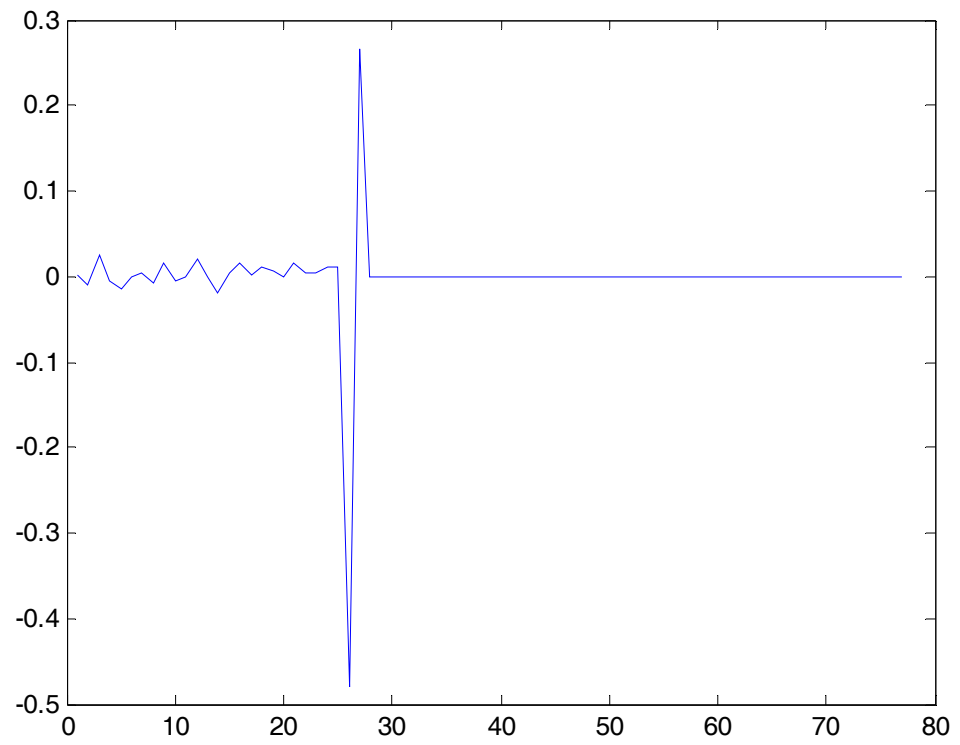
Şekil 6.15. Rbbb Data 3. Seviye Detay Katsayı Değerleri



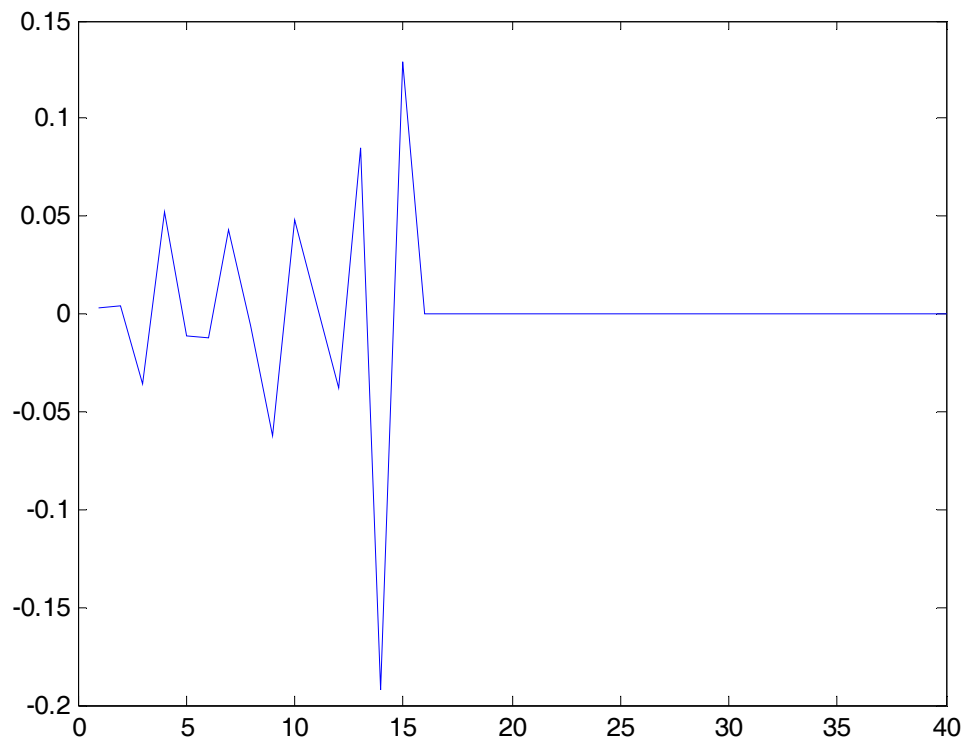
Şekil 6.16. Rbbb Data 4. Seviye Detay Katsayı Değerleri



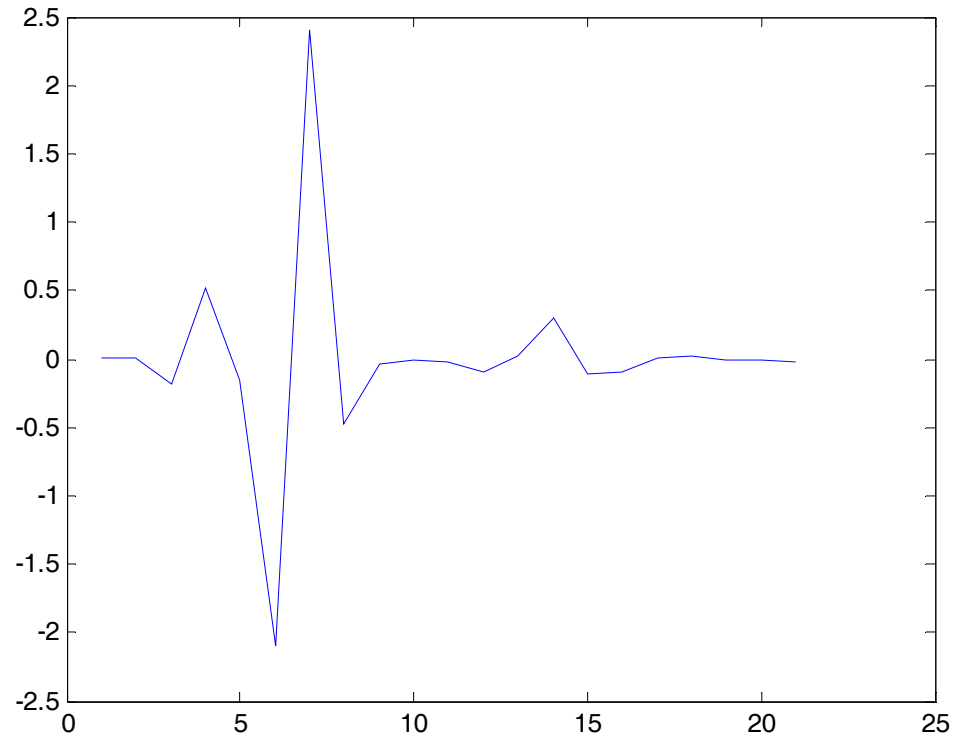
Şekil 6.17. Lbbb Data 1. Seviye Detay Katsayı Değerleri



Şekil 6.18. Lbbb Data 2. Seviye Detay Katsayı Değerleri



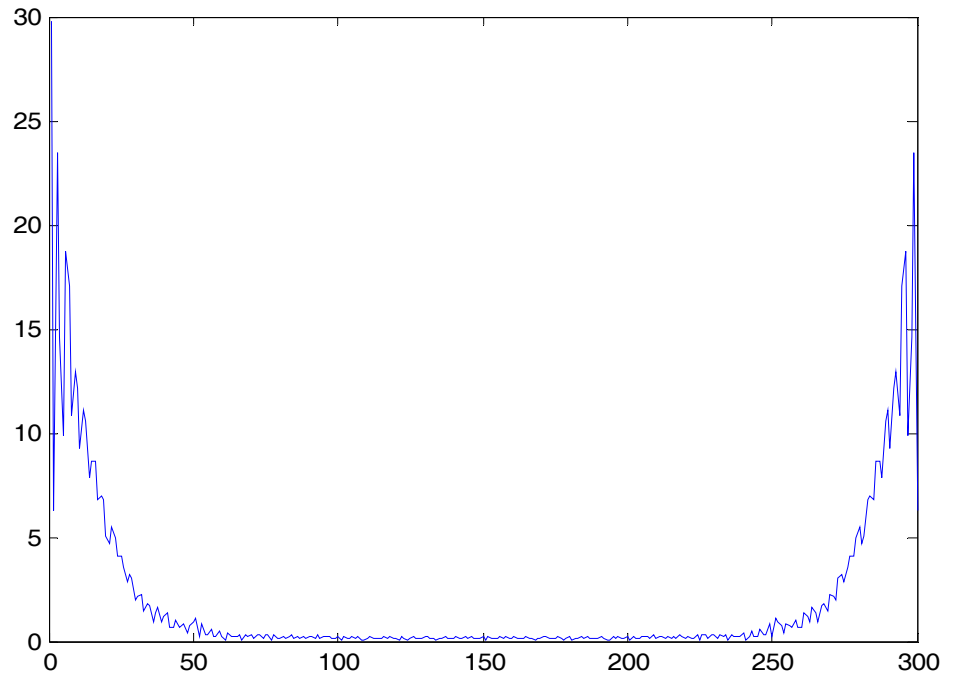
Şekil 6.19. Lbbb Data 3. Seviye Detay Katsayı Değerleri



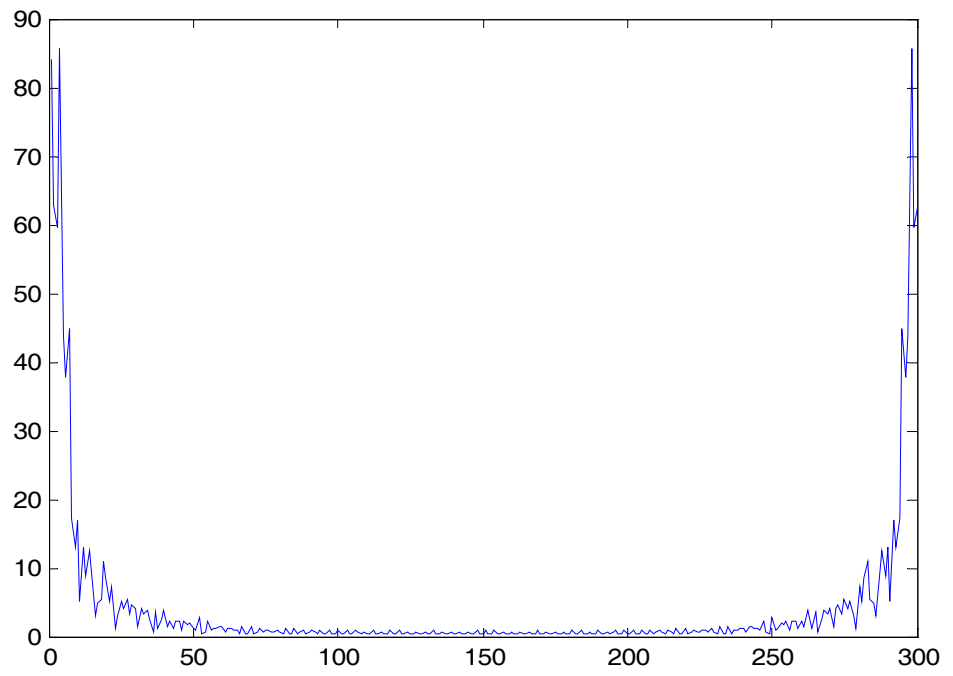
Şekil 6.20. Lbbb Data 4. Seviye Detay Katsayı Değerleri

