

**T.C.
AKDENİZ ÜNİVERSİTESİ**



**TINYML TABANLI GÖMÜLÜ SİSTEM İLE EKG SİNYALİNDEN KARDİYAK
ARİTMİLERİN TESPİTİ**

Doğan Can ÖZBEY

**FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI
YÜKSEK LİSANS TEZİ**

ARALIK 2024

ANTALYA

**T.C.
AKDENİZ ÜNİVERSİTESİ**



**TINYML TABANLI GÖMÜLÜ SİSTEM İLE EKG SİNYALİNDEN KARDİYAK
ARİTMİLERİN TESPİTİ**

Doğan Can ÖZBEY

**FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI
YÜKSEK LİSANS TEZİ**

ARALIK 2024

ANTALYA

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TINYML TABANLI GÖMÜLÜ SİSTEM İLE EKG SİNYALİNDEN KARDİYAK
ARİTMİLERİN TESPİTİ

Doğan Can ÖZBEY
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI
YÜKSEK LİSANS TEZİ

Bu tez 23/12/2024 tarihinde jüri tarafından Oybirliği ile kabul edilmiştir.

Dr. Öğr. Üyesi Yalçın ALBAYRAK (Danışman)

Prof. Dr. Selçuk HELHEL

Prof. Dr. Emine TÜRKMEN

ÖZET

TINYML TABANLI GÖMÜLÜ SİSTEM İLE EKG SİNYALİNDEN KARDİYAK ARİTMİLERİN TESPİTİ

Doğan Can ÖZBEY

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr.Üyesi Yalçın ALBAYRAK

Aralık 2024; 63 sayfa

Bu çalışmada, bir dizi niceleme, özellik çıkarımı ve derin öğrenme algoritmaları kullanılarak STM32H750B kartı üzerinde TinyML ile EKG (elektrokardiyogram) aritmi sınıflandırması yapılmıştır. MIT-BIH Aritmi veri tabanından segmentasyon sonucu elde edilen EKG sinyallerine dalgacık dönüşümü uygulanmış ve elde edilen özellikler kullanılarak LSTM ağının eğitimi gerçekleştirilmiştir. Verinin ön işlenmesi ve modelin geliştirilmesi için MATLAB programı kullanılmıştır. Zaman serileri için tasarlanmış olan LSTM modeli ve sinyalin zaman ve frekans bilgisini aynı anda içeren Morlet dalgacığı fonksiyonunun birlikte kullanımı, ritmik ve ritmik olmayan EKG sinyallerindeki özelliklerin başarılı bir şekilde öğrenilmesini sağlamıştır. Elde edilen modelin kaynakları kısıtlı olan bir cihaz üzerinde (STM32H750B-DK) çalıştırılmasını sağlayacak optimizasyonlar için TinyML teknolojisi kullanılmıştır.

Morlet dalgacık dönüşümü hesaplamalarının gömülü sistem üzerinde de yapılabilmesi için MATLAB CODER eklentisi kullanılarak, MATLAB kütüphanesi içerisinde tanımlı dalgacık dönüşümü algoritması, STM32H750B-DK geliştirme kartı için yeniden oluşturulmuştur. Geliştirme kartının ihtiyaç duyduğu hafızayı düşürecek bir dizi yazılım optimizasyondan sonra, Morlet dalgacık dönüşüm hesapları ve yapay zeka algoritmalarının STM32H750B-DK geliştirme kartı üzerinde birlikte çalıştırılmıştır.

Çalışmanın temel hedefi, EKG verisinde bulunan aritmileri TinyML desteği ile yüksek doğrulukta sınıflandıran, gömülü sistemler üzerinden çalışan bir model geliştirmektedir. Çalışma sonunda gömülü sistem üzerinde çalışan modelin doğruluğu tatmin edici seviyelere ulaşmıştır. Bu çalışma, TinyML'in yalnızca fizyolojik sinyallerde değil, aynı zamanda karmaşık verilerin kaynak kısıtlı cihazlar üzerinde analiz edilmesindeki potansiyelini göstermektedir.

