DOKUZ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EKG SİNYALLERİNİN HASTALIK TEŞHİSİ İÇİN ANALİZİ

Fuat SEZER

Ekim, 2023
İZMİR

EKG SİNYALLERİNİN HASTALIK TEŞHİSİ İÇİN ANALİZİ

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi

İstatistik Anabilim Dalı, Veri Bilimi Yüksek Lisans Programı

Fuat SEZER

Ekim, 2023

İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

FUAT SEZER tarafından **PROF. DR. UMAY UZUNOĞLU KOÇER** yönetiminde hazırlanan **"EKG SİNYALLERİNİN HASTALIK TEŞHİSİ İÇİN ANALİZİ"** başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Umay Uzunoğlu KOÇER

Yönetici

Prof. Dr. Neslihan DEMİREL

Jüri Üyesi

Dr. Özge ELMASTAŞ GÜLTEKİN

Jüri Üyesi

Prof. Dr. Okan FISTIKOĞLU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜR

Tez çalışmamın yürütülmesinde, yardımlarını esirgemeyen ve sabırla çalışmama destek sağlayan danışmanım ve hocam sayın PROF. DR. Umay UZUNOĞLU KOÇER'e ve hayatım boyunca benim hep arkamda olan bütün aileme teşekkürü bir borç bilirim.

Fuat SEZER

EKG SİNYALLERİNİN HASTALIK TEŞHİSİ İÇİN ANALİZİ

ÖZ

Makine öğrenmesi yöntemleri veri analizi için son yıllarda birçok alanda yaygın olarak kullanılmaktadır. Sağlık alanı da bu analitik yöntemlerin başarıyla kullanıldığı alanların başında gelir. Bu tez çalışmasının amacı elektrokardiyografi sinyalini işleyerek normalden farklı sinyalleri tespit ederek kardiyologlara hastalık teşhisi için destek olmaktır. Günümüzde kalp hastalıklarından aritmi önemli ve yaygın görülen bir hastalıktır. Aritmi teşhisi için elektrokardiyografi sinyalinin uzun süre boyunca ve sürekli olarak elde edilmesi ve incelenmesi gerekir. Elektrokardiyografi sinyalleri uzun süre boyunca kaydedilmediği için uzun dönemde tespit edilebilecek kalp hastalıkları tespit edilememektedir. Bu nedenle elektrokardiyografi sinyalinden otomatik olarak aritmilerin tespit edilmesi büyük önem taşımaktadır.

Çalışmada kullanılan veri seti hastalardan elde edilen yaklaşık yarım saatlik elektrokardiyografi sinyalleri ve her atıma karşılık gelen sınıf etiketlerini (normal atım, prematüre ventriküler kasılma atımı vb.) içermektedir. Veri seti kullanılarak K-En yakın komşu algoritması ile atımları sınıflandıran model geliştirilmeye çalışılmıştır. Elektrokardiyografi sinyalinden kesikli dalgacık dönüşümü ile elde edilen yedi değişken ile kurulan K-En yakın komşu algoritması kullanılarak normal ve prematüre ventriküler kasılma atımları, yüksek doğruluk ile sınıflandırılabilmiştir. Bu model ile kardiyologlara elektrokardiyografi sinyalinde prematüre ventriküler kasılma atımı olup olmadığına dair hızlıca bir öngörü sunulabilecektir.

Anahtar kelimeler: Elektrokardiyografi, K-en yakın komşu algoritması, dalgacık dönüşümü, aritmi, prematüre ventriküler kasılma, qrs kompleksi

ANALYSIS OF ELECTROCARDIOGRAPHY SIGNALS TO DIAGNOSE DISEASE

ABSTRACT

Machine learning methods have been widely used in many fields in recent years for data analysis. The field of healthcare is one of the areas where these analytical methods are used successfully. The aim of this thesis is to support cardiologists in diagnosing diseases by processing the electrocardiography signal and detecting signals that differ from normal. Today, arrhythmia is an important and common heart disease. To diagnose arrhythmia, the electrocardiography signal must be acquired and examined continuously over a long period of time. Since electrocardiography signals are not recorded for a long time, heart diseases that can be detected in the long term cannot be detected. Therefore, it is of great importance to automatically detect arrhythmias from the electrocardiography signal.

The data set used in the study includes approximately half an hour of electrocardiography signals obtained from patients and class labels corresponding to each beat (normal beat, premature ventricular contraction beat, etc.). Using the data set, an attempt was made to develop a model that classifies the beats with the K-nearest neighbors algorithm. Normal and premature ventricular contraction beats could be classified with high accuracy by using the K-nearest neighbors algorithm, which was established with seven variables obtained from the electrocardiography signal by discrete wavelet transform. Using this model, cardiologists can be provided with a prediction about whether there is a premature ventricular contraction beat in the electrocardiography signal.

Keywords: Electrocardiography, K-nearest neighbors, wavelet transformation, arrhythmia, premature ventricular contraction, qrs complex

İÇİNDEKİLER

Sayfa

SINAV SONUÇ FORMU……..... ii
TEŞEKKÜR……...…… iii
ÖZ …….. iv
ABSTRACT……...…….. v
ŞEKİLLER LİSTESİ……..... viii
TABLOLAR LİSTESİ……...…….. x
KISALTMALAR ……... xi

BÖLÜM BİR – GİRİŞ ……....…… 1

1.1 Çalışma Alanı ……...… 1
1.2 Amaç …… 4

BÖLÜM İKİ – ÖNCEKİ ÇALIŞMALAR ……...… 6

BÖLÜM ÜÇ – YÖNTEM ……... 10

3.1 Makine Öğrenmesi …… 10
3.1.1 Denetimli/Denetimsiz Öğrenme ……..……... 10
3.1.1.1 Denetimli Öğrenme …… 10
3.1.1.2 Denetimsiz Öğrenme ...……... 12
3.1.2 Parametrik Makine Öğrenmesi Modelleri ……. 14
3.1.3 Parametrik Olmayan Makine Öğrenmesi Modelleri .....…….. 14
3.2 k-En Yakın Komşu Algoritması ……... 15
3.2.1 Regresyon Problemi İçin k En Yakın Komşu Algoritması …… 15
3.2.2 Sınıflandırma Problemi İçin k En Yakın Komşu Algoritması ……... 16
3.3 Yanlılık- Varyans Dengesi (Bias-Variance Tradeoff) …….…….. 18
3.4 En iyi k Değerinin Seçimi …….. 19

3.5 Boyut İndirgeme ……...…. 20
3.5.1 Tekil Değer Ayrıştırması ……... 21
3.5.2 Isomap …… 21
3.5.3 Bağımsız Bileşen Analizi ……... 21
3.5.4 Temel Bileşenler Analizi ……... 22
3.6 Dalgacık Dönüşümü ……... 24
3.6.1 Sürekli Dalgacık Dönüşümü …….. 30
3.6.2 Kesikli Dalgacık Dönüşümü …….. 31
3.7 Normalizasyon ……... 33
3.7.1 Minimum-Maksimum Normalizasyon Yöntemi …… 34

BÖLÜM DÖRT – VERİ ANALİZİ ……... 35

4.1 Veri Setinin İncelenmesi ……...…. 35
4.2 Uygulama ……..... 39

BÖLÜM BEŞ - SONUÇLAR ……... 58

KAYNAKLAR ……..... 59

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1 EKG sinyalinin bileşenleri ……..….. 2
Şekil 1.2 Prematüre ventriküler kasılma hastasının ekg sinyali …… 3
Şekil 3.1 Sınıflandırma problemi (spam filtreleme) ……. 11
Şekil 3.2 Regresyon problemi (saçılım grafiği) …… 11
Şekil 3.3 Müşteri haritası …….....……...……... 12
Şekil 3.4 Kümeleme (müşteri segmentasyonu) ……...…… 13
Şekil 3.5 Öklid ve Manhattan uzaklık ölçüleri …...…….. 17
Şekil 3.6 k-NN sınıflandırma algoritması (saçılım grafiği) …….……. 17
Şekil 3.7 Model performansının ile model esnekliği ilişkisini gösteren grafikler … 19
Şekil 3.8 k değeri için çapraz doğrulama grafiği ……..... 20
Şekil 3.9 Scree Diyagramı ……..... 24
Şekil 3.10 Dalgacık analizinde ölçek-zaman alanı ……... 25
Şekil 3.11 Dönüşümlerin zaman-frekans çözünürlüğü bakımından karşılaştırılması…
...…… 26
Şekil 3.12 Örnek dalgacık şekilleri ……... 29
Şekil 4.1 Elektrotların vücuda takıldığı noktalar …….. 37
Şekil 4.2 Sembol açıklama tablosu ……... 38
Şekil 4.3 MIT-BIH aritmi veri tabanı ……... 39
Şekil 4.4 Hastaların bilgileri …… 40
Şekil 4.5 İlk veri seti ……..…….……. 41
Şekil 4.6 Orjinal ve normalleştirilmiş EKG sinyali …….…….…… 42
Şekil 4.7 KDD katsayı ayrıştırma şeması ……...….. 44
Şekil 4.8 KDD sonrası veri seti ……..……...…. 45
Şekil 4.9 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (standart sapma)……... 46
Şekil 4.10 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (ortalama) ……..... 47

Şekil 4.11 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (minimum) ……..... 47
Şekil 4.12 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (maximum) ……..... 48
Şekil 4.13 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (minimum – maximum Farkı)……..... 48
Şekil 4.14 Değişken seçimi sonrası veri seti …….…… 49
Şekil 4.15 Çalışmadaki Temel Bileşen Analizinin Scree Diyagramı ……... 50
Şekil 4.16 Döndürülmüş bileşen grafiği …….…….…. 51
Şekil 4.17 Temel bileşenler ile çizilmiş saçılım grafiği …… 52
Şekil 4.18 k değeri belirleme ……...……. 54

TABLOLAR LİSTESİ

Sayfa

Tablo 4.1 TBA bileşenlerinin özdeğerler tablosu .....…… 49
Tablo 5.1 Karışıklık matrisi (confusion matrix) …….…….. 55
Tablo 5.2 Çalışmada önerilen modele ait karışıklık matrisi ……. 56

KISALTMALAR

EKG	: Elektrokardiyografi
NA+	: Sodyum
K+	: Potasyum
PVK	: Prematüre ventriküler kasılma
K-NN	: K-En Yakın Komşuluk Algoritması
KDD	: Kesikli Dalgacık Dönüşümü
TBA	: Temel Bileşenler analizi
N	: Normal
RBBB	: Sağ Dal Bloğu
APK	: Atriyal Prematüre Kasılma
BBA	: Bağımsız Bileşen Analizi
MIT	: Massachusetts Teknoloji Enstitüsü
BIH	: Beth Israil Hastanesi
SYGMM	: Sürekli Yoğunluk Gizli Markov Modelleri
YSA	: Yapay Sinir Ağları
KSA	: Konvansiyonel Sinir Ağları
UKSB	: Uzun Kısa Süreli Bellek
XAI	: Açıklanabilir Yapay Zeka
KDİ	: Kitle Durağanlık Endeksi
YMA	: Yorumlanabilir Model Agnostik Açıklamalar
SKA	: Shapley Katkı Açıklamaları
ÇKA	: Çok Katmanlı Algılayıcı
RA	: Rassal Ormanlar
LR	: Lineer Regresyon
SRT	: İki Boyutlu Derin Öğrenme Sınıflandırma Ağı
R-CNN	: Bölge Tabanlı Evrişimsel Ağlar
DBSCAN	: Yoğunluk Tabanlı Kümeleme Analizi
TDA	: Tekil Değer Ayrıştırması
KZFD	: Kısa zamanlı Fourier Dönüşümü
SDD	: Sürekli Dalgacık Dönüşümü

DDD	: Durağan Dalgacık Dönüşümü
DPD	: Dalgacık Paket Dönüşümü
IRIG	: Ara Menzil Enstrümantasyon Grubu
CD-ROM	: Küçük Disk Salt Okunur Bellek
TN	: Doğru Negatif
FN	: Yanlış Negatif
FP	: Yanlış Pozitif
TP	: Doğru Pozitif

BÖLÜM BİR
GİRİŞ

1.1 Çalışma Alanı

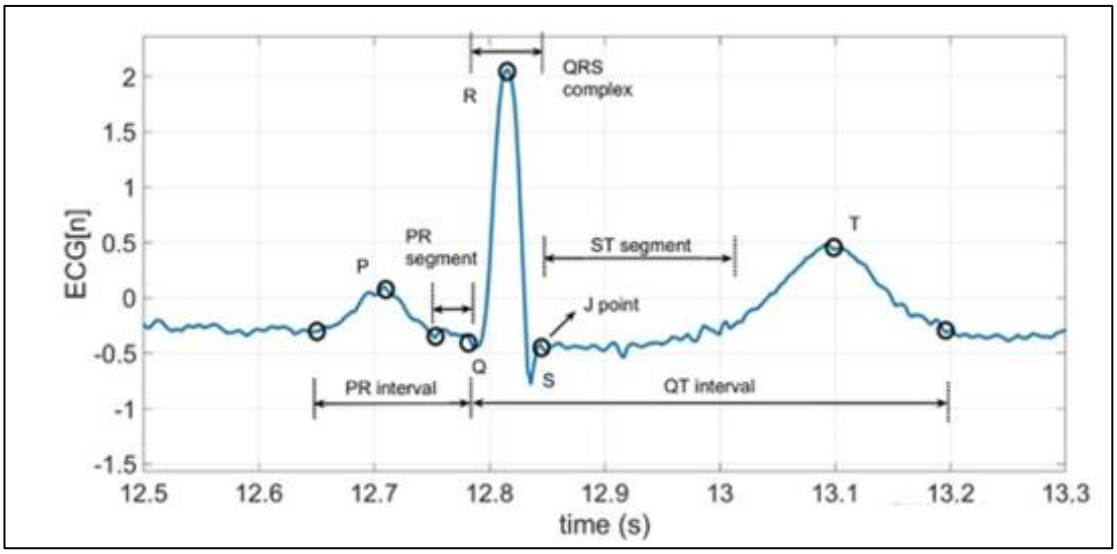
Kalp hastalıkları, dünya genelinde önemli bir halk sağlığı sorunudur ve her yıl milyonlarca insanı etkilemektedir. Bu hastalıkların teşhisinde ve tedavisinde erken tanı büyük bir önem taşımaktadır.

Kalp veya kalp kası, organlara ve özellikle akciğerlere kan pompalama, gaz alışverişini kolaylaştırma, oksijeni emme ve karbondioksiti ortadan kaldırma fonksiyonlarına sahiptir. Atriyal ve ventrikül kasları belirli üretim, iletim ve kasılma işlevleri sunan, kendini tekrar eden elektriksel aktiviteler gerçekleştirebilen kontraktil liflerden oluşur. (Madeiro vd., 2019)

Kalbin elektriksel aktivitesindeki bozuklukları daha iyi anlamak için, kontraktil miyokard hücrelerinin ve kalp pili hücrelerinin elektriksel özellikleri hakkında temel bilgilere sahip olmak önemlidir. Tek bir mikro elektrot kullanımıyla, tek bir hücrenin elektriksel olayı (veya aksiyon potansiyeli) kaydedilebilir.

EKG (Elektrokardiyografi) kalbin elektriksel aktivitesini kaydeden ve bu alanda yapılan araştırma ve teşhis süreçlerinin temelini oluşturan yaygın bir tanı aracıdır. EKG, kalp ritmi ve elektriksel iletimindeki herhangi bir anormalliği tespit etmek ve aritmiler gibi kalp hastalıklarını teşhis etmek için kullanılan güvenilir bir yöntemdir.

Sekil 1.1 de de görüldüğü gibi EKG sinyallerinin bileşenleri P dalgası, QRS Kompleksi, T dalgası ve U dalgası olarak adlandırılan bir dizi dalga ve kompleksten oluşur. EKG sinyalinde bu bileşenler yazılan sıra ile oluşur ve sürekli tekrar eder. (Madeiro vd., 2019)



Şekil 1.1 EKG sinyalinin bileşenleri

Depolarizasyon ve repolarizasyon, sinir ve kas hücrelerindeki elektriksel aktivitenin iki önemli aşamasını ifade eden terimlerdir. Bu süreçler, sinir iletimi ve kasılma gibi temel fizyolojik olayların gerçekleşmesinde önemli rol oynar.

Depolarizasyon, hücre zarındaki elektriksel potansiyelin (membran potansiyeli) geçici olarak değiştiği bir süreçtir. Normal koşullarda, hücre zarı polarize durumdadır, yani iç kısmı negatif yüklü, dış kısmı ise pozitif yüklüdür. Ancak bir uyaran veya stimulus hücre zarına ulaştığında, bu dengeli durum geçici olarak bozulur. Depolarizasyon sürecinde, hücre zarında yer alan iyon kanalları açılır ve genellikle sodyum (Na+) iyonları hücreye doğru hareket eder. Sodyum iyonlarının hücre içine girmesiyle, iç kısım daha pozitif bir hale gelir ve elektriksel potansiyel geçici olarak pozitif yönde değişir. Bu süreç, hücre zarında uyarının iletilmesi için gerekli olan elektriksel sinyalin oluşumunu sağlar.

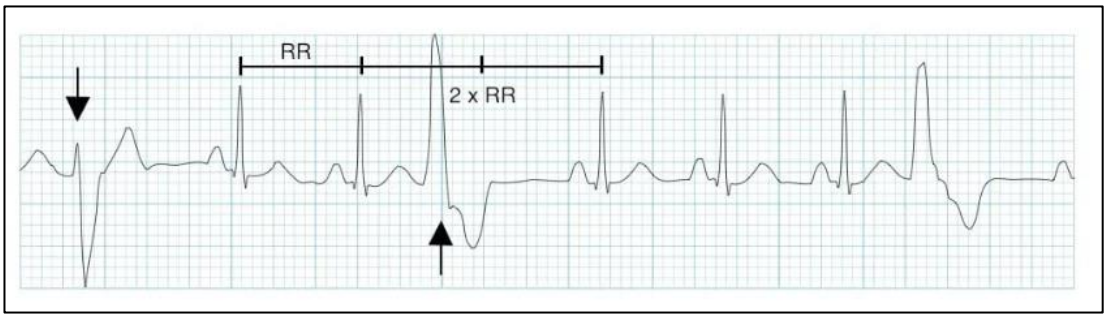
Repolarizasyon, depolarizasyon sürecinden sonra gerçekleşen bir aşamadır. Depolarizasyon sırasında hücre zarında açılan sodyum kanalları kapanırken, potasyum (K+) kanalları açılır. Bu sayede, hücre zarından sodyum çıkışı ve potasyum girişi gerçekleşir. Potasyum iyonlarının hücre dışına çıkmasıyla birlikte,

hücre içindeki pozitif yük miktarı azalır ve hücre zarı tekrar negatif hale döner. Bu süreç, hücrenin dinlenme durumuna geçmesini sağlar ve sinir hücrelerinde yeni bir uyarının iletilmesi için hazır hale gelmesini sağlar.

Atriyum depolarizasyonu P dalgasının oluşmasına neden olur, Ventriküler depolarizasyonu QRS kompleksinin oluşmasına neden olur, Ventriküler repolizasyon T dalgasının oluşmasına neden olur, U dalgasının oluşma nedeni belirsizdir ancak Purkinje sisteminin repolarizasyonundan kaynaklanıyor olabilir. (Hall, 2011)

PVK (Prematüre ventriküler kasılma), kalp kasının normal ritmine uymayan Prematüre bir kasılma olarak tanımlanan bir aritmik olaydır. EKG sinyalinde PVK'ler, belirli bir bileşen olan QRS kompleksi üzerinde belli olurlar. Normal bir QRS kompleksi genellikle dar ve simetriktir. Ancak PVK'lerin varlığında, QRS kompleksi genellikle genişletilmiş ve anormal bir şekle sahip olur. PVK'lerin tanımlanmasında kullanılan bir kural, "en az 0,12 saniyeden daha geniş olan QRS kompleksi" olarak ifade edilir (Burns ve Buttner, 2021). Yani, PVK'lerin varlığında QRS kompleksi normalden daha uzun sürebilir.

Şekil 1.2'de Prematüre ventriküler kasılması olan bir hastanın kısa dönem EKG sinyali gösterilmektedir.



Şekil 1.2 Prematüre ventriküler kasılma hastasının ekg sinyali

1.2 Amaç

Bu yüksek lisans tezi, EKG sinyallerinin analizi ve aritmi teşhisi üzerine odaklanmaktadır. Aritmiler, kalp ritminin normalden sapması durumunda ortaya çıkan önemli sağlık sorunlarıdır ve ciddi komplikasyonlara neden olabilir. Kalp atışlarının düzensiz olması, hızlanması veya yavaşlaması gibi aritmi durumları, kalp krizi, inme ve hatta ani ölüm gibi hayati risklerin artmasına yol açabilir. Bu nedenle, aritmi teşhisinin erken ve doğru bir şekilde yapılması, uygun tedavi ve yönetim stratejilerinin belirlenmesi açısından kritik öneme sahiptir.

Bu çalışmanın amacı k-En Yakın Komşuluk Algoritması (k-NN) kullanılarak kalp atımlarını normal ve Prematüre Ventriküler Kasılma olarak sınıflandırabilmektir. MIT Aritmi veri tabanındaki EKG Sinyalleri ve her atımın sınıf etiketi kullanılarak EKG sinyalindeki her bir atımın Kesikli Dalgacık Dönüşümü (KDD) yöntemi ile 5 seviyeli dalgacık katsayıları çıkartılmıştır. Ardından ise bu dalgacık katsayılarının bazı tanımlayıcı istatistikleri k-NN modeli için girdi değişkenleri olarak kullanılmıştır. Ayrıca girdi değişkenlerini görselleştirilebilmesi için girdi değişkenleri Temel Bileşenler analizi (TBA) kullanılarak 2 boyuta indirilmiştir.

Araştırmanın hedefleri şöyle sıralanabilir,

1. EKG sinyallerini depolamak ve işlemek için uygun donanım ve yazılım araçlarını belirlemek
2. Aritmilerin EKG sinyalleri üzerindeki karakteristik özelliklerini belirlemek ve analiz etmek.
3. Mevcut literatürü inceleyerek EKG sinyallerinde aritmi teşhisi için yaygın kullanılan matematiksel yöntemleri ve algoritmaları araştırmak.
4. Yapay zeka ve makine öğrenmesi tekniklerini kullanarak EKG sinyallerinde aritmi teşhisi için bir sınıflandırma yapmak.
5. Geliştirilen sınıflandırma modelini, gerçek zamanlı EKG sinyalleriyle test etmek ve doğruluk oranını değerlendirmek.