6.2.2. Fourier Dönüşümü ile Özellik Çıkarımı

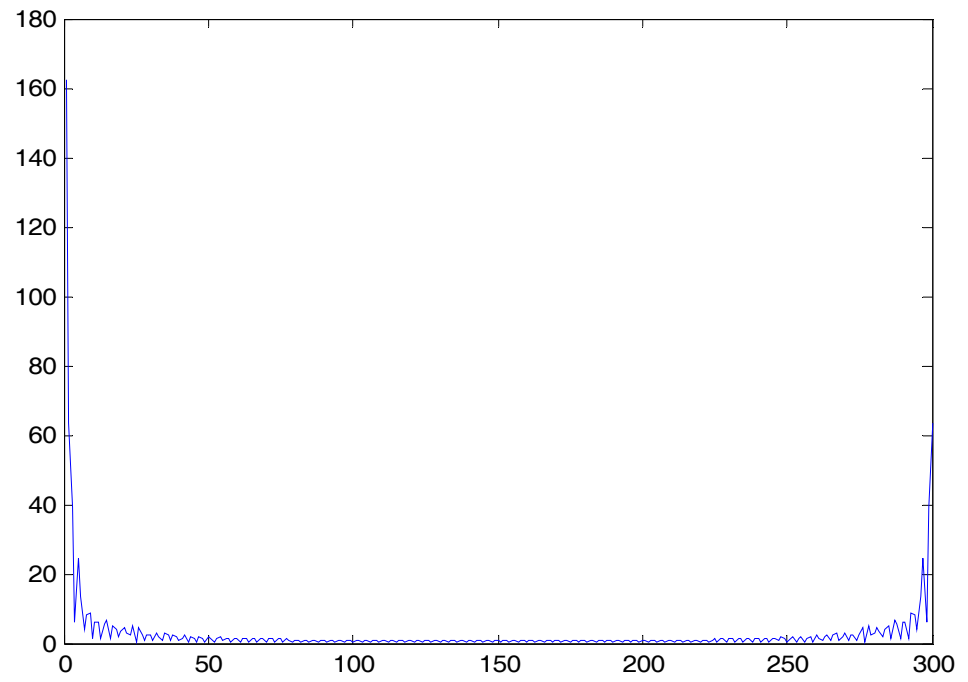
Fourier dönüşümünde yine dalgacık dönüşümünde kullanılan datalar kullanıldı. Bu dataların fourier dönüşümü Matlab paket programı “fft” komutu ile yapıldı. Dönüşümden sonra elde edilen 300 verili özellik vektörüne bakıldı. Bu vektörlerin ilk 100 verisinde anlamlı bilgilerin olduğu görüldü. Bundan dolayı yapay sinir ağırları ile sınıflandırmada 250 vurunun ilk 100'er verisi kullanıldı.



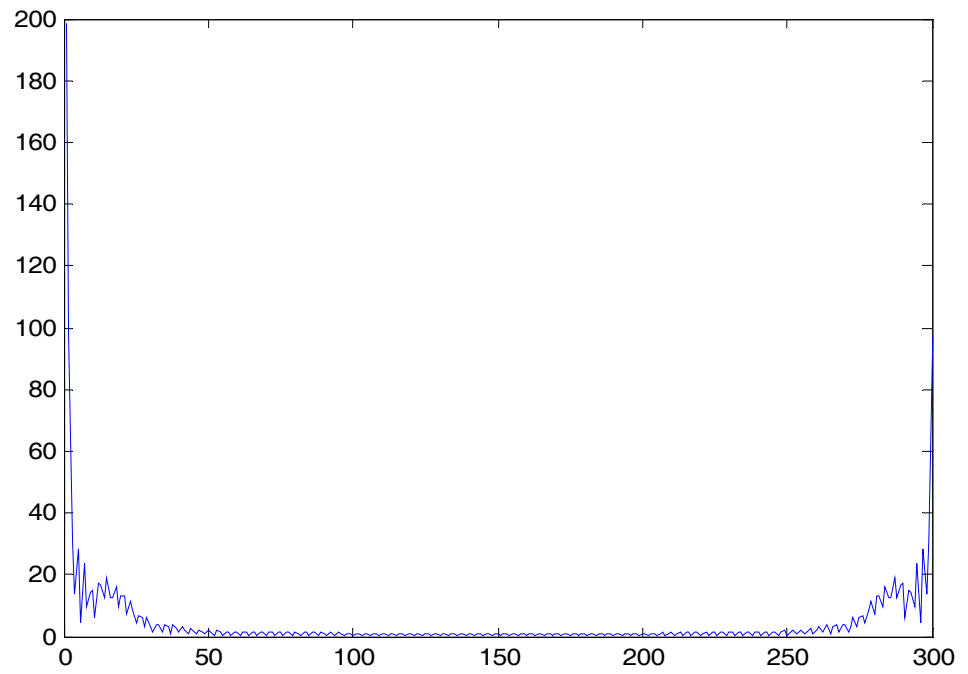
Şekil 6.21. Normal Datanın Fourier Dönüşümü



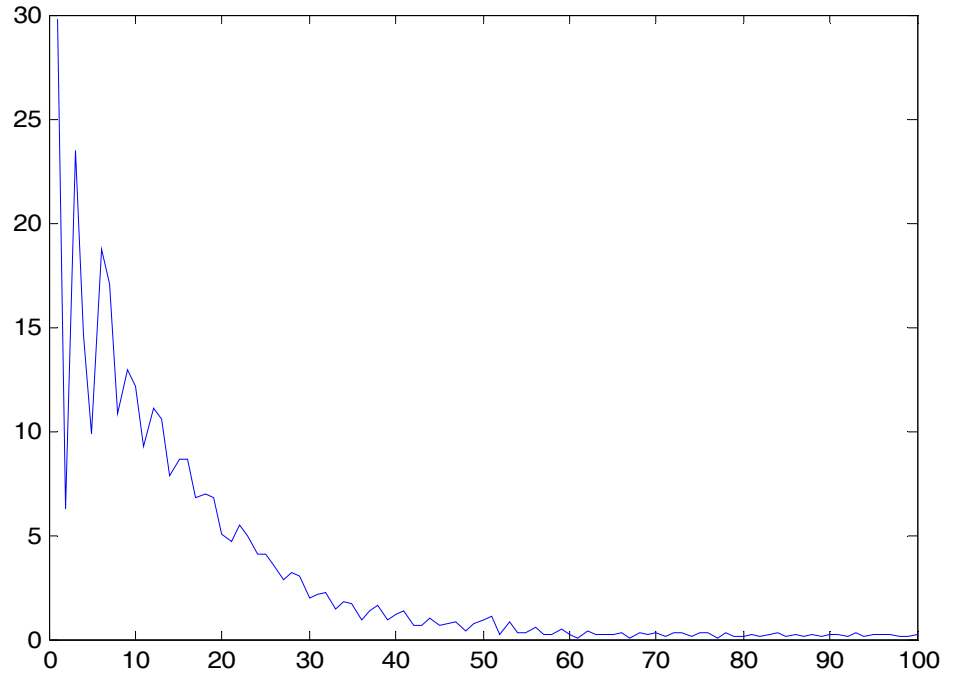
Şekil 6.22. Pace Datanın Fourier Dönüşümü



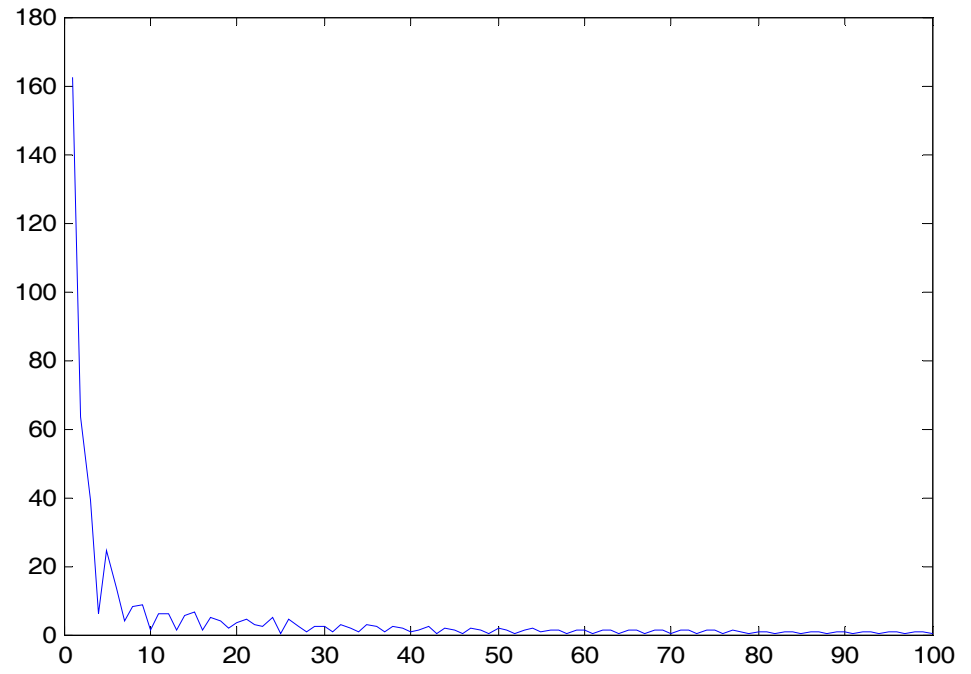
Şekil 6.23. Rbbb Datanın Fourier Dönüşümü