ANAHTAR KELİMELER: Dalgacık Dönüşümü, EKG, Gömülü Sistemler, LSTM, TinyML

JÜRİ: Dr. Öğr. Üyesi Yalçın ALBAYRAK

Prof. Dr. Selçuk HELHEL

Prof. Dr. Emine TÜRKMEN

ABSTRACT

DETECTION OF ARRHYTHMIAS FROM ECG SIGNAL USING TINYML-BASED EMBEDDED SYSTEM

Doğın Can ÖZBEY

MSc Thesis in Department of Electric-Electronic Engineering

Supervisor: Asst. Prof. Dr Yalcin ALBAYRAK

December 2024; 63 pages

In this study, arrhythmia classification of ECG (electrocardiogram) signals using TinyML was performed on the STM32H750B board, utilizing a series of quantization, feature extraction, and deep learning algorithms. Wavelet transform was applied to the ECG signals obtained through segmentation from the MIT-BIH Arrhythmia database, and the extracted features were used to train an LSTM network. MATLAB was used for data preprocessing and model development. The combination of the LSTM model, designed for time series, and the Morlet wavelet function, which contains both time and frequency information of the signal, enabled successful learning of features in both rhythmic and non-rhythmic ECG signals. TinyML technology was employed to optimize the model for execution on a resource-constrained device (STM32H750B-DK).

To perform the Morlet wavelet transform calculations on the embedded system, the MATLAB CODER extension was used to recreate the wavelet transform algorithm, which is predefined in the MATLAB library, for the STM32H750B-DK development board. After a series of software optimizations to reduce the memory requirements of the development board, Morlet wavelet transform calculations and machine learning algorithms were successfully executed together on the STM32H750B-DK development board.

The primary goal of this study is to develop a model that classifies arrhythmias in ECG data with high accuracy using TinyML on embedded systems. By the end of the study, the accuracy of the model running on the embedded system reached satisfactory levels. This study demonstrates the potential of TinyML not only for physiological signals but also for analyzing complex data on resource-constrained devices.

KEYWORDS: ECG, Embedded Systems, LSTM, TinyML, Wavelet Transform

COMMITTEE: Asst. Prof. Dr. Üyesi Yalçın ALBAYRAK

Prof. Dr. Selçuk HELHEL

Prof. Dr. Emine TÜRKMEN

ÖNSÖZ

Bu tezin hazırlanmasında bilgi ve deneyimleriyle bana yol gösteren, her aşamada destek ve sabrını esirgemeyen değerli hocam Dr. Öğr. Üyesi Yalçın ALBAYRAK'a en içten teşekkürlerimi sunarım. Kendisi, yönlendirmeleriyle çalışmamın bilimsel doğruluk ve tutarlılık içinde ilerlemesini sağlamış, aynı zamanda bana araştırma süreçlerinde disiplin ve kararlılığın önemini göstermiştir.

Bu çalışmayı gerçekleştirirken her zaman yanımda olan, beni koşulsuz sevgiyle destekleyen ve her adımında bana güç veren sevgili annem Nilgün ÖZBEY'e ve babam Doğan ÖZBEY'e sonsuz teşekkürlerimi sunuyorum. Verdikleri emek, sabır ve özveri sayesinde bugünlere ulaşabildim. Hayatım boyunca bana inandıkları, hayallerimin peşinden gitmem için cesaret verdikleri ve her zaman yanımda oldukları için onlara minnettarım. Bu tez, onların sevgisi ve desteği olmadan mümkün olamazdı.