Çalışmanın bir sonraki bölümünde, literatürde EKG sinyalini analiz eden mevcut çalışmalara örnekler verilmiştir. Üçüncü bölümde, çalışmada kullanılan yöntemlere altyapı oluşturması bakımından bazı bilgilere yer verilmiştir. Dördüncü bölüm, mevcut veri seti kullanılarak anlatılan yöntemlerin uygulamasını ve elde edilen bulguları özetlemekte, son bölümde ise sonuçlar yer almaktadır.

BÖLÜM İKİ
ÖNCEKİ ÇALIŞMALAR

EKG sinyalindeki atımları sınıflandıran ve EKG Sinyalindeki bileşenleri otomatik tespit etmeye yönelik birçok çalışma yapılmıştır. Yapılan bu çalışmalardan bazıları aşağıda kısaca açıklanmaktadır.

Dalgacık dönüşümü kullanılmış çalışmalardan biri olan (Sangaiah, Arumugam, ve Bian, 2019), EKG sinyalindeki gürültüleri giderebilmek için bir filtre kombinasyonu kullanmışlardır, ardından dalgacık dönüşümünü kullanıp katsayıları elde etmişlerdir. Bu katsayıların minimum, maksimum, ortalama, standart sapma ve medyanını girdi değişkeni olarak kullanıp Gizli Markov Modeli yaratmışlardır. Bu çalışmada, MIT BIH aritmi veri tabanındaki Normal (N), Sağ Dal Bloğu (RBBB), Sol Dal Bloğu (LBBB), Prematüre Ventriküler Kasılma (PVC), Atriyal Prematüre Kasılma (APK) atımları sınıflandırmada %99,7±0,0004 sınıflandırma doğruluk oranına ulaşmışlardır. (Llamedo ve Martinez, 2011), EKG sinyallerindeki iki atımın tepe noktaları (R noktaları) arasındaki mesafe yani RR aralığını, QRS genişliğini ve dalgacık dönüşümünün katsayılarından elde edilen 8 değişkeni kullanarak genelleme kabiliyeti yüksek bir sınıflandırma modeli geliştirmişlerdir ve %93 sınıflandırma doğruluk oranına ulaşmışlardır. (Li vd., 2016), EKG atımlarını sınıflandırma için çok alanlı özellik çıkarımına dayalı bir EKG tanıma sistemi geliştirmişlerdir. EKG sinyalinde gürültüyü ortadan kaldırmak için geliştirilmiş bir dalgacık eşik yöntemi kullanılmıştır ve değişken oluşturmak için çok alanlı özellik çıkartma yöntemi, BBA (Bağımsız Bileşen Analizi) ve KDD kullanmışlardır. Sınıflandırma için Destek Vektör Makineleri algoritması kullanmışlardır ve %98,8 sınıflandırma doğruluk oranına ulaşmışlardır. (Smisek, 2016), QRS kompleksinin birçok özelliğini (R dalgasının amplitüdü, QRS kompleksinin basıklık ve çarpıklığı, QRS kompleksinin genişliği ve altında kalan alan) ve dalgacık dönüşümünün katsayılarından elde edilen birkaç değişkeni girdi değişkeni olarak kullanarak atımları Destek Vektör Makineleri ile sınıflandırmıştır. MIT BIH aritmi veri tabanında %99,17 sınıflandırma doğruluk oranına ulaşılmıştır. (Gomes vd., 2010), üç seviyeli dalgacık dönüşümünü ve Sürekli Yoğunluk Gizli Markov Modellerini (SYGMM) normal, PVK, supra-ventriküler

aritmi, atriyal fibrilasyon ve atriyal flutter atımlarını sınıflandırmak için kullanmışlardır. MIT BIH aritmi veritabanında %99,01 sınıflandırma doğruluk oranına ulaşılmıştır. (Balaskas ve Siozios, 2019), yapay sinir ağlarını (YSA) ve dalgacık dönüşümünü kullanarak atımları sınıflandıracak bir sistem kurmuştur. MIT BIH aritmi veritabanında %96,5 sınıflandırma doğruluk oranına ulaşılmıştır. (Nagaraju ve Srinivas, 2019), EKG sinyalinin bileşenlerinden oluşturduğu değişkenleri kullanarak atımlar segmente etmişlerdir. Bunu yapmak içinde kümeleme analizi kullanılmıştır.

EKG sinyallerinin sınıflandırılmasında derin öğrenme yöntemlerinin kullanıldığı bazı çalışmalar da mevcuttur. (Shobanadevi ve Veeramakali, 2023), konvansiyonel sinir ağları (KSA) ve Uzun kısa süreli bellek (UKSB) gibi derin öğrenme algoritmalarını kullanarak atımları sınıflandırmışlardır. Tek başına KSA kullanılan model normal atımları %99,06 doğruluk ile tahmin ederken, KSA ve Çift Yönlü Uzun Kısa Süreli Bellek algoritmalarını birlikte kullanıldıklarında normal atımı sınıflandırma doğruluğu 99,78'e çıkmıştır. UKSB, uzun ve kısa vadeli bağımlılıkları yakalayabilme yeteneği ile meşhur bir tür sinir ağıdır. Bir UKSB hücresi, özellikle zaman serileri veya sıralı veriler gibi yapısal bağımlılıkları olan problemlerde etkili bir şekilde çalışabilir. Çift Yönlü Uzun Kısa Süreli Bellek, geleneksel UKSB'ye ek olarak geliştirilmiş bir versiyondur. Temel fark, veri akışını tek yönlü olmaktan çıkarıp hem ileri hem de geri yönde işlem yapabilme yeteneğine sahip olmasıdır. Bu özellik, her bir girdi zaman adımı için hem önceki hem de sonraki zaman adımlarındaki bilgileri kullanarak daha kapsamlı bir bağlam yakalama imkanı sağlar. (Neves vd., 2021), atımları normal ve normal olmayan olmak üzere iki sınıfa ayırmışlardır. Model agnostik yöntemi kullanarak kalp atışı sınıflandırmasını daha açıklanabilir hale getirmek için bir Açıklanabilir Yapay Zeka (XAI) çözümüne (Kitle Durağanlık Endeksi (KDİ), Yorumlanabilir Model Agnostik Açıklamalar (YMA), Shapley Katkı Açıklamaları (SKA) odaklanılmıştır. Kullanılan XAI yöntemlerini karşılaştırmak için 1-D Jaccard indeksi kullanılmıştır. Jaccard indeksi, açıklayıcılığı değerlendirmek için bir ölçü olarak kullanılmıştır. Jaccard İndeksi ile ölçülen açıklayıcılık kalitesi SKA için KDİ ve YMA ile karşılaştırıldığında daha düşüktür sonucuna varılmıştır. (Meng vd., 2022), kalp atışının dahili morfolojisinin

değişkenlerinin ağırlığını artırmak için daha güçlü bir yerleştirme yapısı (dikkat mekanizmalı Konvolüsyonel Sinir Ağı) tasarlamışlardır. Sistemlerini PVK ve Supraventriküler Prematüre Atım (SPB) atımlarını sınıflandırmak için kullanmışlar ve MIT BIH veri tabanında %99,32 sınıflandırma doğruluk oranına ulaşmışlardır. (Yakut ve Bolat, 2022), EKG sinyalinin aritmik kalp atışlarını kategori bazlı ve hasta bazlı değerlendirme planına göre otomatik olarak sınıflandıran topluluk öğrenme tabanlı yeni bir yöntem önermişlerdir. Bu çalışmada taban çizgisi kayması sorunu için medyan filtresi uygulanmış ve sınıflandırmadan önce QRS komplekslerinin konumları tespit edilmiştir. Güç Spektral Yoğunluğuna dayalı değişken çıkarımı (extraction) yöntemi kullanılmıştır. Sınıflandırma için topluluk Öğrenmesine dayalı yeni bir yöntem uygulanmıştır. Temel öğrenenler olarak Çok Katmanlı Algılayıcı (ÇKA) ve Rassal Ormanlar (RA) ve meta öğrenici olarak Lineer Regresyon (LR) kullanılmıştır. MIT BIH aritmi veri tabanında %99,88 sınıflandırma doğruluk oranına ulaşılmıştır. (Mastoi vd., 2023), IOT cihazlarından gelen gerçek zamanlı veri işlenmiş ve gerçek zamanlı bir atım sınıflandırıcısı yaratmışlardır. EKG atımlarını sınıflandırmak için, Yinelemeli sinir ağı (YSA) en hızlı genişletilmiş versiyonu kullanılmıştır. MIT BIH aritmi veritabanında %99,97 sınıflandırma doğruluk oranına ulaşılmıştır. (Wu vd., 2023), EKG kalp atışı görüntülerinin iki boyutlu sınıflandırma modellerinin (KSA ve dönüştürücü) mevcut sorunlarının üstesinden gelmek ve model yapısını iyileştirmek için bir KSA ve dönüştürücü-kodlayıcı modülleri içeren bir iki boyutlu derin öğrenme sınıflandırma ağı SRT kullanılmıştır. MIT BIH aritmi veri tabanında %95,7 sınıflandırma doğruluk oranına ulaşılmıştır. (Rahman vd., 2022), farklı transfer öğrenme (AlexNet, SqueezeNet ve ResNet50) yöntemlerinin doğruluk oranı bakımından karşılaştırdıkları bir çalışma yapmışlardır. Transfer öğrenme, bir görevde edinilen bilginin veya deneyimin, farklı bir görevdeki performansı artırmak için kullanılmasıdır. Önerilen yöntemin uygulanması Kaggle platformundan alınan veri üzerinde AlexNet, SqueezeNet ve ResNet50 için sırasıyla %98,8, %90,08 ve %91 sınıflandırma doğruluk oranına ulaşılmıştır. (Qiu vd., 2020), eş zamanlı olarak kalp atışı bileşenlerini (QRS kompleksi, S, T dalgası) ve sınıflandırma yapabilen yeni bir derin öğrenme modeli geliştirmiştir. Ayrıca eş zamanlı segmentasyon ve sınıflandırma elde etmek için, EKG verilerini işlemek üzere özelleştirilmiş ve daha hızlı bir R-CNN tabanlı model elde edilmiştir. MIT

BIH veri tabanında %99,52 sınıflandırma doğruluk oranına ulaşılmıştır. (Kiranyaz vd., 2016), hızlı, doğru ve hastaya özel EKG sınıflandırması ve izleme sistemi yapmışlardır. Modelde 1-D evrişimli sinir ağlarının (KSA) kullanılmıştır. Değişken çıkarımını hastaya özel yapılmıştır ve her hasta için ayrı KSA modeli eğitilmiştir. MIT BIH aritmi veri tabanında %98,9 sınıflandırma doğruluk oranına ulaşılmıştır. (Yan vd., 2019), EKG sinyallerinin zaman dizisi özelliğini göz önünde bulundurarak, nispeten basit bir mimariye ve daha yüksek derecede paralelliğe sahip bir diziden diziye model olan dönüştürücüye dayalı yeni bir kalp atışı sınıflandırma yöntemi kullanılmıştır. Dönüştürücü değişken çıkarımını otomatik olarak EKG sinyallerindeki farklılıklara bakarak yapılmıştır. MIT BIH aritmi veri tabanında %99,62 sınıflandırma doğruluk oranına ulaşılmıştır.

Literatürde gizli Markov modelleri kullanılarak EKG sinyallerinin analiz edildiği çalışmalar da bulunmaktadır. (Hu vd., 2012), sınıflandırma algoritmaları ile gerçek zamanlı giyilebilir bir EKG izleme sisteminin tasarımı yapmışlardır. Ayrıca EKG sinyalindeki gürültüyü gidermek için gerçek zamanlı filtre algoritmaları da kullanılmıştır. İlk önce gerçek zamanlı tıbbi analizi gerçekleştirmek için EKG dijitalleştirilmiştir ve Bluetooth aracılığıyla bir akıllı telefona iletmiştir. Sonra akıllı telefonda, EKG dalga formu görselleştirilmiştir ve çok sayıda kardiyak aritmiyi gerçek zamanlı olarak sınıflandırmak için yeni bir Gizli Katmanlı Markov modeli sorunsuz bir şekilde entegre edilmiştir. Deneysel sonuçlar, temiz ve güvenilir EKG dalga formunun çok sayıda stresli durumda yakalanabileceğini ve kardiyak aritmi üzerinde gerçek zamanlı sınıflandırmanın diğer çalışmalara göre yeterli olduğunu gösterilmiştir.

BÖLÜM ÜÇ
YÖNTEM

Bu bölümde bu çalışmada kullanılmış yöntemler hakkında kısa bilgilere yer verilmiştir.

3.1 Makine Öğrenmesi

Makine öğrenmesi, bilgisayar sistemlerine veri tabanlı deneyimlerden öğrenme ve kendilerini geliştirme yeteneği kazandıran bir yapay zeka dalıdır. Bu yöntem, bilgisayarların belirli bir görevi veya problemi çözmek için veriye dayalı örüntüleri tanıma, sınıflandırma, tahmin etme veya karar verme yeteneği kazanmasını sağlar. (Géron, 2017)

Makine öğrenmesi algoritmaları, büyük miktarda veriye dayalı olarak çalışır. Bu veriler, genellikle insan tarafından etiketlenmiş veya kategorize edilmiş örneklerden oluşur. Makine öğrenmesi, bu veri kümesini analiz eder, istatistiksel modeller oluşturur ve bu modelleri kullanarak yeni verilere dayalı tahminler yapılır. Makine Öğrenmesi yöntemlerini aşağıdaki ölçütlere göre sınıflandırmak mümkündür:

- İnsan gözetiminde eğitilip eğitilmediği (denetimli, denetimsiz ve Takviyeli Öğrenim)
- Modelin Parametrik olup olmadığı

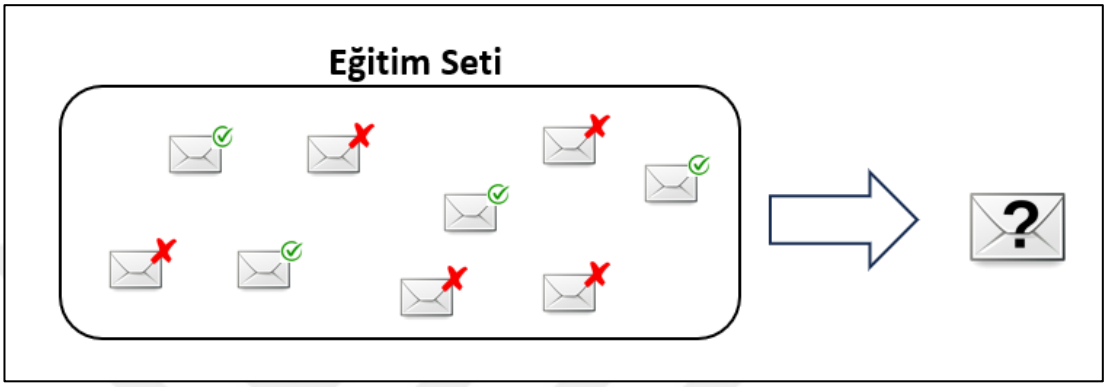
3.1.1 Denetimli/Denetimsiz Öğrenme

Makine Öğrenmesi sistemleri, eğitim sırasında aldıkları denetimin miktarına ve türüne göre sınıflandırılabilir. Dört ana kategori vardır: denetimli öğrenme, denetimsiz öğrenme ve Takviyeli Öğrenme. Burada ilk iki öğrenme yaklaşımı kısaca anlatılacaktır.

3.1.1.1 Denetimli Öğrenme

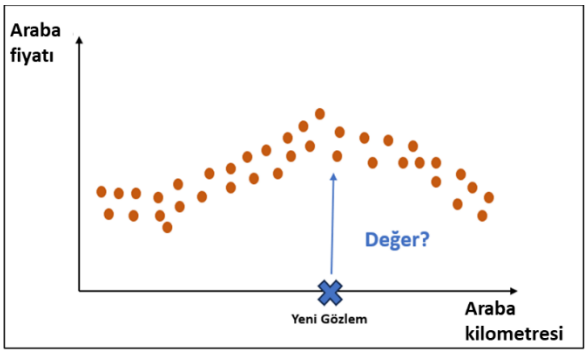
Denetimli öğrenmede, algoritmayı beslediğiniz eğitim veri seti, etiket içerir, yani çıktı değişkenine sahiptir.

Şekil 3.1'de illüstrasyonu gösterildiği gibi tipik bir denetimli öğrenme görevi sınıflandırmadır. Spam e-posta filtresi buna iyi bir örnektir: model etiketiyle (Spam veya gerçek) birlikte birçok örnek e-posta ile eğitilir ve yeni e-postaları nasıl sınıflandıracağını öğrenmesi gerekir. (Géron ,2017)



Şekil 3.1 Sınıflandırma problemi (spam filtreleme)

Bir başka denetimli öğrenme görevi, araç kilometresi, yaşı gibi girdi değişkenleri kullanılıp, arabanın fiyatı gibi bir çıktı sayısal değişkeni tahmin edilmesidir. Bu tür bir göreve regresyon denir. Modeli eğitmek için girdi değişkenleri ve çıktı değişkeni (yani araç fiyatı) olmak üzere birçok arabaya ait veri vermeniz gerekir. Şekil 3.2'de de araç fiyatı ve araç yaşı gibi iki sayısal değişken kullanılarak çizilmiş saçılım grafiği bulunmaktadır.



Şekil 3.2 Regresyon problemi (saçılım grafiği)

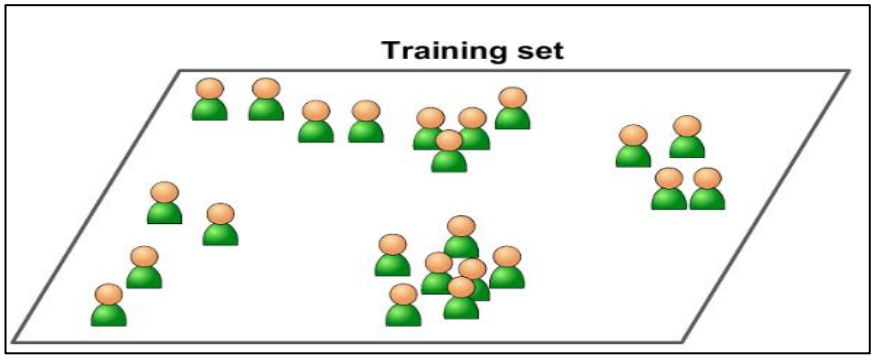
Denetimli öğrenme algoritmalarından bazıları şunlardır (Géron, 2017);

- k- En Yakın komşular Algoritması (k-NN)
- Lineer Regresyon
- Lojistik Regresyon
- Destek Vektör Makineleri
- Karar Ağacı ve Rassal Ormanlar
- Yapay Sinir Ağları (YSA)

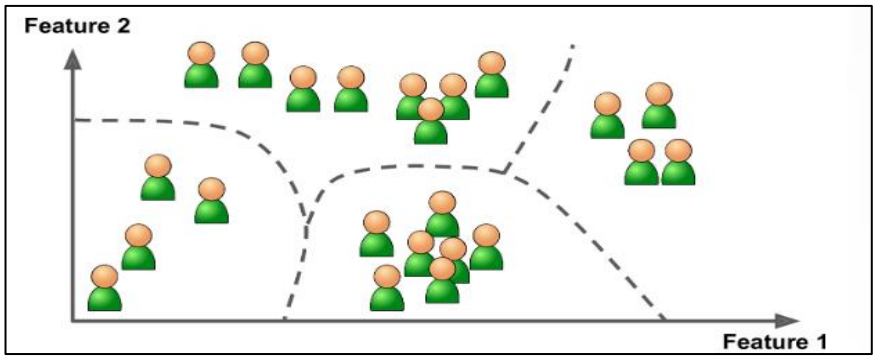
3.1.1.2 Denetimsiz Öğrenme

Denetimsiz öğrenme de eğitim verileri etiketlenmemiştir, bir başka deyişle veriler belli bir ayırıcı özellik bakımından sınıflara ayrılmamıştır. Sistem öğretmensiz öğrenmeye çalışır. Buradaki amaç gözlemleri gruplamak, görselleştirme yapmak için verinin boyutunu küçültmek veya anomalileri tespit etmek olabilir.

Şekil 3.3 ve Şekil 3.4'de bir şirketin müşterilerin önceden belirlenen iki özelliğine göre gruplanması örneklendirilmiştir.



Şekil 3.3 Müşteri haritası (Géron, 2017)



Şekil 3.4 Kümeleme (müşteri segmentasyonu) (Géron, 2017)

Bazı denetimsiz öğrenme algoritmaları;

Kümeleme Analizi için kullanılanlar:

- K-Ortalamalar (K-Means)
- Yoğunluk Tabanlı Kümeleme Analizi (DBSCAN)
- Hiyerarşik Kümeleme Analizi (Hierarchical Cluster Analysis)

Anomali Tespiti için kullanılanlar:

- Tek-Sınıf Destek Vektör Makineleri
- Yalıtım Ortamı (Isolation Forest)

Boyut indirgeme:

- Temel Bileşen Analizi (TBA)
- Bağımsız Bileşenler Analizi
- Kısmi En Küçük Kareler Regresyonu
- Ayırma Analizi

Birliktelik Kuralı için kullanılanlar:

- Apriori
- Eclat

Makine öğrenmesi yöntemlerinin bir başka sınıflandırma biçimi parametrik ve parametrik olmayan modeller olarak verilmektedir.

3.1.2 Parametrik Makine Öğrenmesi Modelleri

Parametrik makine öğrenmesi modelleri, bir veri seti üzerindeki belirli bir parametrik formülasyonu kullanarak bir model oluşturan ve bu modele dayanarak tahminler yapabilen istatistiksel modellerdir. Bu modeller, veri setinin özelliklerini ve hedef değişken arasındaki ilişkiyi modellemek için parametreleri tahmin etmek üzere kullanılmaktadır.