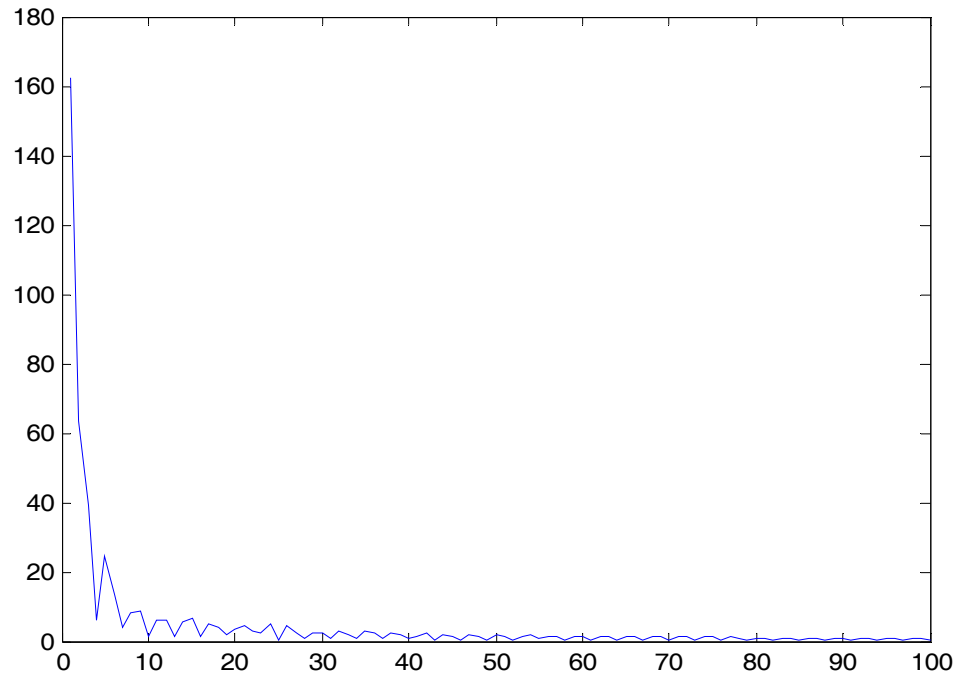
Şekil 6.24. Lbbb Datanın Fourier Dönüşümü



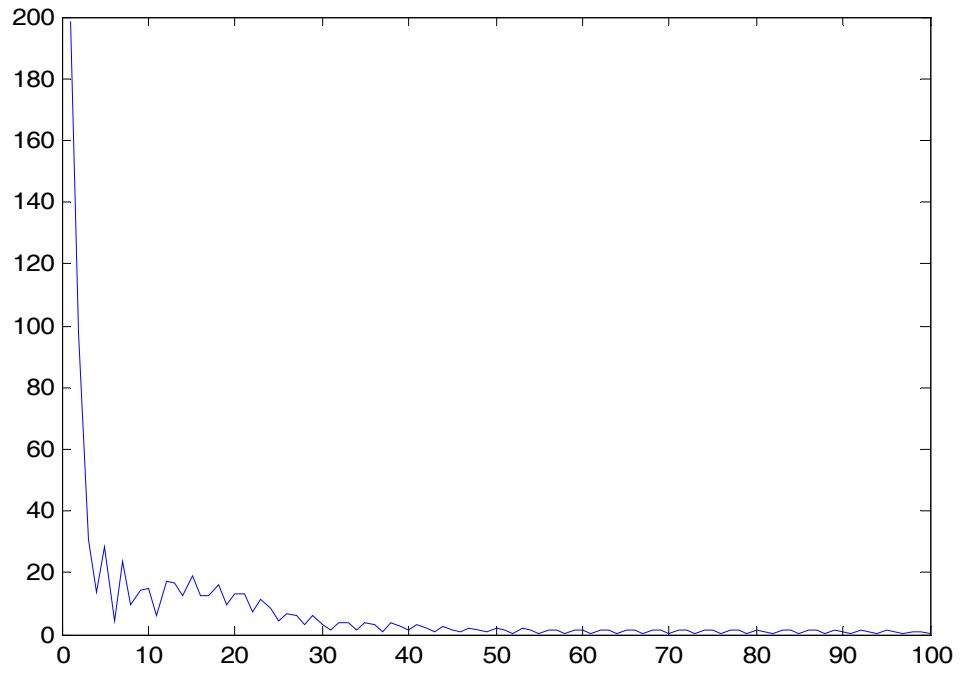
Şekil 6.25. Normal Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü



Şekil 6.26. Pace Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü



Şekil 6.27. Rbbb Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü

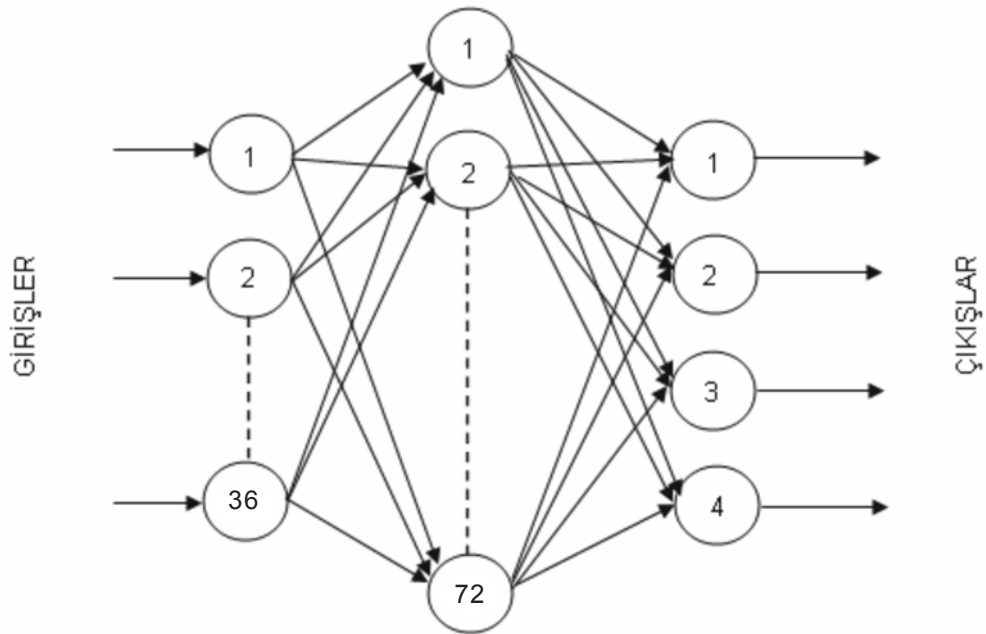


Şekil 6.28. Lbbb Datanın Sınıflandırmada Kullanılan Fourier Dönüşümü

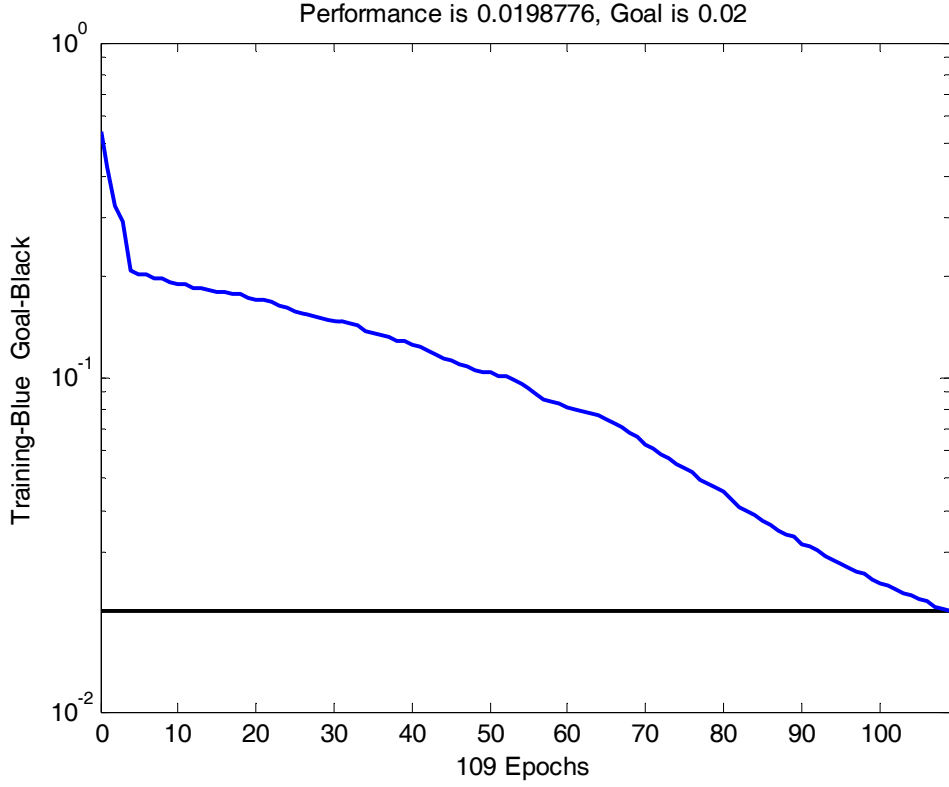
6.3. Yapay Sinir Ağları İle Sınıflandırma

6.3.1. Yapay Sinir Ağının Eğitimi

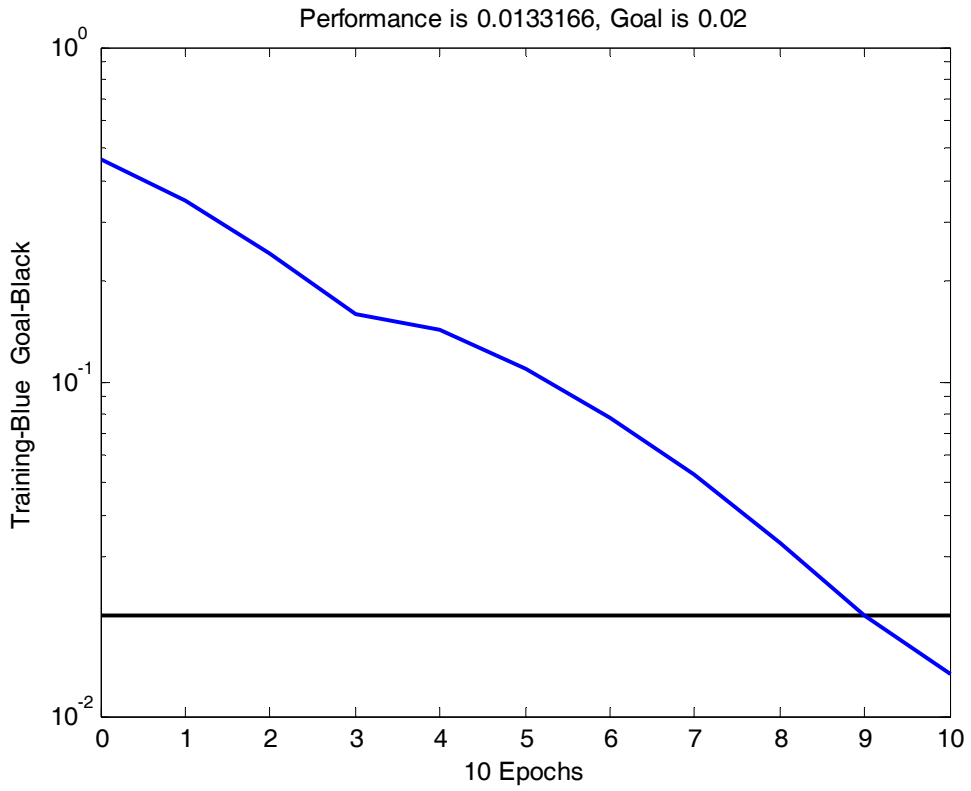
Dalgacık Dönüşümü detay katsayıları ile Fourier Dönüşümü parametreleri zamandan kazanmak için Excel dosyası olarak kaydedildi. Bu kayıtlı özellik vektörleri Matlab “xlsread” komutu ile çağrılarak yapay sinir ağına girildi. Dalgacık dönüşümünde 36 satır, 250 sütunlu giriş vektörü kullanıldı. YSA dört farklı sınıflandırma yapacağından çıkış sayısı 4 olarak belirlendi. Gizli katman sayısı 1, nöron sayısı çeşitli denemelerden sonra 72 olarak belirlendi. Öğrenmesini istediğimiz girişin karşılığı olarak ilgili çıkış 1 yapıldı. Öğrenme modeli olarak Scg, Lm, Gdx ve Bfg fonksiyonları kullanıldı. En yüksek verimin Scg eğitim fonksiyonunda olduğu görüldü. Etkinlik fonksiyonu olarak Hiperbolik tanjant fonksiyonu kullanıldı.