Bu tez, yoğun bir emek ve kararlılık sonucunda ortaya çıkmıştır ve umuyorum ki bilimsel bilgi birikimine anlamlı bir katkı sağlar.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
AKADEMİK BEYAN	vi
SİMGELER VE KISALTMALAR.....	vii
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	xi
1. GİRİŞ	1
2. KAYNAK TARAMASI	6
2.1 EKG Sinyalinin Ön İşlenmesi.....	6
2.2 Makine Öğrenmesi ve EKG	7
2.3 TinyML ve Gömülü Sistemlerde Yapay Zeka Çalışmaları.....	8
2.4 Gömülü Sistemlerde Makine Öğrenmesi ile EKG Sinyali Sınıflandırması.....	11
3. MATERYAL VE METOT	13
3.1 Kalbin Çalışması ve EKG Sinyali.....	13
3.2 Özellik Çıkarımı ve Dönüşüm Teknikleri.....	13
3.2.1 Segmentasyon analizi.....	14
3.2.2 Dalgacık dönüşümü ile öznelilik çıkarımı	14
3.3 MIT-BIH Aritmi Veri Seti	16
3.4 Verinin Ön İşlenmesi	19
3.4.1 Verilerin segmentasyonu ve etiketlenmesi	20
3.4.2 Verilerin alt örneklenmesi.....	21
3.4.3 Morlet dalgacık dönüşümü.....	23
3.4.4 Veri setinde bulunan aritmi sınıfları	24
3.4.5 Veri kümesinin ayrılması	25
3.5 TinyML	26
3.5.1 TinyML tanımı.....	26

3.5.2 TinyML'in tarihçesi	27
3.5.3 TinyML'nin temel özellikleri	28
3.5.4 TinyML'nin yazılım araçları.....	29
3.5.5 TinyML'in zorlukları ve sınırlamaları	30
3.5.6 TinyML'in çalışmamızdaki rolü ve önemi	31
3.6 Yapay Zeka Modelinin Oluşturulması.....	31
3.7 Model Dönüşümü ve Niceleme.....	33
3.8 STM32H750B-DK Tabanlı Platform.....	34
4. BULGULAR VE TARTIŞMA	41
4.1 Alt Örnekleme Sonuçları.....	41
4.2 MATLAB Programında Dalgacık Dönüşümü Sonuçları.....	42
4.3 Model Sonuçları	47
6. SONUÇLAR	57
7. KAYNAKLAR	59
ÖZGEÇMİŞ	

AKADEMİK BEYAN

Yüksek Lisans Tezi olarak sunduğum “TINYML TABANLI GÖMÜLÜ SİSTEM İLE EKG SİNYALİNDEN KARDİYAK ARİTMİLERİN TESPİTİ ” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

23/12/2024

Doğan Can ÖZBEY

SİMGELER VE KISALTMALAR

Simgeler

t : Zaman

Hz : Hertz

Kisaltmalar

AI : Yapay zeka (Artificial Intelligence)

AAMI : Amerikan Akredite Tıbbi Cihazlar Enstitüsü (American Association of Medical Instrumentation)

API : Uygulama programlama arayüzü (Application Programming Interface)

BVP : Kan hacmi pulsasyonu (Blood Volume Pulse)

BLE : Düşük enerji Bluetooth (Bluetooth Low Energy)

BP : Geri Yayılım (Back Propagation)

CNN : Konvolüsyonel sinir ağı (Convolutional Neural Network)

EDA : Elektrodermal Aktivite (Elektrodermal Activity)

EKG : Elektrokardiyogram (Electrocardiogram)

EEG : Elektroensefalogram (Electroencephalogram)

EMG : Elektromiyogram (Electromyogram)

GPU : Grafik işlem birimi (Graphics Processing Unit)

HFpEF : Diyastolik Kalp Yetmezliği (Heart Failure with Preserved Ejection Fraction)

IoT : Nesnelerin interneti (Internet of Things)

LSTM : Uzun-kısa vadeli bellek (Long Short-Term Memory)

MIT-BIH : Massachusetts Institute of Technology - Beth Israel Hospital EKG veri seti (MIT-BIH Arrhythmia Database)

MR : Manyetik rezonans (Magnetic Resonance)

PTB : EKG veri seti (Physikalisch - Technische Bundesanstalt)

PRED. : Tahmin (Prediction)