Parametrik makine öğrenmesi modelleri, veri setindeki desenleri yakalamak için bir dizi parametrik fonksiyon kullanılmaktadır ve bu fonksiyonlarla ilişkili parametreleri öğrenerek modele uygulanmaktadır. Bu parametreler, genellikle maksimum olabilirlik veya en küçük kareler gibi istatistiksel tahmin yöntemleri kullanılarak tahmin edilmektedir.

Bazı Parametrik Makine Öğrenmesi Modelleri;

- Lineer Regresyon
- Lojistik Regresyon

3.1.3 Parametrik Olmayan Makine Öğrenmesi Modelleri

Parametrik olmayan makine öğrenmesi modelleri, bir veri seti üzerinde belirli bir parametrik formülasyonu kullanmak yerine, veri setindeki desenleri ve ilişkileri doğrudan tahmin edebilen modellerdir. Bu modeller, veri setindeki özellikleri ve hedef değişkeni doğrudan eşleştirmek veya veri setindeki karmaşık yapıları yakalamak için esnek bir yapıya sahip olabilmektedir. Parametrik olmayan makine öğrenmesi modelleri, genellikle büyük ve karmaşık veri setlerinde veya parametrik modellerin yetersiz kaldığı durumlarda tercih edilmektedir.

- k-En Yakın Komşu Algoritması (k-NN)
- Karar Ağaçları ve Rassal Ormanlar

Parametrik modeller parametrik olmayan modellerden daha az esnek ve kompleks modellerdir fakat parametrik modeller daha yorumlanabilir modellerdir. (Géron, 2017)

Sonraki bölümde çalışmada kullanılan makine öğrenmesi modeli olan k-En Yakın Komşu Algoritmasından bahsedilmektedir.

3.2 k-En Yakın Komşu Algoritması

k-En Yakın Komşu algoritması en basit ve en iyi bilinen parametrik olmayan denetimli makine öğrenmesi algoritmalarından biridir. k en yakın komşu algoritması hem regresyon problemleri için hem de sınıflandırma problemi için kullanılabilmektedir. Çok önemli bir hiper parametresi vardır; k (komşu sayısı). Bu hiper parametre değerine göre algoritmanın performansı çok etkilenmektedir.

3.2.1 Regresyon Problemi İçin k En Yakın Komşu Algoritması

Regresyon yönteminde k-NN, k (komşu sayısı) seçildiğinde ilk önce eğitim veri setindeki karar verilecek x_0 gözlemine en yakın k adet gözlem (N_0) tanımlanmaktadır. Daha sonrasında, N_0 'daki tüm eğitim gözlemlerinin yanıt değişkenlerinin ortalaması kullanılarak $f(x_0)$ tahmin edilmektedir. Eşitlik 3.1 'deki gibi formüle edilmektedir

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i \tag{3.1}$$

Bu algoritmada karar verilmesi gereken bir diğer konu k için en uygun değerin seçilmesidir.

Genel olarak, k için en uygun değer, Yanlılık-varyans dengesine bağlı olacaktır. k için küçük bir değer düşük yanlılığa ancak yüksek varyansa sebep olacak yani modelin aşırı uyum (overfitting) problemine sebep olacaktır. Bu varyansın nedeni, belirli bir bölgedeki tahminin tamamen tek bir gözleme bağlı olmasıdır. Buna

karşılık, daha büyük k değerleri daha yumuşak ve daha az değişken bir uyum sağlamaktadır; bir bölgedeki tahmin birkaç noktanın ortalamasıdır ve bu nedenle bir gözlemi değiştirmenin model performansına daha küçük bir etkisi vardır. Bununla birlikte yumuşatma $f(X)$'teki yapının bir kısmını maskeleyerek yanlılığa neden olabilmektedir. (James vd., 2013)

k-NN regresyonunda optimum k değerini belirlemek için k değerinin birçok olası değeri ile modeller kurup bu modellerden en iyi performans sağlayan model tespit edilmektedir. Yani bir nevi hiper parametre ayarlaması (tuning) yapılmaktadır.

3.2.2 Sınıflandırma Problemi İçin k En Yakın Komşu Algoritması

Sınıflandırma yönteminde k-NN, k (komşu sayısı) seçildiğinde ilk önce eğitim veri setindeki karar verilecek x_0 gözlemine en yakın k adet gözlem (N_0) tanımlanmaktadır. Daha sonrasında, $f(x_0)$ için tahmin N_0'daki tüm eğitim gözlemlerinin yanıt değişkenlerinin maksimum geçen sınıfıdır. (James vd., 2013)

k-NN'de tanımlanmış bir eğitim veri setine gerek duyulmaz. Gözlemlerin sınıflarının bilindiği bir veri seti eğitim seti olarak alınabilir. Sınıflandırma yapılırken gözlemler, eğitim verisi olarak kullanılan veri setindeki gözlemlere olan uzaklıklarına göre sınıflandırılır. Uzaklık olarak kullanılan ölçülerden bazıları şunlardır;

- Öklid Uzaklık ölçüsü
- Manhattan uzaklık ölçüsü

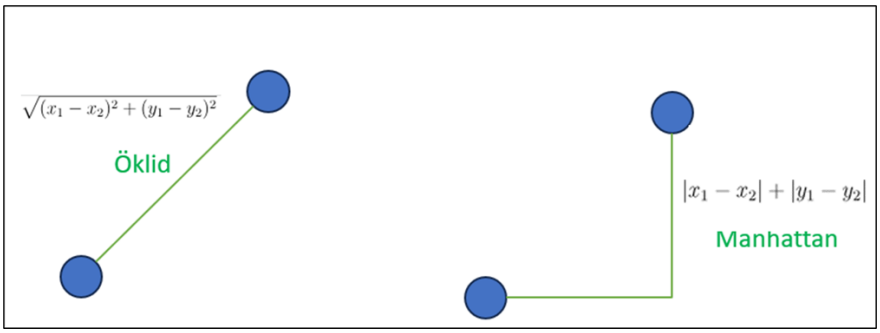
Öklid uzaklığı, sınıflandırma ve kümeleme algoritmalarında en sık kullanılan uzaklık ölçütüdür. Öklid uzaklığı, iki nokta arasındaki doğrusal uzaklık olup herhangi iki nokta, P ve Q arasındaki Öklid uzaklığı $P = (x_1, y_1)$ ve $Q = (x_2, y_2)$ olmak üzere (3.2)'de verildiği gibidir.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.2}$$

Manhattan uzaklığı ise, n boyutlu iki nokta arasındaki farkların mutlak değerlerinin toplamıdır. Herhangi iki nokta, P ve Q arasındaki Manhattan uzaklığı $P = (x_1, y_1)$ ve $Q = (x_2, y_2)$ olmak üzere, (3.3)'de verildiği gibidir.

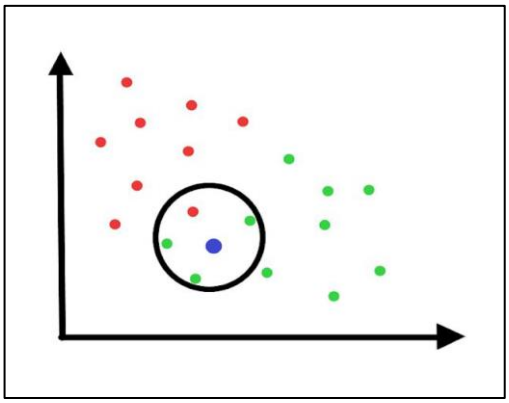
$$|x_1 - x_2| + |y_1 - y_2| \tag{3.3}$$

Şekil 3.5'te Öklit ve Manhattan uzaklıklarının geometrik düzlemde karşılıkları gösterilmektedir.



Şekil 3.5 Öklid ve Manhattan uzaklık ölçüleri

Örneğin Şekil 3.6'da verilen grafikte mavi nokta tahmin etmek istenilen nokta ve bu noktaya en yakın 4 tane noktanın 3'ü yeşil sınıfa, 1'i kırmızı sınıfa ait, yeşil sınıfa ait olanlar daha çok olduğu için tahmin de yeşil sınıf olacaktır.



Şekil 3.6 k-NN sınıflandırma algoritması (saçılım grafiği)

Bir sonraki bölümde k-NN için k komşu değerini seçerken kullanılan metodoloji olan Yanlılık-Varyans Dengesinden bahsedilmektedir

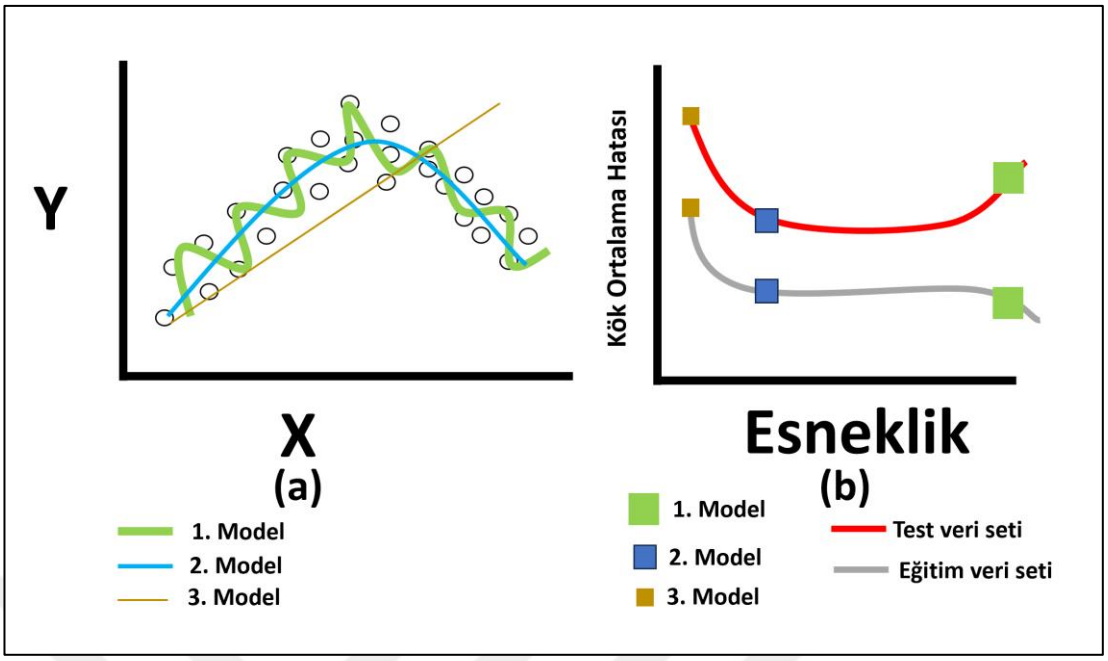
3.3 Yanlılık - Varyans Dengesi (Bias-Variance Tradeoff)

Modelleri oluştururken amaç modelin tahmin başarısını daha doğru değerlendirmektir. Bu varyans ve yanlılık (bias) arasında denge kurarak sağlanmaya çalışılır. Bu dengenin sağlandığı noktada iyi uyum (good-fit) yakalamış model kurmuş olunur.

Eğer modeli fazla optimize edersek, model eğitim setini çok iyi öğrenmiş, bir başka deyişle ezberlemiş olur. Bu k-NN için k değerini bir seçmektir. Yeni gelen gözlem için eğitim verisine en yakın gözlemin çıktı değişkeni değerini tahmin olarak kullanır bu da hem eğitim verisini aşırı öğrenmiş yani aşırı uyum (overfitting) problemine neden olur hem de varyansı yüksek bir model olur. Tam tersi k değerini aşırı yüksek seçersek yanlı bir model olur.

Şekil 3.7 (b)'de iki tane çizgi grafiği görülmektedir. Kırmızı renkli çizgi grafiği test veri setine ait Kök Ortalama Hatasını (regresyon için bir performans metriği), gri çizgi ise eğitim veri setine ait Kök Ortalama Hatasını temsil etmektedir. Kök Ortalama Hatası ne kadar düşükse modelin performansı o kadar yüksek demektir. 3. Model yanlı bir modeldir ve test veri setinde de eğitim veri setinde de performansı düşüktür. 1. Model varyansı yüksek bir modeldir ve eğitim veri setinde performansı yüksek, test veri setinde performansı düşüktür. Burada iyi model 2. modeldir.

Şekil 3.7 (a)'da modellerin regresyon eğrileri görülmektedir. Burada 1. model eğitim veri setine ait noktaları aşırı öğrenmiş ve genelleme yeteneği olmayan bir model olduğu gözükmektedir. 3. model ise eğitim veri setini hiç öğrenememiş ve verideki kalıbı yakalayamamış bir modeldir. Burada 2. model ise hem verideki kalıbı yakalamış hem de genelleme yeteneği yüksek olan iyi modeldir.



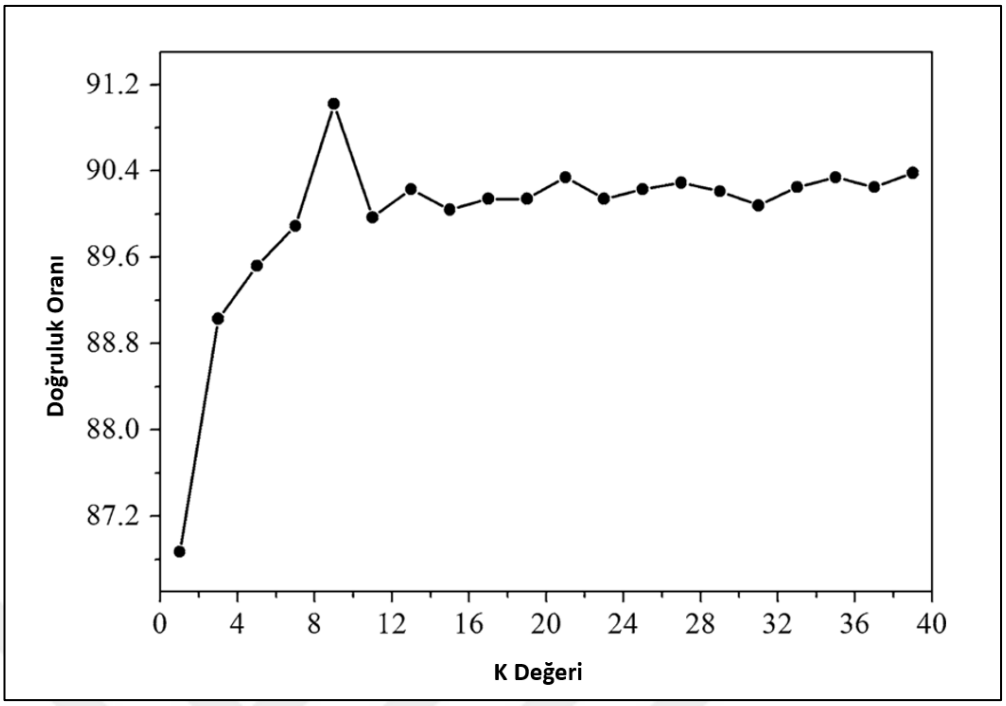
Şekil 3.7 Model performansının ile model esnekliği ilişkisini gösteren grafikler

Sonraki bölümde Yanlılık - Varyans dengesi metodolojisinin k-NN için en iyi k değerini seçmek için nasıl kullanıldığı anlatılmaktadır.

3.4 En iyi k Değerinin Seçimi

k-NN'da en büyük zorluklardan biri uygun k değerinin seçimidir. Küçük k değerleri aşırı öğrenmeye yol açabilirken, büyük k değerleri ise genelleme performansını düşürebilir. Uygun k değeri, çapraz doğrulama yapılarak seçilebilir. Yani birçok olası k değeri için modeller kurup modellerin doğruluk oranlarını kullanarak Şekil 3.8'deki grafik çizdirilebilir. Ardından doğruluk oranı k değeri artsa bile büyük bir yükseliş yaşanmadığı yerdeki k değeri seçilebilir.

Şekil 3.8'de her k değerine sahip modellerin doğruluk oranlarını gösteren bir çizgi grafiği gösterilmektedir. Burada en iyi k değeri olarak doğruluk oranı en yüksek olan 8 seçilmelidir.



Şekil 3.8 k değeri için çapraz doğrulama grafiği

Sonraki bölümde çalışmada verileri görselleştirmek için kullandığımız boyut indirgeme yöntemlerinden bahsedilecektir.

3.5 Boyut İndirgeme

Boyut indirgeme algoritmaları olarak bilinen bir algoritma ailesi, orijinal yüksek boyutlu girdi verilerini düşük boyutlu bir uzaya yansıtır, pek alakalı olmayan değişkenleri filtrelemekte ve ilginç olanların mümkün olduğunca çoğunu tutmaktadır. Bu işlem, verinin analizi, görselleştirilmesi veya makine öğrenmesi algoritmalarının verimliliği için kullanılmaktadır. Boyut azaltma, denetimli öğrenme algoritmaların kalıpları daha etkili bir şekilde tanımlamasına ve büyük ölçekli, hesaplama açısından pahalı sorunları (genellikle görüntü, video, konuşma ve metin içeren) daha verimli bir şekilde çözmesine olanak tanımaktadır.

Boyut İndirgeme algoritmalarının iki ana dalı vardır: doğrusal ve doğrusal olmayan boyut indirgeme algoritmaları. Bazı doğrusal boyut indirgeme algoritmaları şunlardır:

- Temel Bileşenler Analizi (TBA)
- Tekil Değer Ayrıştırması (TDA)

Doğrusal olmayan boyut indirgeme algoritmaları:

- Isomap
- Bağımsız Bileşen Analizi (BBA)

3.5.1 Tekil Değer Ayrıştırması

Verilerin altında yatan yapıyı öğrenmeye yönelik başka bir yaklaşım, orijinal öznitelik matrisinin sırasını daha küçük bir sıraya düşürmektir, böylece orijinal matris, daha küçük sıralı matristeki bazı vektörlerin doğrusal bir kombinasyonu kullanılarak yeniden oluşturulabilmektedir. Bu, Tekil Değer Ayrıştırması olarak bilinmektedir. Daha küçük sıralı matrisi oluşturmak için TDA, orijinal matrisin en çok bilgiye sahip vektörlerini tutar (yani, en yüksek tekil değer). Daha küçük sıralama matrisi, orijinal özellik uzayının en önemli öğelerini yakalar.

3.5.2 Isomap

Isomap, veri noktalarının gerçek uzaklıklarını korumayı hedefleyen bir yöntemdir. Veri noktaları arasındaki en kısa yol uzunluklarını tahmin etmek için grafik teorisi temellerini kullanır. Böylece, veri noktalarının gerçek uzaydaki konumlarına yakın bir şekilde temsil edilir.

3.5.3 Bağımsız Bileşen Analizi (BBA)

BBA, gözlemlenen bir veri setini, istatistiksel olarak bağımsız bileşenlerine ayırmayı hedefler. Bu yöntem, veri setindeki gizli faktörleri ve bileşenleri bulmak için genellikle karmaşık yapıları olan veri kümelerinde kullanılmaktadır. Etiketlenmemiş verilerle ilgili yaygın bir sorun, verilen değişkenlerin sahip olduğu

birçok bağımsız sinyal olmasıdır. Bağımsız bileşen analizi (BBA) kullanarak, bu harmanlanmış sinyalleri kendi bileşenlerine ayrılabilmektedir. Ayırma tamamlandıktan sonra, ürettiğimiz bireysel bileşenlerin bazı kombinasyonlarını bir araya getirerek orijinal özelliklerden herhangi biri yeniden oluşturulabilmektedir. BBA genellikle sinyal işleme görevlerinde kullanılır (örneğin, kalabalık bir ortamda kaydedilen bir videodaki insanların sesleri ayrıştırabilmek için).

3.5.4 Temel Bileşenler Analizi (TBA)

Temel bileşenler analizi denetimsiz istatistiksel öğrenme algoritmalarından biridir ve boyut indirgeme için kullanılan en popüler algoritma olarak görülebilmektedir. Birbiriyle ilişkili değişkenleri, daha az sayıda ilişkisiz değişkenler kümesine dönüştürmek üzere kullanılan matematiksel yönteme TBA denir.

Bir başka deyişle orjinal p değişkenin varyans yapısını daha az sayıda ve bu değişkenlerin doğrusal bileşenleri olan yeni değişkenlerle ifade etme yöntemidir. Aralarında korelasyon bulunan p sayıda değişkenin açıkladığı yapıyı, aralarında korelasyon bulunmayan ve sayıca orjinal değişken sayısından daha az sayıda, orjinal değişkenlerin doğrusal bileşenleri olan değişkenlerle ifade etme yöntemidir.

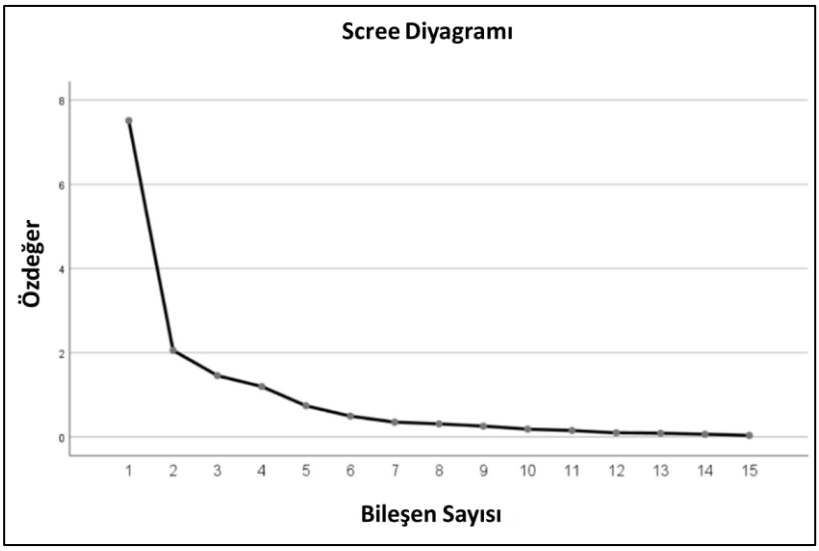
Çok değişkenli verilerde TBA görselleştirme için çok yararlıdır. Hemen hemen her türlü veri analizi için TBA ilk adım olarak önerilmektedir. Çok değişkenli verinin varsayımlarının kontrolü, aykırı gözlemlerin tanımlanması ve konumunun belirlenmesinde kullanılır. Eğer çok değişkenli veride farklı anormallikler varsa TBA bunları ortaya çıkarır. P boyutlu bir veri seti gerçekten p boyuta sahip olmalı mıdır? Eğer değilse, p adet değişken olsa bile, gerçekte verinin boyutu p'den küçüktür. Bu durumda orijinal değişkenler, herhangi bir bilgi kaybına neden olmadan daha az sayıda değişkenle yer değiştirebilir. Böylece daha az sayıda değişken sonraki analizler için kullanılabilir.