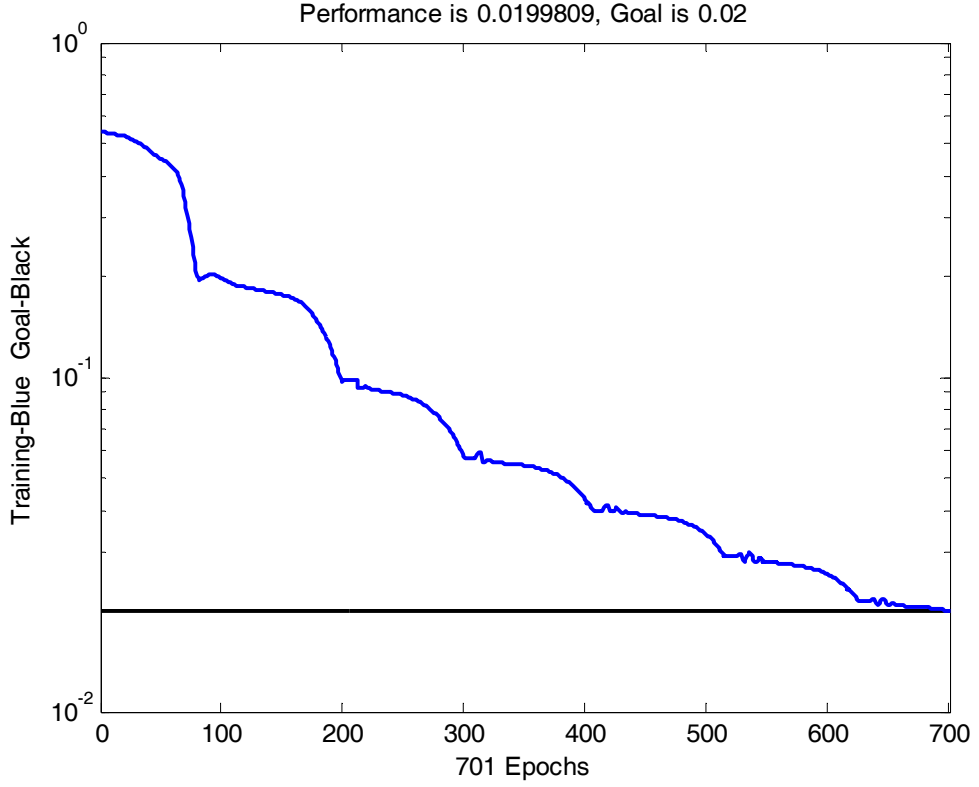
Şekil 6.29. 36 Giriş, 4 Çıkışlı, 1 Gizli Katmana Sahip YSA



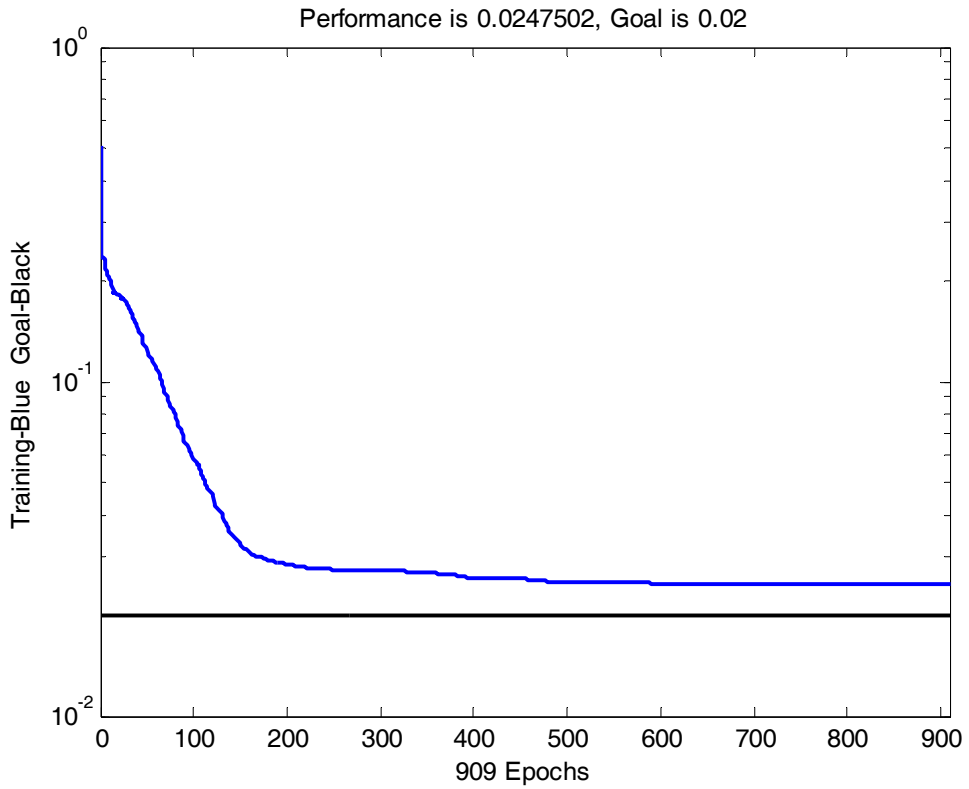
Şekil 6.30. Scg Eğitim Fonksiyonu ile Öğrenme Eğrisi