RAM	: Rastgele erişimli bellek (Random Access Memory)
RNN	: Tekrarlayan sinir ağı (Recurrent Neural Network)
TPU	: Tensör işlem birimi (Tensor Processing Unit)
TFLM	: TensorFlow Lite for Microcontrollers
TinyML	: Mikrodenetleyicilerde makine öğrenmesi (Tiny Machine Learning)
UART	: Evrensel asenkron alıcı-verici (Universal Asynchronous Receiver-Transmitter)



ŞEKİLLER DİZİNİ

Şekil 1.1. Yıllara göre kalp hastalıklarına bağlı toplam ölüm (milyon)	1
Şekil 1.2. Normal EKG dalga formu	2
Şekil 1.3. Çalışmanın İş Akışı	4
Şekil 2.1. Ön işleme teknikleri ve modellerin performans yönünden karşılaştırması	9
Şekil 2.2. Modellerin ve niceleme yöntemlerinin boyut yönünden karşılaştırması.....	9
Şekil 2.3. Niceleme yöntemlerinin toplama ve çıkarma işlemleri esnasında harcadığı enerji ve model doğruluklarının karşılaştırılması.....	11
Şekil 3.1. Physiobank Arayüzü	19
Şekil 3.2. Bir EKG sinyalinin frekans bileşenleri	22
Şekil 3.3. a) Uzunluk Eşitleme Öncesi R-R İntervali; b) Uzunluk eşitleme sonrası R-R intervali	22
Şekil 3.4. Veri setinin aritmi tiplerine göre dağılımı	25
Şekil 3.5. Niceleme seviyesinin model performansına etkisi	28
Şekil 3.6. Model niceleme akış diyagramı.....	34
Şekil 3.7. STM32H750B-DK Geliştirme Kartı	35
Şekil 3.8. Nextion Editor Arayüzü.....	36
Şekil 3.9. MATLAB Coder Arayüzü.....	37
Şekil 3.10. X-CUBE-AI Arayüzü	39
Şekil 4.1. a) “Normal Atım” etiketli R-R aralığı; b) Skalogram	43
Şekil 4.2. a) “Atriyal Erken Atım” etiketli R-R aralığı; b) Skalogram	43
Şekil 4.3. a) “Bozulmuş Atriyal Erken Atım” etiketli R-R aralığı; b) Skalogram	44
Şekil 4.4. a) “Nodal Kaçış Atımı” etiketli R-R aralığı; b) Skalogram	44
Şekil 4.5. a) “Nodal Erken Atım” etiketli R-R aralığı; b) Skalogram.....	44
Şekil 4.6. a) “Ventriküler Kaçış Atımı” etiketli R-R aralığı; b) Skalogram.....	45
Şekil 4.7. a) “Erken Ventriküler Kasılma” etiketli R-R aralığı; b) Skalogram	45