Temel Bileşenler Analizinin 5 adımı vardır,

1. Veri Normalizasyonu: Veri setindeki her bir değişkenin ortalama değeri kullanılarak verinin merkezlenmiş (ortalama değeri 0 olan) hali elde edilir.
2. Kovaryans Matrisi Hesaplama: Merkezlenmiş veri seti üzerinden kovaryans matrisi hesaplanır. Kovaryans, iki değişkenin birlikte değişimlerinin ölçüsüdür ve matris şeklinde ifade edilir, bu matrise de kovaryans matrisi denir.
3. Özdeğer ve Özvektör Hesaplama: Kovaryans matrisinin özdeğer ve özvektörlerini hesaplamak için bir özdeğer-özvektör analizi yapılır. Özdeğerler, kovaryans

   matrisinin karakteristik kökleri olup, verinin hangi yönde daha fazla varyans içerdiğini temsil eder. Özvektörler ise bu yönlere karşılık gelen yeni öznitelik eksenlerini gösterir.
4. Değişken Sıralaması: Değişkenler, büyüklüklerine göre sıralanır. Büyük özdeğerlere karşılık gelen özvektörler, verinin en fazla varyans içeren yönlerini temsil eder. Bu şekilde, bilgi içeren değişkenlerin sıralaması elde edilir.
5. Boyut Seçimi ve Yeni Değişkenlerin Oluşturulması: Boyut seçimi, TBA'nin esas adımlarından biridir. Elde edilen özdeğerlerin sıralamasına göre, kullanılacak boyut sayısı belirlenir. Örneğin, en büyük k değişken ve onlara karşılık gelen özvektörler seçilebilir. Bu seçilen özvektörler, yeni değişkenler oluşturur.

Boyut seçiminde bir yöntem, özdeğeri 1'in üzerinde olan temel bileşenleri seçmek, diğer bir yöntem ise toplam değişimin %75-80'ini açıklayacak sayıda temel bileşen seçmektir. Bir diğer yöntem optimum bileşen sayısını belirlemek üzere özdeğerlerin Scree diyagramını çizmektir. Özdeğer sayısı X ekseninde, özdeğerin değeri de Y ekseninde olacak şekilde grafik çizilmesidir. Grafikte noktalar yatay çizgi oluşturacak şekilde eğilim gösterdiklerinde özdeğerler yeterince sıfıra yaklaşmış demektir ve o özdeğerler göz ardı edilir. (James vd., 2013) Şekil 3.9'da bir scree diyagramı görülmektedir. Bu grafik her bir bileşene karşılık gelen özdeğeri göstermektedir. Bu grafikte bileşen sayısı 6 seçilebilir çünkü 6'dan sonraki bileşenlerin özdeğeri sıfıra çok yakındır ve göz ardı edilebilir.



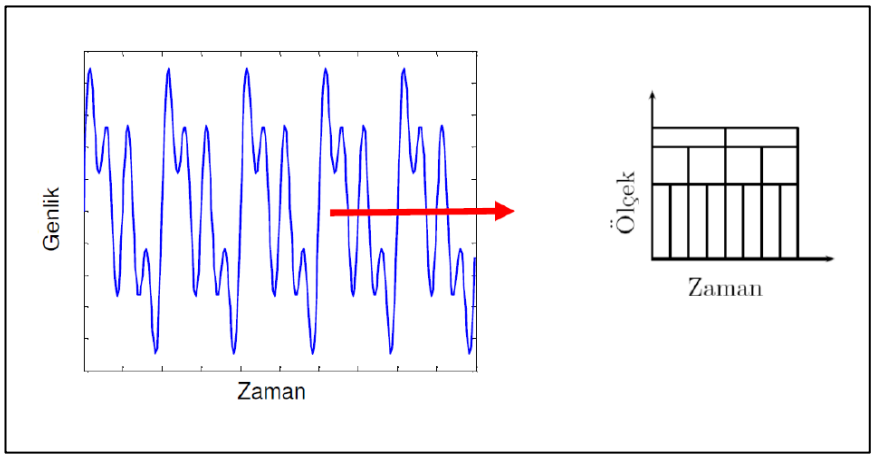
Şekil 3.9 Scree Diyagramı

3.6 Dalgacık Dönüşümü

Birçok sinyal gerçekte zamanın bir fonksiyonu olarak tanımlanabilir. Zaman tanım alanında verilmiş bir sinyal, bir matematiksel dönüşüm kullanılarak değiştirilebilir ve böyle bir değişim literatürde sinyal işleme olarak geçmektedir. Dalgacık dönüşümü, bir işaretin zaman-frekans alanı kullanılarak yapılan bir dönüşüm türüdür. Zaman tanım alanındaki bir sinyalin grafiksel gösteriminde, eksenlerden biri işaretin genliğini (bağımlı değişken), diğer eksen ise zamanı (bağımsız değişken) temsil eder ve bu gösterim zaman-genlik değişimi olarak ifade edilir. Fakat bu gösterim, çoğu sinyal işleme uygulamaları için çok da doğru bir gösterim şekli değildir. Çünkü bu uygulamalar için gerekli olan asıl bilgi sinyalin frekans bileşeninde gizlenmiş olabilir. Bu durumda sinyalin zaman-genlik gösterimi yerine frekans spektrumunun incelenmesi daha doğru olacaktır. Frekans spektrumu, sinyalin içerdiği farklı frekans bileşenlerini ve bu frekanslara ait genlik büyüklüklerini içermektedir. Bu açıklamadan, matematiksel dönüşümlerin kullanım amacının zaman tanım alanında verilmiş sinyalden çıkarılamayan bilgilere sahip olmaktır.

Sismik veriler gibi birçok verinin incelenmesi sırasında ortaya çıkan problemler "Sürekli Dalgacık Dönüşümünün" ortaya çıkmasına olanak sağlamıştır. Daha

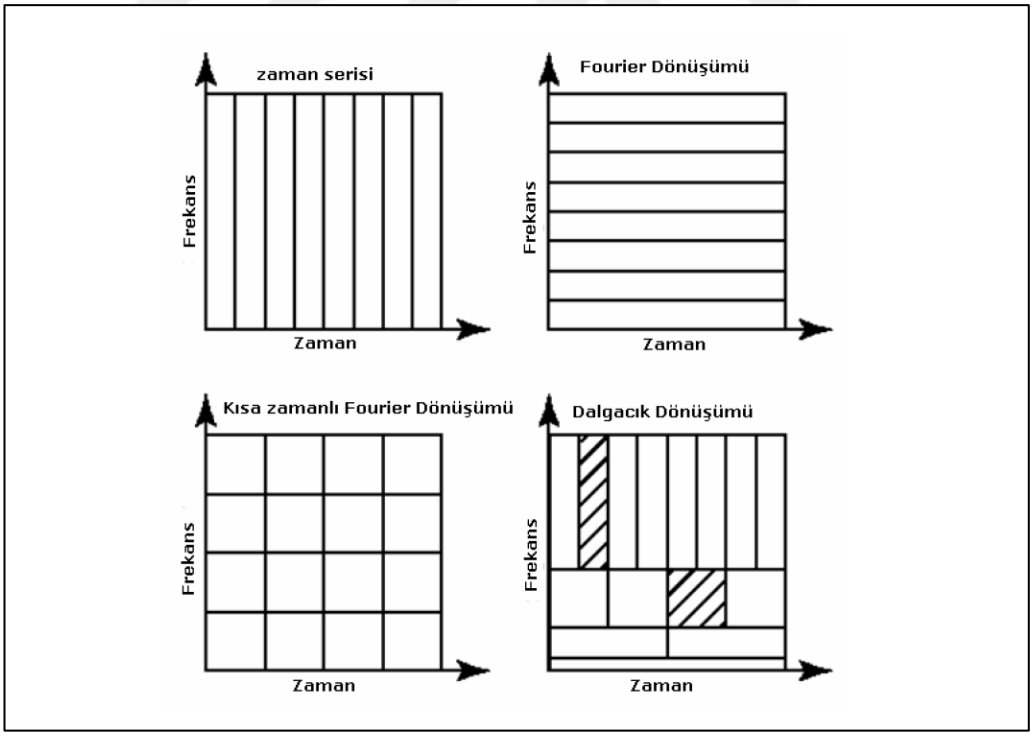
sonraları ise dalgacık analizinin EKG sinyalleri gibi birçok sinyali analizi kullanılabilir olduğu ortaya çıkmıştır. Yapılan çalışmalarda sinyallerde süreksizlik tanısında (kalp grafiklerinin yorumlanması örnek verilebilir) ve sinyalden gürültü giderme de dalgacık dönüşümünden önemli ölçüde yararlanılmıştır. Adından da anlaşıldığı gibi dalgacık dönüşümünde dalgacıklar kullanılmaktadır. Kısa Zamanlı Fourier Dönüşümüne (KZFD) benzer olarak dalgacık dönüşümünde pencere görevini ana dalgacık denilen bir fonksiyon üstlenir, fakat bu ana dalgacık, dönüşüm yönteminde hem ölçeklenir hem de ötelenir bu yüzden de Dalgacık Dönüşümünde ölçekleme ve kaydırma parametreleri kullanılır. Ölçekleme dalganın genişletilip daraltılmasına, öteleme ise zaman ekseninde dalganın kaydırılmasına neden olur. KZFD ile Dalgacık Dönüşümünün farkı, dönüşüm boyunca sabit kalan pencere fonksiyonunun genişliği dalgacık dönüşümünde sürekli değiştiğinden hem zaman hem de frekans alanının çözünürlüğü artmasıdır, bu da dalgacık dönüşümünü Fourier dönüşümünden daha üstün kılmaktadır. Dalgacıklar, kısa süreli ve yerelliği olan temel fonksiyonlardır ve işaretin farklı zaman noktalarında farklı frekans bileşenlerini temsil edebilir. Bu özellik, dalgacık dönüşümünün, zamanda anlık frekans değişikliklerini tespit etme yeteneğini sağlar. Dalgacık analizi; kısa zamanlı Fourier analizinin aksine zaman frekans alanını değil, zaman-ölçek alanını kullanır. Şekil 3.10'da dalgacık analizi sonucunda sinyalde ölçek ve zaman değişimini göstermektedir.



Şekil 3.10 Dalgacık analizinde ölçek-zaman alanı

Diğer taraftan Şekil 3.11'de Fourier, kısa zamanlı Fourier ve dalgacık dönüşümlerinin zaman-frekans domaininde çözünürlükleri gösterilmektedir. Kısaca özetlemek gerekirse; Fourier dönüşümü bize frekans bilgisini verir ama bu frekansların ne zaman oluştuğunu belirtemez. Kısa Zamanlı Fourier dönüşümünde zaman-frekans çözünürlüğü sabittir, yani KZFD'de zaman penceresi bütün frekanslarda değişmeyen büyüklüktedir. Dalgacık dönüşümünde ise bu çözünürlük ayarlanabilmektedir. Dalgacık dönüşümünün en önemli parametresi dalgacıktır. Fourier dönüşümündeki pencere fonksiyonunun görevini, dalgacık dönüşümünde ana dalgacık fonksiyonları yerine getirir. (Öner vd., 2017)

Dalgacık fonksiyonları dalgacık dönüşümü gibi işaret işleme ve veri analizi tekniklerinde kullanılan matematiksel fonksiyonlardır. Dalgacık fonksiyonları, bir işaretin farklı ölçeklerde ve zamanda yerel frekans değişimlerini temsil edebilme özelliğine sahiptir.



Şekil 3.11 Dönüşümlerin zaman-frekans çözünürlüğü bakımından karşılaştırılması

Dalgacık fonksiyonları, genellikle bir ana dalgacık (mother wavelet) fonksiyonunun ölçekleme (scaling) ve kaydırma (translation) operasyonları ile türetilir. Ölçekleme işlemi, dalgacık fonksiyonunun boyutunu değiştirirken, kaydırma işlemi ise onu zamanda hareket ettirir.

Dalgacık fonksiyonlarının genellikle belirli özelliklere sahip olması beklenir. Bu özellikler arasında şunlar yer alır:

1. Ortalama sıfır: Dalgacık fonksiyonunun toplamı sıfıra yakın olmalıdır, yani pozitif ve negatif taraflar dengelenmelidir.
2. Ölçeklenebilirlik: Dalgacık fonksiyonu, ölçekleme operatörü ile boyutunu değiştirebilme yeteneğine sahip olmalıdır. Bu, işaretin farklı ölçeklerde analiz edilebilmesini sağlar.
3. Kaydırılabilirlik: Dalgacık fonksiyonu, kaydırma operatörü ile zamanda hareket ettirilebilme yeteneğine sahip olmalıdır. Bu, işaretin farklı pozisyonlarda analiz edilebilmesini sağlar.

Dalgacık fonksiyonları, farklı matematiksel formüller veya sayısal filtreler kullanılarak elde edilebilir. Her dalgacık fonksiyonu farklı frekans, zaman ve ölçek özelliklerine sahip olabilir. Bu nedenle, dalgacık fonksiyonlarının seçimi, analiz edilmek istenen sinyalin özelliklerine uygun olmalıdır.

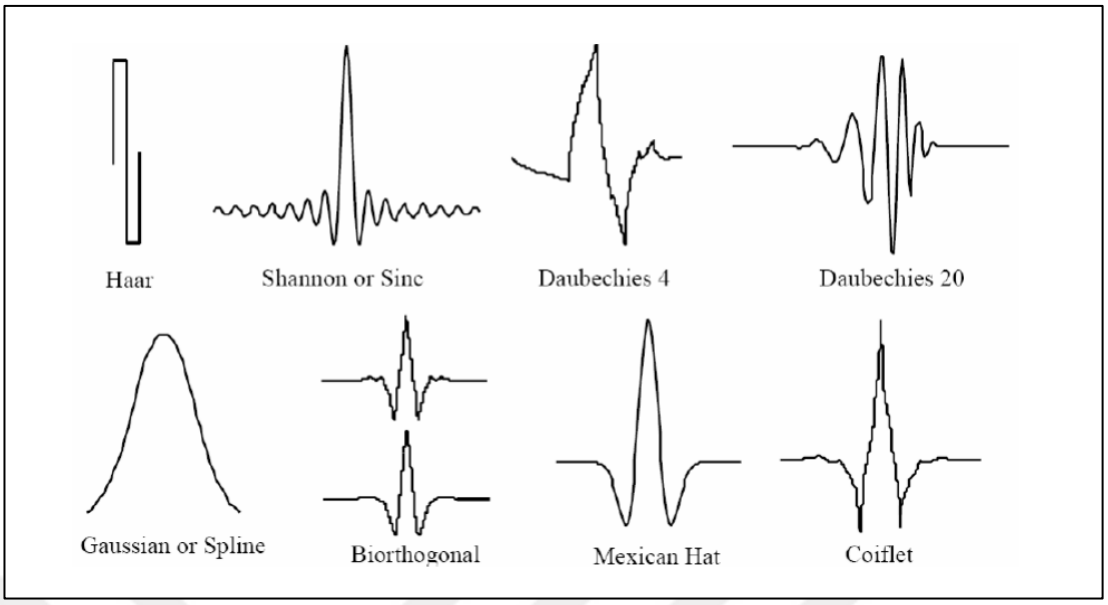
Bazı önemli dalgacık fonksiyonları:

1. Haar Dalgacığı: Haar dalgacığı, en basit ve temel dalgacık dönüşümüdür. Adım (step) fonksiyonundan türetilmiştir ve sürekli bir sinyalin belirli özelliklerini temsil eder. Dönüşüm sırasında sinyalin düzeyleri ikiye bölünür ve ikiye bölünen sinyaller arasındaki farklar analiz edilir.
2. Daubechies Dalgacıkları: Daubechies dalgacıkları, daha pürüzsüz ve simetrik bir dalgacık setidir. Daubechies dalgacıkları, belirli bir ölçekte ve düzeyde sinyalin zamanda yerel frekans değişikliklerini yakalamak için kullanılır. Daha yüksek seviyeli Daubechies dalgacıkları, daha fazla detay yakalayabilen daha uzun filtreler içerir.

3. Symlet Dalgacıkları: Symlet dalgacıkları, simetrik ve pürüzsüz olmaları açısından Daubechies dalgacıklarına benzer. Ancak, Symlet dalgacıkları, daha geniş frekans bantlarında daha iyi performans gösterir ve daha düşük frekanslarda daha fazla detayı korur.
4. Coiflet Dalgacıkları: Coiflet dalgacıkları, düşük geçiş bölgesi ve daha iyi frekans tutarlılığına sahip dalgacıklardır. Coiflet dalgacıkları, işaretlerin kenarlarını daha iyi temsil edebilen filtreler kullanır.
5. Beyaz Gürültü Dalgacıkları: Beyaz gürültü dalgacıkları, genellikle sinyalin gürültüsünü azaltmak veya gidermek için kullanılır. Beyaz gürültü dalgacıkları, beyaz gürültü spektrumunun bir benzetmesini oluşturur ve sinyalin spektral bileşenlerini analiz eder.
6. Biorthogonal Dalgacıkları: Biorthogonal dalgacıklar, kesikli dalgacık dönüşümü (DWT) için kullanılan özel bir dalgacık fonksiyonu ailesidir. DWT, bir sinyali farklı ölçek düzeylerinde ve zaman noktalarında analiz etmek için kullanılan bir dönüşüm yöntemidir. Biorthogonal dalgacık fonksiyonları, DWT işleminde kullanılan ölçeklendirme (scaling) ve dalgacık fonksiyonlarının bir çiftini ifade eder.

Bu, sadece birkaç örnek olup dalgacık dönüşümünde kullanılan dalgacık fonksiyonlarının sadece bir kısmını temsil etmektedir. Dalgacık dönüşümünde kullanılan dalgacık türleri, sinyalin veya görüntünün özelliklerine ve analiz amacına bağlı olarak değişir. Her bir dalgacık türü, farklı frekans bölgelerini, kenarları veya diğer özellikleri vurgulayabilir. Bu nedenle, doğru dalgacık türünün seçimi analiz için çok önemlidir. Çalışmada Biorthogonal dalgacıklar kullanılmıştır çünkü veri sıkıştırma ve gürültü azaltmada en etkili dalgacık fonksiyonundan biridir.

Şekil 3.12'de de birkaç dalgacık fonksiyonu gösterilmektedir.



Şekil 3.12 Örnek dalgacık şekilleri

Dalgacık dönüşümü türleri ise dönüşümde kullanılan dalgacık fonksiyonunun özelliklerine ve uygulama alanına bağlı olarak değişir. İşte yaygın olarak kullanılan dalgacık dönüşümü türlerinin bazıları:

- Sürekli Dalgacık Dönüşümü (SDD)
- Kesikli Dalgacık Dönüşümü (KDD)
- Durağan Dalgacık Dönüşümü (DDD)
- Dalgacık Paket Dönüşümü (DPD)

Durağan Dalgacık Dönüşümü, sinyalin zaman çözünürlüğünü koruyan bir dönüşüm sağlamaktadır. DDD, işaretin farklı ölçeklerde ayrıştırılmasını ve yeniden oluşturulmasını gerçekleştirirken, sürekli dalgacık dönüşümünde kaybolabilen zaman bilgisini korumaktadır. Bu nedenle, DDD, zamanla değişen işaretlerin analizi ve özellik çıkarımı için daha uygundur. Dalgacık Paket Dönüşümü, sinyali daha derin bir düzeyde ayrıştırma yeteneği sağlamaktadır. DPD, kesikli dalgacık dönüşümünün genişletilmiş bir versiyonudur ve sinyalin tüm alt bantlarını ayrı ayrı analiz edilmektedir. Bu, daha fazla detay ve frekans bilgisine sahip olmayı sağlar, ancak daha yüksek hesaplama maliyeti gerektirmektedir.

3.6.1 Sürekli Dalgacık Dönüşümü

Sürekli Dalgacık Dönüşümü, sinyalin sürekli zaman ve frekans uzayında analiz edilmesini sağlamaktadır. Bu dönüşüm, işaretin her noktasında dalgacık fonksiyonunun sürekli bir ölçek aralığında kaydırılmasıyla gerçekleştirilmektedir. SDD, zaman ve frekans çözünürlüğü arasında bir değiş-tokuş yapar ve genellikle sinyal analizi ve zaman-frekans analizi için kullanılmaktadır. Formül şu şekildedir:

$$WT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi^*(\frac{t-b}{a})dt$$

(3.4)

$x(t)$,dönüşümü yapılacak olan sinyalin zamana bağlı fonksiyonudur.
$\psi^*(t)$, dalgacık fonksiyonudur ve genellikle belirli özelliklere sahip bir ana dalgacık fonksiyonu seçilmektedir.
a ve b ise dalgacık fonksiyonunun ölçekleme ve kaydırma parametreleridir.

Dalgacık dönüşümü, ölçekleme ve kaydırma operatörleri kullanılarak gerçekleştirilir. Ölçekleme operatörü ($\bar{a}$) dalgacık fonksiyonunun boyutunu değiştirirken, kaydırma operatörü ($\bar{b}$) dalgacık fonksiyonunu işaret üzerinde hareket ettirir. İntegral işlemi, işaretin dalgacık fonksiyonu ile çarpımı ve tüm zaman boyunca entegrasyonunu temsil eder.

Dalgacık dönüşümü sonucunda elde edilen *WT(a,b)* değerleri, ölçekleme ve kaydırma parametrelerine bağlı olarak elde edilen katsayılardır. Bu katsayılar, sinyalin zaman-frekans uzayında farklı bileşenlerini temsil etmektedir. Daha sonra, bu katsayılar kullanılarak sinyalin zaman ve frekans özellikleri analiz edilebilmektedir. Dalgacık dönüşümü, genellikle ölçekleme ve kaydırma parametrelerini değiştirerek farklı ölçeklerde ve konumlarda dönüşümü gerçekleştirerek işaretin farklı bileşenlerini analiz etmek için çoklu ölçekli analizde kullanılmaktadır.