Şekil 6.31. Lm Eğitim Fonksiyonu ile Öğrenme Eğrisi

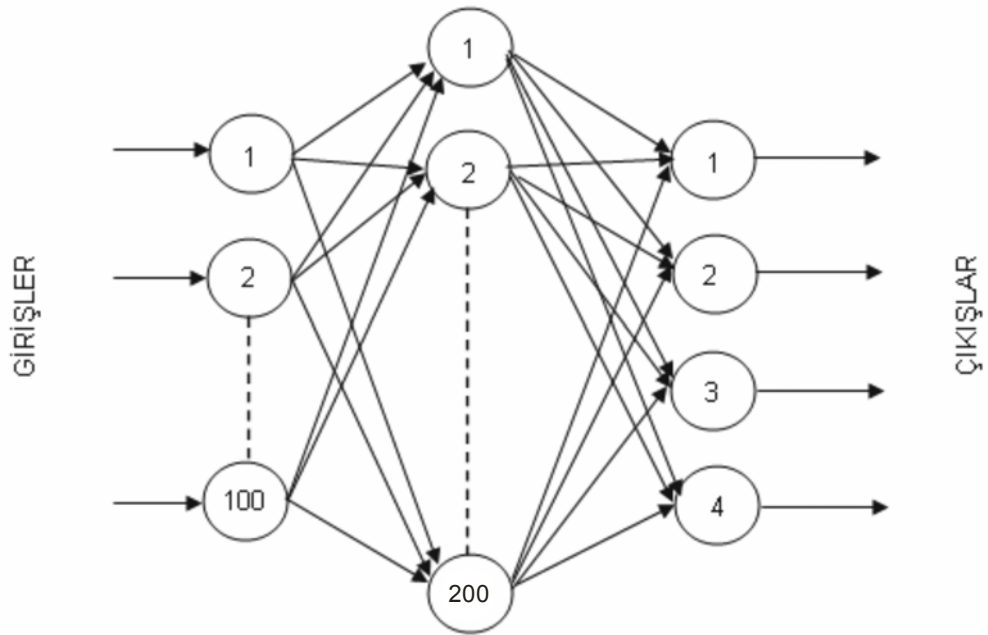


Şekil 6.32. Gdx Eğitim Fonksiyonu ile Öğrenme Eğrisi

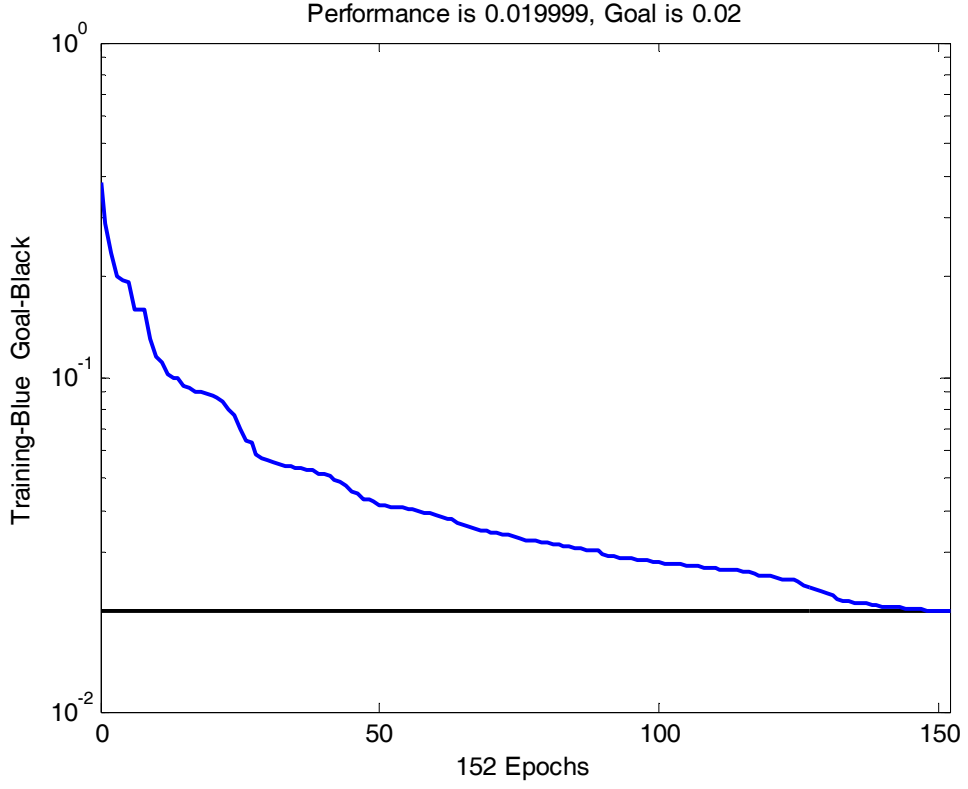


Şekil 6.33. Bfg Eğitim Fonksiyonu ile Öğrenme Eğrisi

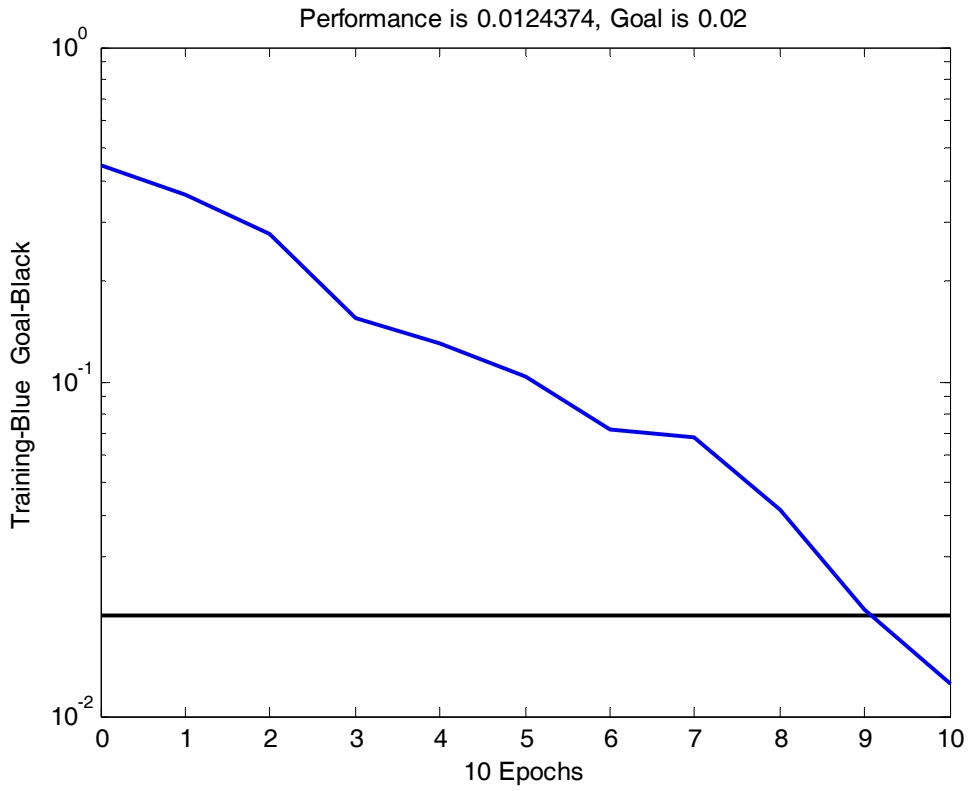
Fourier Dönüşümünde ise 300X250'lik matrisin, anlamlı olduğu için ilk 100X250'lik kısmı alındı. Bu matrisler Excel dosyası olarak kaydedildi. Bu matrisler Matlab "xlsread" komutu ile çağrılarak kullanıldı. Eğitimde data ile ilgili anlamlı bilgi taşıyan bu 100X250'lik veri YSA'nın girişine verildi. Yani 100 girişli bir YSA tasarlandı. Bu YSA 100 giriş, 4 çıkış ve 1 gizli katmandan oluşturuldu. Gizli katman nöron sayısı 200 olarak belirlendi. Öğretilecek olan girişin çıkışı 1 olarak belirlendi. Öğrenme modeli olarak yine Scg, Lm, Gdx ve Bfg kullanıldı. Yine Scg eğitim fonksiyonu ile yüksek sınıflandırma başarısı elde edildi. Etkinlik fonksiyonu olarak Hiperbolik tanjant fonksiyonu kullanıldı. 1000 denemede %2 hata ile öğrenme gerçekleştirildi.



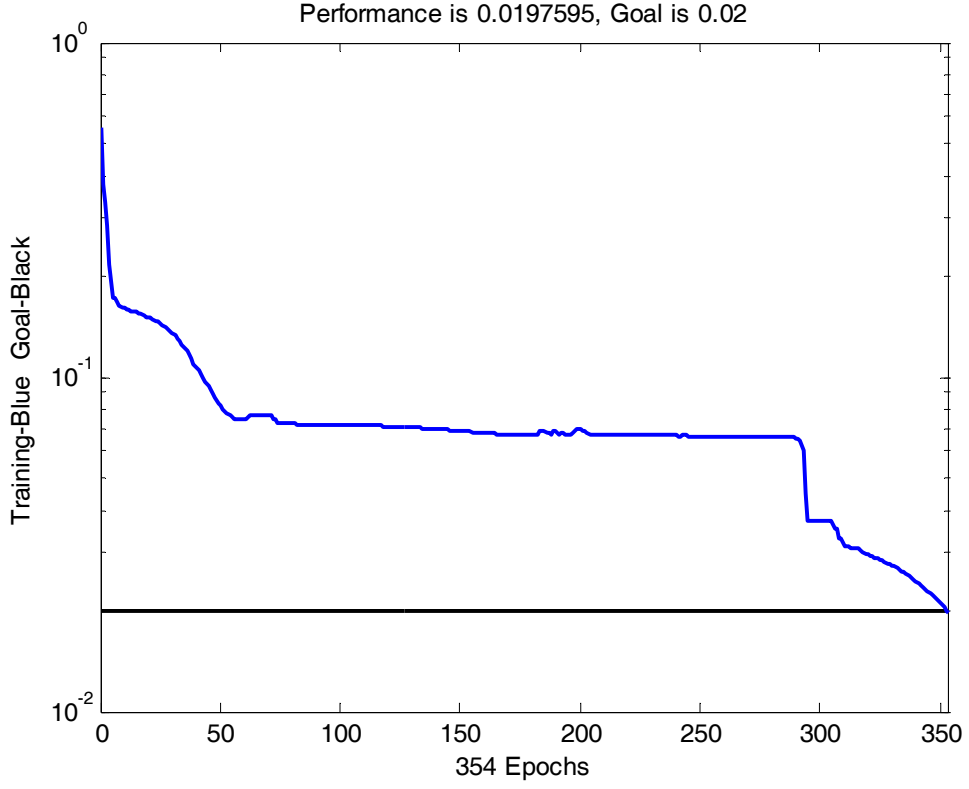
Şekil 6.34. 100 Giriş, 4 Çıkışlı, 1 Gizli Katmana Sahip YSA



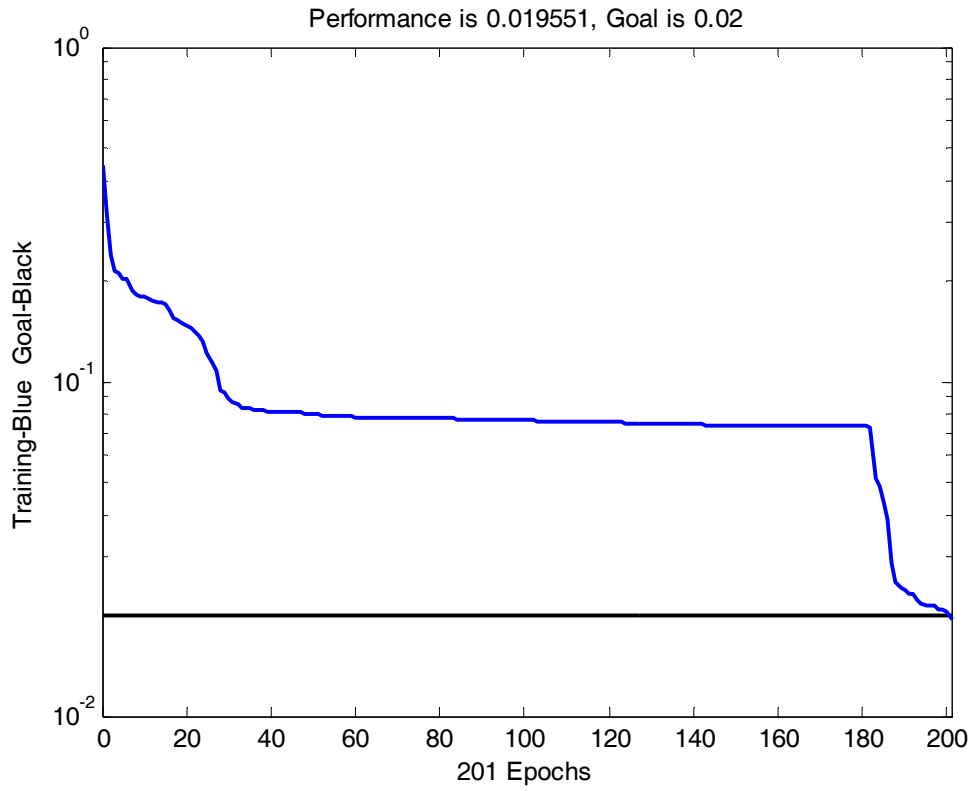
Şekil 6.35. Scg Eğitim Fonksiyonu ile Öğrenme Eğrisi



Şekil 6.36. Lm Eğitim Fonksiyonu ile Öğrenme Eğrisi



Şekil 6.37. Gdx Eğitim Fonksiyonu ile Öğrenme Eğrisi



Şekil 6.38. Bfg Eğitim Fonksiyonu ile Öğrenme Eğrisi

6.3.2. Yapay Sinir Ağları ile Sınıflandırma

YSA eğitildikten sonra girişlere sırasıyla Normal, Pace, Rbbb, Lbbb verileri girilir. Bu datalar ağ tarafından sınıflandırılır. Test dataları Sınıflandırma sonucu 4X250'lik bir matris elde edilir. Bu matrisin 1. satırı eğitim datalarından farklıdır. Normal sonuçları, 2. satırı Pace hastalığı sonuçlarını, 3. satırı Rbbb hastalığı sonuçlarını, 4. satırı Lbbb hastalığı sonuçlarını verir. Bu sonuçlar YSA'nın giriş ve çıkışlar arasında oluşturduğu ağırlık vektörleridir. Bu değerler datalar ile ilgili bilgileri temsil eder. Test işleminde 0.4 sayısı eşik değeri olarak seçilmiştir. 4X250'lik matrisin her satırı farklı data grubunu temsil etmektedir. Girişlere göre ilgili satırdaki 0.4'den büyük değerlerin sayısı başarımlar tablosunu oluşturmaktadır. Aşağıdaki tablolarda 250 sütunun sadece 10'u sınıflandırma sonucuna örnek gösterilmiştir. En son sütundaki değerler 250 sütun üzerinden hesaplanan 0.4'den büyük değerlerin sayısıdır.