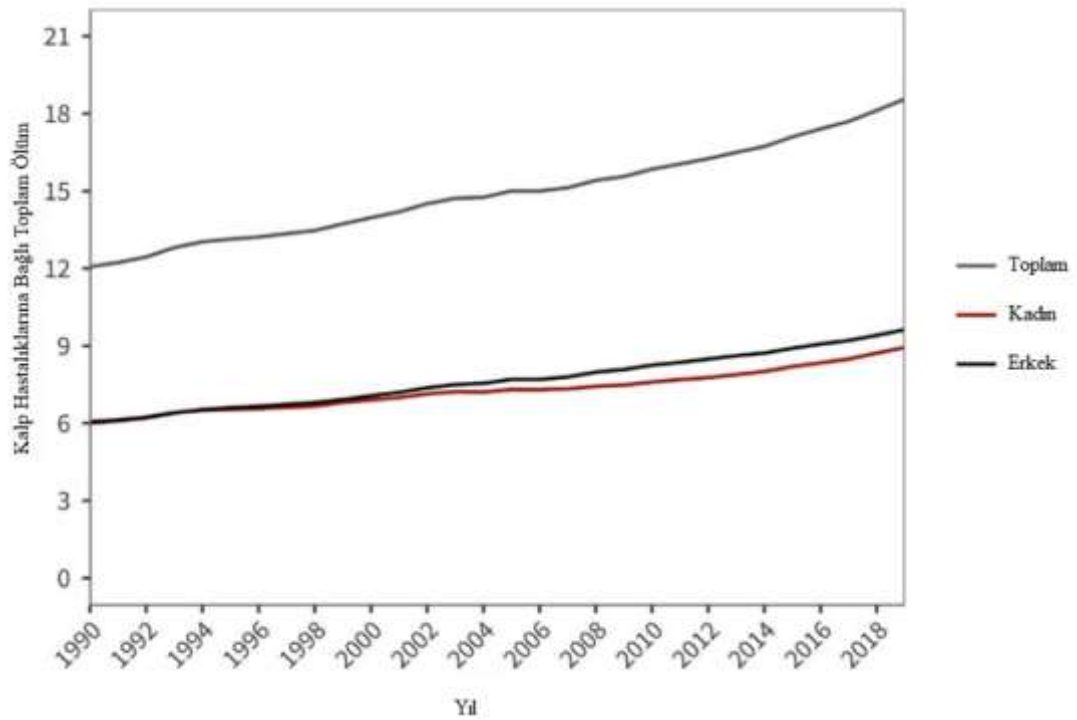
Şekil 4.8. a) “Sol Dal Bloğu Atımı” etiketli R-R aralığı; b) Skalogram	45
Şekil 4.9. a) “Sağ Dal Bloğu Atımı” etiketli R-R aralığı; b) Skalogram	46
Şekil 4.10. a) “Ventriküler ve Normal Atımın Birleşimi” etiketli R-R aralığı; b) Skalogram.....	46
Şekil 4.11. a) “Kalp Pili ve Normal Atımın Birleşimi” etiketli R-R aralığı; b) Skalogram.....	46
Şekil 4.12. LSTM Modeli Eğitim Süreci ve Performans Grafikleri	48
Şekil 4.13. CNN Modeli Eğitim Süreci ve Performans Grafikleri	49
Şekil 4.14. a) CNN Modeli Katman Yapısı b) LSTM Modeli Katman Yapısı	50
Şekil 4.15. a) Optimizasyon önce modellerin boyut karşılaştırması; b) Optimizasyon sonrası modellerin boyut karşılaştırması; c) Optimizasyon önce modellerin doğruluk karşılaştırması; d) Optimizasyon sonrası modellerin doğruluk karşılaştırması	53
Şekil 4.16. a) Yapılan çıkarımın MATLAB Programındaki Sonuçları (Yüzde olarak); b) Yapılan çıkarımın STM32H750B Geliştirme Kartı üzerindeki Sonuçları (Yüzde olarak)	54
Şekil 4.17. HMI Ekran Tasarımı.....	55
Şekil 4.18. HMI ekran üzerinde örnek analiz sonucu	56

ÇİZELGELER DİZİNİ

Çizelge 2.1. MIT-BIH veri Seti Sınıfları ve AAMI Sınıflarının Karşılaştırılması	12
Çizelge 3.1. Fourier Dönüşümü ve Dalgacık Dönüşümünün Karşılaştırılması.....	15
Çizelge 3.2. MIT-BIH Aritmi Verisetinde Bulunan Etiketler ve Açıklamaları.....	18
Çizelge 3.3. MIT-BIH Veri Seti Anotasyon Dosyası Örneği	20
Çizelge 4.1. EKG sinyali uzunluğu ve dalgacık dönüşümü sonucunun karşılaştırılması	42
Çizelge 4.2. MATLAB programında dalgacık dönüşümü sonuç matrisi	47
Çizelge 4.3. Optimizasyon öncesi ve sonrasında performans parametrelerinin karşılaştırılması	53