3.6.2 Kesikli Dalgacık Dönüşümü

Kesikli Dalgacık Dönüşümü (KDD), sinyalin belirli ölçeklerde ve düzeylerde analiz edilmesini sağlar. KDD, sinyalin örnekleme noktalarının ölçeklendirilmesi ve kaydırılmasıyla gerçekleştirilir. Bu, sinyalin çözünürlüğünün farklı düzeylerde ayrıştırılmasına ve yeniden oluşturulmasına olanak tanır. KDD, sinyal sıkıştırma ve gürültü azaltma gibi uygulamalarda yaygın olarak kullanılır.

Dalgacık dönüşümünün formülasyonu aşağıdaki gibi gösterilebilmektedir:

$$WT(j,k) = \sum_{n=0}^{N-1} X[n].\psi_{j,k}[n] \tag{3.5}$$

$x[n]$,dönüşümü yapılacak olan kesikli sinyalin veri noktalarıdır.
j ve k, ölçekleme ve kaydırma parametreleridir ve j ölçek numarasını, k ise pozisyon numarasını temsil eder.
$\psi_{j,k}[n]$, ölçekleme ve kaydırma parametrelerine bağlı olarak oluşturulan dalgacık fonksiyonudur.

Kesikli Dalgacık dönüşümü, öncelikle ölçekleme ve kaydırma operatörlerinin uygulanmasıyla gerçekleştirilir. Ölçekleme operatörü, sinyalin veya görüntünün boyutunu değiştirirken, kaydırma operatörü onu farklı pozisyonlarda uygular. Daha sonra, veri noktaları $x[n]$ ve dalgacık fonksiyonu $\psi_{j,k}[n]$ çarpılarak toplanır.

Kesikli Dalgacık Dönüşümü sonucunda elde edilen WT(j,k) değerleri, farklı ölçeklerde ve pozisyonlarda dönüşüm katsayılarını temsil eder. Bu katsayılar, farklı bileşenlerini zamanda ve frekansta temsil eder. Daha sonra, bu dönüşüm katsayıları kullanılarak işaretin veya görüntünün farklı özellikleri analiz edilebilmektedir.

Kesikli Dalgacık Dönüşümü, genellikle sinyalin veya görüntünün çözünürlüğünü farklı ölçeklerde analiz etmek için çok seviyeli bir şekilde uygulanır. Bu, işaretin

veya görüntünün farklı frekans bileşenlerinin ayrıştırılması ve yeniden oluşturulması için kullanılmaktadır.

Kesikli Dalgacık Dönüşümünde ölçeklendirme ve dalgacık katsayıları ise aşağıdaki formüller ile hesaplanmaktadır;

Ölçeklendirme (Scaling) Katsayıları (a_k, n):

$$cA(k,n) = \sum [x(i)\varphi(\frac{(i-2^k.n)}{2^k})]$$

(3.6)

Dalgacık (Dalgacık) Katsayıları (d_k, n):

$$cD(k,n) = \sum [x(i)\psi(\frac{(i-2^k.n)}{2^k})]$$

(3.7)

$x(i)$, sinyalin i. örnekleme noktasıdır.
φ ve ψ, ölçeklendirme ve dalgacık fonksiyonlarıdır.
k, ölçek parametresini temsil eder ve işaretin ölçeklenme düzeyini belirler.
n, kaydırma parametresini temsil eder ve işaretin zaman noktasındaki konumunu belirler.
$cA(k,n)$, A bileşenini temsil eder ve sinyalin düşük frekans bileşenini ifade eder.
$cD(k,n)$, D bileşenini temsil eder ve sinyalin yüksek frekans bileşenini ifade eder.

KDD, sinyali belirli bir ölçek ve kaydırma kombinasyonunda dalgacık fonksiyonları ile çapraz korelasyon yaparak ölçeklendirme ve dalgacık katsayılarını hesaplar. Sinyalin örnekleme noktaları, ölçeklendirme ve kaydırma operatörleriyle ölçeklenir ve kaydırılır. Ardından, sinyalin her bir örnekleme noktası, ilgili ölçek ve kaydırma parametreleri ile dalgacık fonksiyonlarıyla çapraz korelasyon yapılır ve katsayılar hesaplanmaktadır.

KDD, ölçek ve kaydırma parametreleri aracılığıyla sinyalin farklı düzeylerde ayrıştırılmasını ve yeniden oluşturulmasını sağlar. Bu sayede, işaretin frekans ve zaman özellikleri farklı bileşenler aracılığıyla temsil edilir ve analiz edilebilir.

Sonraki bölümde hastaların EKG sinyallerini standartlaştırmak için kullanılan normalizasyon yönteminden bahsedilecektir.

3.7 Normalizasyon

Normalizasyon sıklıkla kullanılan veri ölçekleme ve haritalama tekniğidir. Verilerin özelliklerini normalleştirme, önceden karar verilen aralıklardaki tüm özelliklerin değerlerini sınırlamak için yararlı bir adımdır. Özellikle uzaklık tabanlı makine öğrenmesi algoritmaları büyük değerleri veya büyük değer aralığı olan değişkenlerden daha çok etkilenme eğilimindedir. Normalizasyonda burada her bir değişkenin değer aralığını belli bir yerde sabitleyerek bu sorunun önüne geçmiş olur.

Bir diğer kullanılma nedeni veri setinin özellik çıkarımından sonra oluşturulan yeni veri setinin boyutu fazla olabilir. Veri setinde ilgisiz, fazla ve gürültülü özellikler olabilir. Bu özellikler sınıflama performansını azaltabilir, sınıflandırıcının hesaplama maliyetini artırabilir ve sonuçların kalitesini düşürebilir.

Bir diğer kullanılma nedeni ise aykırı değerlerin etkisini azaltmaktır. Aykırı değerler, veri setinde nadir görülen veya beklenen değerlerden önemli ölçüde farklı olan değerlerdir. Aykırı değerler, normalleştirme yapılmadan önce veri setinin ölçeğini ve dağılımını etkileyebilir. Normalizasyon, aykırı değerlerin etkisini azaltır veya ortadan kaldırır, böylece daha sağlıklı bir analiz yapılmasını sağlar.

Bir diğer kullanılma nedeni ise analiz hızını arttırmaktır. Normalizasyon, bazı hesaplama yöntemlerini hızlandırabilir. Örneğin, Öklid mesafesi hesaplamaları veya nokta çarpımı işlemleri normalleştirilmiş veriler üzerinde daha hızlı gerçekleştirilebilir.

3.7.1 Minimum-Maksimum Normalizasyon Yöntemi

Bu normalizasyon yöntemi genellikle mühendislik problemlerinde kullanılan en yaygın yöntem olup veri setini 0 ile 1 arasına sıkıştırır. Normalizasyon işlemi sonrasında eğer veride negatif bir değişken varsa bile artık pozitif değer alacaktır. Ayrıca minimum maksimum normalizasyon yöntemi aykırı değerlere iyi odaklanamaz. Formül aşağıdaki gibidir, (Yüce ve Özkan, 2021)

$$x' = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} \tag{3.8}$$

x', Normalize edilmiş değer

x_i, Normalize edilecek değer

x_{min}, Verideki en küçük değer

x_{max}, Verideki en büyük değer

BÖLÜM DÖRT
VERİ ANALİZİ

Bu bölümde çalışmada kullanılan veri setinden ve verinin nasıl analiz edildiğinden bahsedilecektir.

4.1 Veri Setinin İncelenmesi

Bu çalışmada MIT-BIH Aritmi Veri tabanında (MIT-BIH Arrhythmia Database) depolanan veriler kullanılmıştır.

1975'ten beri, Boston'daki Beth Israel Hastanesi'ndeki (şimdiki adı Beth Israel Deaconess Tıp Merkezi) ve MIT'deki laboratuvarlar, aritmi analizi ve ilgili konulardaki kendi araştırmalarını desteklemektedir. Bu çabanın ilk önemli ürünlerinden biri, 1980 yılında tamamlayıp dağıtmaya başlanılan MIT-BIH Aritmi Veri Tabanıdır. Veri tabanı, ritim bozukluğu dedektörlerinin değerlendirilmesi için genel olarak mevcut ilk standart test materyali setidir ve bu amaçla kullanılmaktadır. Dünya çapında 500'den fazla tesiste kardiyak dinamiklerle ilgili temel araştırmaların yanı sıra aritmi dedektörlerinin değerlendirilmesi için de kullanılabilir. Başlangıçta, veri tabanını 800 ve 1600 bpi'de 9 kanallı yarım inçlik dijital bant ve çeyrek inçlik IRIG biçimli FM analog bant üzerinde dağıtılmış, Ağustos 1989'da veri tabanının CD-ROM versiyonu çıkarılmıştır. (Moody ve Mark, 2001)

MIT-BIH Aritmi Veritabanı, 1975 ile 1979 yılları arasında BIH Aritmi Laboratuvarı tarafından incelenen 47 denekten elde edilen iki kanallı ambulatuar EKG kayıtlarından yarım saatlik 48 kayıt içerir. Bu kayıtların 23'ü 4000 tane kayıttan rastgele seçilmiştir ve kayıtların sahibi olduğu hastaların yaklaşık %60'ı yatan hastalar ve diğer %40 hasta ayakta tedavi edilen hastadır. 48'den geriye kalan 25 kayıt yine bu 4000 tane kayıttan seçilmiştir ama bu 25 nadir görülebilecek türde ve anlamlı aritmileri içeren kayıtlardır. (Moody ve Mark, 2001)

Kayıtlar, 10 mV aralığında 11 bit çözünürlükte kanal başına saniyede 360 (örnekleme hızı) örnekte sayısallaştırılır. İki veya daha fazla kardiyolog bağımsız

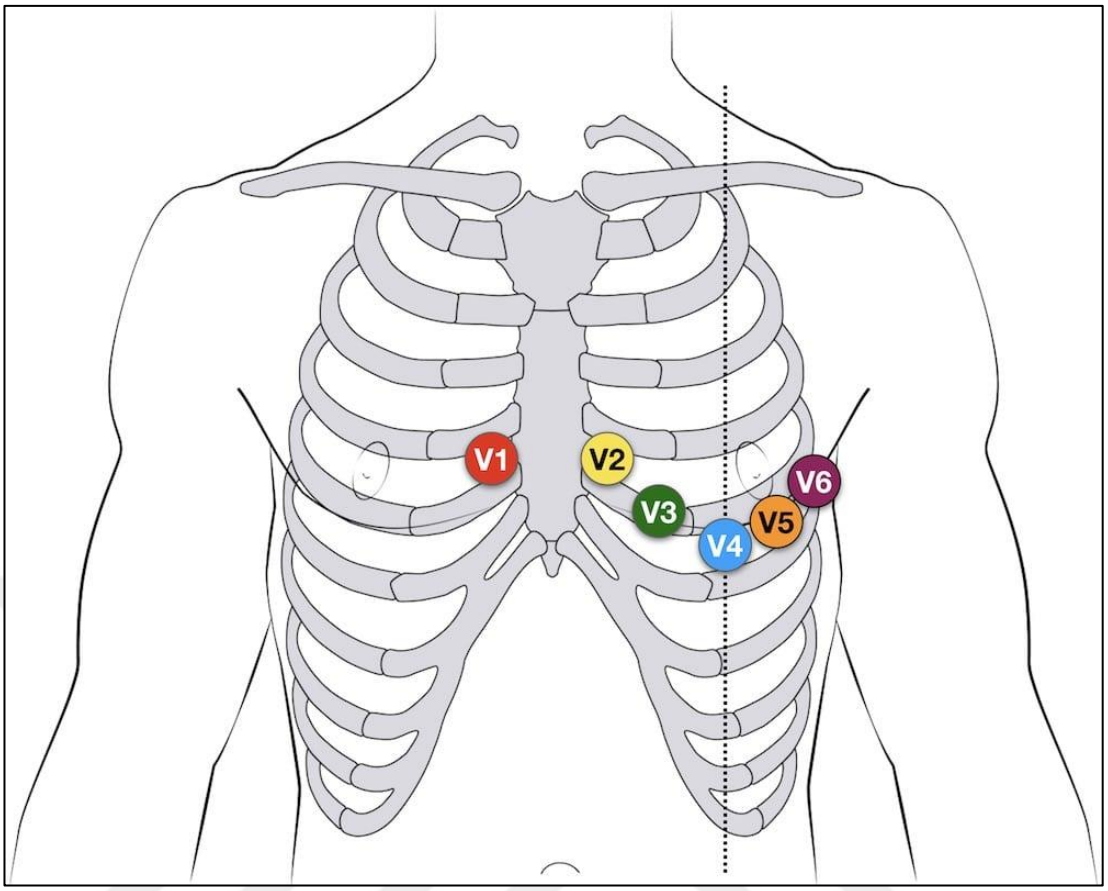
olarak her kayda açıklama eklemiştir, veri tabanına dahil edilen her vuruş için (toplamda yaklaşık 110.000 açıklama) bilgisayar tarafından okunabilir referans açıklamaları elde etmek için çözülmüştür. (Moody ve Mark, 2001)

Bu dizin, MIT-BIH Aritmi Veritabanının tamamını içerir. PhysioNet'in Eylül 1999'daki başlangıcından bu yana bu veritabanının yaklaşık yarısı (48 tam kaydın 25'i ve 48 kaydın tümü için referans ek açıklama dosyaları) burada ücretsiz olarak mevcuttur. Yalnızca MIT-BIH Aritmi'de bulunan kalan 23 sinyal dosyası Veritabanı CD-ROM'u, Şubat 2005'te bu veri tabanına kaydedilmiştir. (Moody ve Mark, 2001)

EKG, elektrot adı verilen metal tabakalar aracılığıyla vücut yüzeyine yerleştirilen elektrodlar tarafından gerçekleştirilir. Bu elektrodlar, kalp atışları sırasında oluşan elektriksel sinyalleri algılar ve bu sinyaller bir EKG makinesi tarafından kaydedilir.

Şekil 4.1'de de gösterildiği gibi EKG'yi kaydetmek için kullanılan elektrotlardan her biri, kalbin farklı bir bölgesinden elektriksel aktiviteyi ölçmek için tasarlanmıştır. EKG'nin standart 12 derivasyonu vardır ve bu derivasyonlardan altısı göğüs üzerine yerleştirilen elektrotlardan elde edilir. Bu altı derivasyon V1, V2, V3, V4, V5 ve V6 olarak adlandırılır.

V1 Derivasyonu: Sağ sternum (göğüs kemiği) kenarına yerleştirilen elektrot ile elde edilir.
V2 Derivasyonu: Sol sternum kenarına yerleştirilen elektrot ile elde edilir.
V3 Derivasyonu: V2 derivasyonunun yanına yerleştirilen elektrotla elde edilir.
V4 Derivasyonu: Sol meme çizgisi beşinci kaburga aralığına yerleştirilen elektrotla elde edilir.
V5 Derivasyonu: Solaksı kaburgaların beşinci kaburga aralığına yerleştirilen elektrotla elde edilir.
V6 Derivasyonu: Solaksı kaburgaların altıncı kaburga aralığına yerleştirilen elektrotla elde edilir.



Şekil 4.1 Elektrotların vücuda takıldığı noktalar

Her bir atım kardiyologlar tarafından aşağıdaki etiketlerle etiketlenmiş, bu etiketlerin bazıları bu çalışmada yanıt değişkeni olarak kullanılmıştır.

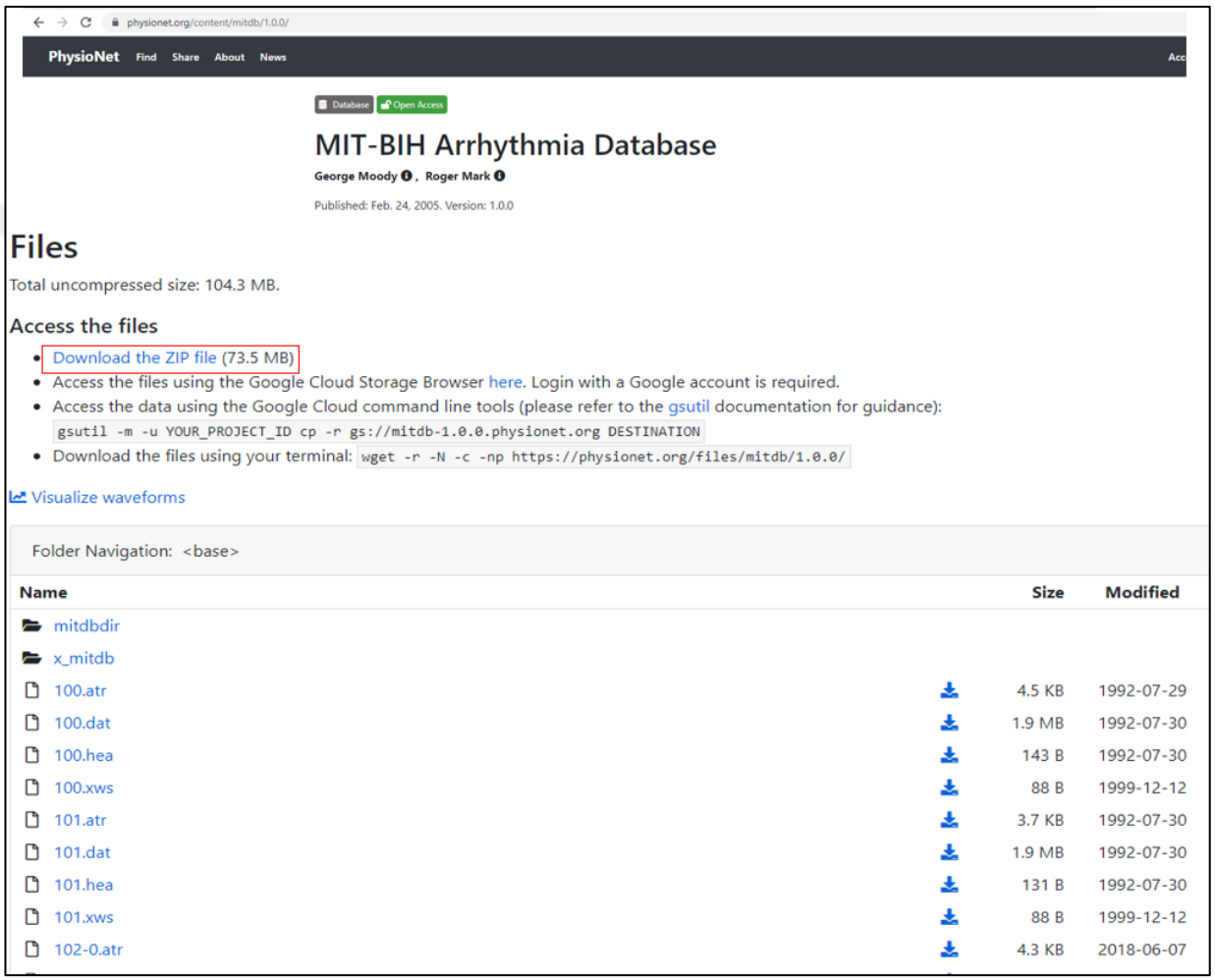
	label_store	symbol	description
0	0		Not an actual annotation
1	1	N	Normal beat
2	2	L	Left bundle branch block beat
3	3	R	Right bundle branch block beat
4	4	a	Aberrated atrial premature beat
5	5	V	Premature ventricular contraction
6	6	F	Fusion of ventricular and normal beat
7	7	J	Nodal (junctional) premature beat
8	8	A	Atrial premature contraction
9	9	S	Premature or ectopic supraventricular beat
10	10	E	Ventricular escape beat
11	11	j	Nodal (junctional) escape beat
12	12	/	Paced beat
13	13	Q	Unclassifiable beat
14	14	~	Signal quality change
16	16	|	Isolated QRS-like artifact
18	18	s	ST change
19	19	T	T-wave change
20	20	*	Systole
21	21	D	Diastole
22	22	"	Comment annotation
23	23	=	Measurement annotation
24	24	p	P-wave peak
25	25	B	Left or right bundle branch block
26	26	^	Non-conducted pacer spike
27	27	t	T-wave peak
28	28	+	Rhythm change
29	29	u	U-wave peak
30	30	?	Learning
31	31	!	Ventricular flutter wave
32	32	[	Start of ventricular flutter/fibrillation
33	33	]	End of ventricular flutter/fibrillation
34	34	e	Atrial escape beat
35	35	n	Supraventricular escape beat
36	36	@	Link to external data (aux_note contains URL)
37	37	x	Non-conducted P-wave (blocked APB)
38	38	f	Fusion of paced and normal beat
39	39	(	Waveform onset
40	40	)	Waveform end
41	41	r	R-on-T premature ventricular contraction

Şekil 4.2 Sembol açıklama tablosu

Sonraki bölümde çalışmada yapılan uygulama ve veri analizi anlatılmaktadır.

4.2 Uygulama

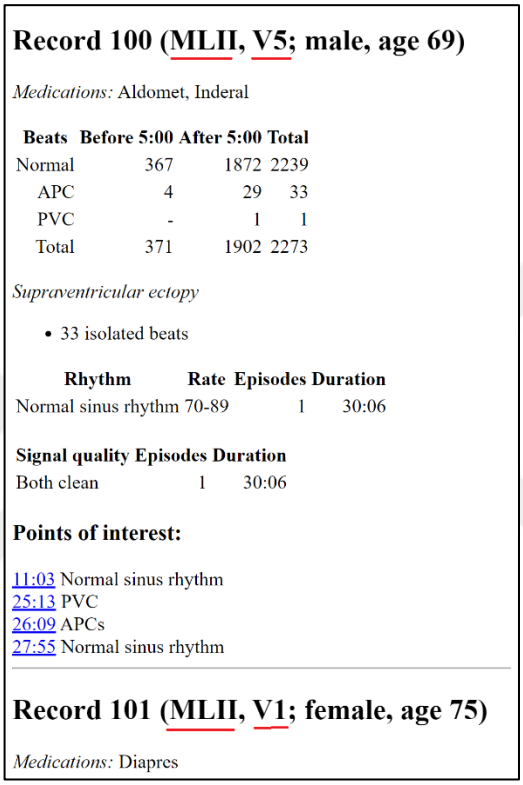
Uygulama bölümünde MIT-BIH Aritmi Veri Tabanından (https://physionet.org/content/mitdb/1.0.0/) indirilen veriler kullanılmıştır. Verilerin elde edildiği web sitesinin görseli Şekil 4.3'te verilmiştir. Verilerin analizi Python programa dili kullanılarak gerçekleştirilmiştir.