Tablolarda sınıflandırma sonuçlarının toplamına bakıldığında 250 veriden az ya da çok olduğu görülmektedir.

Az olmasının sebebi sınıflandırmada kullanılan 0.4 eşik değeridir. Yani ilgili sınıfa ait olup ta 0.4'ten küçük sayılar olabilir. Sınıflandırmada 0.4'ten büyük değerler o sınıfa ait kabul edildiğinden bazı değerler sınıflandırma dışı kalmaktadır.

Fazla olmasının nedeni ise datalarda diğer sınıflara benzer değerlerin olması ve birden fazla sınıfa dahil edilmesidir.

Tablo 6.1. Normal Data Girilerek Elde Edilen Sınıflandırma Sonuçları

1	0,86614	1	0,96867	0,020152	0,7947	6,67E-07	0,99998	0,42823	204
8,48E-32	5,62E-12	1,46E-16	1,64E-13	1,92E-20	3,13E-21	4,74E-14	8,97E-25	5,37E-25	5
0,26991	1,37E-18	3,22E-07	6,03E-16	9,60E-05	2,52E-09	0,054795	4,70E-13	0,000211	2
4,54E-12	0,078178	1,36E-14	0,07498	0,005711	0,002464	1,32E-19	0,003123	0,37485	18

Tablo 6.2. Pace Data Girilerek Elde Edilen Sınıflandırma Sonuçları

0,07135	2,66E-09	2,24E-15	3,23E-10	1,96E-07	6,92E-10	1,02E-19	9,41E-07	4,29E-07	11
1	0,92018	0,99839	0,86199	0,9913	0,99984	1	1	1	173
1,12E-18	1,25E-08	8,61E-12	3,93E-18	6,19E-19	5,23E-22	7,26E-06	1,64E-13	2,12E-26	25
8,76E-26	5,11E-28	1,19E-09	0,005497	4,57E-27	5,31E-06	1,27E-37	4,31E-21	2,00E-32	20

Tablo 6.3. Rbbb Data Girilerek Elde Edilen Sınıflandırma Sonuçları

1	1,72E-18	2,42E-38	2,07E-12	2,18E-21	0,34094	7,18E-07	2,03E-07	4,96E-09	57
1,73E-30	6,76E-21	1,72E-15	6,88E-16	2,71E-08	9,31E-07	2,72E-06	5,70E-11	1,50E-21	18
0,66999	1	1	0,99503	0,89165	6,38E-07	5,07E-05	0,96471	0,93453	181
5,09E-12	3,38E-20	2,63E-31	0,002018	4,20E-07	1,37E-28	1,06E-17	5,19E-21	0,083288	7

Tablo 6.4. Lbbb Data Girilerek Elde Edilen Sınıflandırma Sonuçları

0,026792	0,000292	1,10E-27	5,76E-20	0,048544	7,09E-13	0,016047	4,42E-18	0,044239	72
0,02321	5,50E-05	6,26E-05	5,25E-16	1,20E-16	1,32E-08	0,067922	0,002003	7,17E-11	19
0,089717	5,50E-20	1,02E-09	0,000286	3,19E-18	1,50E-13	2,21E-13	1,09E-13	8,46E-14	22
6,40E-09	0,99804	0,96161	0,94131	1	0,95576	6,83E-19	0,94603	0,99728	164

Yukarıdaki tablolarda YSA'nın çıkışlarından bir kesit görülmektedir. 250 sütunun 9'u örnek olarak gösterilmiştir. Her satırdaki değerler 0.4 eşik değeri ile karşılaştırılmış, büyük olanlar o sınıfa ait sayılmıştır. Koyu rakamlar bu sayıları göstermektedir.

Tablo 6.5. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	93	33	39	105	0.37
Pace	6	200	9	20	0.80
Rbbb	36	15	179	22	0.71
Lbbb	89	35	36	100	0.40

Tablo 6.6. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	91	36	31	112	0.36
Pace	9	212	13	20	0.84
Rbbb	32	20	193	13	0.77
Lbbb	121	26	37	119	0.47

Tablo 6.7. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	117	26	35	115	0.45
Pace	8	205	19	15	0.82
Rbbb	26	9	176	19	0.70
Lbbb	81	36	23	117	0.46

Tablo 6.8. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	82	23	39	99	0.32
Pace	9	200	19	16	0.80
Rbbb	39	20	152	23	0.60
Lbbb	84	30	37	102	0.40

Tablo 6.9. Dalgacık Dönüşümü (db-2) Başarım Tablosu

	SCG	LM	GDX	BFG	Vuru Ortalaması
Normal	0.37	0.36	0.45	0.32	0.38
Pace	0.80	0.84	0.82	0.80	0.82
Rbbb	0.71	0.77	0.70	0.60	0.70
Lbbb	0.40	0.47	0.46	0.40	0.43
Ortalama Başarım	%57	%61	%61	%53	
Süre	21 sn	209 sn	40 sn	1064 sn	

Yukarıdaki tablolarda db-2 dalgacığı kullanılarak özellik çıkarımı yapılmıştır. Buna göre elde edilen sınıflandırma sonuçları, sınıflandırma yüzdeleri ve sınıflandırma süreleri verilmiştir.

Tablo 6.10. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	204	11	57	72	0.82
Pace	5	173	18	19	0.69
Rbbb	2	25	181	22	0.72
Lbbb	18	20	7	164	0.65

Tablo 6.11. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	173	26	51	68	0.69
Pace	15	163	25	26	0.65
Rbbb	5	41	149	29	0.59
Lbbb	30	36	37	134	0.53

Tablo 6.12. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	199	20	69	72	0.80
Pace	6	198	15	19	0.79
Rbbb	5	17	192	18	0.76
Lbbb	12	16	6	165	0.66

Tablo 6.13. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	164	30	30	46	0.65
Pace	18	140	32	25	0.56
Rbbb	16	24	160	25	0.64
Lbbb	23	20	14	132	0.52

Tablo 6.14. Dalgacık Dönüşümü (db-10) Başarım Tablosu

	SCG	LM	GDX	BFG	Vuru Ortalaması
Normal	0.82	0.69	0.80	0.65	0.74
Pace	0.69	0.65	0.79	0.56	0.67
Rbbb	0.72	0.59	0.76	0.64	0.68
Lbbb	0.65	0.53	0.66	0.52	0.59
Ortalama Başarım	0.72	0.62	0.75	0.59	
Süre	22 sn	689 sn	56 sn	13903 sn	

Yukarıdaki tablolarda db-10 dalgacığı kullanılarak özellik çıkarımı yapılmıştır. Buna göre elde edilen sınıflandırma sonuçları, sınıflandırma yüzdeleri ve sınıflandırma süreleri verilmiştir.

Yukarıdaki tablolarda görüldüğü gibi db-2 ve db-10 adlı iki farklı dalgacık kullanılmıştır. db-10 dalgacığı kullanıldığında sınıflandırma başarısı artmıştır.

Dört farklı eğitim fonksiyonu karşılaştırılmış, sınıflandırma ve süre bakımından trainscg eğitim fonksiyonunun daha başarılı olduğu görülmüştür.

Ayrıca sınıflandırma Fourier dönüşümü ile de yapılmış olup, sınıflandırma sonuçları aşağıdaki gibi olmuştur.

Tablo 6.15. SCG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	250	1	1	0	1
Pace	0	237	0	0	0.94
Rbbb	0	0	241	1	0.96
Lbbb	0	8	5	235	0.94

Tablo 6.16. LM Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	243	5	3	2	0.97
Pace	1	215	11	11	0.86
Rbbb	1	11	226	10	0.90
Lbbb	0	2	3	212	0.84

Tablo 6.17. GDX Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	250	0	1	0	0.97
Pace	0	233	9	0	0.86
Rbbb	0	0	247	6	0.90
Lbbb	0	8	4	243	0.84

Tablo 6.18. BFG Eğitim Fonksiyonu Kullanılarak Elde Edilen Sonuçlar

	Normal	Pace	Rbbb	Lbbb	Başarım
Normal	246	5	0	5	0.98
Pace	0	229	5	27	0.91
Rbbb	1	7	238	2	0.95
Lbbb	3	9	7	216	0.86

Tablo 6.19. FFT Başarım Tablosu

	SCG	LM	GDX	BFG	Vuru Ortalaması
Normal	1	0.97	0.97	0.98	0.98
Pace	0.94	0.86	0.86	0.91	0.89
Rbbb	0.96	0.90	0.90	0.95	0.93
Lbbb	0.94	0.84	0.84	0.86	0.87
Ortalama Başarım	0.96	0.89	0.89	0.92	
Süre	60 sn	333 sn	84 sn	1205 sn	

Yukarıdaki tablolarda Fourier Dönüşümü kullanılarak özellik çıkarımı yapılmıştır. Buna göre elde edilen sınıflandırma sonuçları, sınıflandırma yüzdeleri ve sınıflandırma süreleri verilmiştir.

Tablo 6.20. Dalgacık Dönüşümü (db-10) Sonuçları ile FFT 'nin Karşılaştırılması

	Wavelet (db-10) (trainscg)	FFT (trainscg)
Normal	0.82	1
Pace	0.69	0.94
Rbbb	0.72	0.96
Lbbb	0.65	0.94
Ortalama başarıml	0.72	0.96
Süre	22 sn	60 sn
YSA giriş sayısı	36	100

Tablo 6.20’de Dalgacık Dönüşümü ile Fourier Dönüşümü sonuçları karşılaştırılmıştır. Dalgacık Dönüşümünde en iyi sınıflandırmanın db-10 dalgacığı ve trainscg eğitim fonksiyonu kullanılarak elde edildiğı görülmüştür. Fourier Dönüşümünde ise en iyi sınıflandırma yine Scg eğitim fonksiyonunda olmuştur.

Dalgacık Dönüşümünün sınıflandırma başarısı FFT’ye göre biraz daha düşük gibi görünse de YSA giriş sayısı ve sınıflandırma süresi göz önüne alındığında 36 giriş ile 22 saniyede %72’lik bir sınıflandırma başarısı elde edilmiştir. Fourier Dönüşümünde 100 giriş ile ancak 60 saniye sonra sonuç alınabilmektedir.

7. SONUÇ VE GELECEKTE YAPILABİLECEK ÇALIŞMALAR

EKG cihazlarından alınan analog sinyal kalp ile ilgili bilgiler taşır. Bu bilgileri ortaya çıkarabilmek için analog işaret, sayısal işaret işleme teknikleri ile önce sayısal işarete dönüştürülür. Bu sayısal işarete kalp ile ilgili bilgiler vardır. Bu bilgileri sayısal işareten doğrudan almak mümkün değildir. Sayısal işaretin dönüşüm işlemine tabi tutulması gerekmektedir.