1. GİRİŞ

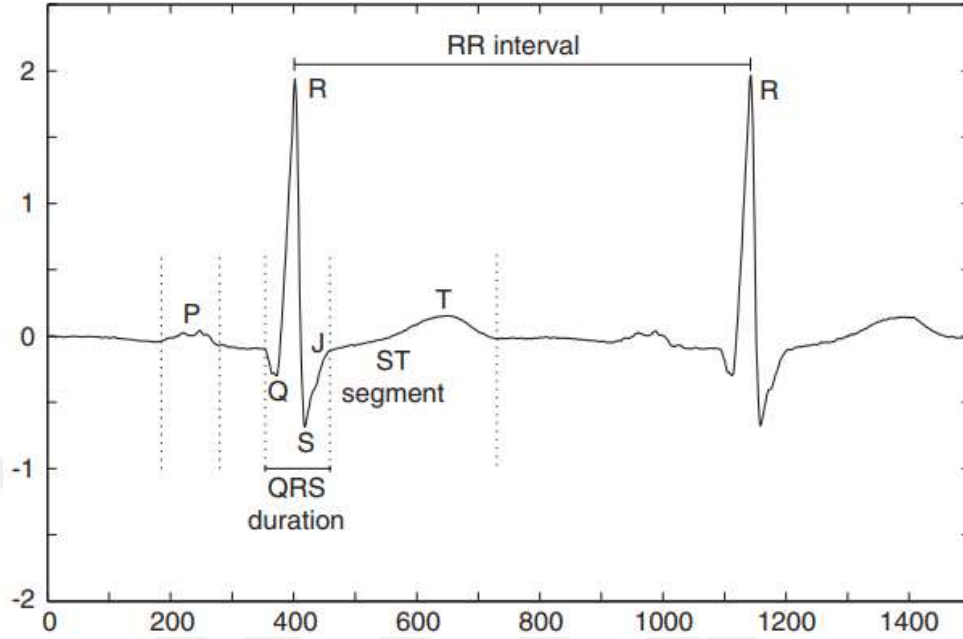
Dünya Sağlık Örgütü'ne göre, kardiyovasküler hastalıklar Dünya genelinden ölümlerin başlıca sebebidir. 2019 yılına ait verilere göre, kardiyovasküler hastalıklar 17.9 milyon kişinin ölümüne sebep olmuştur. Bu, Dünya'daki tüm ölümlerin %32'sini oluşturmaktadır. Erken ölüm olarak kabul edilen 70 yaş altındaki ölümlerin %39'u kardiyovasküler hastalıklar sebeplidir. Dünya Sağlık Örgütü, kardiyovasküler hastalıkların olabilecek en erken aşamada tespit edilmesinin çok önemli olduğunu belirtmektedir. Ancak geleneksel teşhis yöntemleri pahalı ve karmaşıktır ve ulaşılması özellikle toplumun bazı kesimleri için zor olabilir (Anonymous 1). Şekil 1.1'de Dünya Kalp Dederasyonu 2023 raporuna göre, 1990 yılında 12 milyon olan kalp hastalıkları ile ilgili ölüm sayısı, 2023 yılında 18 milyonu aşmıştır (Anonymous 2).



Şekil 1.1. Yıllara göre kalp hastalıklarına bağlı toplam ölüm (milyon) (Anonymous 2)

İnsan kalbi, bir insan ömrü boyunca yaklaşık olarak 2.5 milyar kez atar (Park ve Fishman 2011). Bu atımlar, sinoatrial düğümün ürettiği otomatik sinyaller ile düzenlenir, bu periyodik uyarımlar kardiyak döngü olarak isimlendirilir (Joung ve Chen 2015). Her bir atım kardiyak iletim sistemi hücreleri tarafından kalp kaslarına iletilir. Kalp kaslarının aldıkları sinyaller sonucunda verdikleri tepki, elektrokardiyogram (EKG) sinyali olarak kaydedilir ve her bir kardiyak döngü boyunca belirgin fazlar oluşturur. Bu fazlar P dalgası, QRS kompleksi, ST segmenti ve T dalgası olarak adlandırılır (Chutchavong 2019). Elektrokardiyogram, son yüzyılın kardiyovasküler tarama, kayıt, değerlendirme ve tanı koymada temel araçlardan biridir (Rafie vd. 2021). 12 kanallı elektrokardiyogram, kardiyovasküler tanı yöntemleri arasında en sık

kullanılan yöntemdir ve Dünya çapında yılda 200 milyon EKG kaydedilmektedir (Reichlin vd. 2016). Şekil 1.2’de normal bir EKG sinyalinin fazları bulunmaktadır.