Şekil 4.3 MIT-BIH aritmi veri tabanı

Her bir hastanın verileri kendi hasta no'su ile adlandırılmıştır. Örn: (100 nolu hastanın verileri 100.atr, 100.dat, 100.hea ve 100.xws). Her hastanın atımlarının etiketleri .atr uzantılı dosyalarda EKG sinyallerinin verileri de diğer dosyalarda tutulmaktadır. Şekil 4.3'te de görülüğü gibi her hastaya ait ikisi de farklı derivasyon

olan ikişer EKG sinyali bulunmaktadır. Bunun gibi veri hakkında bilgiler ve Şekil 4.4'teki gibi atım sembollerinin açıklama karşılık tablosu gibi meta veriler "mitdbdir" klasörünün içinde bulunmaktadır. Bu çalışmada MLII (V2'nin modifiye edilmiş türü) (Xie, 2020) derivasyon EKG sinyalleri kullanılmıştır. MIT Veri tabanında 46 adet hastanın MLII derivasyon EKG sinyali vardır.



Record 100 (MLII, V5; male, age 69)

Medications: Aldomet, Inderal

Beats	Before 5:00	After 5:00	Total
Normal	367	1872	2239
APC	4	29	33
PVC	-	1	1
Total	371	1902	2273

Supraventricular ectopy

- 33 isolated beats

Rhythm	Rate	Episodes	Duration
Normal sinus rhythm	70-89	1	30:06

Signal quality	Episodes	Duration
Both clean	1	30:06

Points of interest:

11:03 Normal sinus rhythm
25:13 PVC
26:09 APCs
27:55 Normal sinus rhythm

Record 101 (MLII, V1; female, age 75)

Medications: Diapres

Şekil 4.4 Hastaların bilgileri

Sonraki aşamada verilerin analiz edilebilmesi için Python'da içe aktarmamız gereklidir. Pythonın bunun için yaratılmış "wfdb" adlı bir kütüphanesi bulunmaktadır. Bu kütüphane ve "pandas" adlı kütüphane kullanılarak Sekil 4.5'te birkaç satırının bulunduğu her satır bir atımı temsil edecek şekilde matris formatta bir veri oluşturulmuştur.

Veri seti 46 hastanın MLII derivasyon EKG sinyalinden toplam 108098 atımdan oluşmaktadır. Veri setindeki "Symbol" değişkeni atımın sınıfını bir başka değişle normal mi abnormal mi olduğu (N: Normal, V: PVK gibi) bilgisini taşır. Bu değişken

sınıflandırma algoritması olan k-NN modeli için bağımlı değişken olmuştur. “Index” değişkeni atımın QRS kompleksinin, “Min_Index” değişkeni atımın başlangıcının, “Max_Index” ise atımın bitişinin zaman indeksidir. “Signal_Array” değişkeni ise içinde atımın büyüklük değerlerini tutan dizileri temsil eder. “Normal_Signal_Array” değişkeni “Signal_Array” değişkeninin Minimum-Maksimum dönüşümü ile normalleştirilmiş formatını tutar. “Patient_No” değişkeni ise atımın hangi hastanın atımı olduğu bilgisini tutar. Bundan sonraki analizlerde veri setindeki “Normal_Signal_Array” ve “Symbol” değişkenleri kullanılmıştır.

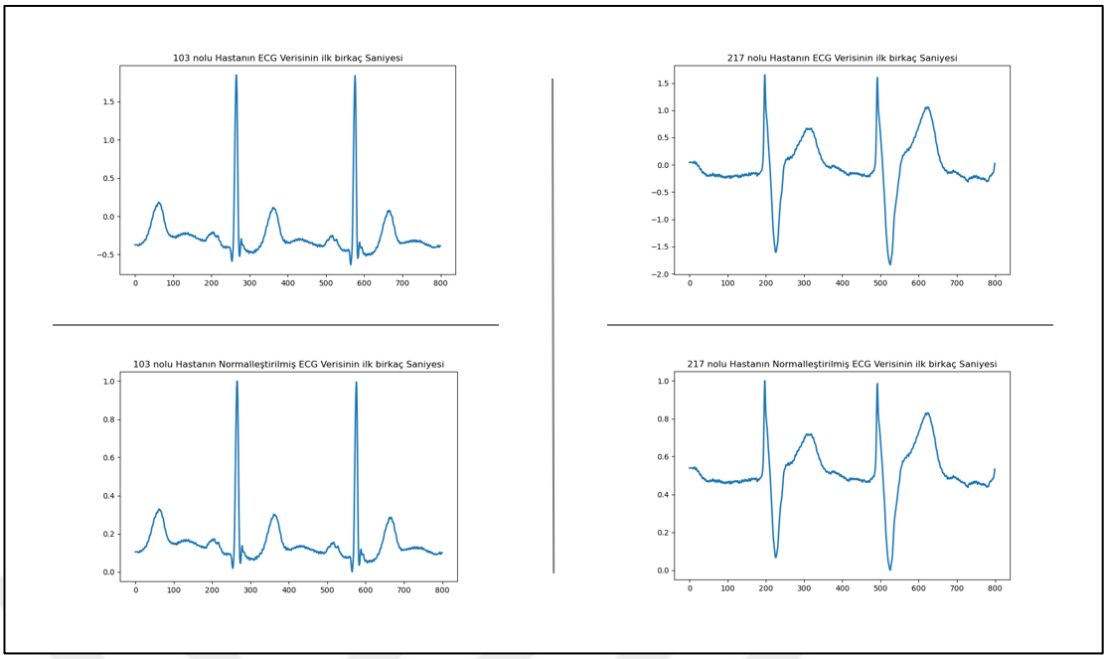
	Symbol	Index	Min_Index	Max_Index	Signal_Array	Normal_Signal_Array	Patient_No
0	N	77	38	224	[-0.255, -0.265, -0.275, -0.29, -0.29, -0.29, ...	[0.17358490566037735, 0.1660377358490566, 0.15...	100
1	N	370	224	516	[-0.26, -0.275, -0.28, -0.29, -0.275, -0.27, -...	[0.1864406779661017, 0.17627118644067796, 0.17...	100
2	N	662	516	804	[-0.31, -0.295, -0.29, -0.305, -0.31, -0.32, -...	[0.1699346405228758, 0.17973856209150327, 0.18...	100
3	N	946	804	1088	[-0.32, -0.32, -0.32, -0.325, -0.35, -0.35, -0...	[0.21594684385382062, 0.21594684385382062, 0.2...	100
4	N	1231	1088	1373	[-0.305, -0.315, -0.31, -0.315, -0.295, -0.3, ...	[0.18772563176895307, 0.18050541516245486, 0.1...	100
...	...	...	...	...	...	...	...
108093	N	648797	648676	648918	[-0.255, -0.255, -0.25, -0.24, -0.245, -0.25, ...	[0.05263157894736842, 0.05263157894736842, 0.0...	234
108094	N	649040	648918	649166	[-0.235, -0.225, -0.215, -0.21, -0.21, -0.22, ...	[0.08373205741626795, 0.08851674641148324, 0.0...	234
108095	N	649292	649166	649414	[-0.295, -0.285, -0.28, -0.285, -0.285, -0.275...	[0.0685279187817259, 0.0736040609137056, 0.076...	234
108096	N	649536	649414	649654	[-0.22, -0.24, -0.255, -0.26, -0.28, -0.275, -...	[0.06435643564356434, 0.05445544554455445, 0.0...	234
108097	N	649772	649654	649999	[-0.22, -0.21, -0.2, -0.215, -0.215, -0.225, -...	[0.09732360097323599, 0.1021897810218978, 0.10...	234

108098 rows × 7 columns

Şekil 4.5 İlk veri seti

Atımların büyüklüklerine normalizasyon uygulanmasının 3 sebebi vardır:

1. Sinyalleri karşılaştırabilmek için, farklı hastaların EKG sinyalleri, genellikle farklı büyüklük ölçeğine sahiptir. Bu durum farklı hastaların EKG Sinyalleri arasında karşılaştırmalar yaparken zorluklara neden olabilir. Normalleştirme, sinyalleri benzer bir ölçekte temsil ederek karşılaştırılabilirliği artırır.



Şekil 4.6 Orjinal ve normalleştirilmiş EKG sinyali

Şekil 4.6'da 103 ve 217 nolu hastaların orjinal ve normalleştirilmiş EKG sinyalleri görülmektedir. Orjinal sinyallere dikkatli bakılırsa büyüklük değerleri aralıkları iki hasta için farklıdır. Bu da yapılacak analizlerde yanılmaya neden olabilecek bir unsurdur. Bu nedenle EKG sinyallerine normalleştirme uygulanarak her hastanın sinyalinin büyüklük aralığı 0 ile 1 aralığına getirilir.

2. Gürültüyü azaltma: EKG sinyalleri, elektromanyetik girişimler, kas hareketleri ve diğer etkenler nedeniyle gürültü içerebilir. Normalleştirme, sinyaldeki gürültüyü azaltmak için sinyali belirli bir aralığa sıkıştırabilir veya gürültü oranını azaltabilir.
3. Özellik çıkarma: EKG sinyalleri üzerinde gerçekleştirilecek olan ileri işleme ve analiz yöntemleri, genellikle belirli özelliklerin çıkarılmasını gerektirir. Sinyallerin normalleştirilmesi, belirli özelliklerin daha kolay ayırt edilebilmesini sağlar. Burada aslında dalgacık dönüşümü özellik çıkarmak için kullanılmaktadır.

Buraya kadar anlatılanları yapan Python kodu aşağıda verilmiştir.

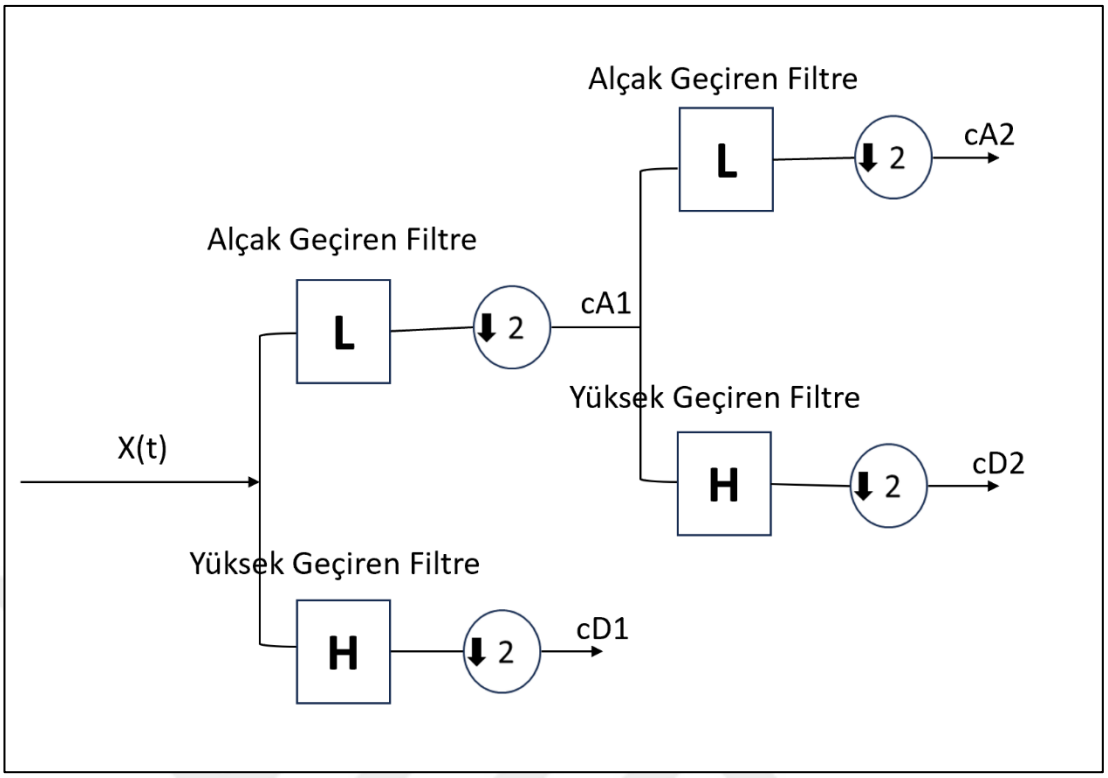
```
def prep_data(tuple_):
    patient_no,seq = tuple_
    record = wfdb.rdsamp(f'V1 ECG/{patient_no}')
    signal = record[0][:,seq]
    ann_ref = wfdb.rdann(f'V1 ECG/{patient_no}','atr')
    symbol = pd.Series(ann_ref.symbol)[1:]
    index = pd.Series(ann_ref.sample)[1:]
    df = pd.DataFrame({"Symbol":symbol,"Index":index})
    df["Lag"] = df["Index"].shift(1).fillna(0).astype(int)
    df["Min_Index"] = round(df["Index"] - ((df["Index"] - df["Lag"]) / 2),0).astype(int)
    df["Max_Index"] = df["Min_Index"].shift(-1).fillna(len(signal)-1).astype(int)
    df.index = range(len(df))
    df["Signal_Array"] = ""
    df["Normal_Signal_Array"] = ""
    for i in range(len(df)):
        signal_d = signal[df["Min_Index"][i]:df["Max_Index"][i]]
        df["Signal_Array"][i] = signal_d
        df["Normal_Signal_Array"][i] = (signal_d - np.min(signal_d)) / (np.max(signal_d) - np.min(signal_d))


     del df["Lag"]
    df["Patient_No"] = patient_no
    return df


tuple_list = [(100,0),(101,0),(103,0),(105,0),(106,0),(107,0),(108,0),(109,0),(111,0),(112,0),
            (113,0),(114,1),(115,0),(116,0),(117,0),(118,0),(119,0),(121,0),(122,0),(123,0),(124,0),
             (200,0),(201,0),(202,0),(203,0),(205,0),(207,0),(208,0),(209,0),(210,0),(212,0),(213,0),
             (214,0),(215,0),(217,0),(219,0),(220,0),(221,0),(222,0),(223,0),(228,0),(230,0),(231,0),
             (232,0),(233,0),(234,0)]

df_full = pd.DataFrame()
for z in tuple_list:
    df = prep_data(z)
    df_full = pd.concat([df_full,df],axis=0)
```

Zaman ortamında, alçak ve yüksek geçiren süzgeçlerden oluşan bir süzgeç bankası kullanılarak KDD katsayılarının elde edilmesi mümkündür (Didiot vd., 2010). Şekil 4.7'de iki seviyeli bir KDD'nin değişken çıkarımını nasıl yaptığı gösterilmektedir.



Şekil 4.7 KDD katsayı ayrıştırma şeması

Şekil 4.7'ye göre, her filtreleme katmanından sonra filtrelerin çıkışlarındaki işaretlerin örnek sayısı yarı yarıya azaltılmaktadır. Alçak geçiren (L) filtrelerin çıkışlarındaki katsayılar yaklaşım katsayıları ve yüksek geçiren (H) filtrelerin çıkışındaki katsayılar ayrıntı veya dalgacık katsayıları olarak isimlendirilmektedir. Yaklaşım katsayıları işaretin yerel ortalamalarını verirken, ayrıntı katsayıları ise bu yerel ortalamalar arasındaki farkı ifade etmektedir. (Düzenli vd., 2015)

Sonraki aşamada her atımın normalleştirilmiş EKG Sinyaline 5 Seviyeli Biorthogonal dalgacık fonksiyonlu Kesikli Dalgacık Dönüşümü uygulanarak yukarıda bahsedilen katsayılar hesaplanmıştır (cD1, cD2, cD3, cD4, cd5, cA5). Ayrıştırma sonucu oluşan bu katsayıların ortalama standart sapma, minimum, maksimum, aralık (maksimum – minimum) gibi istatistiksel ölçümleri kullanılarak 30 adet değişken elde edilmiştir. Veri setinin son halinin birkaç satırı ve birkaç sütunu Şekil 4.8'de görülmektedir.

Normal_Signal_Array	Patient_No	cA5_Min	cA5_Max	cA5_Min_Max_Diff	cA5_Mean	cA5_Std	cD5_Min	cD5_MDx	cD5_Min_Max_Diff	cD5_Mean	cD5_Std	cD
0.17358490566037735, 0.1660377358490566, 0.15...	100	0.539176	1.702099	1.162923	0.889390	0.266477	-0.073222	0.778442	0.851664	0.089790	0.237081	-0.(
[0.1864406779661017, 0.17627118644067796, 0.17...	100	0.409248	1.281441	0.872193	0.888007	0.200653	-1.460192	0.133833	1.594025	-0.087247	0.347099	-0.1
[0.1699346405228758, 0.17973856209150327, 0.18...	100	0.566300	1.107962	0.541661	0.921136	0.138765	-1.252810	0.109233	1.362042	-0.068388	0.298050	-0.1
0.21594684385382062, 0.21594684385382062, 0.2...	100	0.975241	1.390703	0.415462	1.203288	0.110688	-0.725046	0.136124	0.861170	-0.036668	0.179432	-1.1
0.18772563176895307, 0.18050541516245486, 0.1...	100	0.548608	1.296949	0.748341	0.990585	0.165216	-0.702366	0.136204	0.838570	-0.033298	0.177332	-0.9
...	...	...	...	...	...	...	...	...	...	...	...	
0.05263157894736842, 0.05263157894736842, 0.0...	234	-0.007818	1.259988	1.267806	0.430647	0.268489	-1.082329	0.317094	1.399423	-0.056791	0.282592	-1.(

Şekil 4.8 KDD sonrası veri seti

KDD kullanılarak ayrıştırma katsayılarının çıkartılması işlemi, Python'ın "pywt" kütüphanesinin "wavedec" fonksiyonu kullanılarak yapılmıştır. Katsayıların istatistiksel ölçümlerinin elde edilmesi işlemi için ise "numpy" ve "pandas" kütüphaneleri kullanılmıştır.

Kullanılan Python kodu aşağıdadır.

```
(df["cA5_Min"] = 0, df["cA5_Max"] = 0,df["cA5_Min_Max_Diff"] = 0,df["cA5_Mean"] = 0, df["cA5_Std"] = 0,df["cA5_Min"] = 0,
df["cD5_Min"] = 0,df["cD5_MDx"] = 0,df["cD5_Min_Max_Diff"] = 0,df["cD5_Mean"] = 0,df["cD5_Std"] = 0,df["cD5_Min"] = 0,
df["cD4_Min"] = 0,df["cD4_MDx"] = 0,df["cD4_Min_Max_Diff"] = 0,df["cD4_Mean"] = 0,df["cD4_Std"] = 0,df["cD4_Min"] = 0,
df["cD3_Min"] = 0,df["cD3_MDx"] = 0,df["cD3_Min_Max_Diff"] = 0,df["cD3_Mean"] = 0,df["cD3_Std"] = 0,df["cD3_Min"] = 0,
df["cD2_Min"] = 0,df["cD2_MDx"] = 0,df["cD2_Min_Max_Diff"] = 0,df["cD2_Mean"] = 0,df["cD2_Std"] = 0,df["cD2_Min"] = 0,
df["cD1_Min"] = 0,df["cD1_MDx"] = 0,df["cD1_Min_Max_Diff"] = 0,df["cD1_Mean"] = 0,df["cD1_Std"] = 0,df["cD1_Min"] = 0)
for i in range(len(df)):
    cA5, cD5, cD4, cD3, cD2, cD1 = wavedec(list(df["Normal_Signal_Array"][i]), 'bior4.4', level=5)
    df["cA5_Min"][i] = min(cA5),df["cA5_Max"][i] = max(cA5)
    df["cA5_Min_Max_Diff"][i] = max(cA5) - min(cA5)
    df["cA5_Mean"][i] = np.mean(cA5)
    df["cA5_Std"][i] = np.std(cA5)

    df["cD5_Min"][i] = min(cD5)
    df["cD5_MDx"][i] = max(cD5)
    df["cD5_Min_Max_Diff"][i] = max(cD5) - min(cD5)
    df["cD5_Mean"][i] = np.mean(cD5)
    df["cD5_Std"][i] = np.std(cD5)

    df["cD4_Min"][i] = min(cD4)
    df["cD4_MDx"][i] = max(cD4)
    df["cD4_Min_Max_Diff"][i] = max(cD4) - min(cD4)
    df["cD4_Mean"][i] = np.mean(cD4)
    df["cD4_Std"][i] = np.std(cD4)

    df["cD3_Min"][i] = min(cD3)
    df["cD3_MDx"][i] = max(cD3)
    df["cD3_Min_Max_Diff"][i] = max(cD3) - min(cD3)
    df["cD3_Mean"][i] = np.mean(cD3)
    df["cD3_Std"][i] = np.std(cD3)

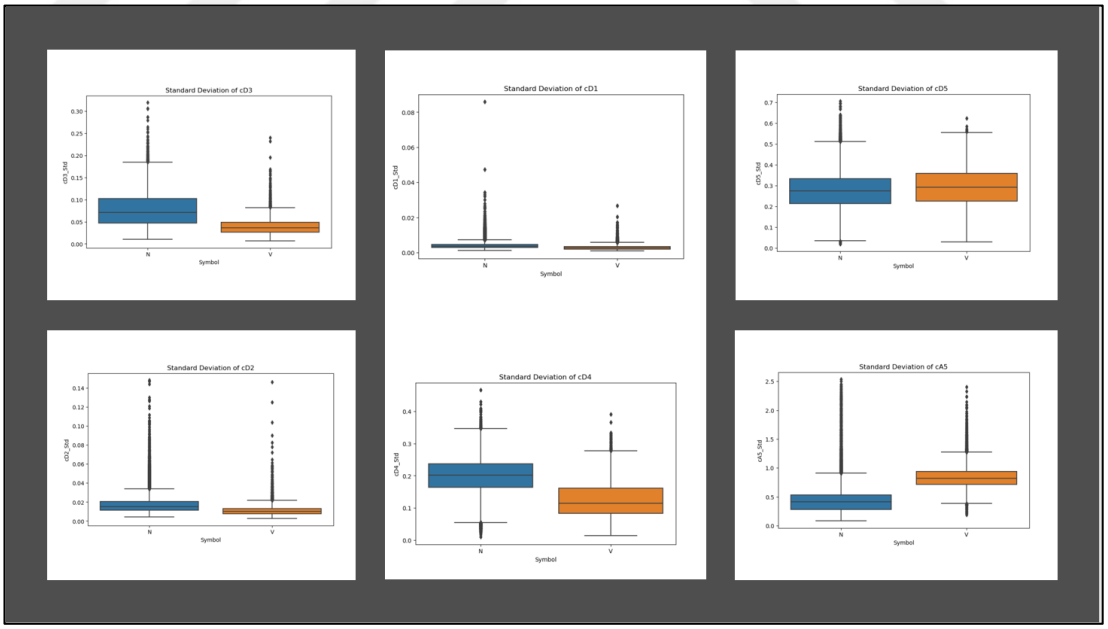
    df["cD2_Min"][i] = min(cD2)
    df["cD2_MDx"][i] = max(cD2)
    df["cD2_Min_Max_Diff"][i] = max(cD2) - min(cD2)
    df["cD2_Mean"][i] = np.mean(cD2)
    df["cD2_Std"][i] = np.std(cD2)

    df["cD1_Min"][i] = min(cD1)
    df["cD1_MDx"][i] = max(cD1)
    df["cD1_Min_Max_Diff"][i] = max(cD1) - min(cD1)
    df["cD1_Mean"][i] = np.mean(cD1)
    df["cD1_Std"][i] = np.std(cD1)
```