Dönüşüm işlemlerinde kullanılan değişik yöntemler vardır. Bu yöntemlerin başında Fourier Dönüşümleri gelmektedir. Fourier Dönüşümleri genellikle durağan sinyallerde kullanılır. Bu yöntem sayısal işaretin frekansı ve genliği ile ilgili bilgi verir, zamanı ile ilgili bilgi vermez. İşaretlerin içinde zamana bağlı bilgiler de olabilir. Bundan dolayı son zamanlarda Dalgacık Dönüşümü ön plana çıkmıştır.

Fourier Dönüşümünde işaret sinüs fonksiyonu ile temsil edilirken, Dalgacık Dönüşümünde ana dalgacık olarak adlandırılan dalgacık fonksiyonu ile temsil edilir. Ana dalgacık işaret üzerinde gezdirilerek işaret ile ilgili dönüşüm yapılmaktadır. Sayısal işarete uygun bir dalgacık fonksiyonu seçilirse dönüşüm işlemi daha iyi olmaktadır. Bu çalışmada db-2 ve db-10 adlı dalgacıklar kullanılmıştır. Sınıflandırma sonuçlarına göre db-10 dalgacı ile yapılan dönüşümün daha iyi olduğu görülmüştür. Bunun sebebi db-10 dalgacığının ekg sinyaline daha çok benzemesidir. Sınıflandırma yapay sinir ağları ile yapılmış, ağ eğitim fonksiyonu olarak trainscg, trainlm, traingdx ve trainbfg fonksiyonları kullanılmıştır. Sınıflandırma başarısının, trainscg fonksiyonuyla daha yüksek olduğu gözlenmiştir.

Bu çalışmada dalgacık dönüşümü ile özellik çıkarımının EKG işaretlerini sınıflandırmadaki başarısının gösterilmesi hedeflenmiştir. Bu amaçla sınıflandırma için tıbbi açıdan önemli sayılan Lbbb, Rbbb, Pace ve Normal EKG sinyalleri seçilmiştir. MIT-BIH veri tabanından alınarak işlenen işaretler, db-10 ile dalgacık dönüşümünden geçirilerek, normal EKG işaretleri %82, Pace %69 , Rbbb %72 ve Lbbb %65 başarı ile sınıflandırılmıştır. Özellik çıkarım FFT ile yapıldığında normal işaretler %100, Pace %94, Rbbb %96 ve Lbbb %94 başarı ile sınıflandırılmıştır. FFT ile dalgacık dönüşümü kıyaslandığında sınıflandırma başarısı FFT’de daha iyi

gözükmektedir. Ancak Dalgacık dönüşümünde sınıflandırmada kullanılan işaretin boyutu 36 veriden oluşmakta iken, FFT'de işaretin boyutu 100 veridir. İşaretin boyutunun fazla olması işlem yükünü arttırmakta ve FFT ile sınıflandırma süresi Dalgacık Dönüşümü'nden yaklaşık üç kat daha fazla olmaktadır.

Dönüşümden geçmiş işaretler istatistiksel sınıflandırıcılarla da sınıflandırılarak sonuçlar karşılaştırılabilir. Gelecekte gerçek zamanlı olarak alınan EKG sinyallerini sınıflandırmak için yapılacak çalışmalarda Dalgacık Dönüşüm yönteminin kullanılması önerilir.

KAYNAKÇA

[1] Fujimura, S., Kiyasu, S. 2000. "Application of Feature Extraction Scheme to the Discrimination of Electrocardiogram". Teknik Rapor, Department of Computer Science and Engineering, The University of Tokyo, Japan.

[2] Zhao, Q., Zhang, L. 2000. "Ecg Feature Extraction and Classification Using Wavelet Transform and Support Vector Machines". Teknik Rapor, Department of Computer Science and Engineering, Shanghai Jiaotong University, China.

[3] Wagner, G., Schwaibold, M. 2002. "Application of Various Feature Extraction Techniques to Automated Sleep Analysis and Online ECG Classification". Teknik Rapor, Medical Information Technology, FZI Forschungszentrum Informatik. University of Karlsruhe, Germany.

[4] Engin, M. "Feature Measurements of ECG Beats Based on Statistical Classifiers". Teknik Rapor. Department of Electronics Engineering, Ege University, İzmir.

[5] Jiang, X., Zhang, L. 2002. "ECG Arrhythmias Recognition System Based on Independent Component Analysis Feature Extraction". Teknik Rapor, Department of Computer Science and Engineering, Shanghai Jiaotong University, China.

[6] Rangayyan, R. M. 2002. "Biomedical Signal Anaysis". University of Calgary, Calgary, Alberta, Canada.

[7] Çete, Y. "Tatep, Türkiye Acil Tıp Eğitimi Kongresi". <http://www.tatep.org>. 12/07/2007

[8] Nizam, A. 2000. "EKG İşaretlerini Gerçek Zamanda Bilgisayara Aktaran ve İşleyen Arayüz Programı". Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, İstanbul.

[9] Dokur, Z. 2000. “Yapay Sinir Ağları ve Genetik Algoritmalar Kullanılarak EKG Vurularının Sınıflandırılması”. Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, İstanbul.

[10] Akıncı, T. Ç. “Dalgacık Analizinin EKG Sinyallerine Uygulanması ve Sinyal Sıkıştırma. ”Marmara Üniversitesi Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, İstanbul.

[11] Batar, H. 2005. “EEG İşaretlerinin Dalgacık Analiz Yöntemleri Kullanılarak Yapay Sinir Ağları ile Sınıflandırılması”. Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Kahramanmaraş.

[12] Mahmoodabadi, S. Z., Ahmadian, A. , Abolhasani, M. D., Eslami, M., Bidgoli, J. H. 2005. “Ecg Feature Extraction Based on Multiresolution Wavelet Transform”. Teknik Rapor, Tehran University of Medical Scences, Tehran, İran.

[13] Engin, E. Z. 2003. “Sayısal Ses İşlemenin Tıbbi Tanıda Kullanılması”. Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, İzmir.

[14] Kurt, M. B., Çerezci, O., Ari, N. “Dalgacık Dönüşümü Kullanılarak Elektromanyetik Saçılım Problemlerindeki Entegral Denklemlerin Hızlı Çözülmesi”. Teknik Rapor, Dicle Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Diyarbakır.

[15] Amet, A. L. “Doku İmgelerinde Hata Tespiti İçin Yeni Bir Yöntem:Altbant Uzayında Eş-Oluşum Matrisleri”. Boğaziçi Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, İstanbul.

[16] Alexakis, C., Nyongesa, H., Saatchi, R., Heller, S., Davies, C., Harris, N., 2003. “Feature Extraction and Classification of ECG Signals Related to Type 1

Diabetic Patients”. Teknik Rapor, University of Exeter, Networks and Computing Science, England.

[17] Avcı, E. 2006. “Doku Tipi İmgelerin Sınıflandırılması İçin Bir Uyarlamalı Entropi Tabanlı Dalgacık Yapay Sinir Ağı Sistemi”. Teknik Rapor, Fırat Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, Elazığ.

[18] Elmas, Ç. 2003. “Yapay Sinir Ağları”. Seçkin Yayıncılık.

[19] Yılmaz, A. S., Oysal, Y., Köklükaya, E. “Bir Dinamik Bulanık Ağ ile Yüksek Frekans Kontrolü Uygulaması”. Teknik Rapor, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Kahramanmaraş.

[20] Coşkun, Ö. “Wavelet Teorisinin Medikal Alana Uygulanması Üzerine Bir Ön Çalışma”. Teknik Rapor, Süleyman Demirel Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Isparta.

[21] Ağaoğlu, Erdem. “Ders Notları”. <http://agaoglu.name.tr/dsp/>. 12/07/2007

[22] Haşiloğlu, A. 1999. “Dalgacık Dönüşümü ve Yapay Sinir Ağları ile Döndürmeye Duyarsız Doku Analizi ve Sınıflandırma”. Teknik Rapor, Atatürk Üniversitesi, Mühendislik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Erzurum.

[23] Özden, M. 2005. “Ortalama Kayma Algoritmasının Geliştirilerek Görüntü Dizilerinde Hareketli Nesne Takibi ve Görüntü Kesimleme Amaçlı Kullanılması”. Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Kahramanmaraş.

[24] Mörchen, F. 2003. “Time Series Feature Extraction for Data Mining Using DWT and DFT”. Philipps University, Marburg.

[25] Dokur, Z., Ölmez, T., Yazgan, E. 1999. "Ecg Waveform Classification Using The Neural Network and Wavelet Transform". Department of Electronics Engineering, İstanbul Technical University, İstanbul.

EKLER

Ek-A: Dataların Elde Edilmesi İçin Yazılan Program Kodu

```

clear all;
for ze=1:4
M= data2(ze);
switch num2str(ze)
case {num2str(1)}
M=M*[1;0];
Ma=[ M(14264:112332); M(113028:213840); M(214188:529572);
M(530268:650000)];
M=M';
case{num2str(2)}
M=M*[1;0];
Mb=[ M(10004:269000); M(271000:428840); M(430840:557720);
M(559720:650000)]; M=M';
case{num2str(3)}
M=M*[1;0];
Mc=[ M(15306:183612); M(184308:650000)];
M=M';
case{num2str(4)}
M=M*[1;0];
Md=[ M(9:77840); M(79840:296720); M(298720:485720);
M(487720:504800); M(506800:553760); M(555760:568880);
M(570880:624680); M(626680:644480); M(646480:650000)];
M=M';
end
end

```

Ek-B: Dalgacık Dönüşümü Yapan Program(Eğitim için)

```

a=[]
normal=Ma(150000:200000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';

% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:, i), 4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
a=[a cD4]
xlswrite('Maa10.xls',a)
end

% pace datalardan buffer ile beat'lerin elde edilmesi
b=[]
normal=Mb(150000:200000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';

```

```

% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
b=[b cD4]
xlswrite('Mbb10.xls',b)
end

% rbbb datalardan buffer ile beat'lerin elde edilmesi
c=[]
normal=Mc(150000:200000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';
% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
c=[c cD4]
xlswrite('Mcc10.xls',c)
end

% lbbb datalardan buffer ile beat'lerin elde edilmesi
d=[]
normal=Md(150000:200000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';
% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
d=[d cD4]
xlswrite('Mdd10.xls',d)
end

```

Ek-C: Dalgacık Dönüşümü Yapan Program(Test için)

```

tn=[]
normal=Ma(350000:400000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';

% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
tn=[tn cD4]
xlswrite('Mtn.xls',tn);
end

```

```

% pace test dataları
tp=[]
normal=Mb(350000:400000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';

% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
tp=[tp cD4]
xlswrite('Mtp.xls',tp);
end

% rbbb test dataları
tr=[]
normal=Mc(350000:400000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';
% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
tr=[tr cD4]
xlswrite('Mtr.xls',tr)
end

% lbbb test dataları
t1=[]
normal=Md(350000:400000)
normalbuffer=buffer(normal,300,100)
cik=normalbuffer.*repmat(hamming(300)',size(normalbuffer,2),1)';
% wavelet alınması
for i=1:size(normalbuffer,2)
[C,L]=wavedec(normalbuffer(:,i),4,'db10');
cD4=detcoef(C,L,4);
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
t1=[t1 cD4]
xlswrite('Mtl.xls',t1)
end