Şekil 1.2. Normal EKG dalga formu (Sörnmo ve Laguna, 2006)

Sinoatrial düğümde ya da iletim yollarındaki bir problem kalp sinyalinin olması gereken şekilde kaslara ulaşamamasına sebep olabilir, bu durum aritmi olarak isimlendirilir. Kalp aritmisi, kalpte anormal elektriksel aktivite ile karakterize edilen durumlara verilen genel tanımdır (Humphreys vd. 2011). Kardiyak aritmilerin büyük kısmı, miyokardiyal hastalıklar sonucunda ya da çeşitli genetik ve çevresel faktörlere bağlı olarak ortaya çıkar (Kingma vd. 2023). Kalp aritmileri erken teşhis edilmediği durumlarda ciddi sağlık problemlerine yol açabilir, hatta ölüme sebep olabilir. Bir aritminin tespiti genellikle elektrokardiyogram (EKG) ile yapılır.

Yapay zeka, insan zekasının taklit edilmesiyle öğrenme, problem çözme ve karar verme gibi görevleri gerçekleştiren algoritmalar bütünü olarak tanımlanabilir. Yapay zeka, teknolojinin ilerlemesi ile birlikte hayatımıza her alanda dahil olan bir kavram olmuştur. Eğitimden ulaşma, iş dünyasında sanat ve yaratıcılığa kadar bir çok alanda kendini göstermektedir. Sağlık alanında teşhis ve tedavi süreçlerinde de devrim yaratmıştır. Görüntü işleme yeteneği ile MR ve tomografide, sinyal işleme yeteneği ile fizyolojik sinyallerin analizinde yaygın olarak kullanılmaktadır.

Literatürde, EKG sinyalinin tahmininde yapay zeka yaklaşımını benimseyen bir çok çalışma bulunmaktadır. Bu çalışmalar özellikle kalp ritim bozukluklarının erken teşhisi için yapay zeka teknolojilerinin etkinliğini incelemektedir. Özellikle büyük veri setleriyle çalışırken, hızlı ve güvenilir sonuçlar sunabilmektedir. Bu, zamanında müdahale edilmesi gereken durumlar için kritik bir avantaj sunabilir. Ayrıca bu kapsamda, hekimler ve hemşireler başta olmak üzere sağlık profesyonellerinin

zamanında ve etkili girişimde bulunabilmesi için yapay zeka algoritmalarının yardımı ile karar destek mekanizmaları oluşturulabilir. Bu şekilde, yapay zeka teknolojileri sadece hastanın sağlığını korumak ve iyileştirmekle kalmaz, aynı zamanda sağlık çalışanlarının iş yükünü hafifletebilir. Ancak büyük ölçekli yapay zeka modellerin EKG sinyali tahmininde kullanımının avantajları yanında bazı sınırlamaları ve dezavantajları da bulunmaktadır. Bu tür büyük yapay zeka modelleri genellikle yüksek hesaplama gücü, hafıza ve enerjiye ihtiyaç duyarlar. Bu durum donanım maliyetini yükseltir ve sistemi bir bulut sunucuya bağımlı hale getirebilir. Yapay zeka modellerinin medikal cihazlarda yaygın bir şekilde kullanılabilmesi için maliyet, sunucuya bağımlılık, enerji gibi problemlerin çözülmesi gerekmektedir.