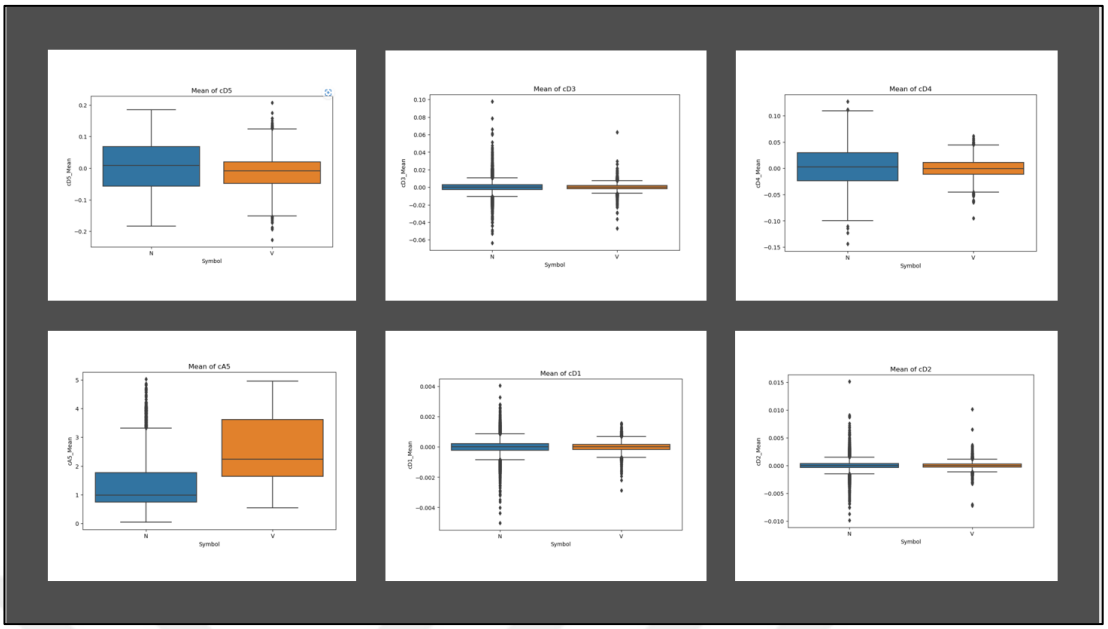
Keşifsel Veri Analizi yaptığımızda aşağıdaki şekillerden de görüldüğü gibi bazı değişkenler normal atımlar (N) ile Prematüre ventriküler kasılma olan atımları (V) iyi bir şekilde ayırabilmektedir. Bu değişkenler,

5. Seviye Ölçeklendirme katsayısının standart Sapması,
5. Seviye Ölçeklendirme katsayısının Maksimum Değeri,
5. Seviye Ölçeklendirme katsayısının Minimum değeri ile maksimum değeri arasındaki fark (Range / Açıklık değeri),
2. Seviye Dalgacık katsayısının Minimum değeri ile maksimum değeri arasındaki fark (Açıklık değeri),
3. Seviye Dalgacık katsayısının Minimum değeri ile maksimum değeri arasındaki fark (Açıklık değeri),
4. Seviye Dalgacık katsayısının Minimum değeri ile maksimum değeri arasındaki fark (Açıklık değeri),

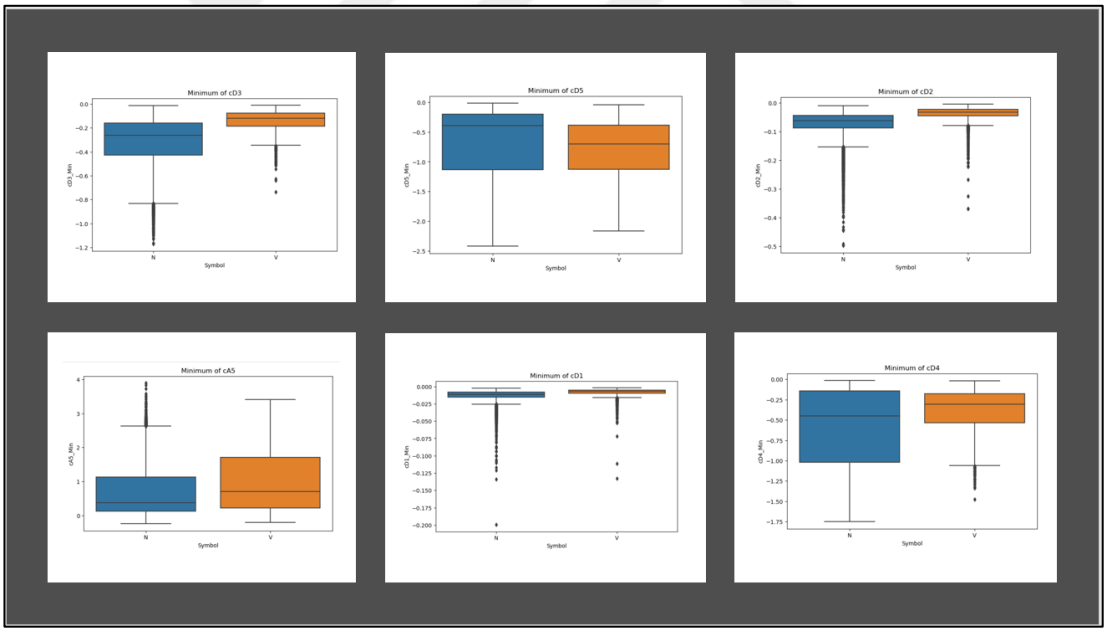
Şekil 4.9, 4.10, 4.11, 4.12, 4.13'te de Normal ve PVK atımları arasında bu 30 değişken bazında farklar kutu grafikleri ile gösterilmektedir.



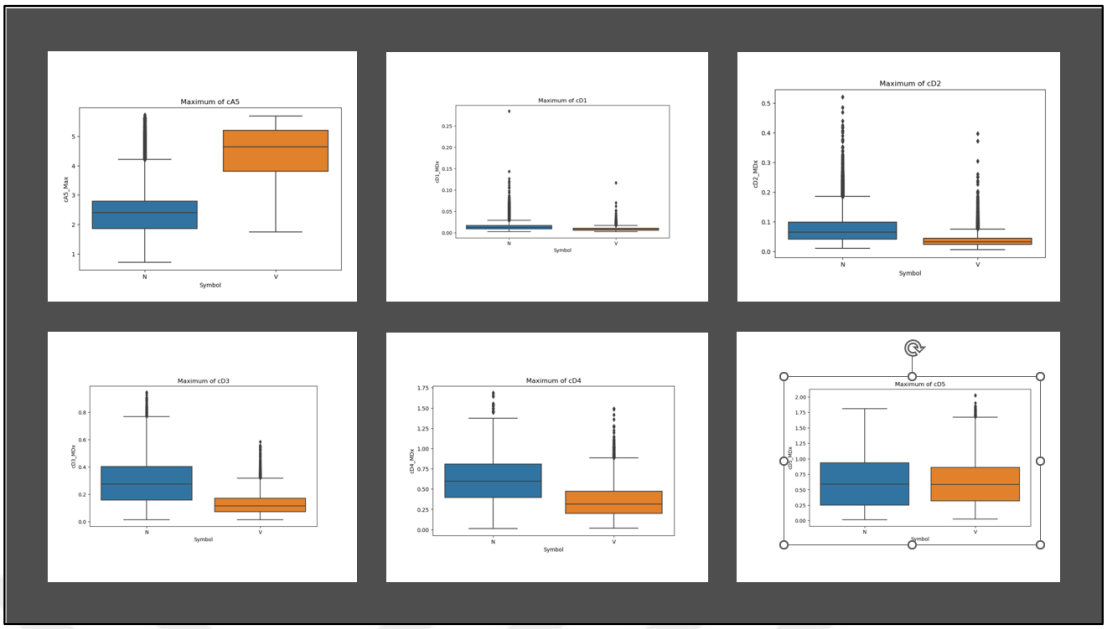
Şekil 4.9 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (standart sapma)



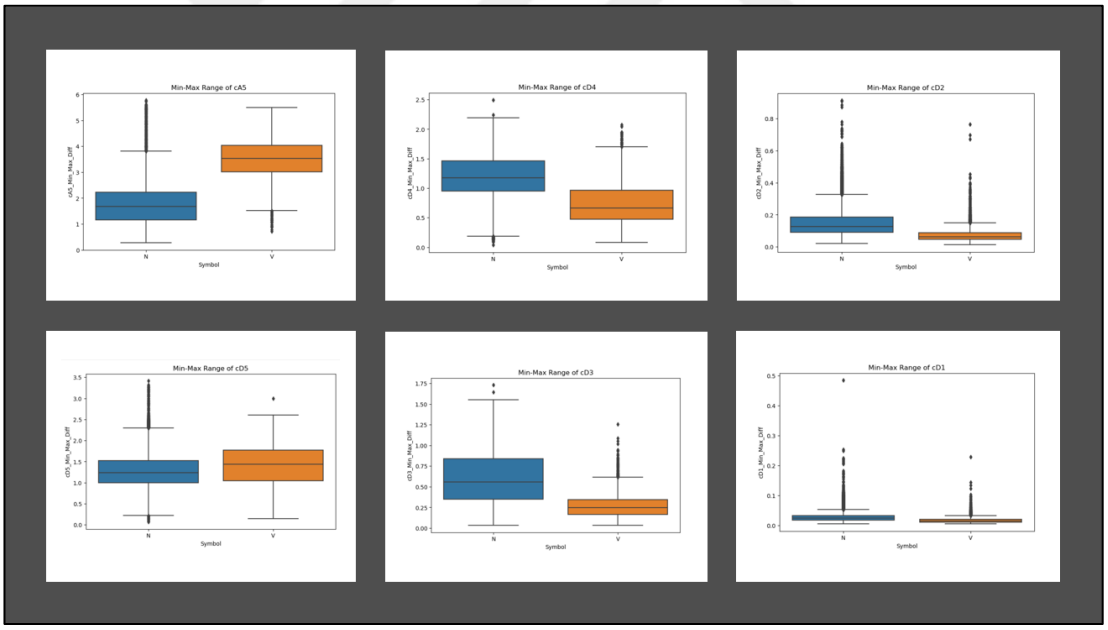
Şekil 4.10 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (ortalama)



Şekil 4.11 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (minimum)



Şekil 4.12 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (maximum)



Şekil 4.13 Atım sınıfları arasındaki farkları görmek için kutu grafikleri (minimum – maximum farkı)

Bu aşamadan sonra Şekil 4.14'te görüldüğü gibi analize yukarıda verilen 6 değişken ve Sembolü N (Normal atımlar) ve V (PVK Atımlar) olan 81914 gözlem ile devam edilecektir.

	cA5_Std	cA5_Max	cA5_Min_Max_Diff	cD4_Min_Max_Diff	cD3_Min_Max_Diff	cD2_Min_Max_Diff	Symbol
0	0.266477	1.702099	1.162923	1.017396	1.177031	0.164482	N
1	0.200653	1.281441	0.872193	1.146602	0.867972	0.140168	N
2	0.138765	1.107962	0.541661	0.996281	0.918590	0.209055	N
3	0.110688	1.390703	0.415462	1.372051	1.055739	0.176502	N
4	0.165216	1.296949	0.748341	1.214367	1.287508	0.147873	N
...	...	...	...	...	...	...	...
81909	0.268489	1.259988	1.267806	1.597943	0.969387	0.143386	N
81910	0.294962	1.389462	1.422526	1.672247	0.775405	0.101657	N
81911	0.328790	1.641821	1.718043	1.640195	0.784177	0.117948	N
81912	0.297626	1.405913	1.460162	1.819268	0.547661	0.099650	N
81913	0.295492	1.058238	1.003228	0.939082	0.877730	0.137643	N

81914 rows × 7 columns

Şekil 4.14 Değişken seçimi sonrası veri seti

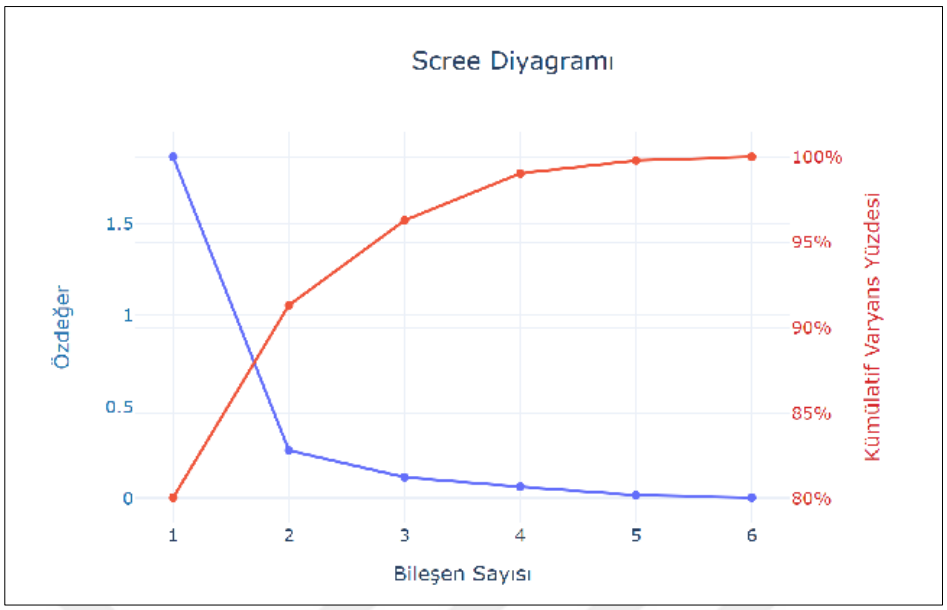
Sonrasında bu seçilen 6 değişken üzerinde Temel Bileşen Analizi uygulanmıştır. Buradaki amaç veriyi daha az boyuta indirip görselleştirebilmektir.

Temel Bileşenler Analizi uygulandıktan sonra Tablo 4.1'de görüldüğü gibi ilk bileşen 1,863, ikinci bileşen ise 0,262 özdeğer değerini almıştır ve bu iki bileşen tüm veri setinin kümülatif olarak %91,309'unu açıklamaktadır.

Tablo 4.1 TBA bileşenlerinin özdeğerler tablosu

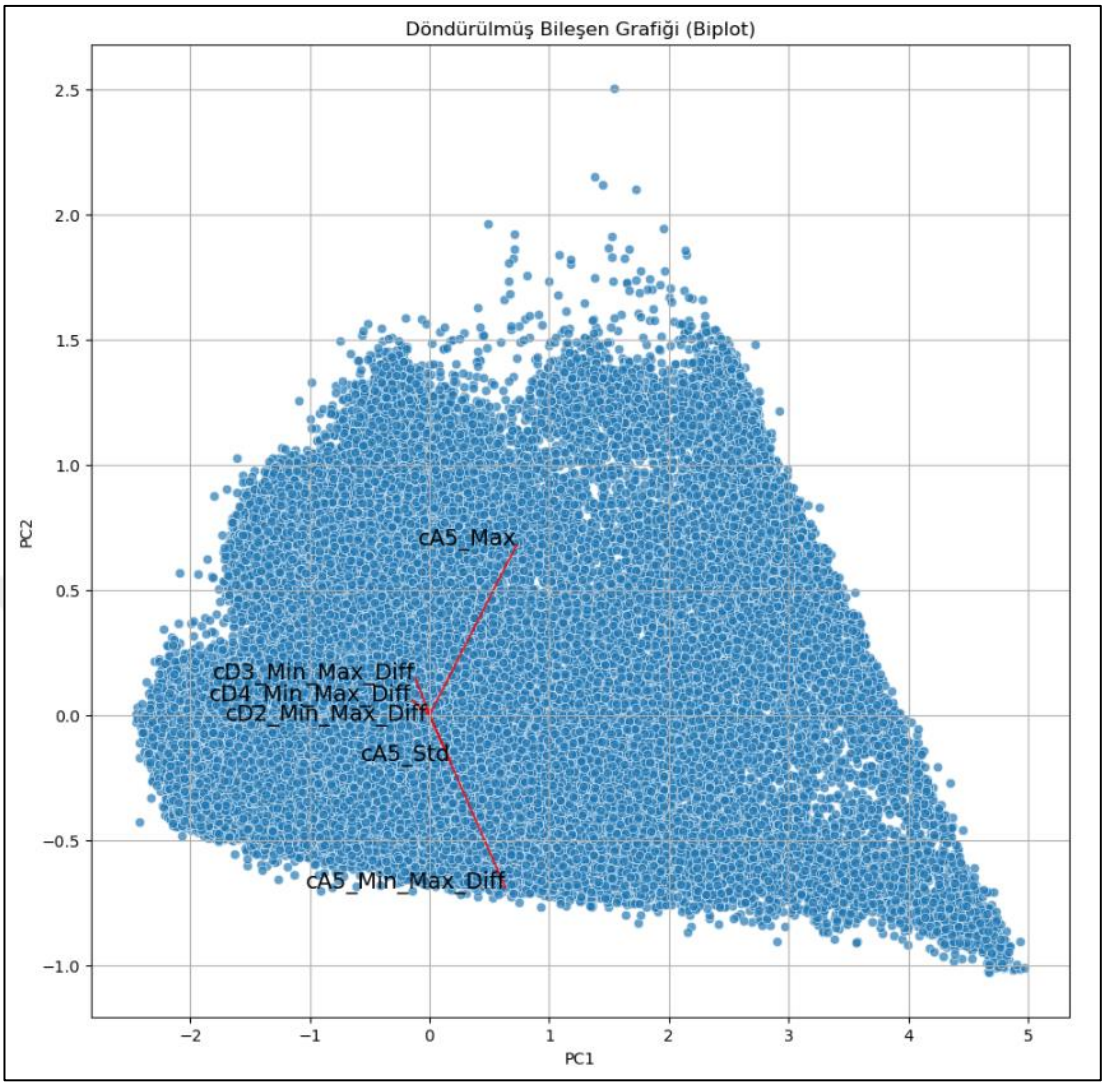
Bileşen	Özdeğerler	Varyans Yüzdesi	Kümülatif Yüzde
1	1,863	80,063	80,063
2	0,262	11,247	91,309
3	0,116	4,973	96,282
4	0,064	2,748	99,03
5	0,018	0,755	99,785
6	0,005	0,215	100

Şekil 4.15'de Görülebileceği gibi iki bileşen tüm veri setinin kümülatif olarak %91,309'unu açıklamaktadır.



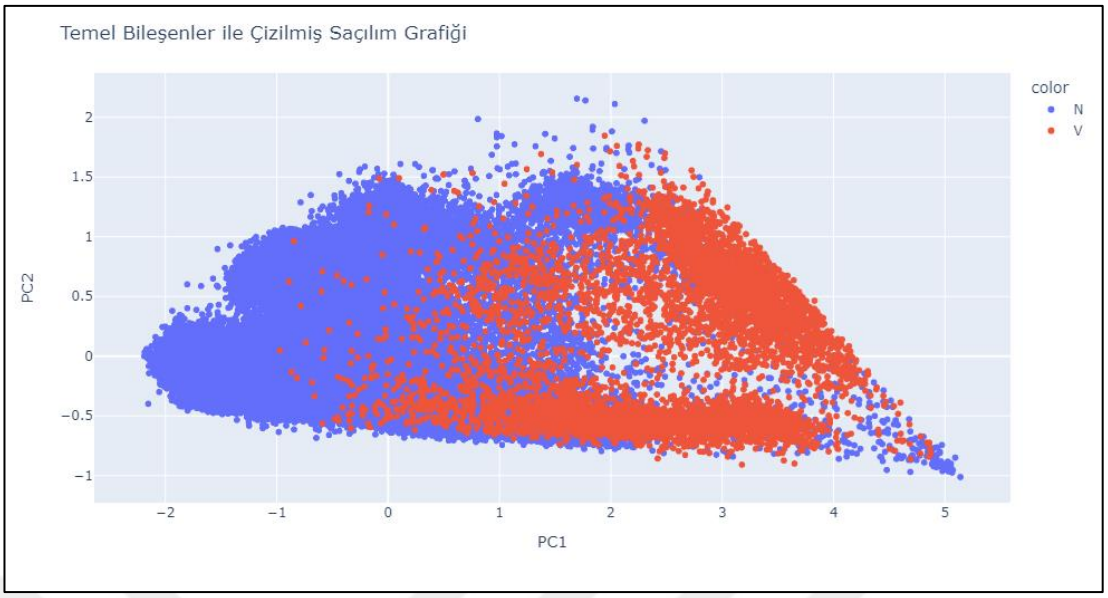
Şekil 4.15 Çalışmadaki Temel Bileşen Analizinin Scree Diyagramı

Şekil 4.16'te döndürülmüş bileşenlerin bileşen 1 ve bileşen 2'ye olan konumları yer almaktadır. Grafikte görüldüğü gibi, "2. Seviye Dalgacık katsayısının minimum maksimum farkı", "3. Seviye Dalgacık katsayısının minimum maksimum farkı", "4. Seviye Dalgacık katsayısının minimum maksimum farkı" değişkenleri bileşen 1 ile ilişkilidir, "5. Seviye Ölçeklendirme katsayısının standart sapması", "5. Seviye Ölçeklendirme katsayısının maksimum değeri", "5. Seviye Ölçeklendirme katsayısının minimum maksimum Farkı" değişkenleri ise bileşen 2 ile ilişkilidir. "2. Seviye Dalgacık katsayısının minimum maksimum farkı", "3. Seviye Dalgacık katsayısının minimum maksimum farkı", "4. Seviye Dalgacık katsayısının minimum maksimum farkı" değişkenleri artarken bileşen 1 azalır. "5. Seviye Ölçeklendirme katsayısının maksimum değeri" değişkeni artarken bileşen 2 artar. "5. Seviye Ölçeklendirme katsayısının standart sapması" ve "5. Seviye Ölçeklendirme katsayısının minimum maksimum Farkı" artarken bileşen 2 azalır.