```

Ek-D: Fourier Dönüşümü Yapan Program(Eğitim için)

```

Ppyn=[];
fn=Ma(150000:200000);
fnbuffer=buffer(fn,300,100);
for i=1:250

```

```

Yn=fft(fnbuffer(:,i));
Pfyyn=abs(Yn(1:100));
P yyn=[P yyn Pfyyn];
xlswrite('Mfaa.xls',P yyn)
end

P yyp=[];
fp=Mb(150000:200000);
fpbuffer=buffer(fp,300,100);
for i=1:250
Yp=fft(fpbuffer(:,i));
Pfyyp=abs(Yp(1:100));
P yyp=[P yyp Pfyyp];
xlswrite('Mfbb.xls',P yyp)
end

P yyr=[];
fr=Mc(150000:200000);
frbuffer=buffer(fr,300,100);
for i=1:250
Yr=fft(frbuffer(:,i));
Pfyyr=abs(Yr(1:100));
P yyr=[P yyr Pfyyr];
xlswrite('Mfcc.xls',P yyr)
end

P yyl=[];
fl=Md(150000:200000);
flbuffer=buffer(fl,300,100);
for i=1:250
Yl=fft(flbuffer(:,i));
Pfyyl=abs(Yl(1:100));
P yyl=[P yyl Pfyyl];
xlswrite('Mfdd.xls',P yyl)
end

```

Ek-E: Fourier Dönüşümü Yapan Program(Test için)

```

P yyn=[];
fn=Ma(350000:400000);
fnbuffer=buffer(fn,300,100);
for i=1:250
Yn=fft(fnbuffer(:,i));
Pfyyn=abs(Yn(1:100));
P yyn=[P yyn Pfyyn];
xlswrite('Mtfn.xls',P yyn)
end

P yyp=[];
fp=Mb(350000:400000);
fpbuffer=buffer(fp,300,100);
for i=1:250
Yp=fft(fpbuffer(:,i));
Pfyyp=abs(Yp(1:100));
P yyp=[P yyp Pfyyp];
xlswrite('Mtfp.xls',P yyp)
end

P yyr=[];
fr=Mc(350000:400000);

```



```

frbuffer=buffer(fr,300,100);
for i=1:250
Yr=fft(frbuffer(:,i));
Pfyvr=abs(Yr(1:100));
Pyyr=[Pyyr Pfyvr];
xlswrite('Mtfv.xls',Pyyr)
end

Pyy1=[];
fl=Md(350000:400000);
flbuffer=buffer(fl,300,100);
for i=1:250
Y1=fft(flbuffer(:,i));
Pfy1=abs(Y1(1:100));
Pyy1=[Pyy1 Pfy1];
xlswrite('Mtf1.xls',Pyy1)
end

```

Ek-F: Yapay Sinir Ağlarında Eğitmeyi Yapan Program(Dalgacık Dönüşümü için)

```

Maa=xlsread('Maa10.xls');%Sağlıklı
Mbb=xlsread('Mbb10.xls');%Hastalıklı pace
Mcc=xlsread('Mcc10.xls');%Hastalıklı rbbb
Mdd=xlsread('Mdd10.xls');%Hastalıklı lbbb

S1=72; %Hidden nöron sayısı
S2=4; %çıkış katmanı nöron sayısı
P=[]; T=[];
for i=1:250
a=[Maa(:,i) Mbb(:,i) Mcc(:,i) Mdd(:,i)];
P=[P a];
b=[[1 0 0 0]' [0 1 0 0]' [0 0 1 0]' [0 0 0 1]'];
T=[T b];
end
net = newff(minmax(P),[S1 S2],{'tansig' 'logsig'},'traingdx'); %neti oluştur

net.trainParam.epochs = 1000; % Deneme sayısı
net.trainParam.show = 100; % Deneme sayısının ekranda gösterilme adımı
net.trainParam.goal = 0.02; % Öğrenme hata yüzdesi
[net, tr] = train(net, P, T); % Eğitim yap

```

Ek-G: Yapay Sinir Ağlarında Eğitmeyi Yapan Program(Fourier Dönüşümü için)

```

Pyyv=xlsread('Mfaa.xls');%Sağlıklı
Pyy1=xlsread('Mfbb.xls');%Hastalıklı pace
Pyyr=xlsread('Mfcc.xls');%Hastalıklı rbbb
Pyy1=xlsread('Mfdd.xls');%Hastalıklı lbbb
S1=200; %Hidden nöron sayısı
S2=4; %çıkış katmanı nöron sayısı
P=[]; T=[];
for i=1:250
a=[Pyyv(:,i) Pyy1(:,i) Pyyr(:,i) Pyy1(:,i)];
P=[P a];
b=[[1 0 0 0]' [0 1 0 0]' [0 0 1 0]' [0 0 0 1]'];

```

```

    T=[T b];
end
net = newff(minmax(P),[S1 S2],{'tansig' 'logsig'},'trainscg'); %neti
oluştur

net.trainParam.epochs = 1000; % Deneme sayısı
net.trainParam.show = 100; % Deneme sayısının ekranda
gösterilme adımı
net.trainParam.goal = 0.02; % Öğrenme hata yüzdesi
[net, tr] = train(net, P, T); % Eğitim yap

```

Ek-H: Yapay Sinir Ağlarında Sınıflandırmayı Yapan Program(Dalgacık Dönüşümü için)

```

%sağlıklıların test başarılarının bulunması
sayn1=0
sayn2=0
sayn3=0
sayn4=0
Mn=xlsread('Mtn.xls')
basn1=sim(net,Mn); %network çıkışının bulunması
for i=1:250
    if basn1(1,i)>0.4
        sayn1=sayn1+1
        saytn1=sayn1
    end
    if basn1(2,i)>0.4
        sayn2=sayn2+1
        saytn2=sayn2
    end
    if basn1(3,i)>0.4
        sayn3=sayn3+1
        saytn3=sayn3
    end
    if basn1(4,i)>0.4
        sayn4=sayn4+1
        saytn4=sayn4
    end
end

%pace hastalığının test başarılarının bulunması
sayp1=0
sayp2=0
sayp3=0
sayp4=0
Mp=xlsread('Mtp.xls')
basp1=sim(net,Mp); %network çıkışının bulunması
for i=1:250
    if basp1(1,i)>0.4
        sayp1=sayp1+1
        saytp1=sayp1
    end
    if basp1(2,i)>0.4
        sayp2=sayp2+1
        saytp2=sayp2
    end
    if basp1(3,i)>0.4
        sayp3=sayp3+1
        saytp3=sayp3
    end
end

```

```

    if basp1(4,i)>0.4
        sayp4=sayp4+1
        saytp4=sayp4
    end
end

% rbbb hastalığının test başarılarının bulunması
sayr1=0
sayr2=0
sayr3=0
sayr4=0
Mr=xlsread('Mtr.xls')
basr1=sim(net,Mr); %network çıkışının bulunması
for i=1:250
    if basr1(1,i)>0.4
        sayr1=sayr1+1
        saytr1=sayr1
    end
    if basr1(2,i)>0.4
        sayr2=sayr2+1
        saytr2=sayr2
    end
    if basr1(3,i)>0.4
        sayr3=sayr3+1
        saytr3=sayr3
    end
    if basr1(4,i)>0.4
        sayr4=sayr4+1
        saytr4=sayr4
    end
end

% lbbb hastalığının test başarılarının bulunması
sayl1=0
sayl2=0
sayl3=0
sayl4=0
Ml=xlsread('Mtl.xls')
basl1=sim(net,Ml); %network çıkışının bulunması
for i=1:250
    if basl1(1,i)>0.4
        sayl1=sayl1+1
        saytl1=sayl1
    end
    if basl1(2,i)>0.4
        sayl2=sayl2+1
        saytl2=sayl2
    end
    if basl1(3,i)>0.4
        sayl3=sayl3+1
        saytl3=sayl3
    end
    if basl1(4,i)>0.4
        sayl4=sayl4+1
        saytl4=sayl4
    end
end
end

```

Ek-I: Yapay Sinir Ağlarında Sınıflandırmayı Yapan Program(Fourier Dönüşümü için)

```

%sağlıklıların test başarılarının bulunması
sayn1=0
sayn2=0
sayn3=0
sayn4=0
Mfn=xlsread('Mtfm.xls')
basn1=sim(net,Mfn); %network çıkışının bulunması
for i=1:250
    if basn1(1,i)>0.4
        sayn1=sayn1+1
        saytn1=sayn1
    end
    if basn1(2,i)>0.4
        sayn2=sayn2+1
        saytn2=sayn2
    end
    if basn1(3,i)>0.4
        sayn3=sayn3+1
        saytn3=sayn3
    end
    if basn1(4,i)>0.4
        sayn4=sayn4+1
        saytn4=sayn4
    end
end
end

%pace hastalığının test başarılarının bulunması
sayp1=0
sayp2=0
sayp3=0
sayp4=0
Mfp=xlsread('Mtfp.xls')
basp1=sim(net,Mfp); %network çıkışının bulunması
for i=1:250
    if basp1(1,i)>0.4
        sayp1=sayp1+1
        saytp1=sayp1
    end
    if basp1(2,i)>0.4
        sayp2=sayp2+1
        saytp2=sayp2
    end
    if basp1(3,i)>0.4
        sayp3=sayp3+1
        saytp3=sayp3
    end
    if basp1(4,i)>0.4
        sayp4=sayp4+1
        saytp4=sayp4
    end
end
end

% rbbb hastalığının test başarılarının bulunması

sayr1=0
sayr2=0
sayr3=0
sayr4=0

```

```

Mfr=xlsread('Mtfr.xls')
basr1=sim(net,Mfr); %network çıkışının bulunması
for i=1:250
    if basr1(1,i)>0.4
        sayr1=sayr1+1
        saytr1=sayr1
    end
    if basr1(2,i)>0.4
        sayr2=sayr2+1
        saytr2=sayr2
    end
    if basr1(3,i)>0.4
        sayr3=sayr3+1
        saytr3=sayr3
    end
    if basr1(4,i)>0.4
        sayr4=sayr4+1
        saytr4=sayr4
    end
end

% lbbb hastalığının test başarılarının bulunması
sayl1=0
sayl2=0
sayl3=0
sayl4=0
Mfl=xlsread('Mtfl.xls')
basl1=sim(net,Mfl); %network çıkışının bulunması
for i=1:250
    if basl1(1,i)>0.4
        sayl1=sayl1+1
        saytl1=sayl1
    end
    if basl1(2,i)>0.4
        sayl2=sayl2+1
        saytl2=sayl2
    end
    if basl1(3,i)>0.4
        sayl3=sayl3+1
        saytl3=sayl3
    end
    if basl1(4,i)>0.4
        sayl4=sayl4+1
        saytl4=sayl4
    end
end
end

```