Şekil 4.16 Döndürülmüş bileşen grafiği

Şekil 4.17'de iki temel bileşen ile çizilen grafikte görüleceği üzere bu seçilen değişkenler ile bir k-NN ile makine öğrenmesi modeli kurarsak normal ve PVK atımlarını tahminlemekte başarılı olabilmektedir.



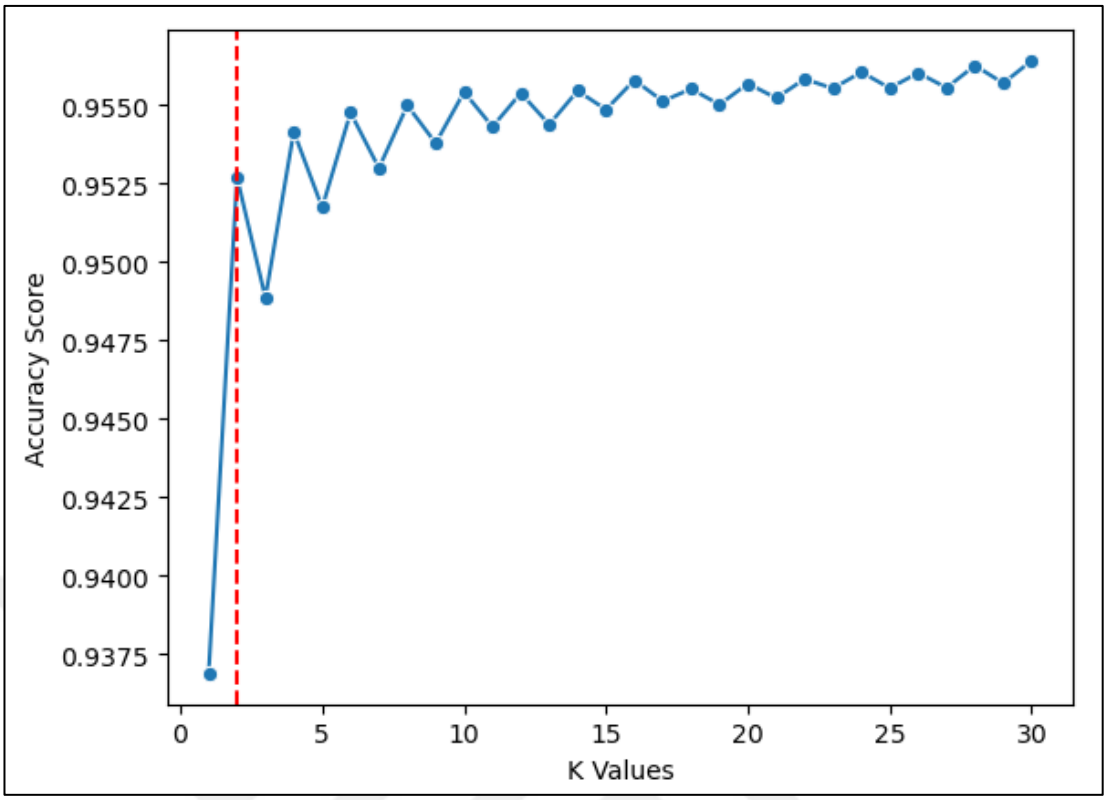
Şekil 4.17 Temel bileşenler ile çizilmiş saçılım grafiği

Temel Bileşenler Analizi Python'da "scikit-learn" kütüphanesi kullanılarak yapılmıştır. Grafik ve görsellerin oluşturulması için ise "matplotlib" ve "plotly" kütüphaneleri kullanılmıştır. Temel Bileşenler Analizi için kullanılan Python kodu aşağıdadır:

```
from sklearn.decomposition import PCA
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
X = df[["cA5_Std","cA5_Max",'cA5_Min_Max_Diff','cD4_Min_Max_Diff','cD3_Min_Max_Diff','cD2_Min_Max_Diff']]
################### Biplot ###################################
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
import pandas as pd
df_pca = pd.DataFrame(data=X_pca, columns=['PC1', 'PC2'])
plt.figure(figsize=(11, 11))
sns.scatterplot(x='PC1', y='PC2', data=df_pca, alpha=0.7)
for i, var in enumerate(pca.components_.T):
    plt.arrow(0, 0, var[0], var[1], color='r', alpha=0.8)
    if X.columns[i] == "cD2_Min_Max_Diff":
        plt.text(var[0], var[1]-0.05, f'{X.columns[i]}', fontsize=14, ha='right')
    else:
        plt.text(var[0], var[1], f'{X.columns[i]}', fontsize=14, ha='right')

plt.title('Döndürülmüş Bileşen Grafiği (Biplot)')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.grid(True);
###################### Özvektör Tablosu ##################
pca = PCA(n_components=6)
X_pca = pca.fit_transform(X)
ozvektor = pca.explained_variance_
varyans_yuzde = (pca.explained_variance_ratio_*100).round(3)
kum_yuzde = (pca.explained_variance_ratio_.cumsum()*100).round(3)
özvek_tablo =pd.DataFrame({"Bileşen":[1,2,3,4,5,6], "Özvektörler":ozvektor.round(3),"Varyans Yüzdesi":varyans_yuzde,
             "Kümülatif Yüzde":kum_yuzde})
################ Saçılım Grafiği ##########################
components = pca.fit_transform(X)
import plotly.express as px
fig = px.scatter(components, x=0, y=1, color=df['Symbol'])
fig.update_layout(
    xaxis_title="PC1", yaxis_title="PC2",
    title="Temel Bileşenler ile Çizilmiş Saçılım Grafiği"
);
```

Son olarak veri seti eğitim ve test olmak üzere ikiye ayrılmaktadır. Gözlemlerin %67'i eğitim veri setinde ve %33'si test veri setinde bulunmaktadır. Şekil 4.17'de görüldüğü gibi k değeri 2 olarak belirlenerek eğitim veri setini kullanarak k değeri 2 olan k-NN eğitilmektedir. Bu modeli test veri setindeki gözlemleri tahmin etmek için kullanıldığında atımların sınıflarını %97,5 doğruluk ile tahmin edilebilmektedir. Şekil 4.18'de k değerini seçmek için kullanılan k değeri ve doğruluk oranı grafiği gösterilmiştir.



Şekil 4.18 k değeri belirleme

k-NN Python'da uygulanması işlemi "scikit-learn" kütüphanesi ile yapılmıştır. Grafiğin oluşturulması için ise "matplotlib" kütüphanesi kullanılmıştır. k-NN için kullanılan Python kodu aşağıdadır:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.model_selection import cross_val_score
X = df[["cA5_Std","cA5_Max",'cA5_Min_Max_Diff','cD4_Min_Max_Diff','cD3_Min_Max_Diff','cD2_Min_Max_Diff']]
y = df["Symbol"]
################ Veri Setinin Eğitim Test Veri setine Bölünmesi ###############
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
################ KEYKA Uygulama ########################################
neigh = KNeighborsClassifier(n_neighbors=2)
neigh.fit(X_train, y_train)
y_pred = neigh.predict(X_test)
accuracy_score(y_test, y_pred)
confusion_matrix(y_test, y_pred)
############# Dirsek Metodu için Çizgi Grafiği ######################
k_values = [i for i in range (1,31)]
scores = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X, y, cv=5)
    scores.append(np.mean(score))

sns.lineplot(x = k_values, y = scores, marker = 'o')
plt.xlabel("K Values")
plt.axvline(x=2, color='red', linestyle='--')
plt.ylabel("Accuracy Score");
```

Sınıflandırma modelinin performansını değerlendirmek için Karışıklık Matrisi, doğruluk oranı, Özgüllük (Specifity) ve Duyarlılık (Sensitivity) oranları kullanılmıştır.

Tablo 5.1 Karışıklık matrisi (Confusion matrix)

		GERÇEK	
		POZİTİF	**NEGATİF**
TAHMİN	**POZİTİF**	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
	NEGATİF	Yanlış Negatif (FN)	Doğru Negatif (TN)

Kurulan bir sınıflandırma modelinin sonuçlarını yorumlayabilmek, gerçek ve tahmin edilen değerler arasında yer alan ilişkideki hataları çapraz olarak inceleyebilmek için karışıklık matrisi kullanılır.

Karışıklık matrisi, modelin yanılma durumunu özetleyen bir matristir. Tablo 5.1.'de genel formu verilen karışıklık matrisinde sütunlar gerçek duruma, satırlar tahmin edilen gözlem sayılarına karşılık gelir. Doğru Pozitif (TP), gerçekte pozitif olanın pozitif olarak tahmin edildiği, Doğru Negatif (TN) ise gerçekte negatif olanın

negatif olarak tahmin edildiği değerlerdir. Bu değerler "isabetli" yapılan tahminleri gösterir ve bu değerler ne kadar büyükse modelin o kadar iyi tahminde bulunduğu söylenebilir. Gerçekte pozitif olan bir sınıfın negatif olarak, gerçekte negatif olan bir sınıfın pozitif olarak tahmin edildiği değerler modelin yanıldığı değerleri gösterir.

Çalışmada pozitif olma durumu normal kalp atımlarına karşılık gelirken negatif durumu PVK atıma karşılık gelir. Bu durumda TP atım gerçekte normal bir atımken modelin tahmini de normal olduğu atım sayısıdır. Tablo 5.2'de çalışmada sunulan model kullanılarak elde edilen karışıklık matrisi sunulmuştur. TN atım gerçekte PVK bir atımken modelin tahmininin de PVK olduğu atım sayısıdır. TP ve TN değerlerinin görece yüksek olması modelin iyi bir sınıflama yaptığını gösterir. FP atım gerçekte PVK bir atımken modelin tahmininin normal olduğu yani modelin yanlış tahmin ettiği atım sayısıdır. FN atım gerçekte normal bir atımken modelin tahmininin PVK olduğu yani yanlış tahmin ettiği atım sayısıdır.

Tablo 5.2 Çalışmada önerilen modele ait karışıklık matrisi

		Gerçek: Normal	Gerçek: PVK
Tahmin	Normal	24423	399
	PVK	228	1982

Modelin sınıflama başarımını değerlendirmek için doğruluk, özgüllük ve duyarlılık oranları kullanılmıştır. Doğruluk oranı, tüm gözlemler içinde doğru yapılan tahminlerin oranını gösterir ve (4.1)'de verildiği gibi hesaplanır. Özgüllük negatif tahmin edilen gözlemlerin ne kadarının gerçekten negatif olduğunu gösteren bir ölçüt olup (4.2) ile verilen formülle hesaplanır. Duyarlılık ise gerçekte pozitif olan gözlemlerden ne kadarının pozitif olarak tahmin edildiğini gösteren bir ölçüt olup (4.3) formülü ile hesaplanır.

Modelin doğruluk, özgüllük ve duyarlılık oranları ve bu değerler kullanılarak hesaplanmaktadır. Doğruluk, özgüllük ve duyarlılık oranlarının formülleri formül 4.1, 4.2 , 4.3 ve 4.4'te verilmektedir.

$$\text{Doğruluk Oranı} = (TN + TP) / (TN+FP+FN+TP) \quad (4.1)$$

$$\text{Özgüllük Oranı} = (TN) / (TN + FP) \quad (4.2)$$

$$\text{Duyarlılık Oranı} = (TP) / (TP + FN) \quad (4.3)$$

Oluşturulan k-En yakın komşu modeli ile atımlar %97,5 doğruluk oranı ile Normal ve Prematüre Ventriküler Kasılma atımları sınıflandırılabilmektedir. Tablo 5.2'de de görülebileceği gibi test veri setimizde gerçekte normal olan 24651 adet atımın 24423 adetini doğru tahmin edilmektedir, Özgüllük (Specifity) yüzdesi %83,2 Gerçekte PVK (PVC) olan 2381 adet atımın 1982 adetini doğru tahmin edilmektedir, duyarlılık (Sensitivity) yüzdesi %99,1 bulunmuştur.

BÖLÜM BEŞ
SONUÇ

Bu çalışmada EKG sinyalleri analiz edilerek, bir hastanın aritmi gibi bir kalp hastalığı olup olmadığını tahmin eden bir model üzerine çalışılmıştır. k-En Yakın Komşuluk Algoritması (k-NN) kullanılarak kalp atımlarını normal ve Prematüre Ventriküler Kasılma olarak sınıflandırılmıştır. MIT Aritmi veri tabanındaki EKG Sinyalleri ve her atımın sınıf etiketi kullanılarak EKG sinyalindeki her bir atımın Kesikli Dalgacık Dönüşümü (KDD) yöntemi ile 5 seviyeli dalgacık katsayıları çıkartılmıştır. Ardından ise bu dalgacık katsayılarının bazı tanımlayıcı istatistikleri K-NN modeli için girdi değişkenleri olarak kullanılmıştır. Ayrıca girdi değişkenlerini görselleştirilebilmesi için girdi değişkenleri Temel Bileşenler analizi (TBA) kullanılarak 2 boyuta indirilmiştir.

Oluşturulan k-En yakın komşu modeli ile atımlar %97,5 doğruluk oranı ile Normal ve Prematüre Ventriküler Kasılma atımları olarak sınıflandırılabilmektedir. Daha optimize bir k-NN modeli ya da derin sinir ağları ile bu oranlar yükseltilebilir.

Oluşturulan bu k-NN modeli ile Prematüre Ventriküler Kasılma atımlarının normal atımlardan otomatik bir şekilde ayrılabileceği ortaya konulmuştur. Bu algoritma sayesinde kardiyologların iş yükü azaltılabilirken teşhis doğruluğu arttırılabilir. İş yükü fazla olduğu için kısa süreli kaydedilen EKG sinyalleri bu sistem sayesinde daha uzun süreli toplanıp işlenebilir.

MIT BIH Aritmi veritabanı üzerinde başarılı sonuçlar elde edilmiştir. Bu sistemin daha da geliştirilebilmesi için daha fazla EKG sinyali elde edilip daha farklı EKG örüntülerini de ele alınabilir.

KAYNAKLAR

Balaskas, K. ve Siozios, K. (2019). ECG Analysis and Heartbeat Classification Based on Shallow Neural Networks. *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST*), Thessaloniki, Greece, 2019 1-4. https://doi.org/10.1109/MOCAST.2019.8742072

Burns, E. ve Buttner, R. (2021, Haziran 2) Premature Ventricular Complex (PVC). Life in the Fastline. https://litfl.com/premature-ventricular-complex-pvc-ecg-library/

Didiot, E., Illina I., Fohr, D. ve Mella, O (2010). A Wavelet-Based Parameterization for Speech /Music Discrimination. *Computer Speech and Language*, 24(2), 341–357. https://doi.org/10.1016/j.csl.2009.05.003

Düzenli, T., Doğan, H. ve Özkurt, N. (2015). Konuşma/Müzik Ayrıştırması için Kesikli Dalgacık Dönüşümü Tabanlı Öznitelik Çıkarımı. *Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 30(2), 103-114. https://doi.org/10.21605/cukurovaummfd.242769

Géron, A. (2017) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.

Gomes, R., Soares, F., Correia J. ve Lima C. (2010). ECG Data-Acquisition and classification system by using wavelet-domain Hidden Markov Models. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, Argentina, 2010* 4670-4673. https://doi.org/*10*.1109/IEMBS.2010.5626456

Hall, J. (2011) *Guyton and Hall Textbook of Medical Physiology*, Saunders Elsevier, Philadelphia.

Hastie, T., Tibshirani, R. ve Friedman, J. (2001). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer New York, NY

Hu, S., Wei, H., Chen, Y. ve Tan, J. (2012). A real-time cardiac arrhythmia classification system with wearable sensor networks. *Sensors (Basel) 2012*, 12(9), 12844-12869 https://doi.org/10.3390/s120912844

James, G., Witten D., Hastie, T. ve Tibshirani, R. (2013) *An Introduction to Statistical Learning with Applications in R*. Springer New York

Kiranyaz, S., Ince, T., ve Gabbouj, M. (2016) Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering*, 63(3), 664-675. https://doi.org/10.1109/TBME.2015.2468589.

Li, H., Wang, Y., Cui, D. ve Cao, L. (2016). Arrhythmia Classification Based on Multi-Domain Feature Extraction for an ECG Recognition System. *Sensors (Basel)*, 16(10) 1744. https://doi.org/10.3390/s16101744

Llamedo, M. ve Martinez, J. (2011). Heartbeat Classification Using Feature Selection Driven by Database Generalization Criteria. *IEEE Transactions on Biomedical Engineering*, 58(3), 616-625. https://doi.org/10.1109/TBME.2010.2068048

Madeiro, J., Cortez, P., Filho, P. ve Brayner, A. (2019). João Paulo do Vale Madeiro, Paulo César Cortez, José Maria da Silva Monteiro Filho ve Angelo Roncalli Alencar Brayner (Ed.), *Developments and Applications for ECG Signal Processing*, içinde (1-28) Academic Press. https://doi.org/10.1016/B978-0-12-814035-2.00007-4

Mastoi, Q., Shaikh, A., Reshan, M., Sulaiman, A., Elmagzoub, M. ve AlYami, S. (2023). A fully automatic model for premature ventricular heartbeat arrhythmia classification using the Internet of Medical Things. *Biomedical Signal Processing and Control*, 83(1), 104697. https://doi.org/10.1016/j.bspc.2023.104697

Meng, L., Tan, W., Ma, J. Wang, R., Yin, X. ve Zhang, Y. (2022). Enhancing dynamic ECG heartbeat classification with lightweight transformer model. *Artificial Intelligence in Medicine*, 124(1), 102236. https://doi.org/10.1016/j.artmed.2022.102236.

Moody, G. ve Mark, R. (2001). The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3), 45-50. https://doi.org/10.1109/51.932724

Nagaraju, B. ve Srinivas, K. (2019). Arrhythmia Detection from ECG based Heartbeat Classification using Deep Learning Networks. *International Journal of Computer Applications*, 178(26), 11-16. https://doi.org/10.1155/2022/1094830

Neves, I., Folgado, D., Santos, S, Barandas, M., Campagner, A., Ronzio, L., Cabitza, F. ve Gamboa, H. (2021) Interpretable heartbeat classification using local model-agnostic explanations on ECGs. *Computers in Biology and Medicine*, 133(1), 104393. https://doi.org/10.1016/j.compbiomed.2021.104393

Öner, İ., Yeşilyurt, M. ve Yılmaz, E. (2017). Wavelet analiz tekniği ve uygulama alanları . *Ordu Üniversitesi Bilim ve Teknoloji Dergisi*, 7(1) , 42-56. https://dergipark.org.tr/tr/pub/ordubtd/issue/30633/331389

Rahman, A., Asif, R., Sultan, K., Alsaif, S., Abbas, S., Khan, M. ve Mosavi, A. (2022). ECG Classification for Detecting ECG Arrhythmia Empowered with Deep Learning Approaches. *Computational Intelligence and Neuroscience*, Sayı. 2022, Article 6852845, 12 sayfa, 2022. https://doi.org/10.1155/2022/6852845

Qiu, X., Liang, S. ve Zhang, Y. (2020). Lauw, Hady W., Wong, Raymond Chi-Wing and Ntoulas, Alexandros, Lim, Ee-Peng, Ng, See-Kiong ve Pan, Sinno Jialin Simultaneous (Ed.), *Advances in Knowledge Discovery and Data Mining,* içinde (371-383) Springer International Publishing. https://doi.org/10.1007/978-3-030-47436-2_28

Sangaiah A., Arumugam, M. ve Bian, G. (2019). An intelligent learning approach for improving ECG signal classification and arrhythmia analysis. *Artificial Intelligence in Medicine,* 103(1), 101788. https://doi.org/10.1016/j.artmed.2019.101788

Shobanadevi, A. ve Veeramakali, T. (2023). *Classification and interpretation of ecg arrhythmia through deep learning techniques*. Research Square, https://doi.org/10.21203/rs.3.rs-2935678/v1

Smisek, R. (2016). *ECG Signal Classification Based on SVM* [Doktora Tezi] Brno University of Technology

Wu, W., Huang, Y. ve Wu, X. (2023). SRT: Improved transformer-based model for classification of 2D heartbeat images, *Biomedical Signal Processing and Control,* 105017. https://doi.org/10.1016/j.bspc.2023.105017.

Xie, C. (2020). Celi, Leo Anthony, Majumder, Maimuna S., Ordóñez, Patricia, Osorio, Juan Sebastian, Paik, Kenneth E. ve Somai, Melek (Ed.), *Leveraging Data Science for Global Health* içinde (285-303). Springer International Publishing. https://doi.org/10.1007/978-3-030-47994-7_17

Yakut, Ö., ve Doğru Bolat, E. (2022). A high-performance arrhythmic heartbeat classification using ensemble learning method and PSD based feature extraction approach. *Biocybernetics and Biomedical Engineering*, 42(2), 667-680. https://doi.org/10.1016/j.bbe.2022.05.004.

Yan, G., Liang, S., Zhang, Y. ve Liu, F. (2019). Fusing Transformer Model with Temporal Features for ECG Heartbeat Classification. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 898-905. https://doi.org/10.1109/BIBM47256.2019.8983326

Yüce, H. ve Özkan, A. (2021). Normalizasyon Yöntemlerinin Biyomedikal Verilerde Sınıflandırma Performansına Etkisi. *Avrupa Bilim ve Teknoloji Dergisi*, Ejosat Özel Sayı 2021 (ICCEES), 35-43. https://doi.org/10.31590/ejosat.1011723