Kalp sinyalinin gerçek zamanlı olarak takip eden ve aritmi tespiti yapan bir cihazın bir sunucuya bağımlı olmaması, düşük enerji tüketmesi, taşınabilir olması ve kesintisiz çalışması önemli avantajlar sağlar. Sunucuya bağımlı olmaması toplanan verilerin yerel olarak işlenmesi ve veri gizliliğini sağlar, dahası sunucu kaynaklı gecikmeleri azaltır. Ayrıca bu tür cihazların hesaplamalarını sunucu bağımsız olarak yapması, geliştirilen sistemin bağlantı probleminin yaşandığı bölgelerde etkili bir biçimde kullanılmasını olanaklı kılar. Düşük enerji tüketimi cihazın taşınabilir olmasını sağlayarak kullanıcının cihazı rahatlıkla kullanmasını mümkün kılar. Tüm bu özellikler hasta sağlığını korumakla kalmaz, aynı zamanda sağlık sisteminin verimliliğine ve etkinliğine katkıda bulunur.

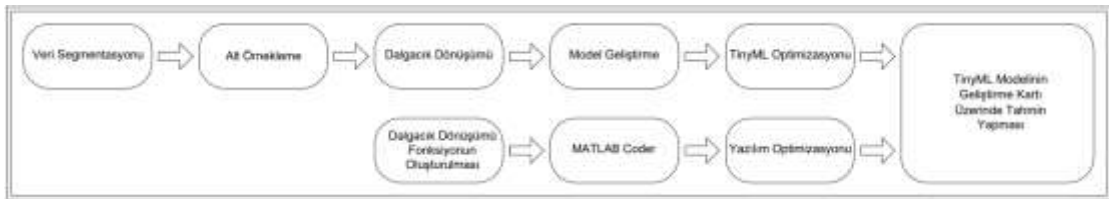
Bu çalışmada, geleneksel teşhis yöntemlerinin ve büyük yapay zeka modellerinin sebep olduğu problemlere alternatif olarak, TinyML tabanlı derin öğrenme algoritmaları kullanarak gömülü sistemler üzerinde aritminin tespit edilmesine yönelik bir çözüm önerilmiştir. Çalışma, TinyML'in sağladığı daha verimli modellerin oluşturulması ile düşük güç tüketimli, taşınabilir, sunucu bağımsız sistemlerin potansiyelinin incelenmesini ve bu sistemlerin aritmi tespiti konusunda nasıl kullanılacağını araştırmayı amaçlamaktadır.

TinyML, yapay zeka modelinin uygun boyuta sıkıştırılması ile kaynak kısıtlı uç cihazlarda çalıştırılmasına olanak sağlayan yazılım ve donanım paketleri olarak ifade edilebilir (Saha vd. 2022). Makine öğrenmesi modellerinin doğrudan uç cihaz üzerinde çalışmasını sağlayarak, bulut sunuculara olan bağımlılığı azaltır (Banbury vd. 2020). Ayrıca düşük gecikme süresi, gizlilik ve enerji verimliliği sunar. Bu teknoloji özellikle sürekli internet bağlantısının olmadığı ya da kısıtlı bant genişliğine sahip olan bölgeler için kritik öneme sahiptir. TinyML'in temelinde, yapay zeka modellerinin boyut ve hesaplama gereksinimlerini azaltan niceleme (quantization) ve budama (pruning) işlemleri yer almaktadır (Capogrosso vd. 2023). Bir TinyML algoritmasının sonucu, model eğitimi öncesinde yapılan optimizasyon işlemleri ve TinyML dönüşümü esnasında yapılan işlemlere bağlıdır. Önceki çalışmaların gösterdiği biçimde, TinyML ile dönüştürülen bir modelin performansının, küçültme oranına bağlı olarak, düşüşü kaçınılmazdır (Lane vd. 2017). Ancak verim odaklı düşünüldüğünde TinyML, kaynak

sınırlı sistemlerde yapay zeka algoritmalarının çalıştırılmasını mümkün kılan devrim niteliğinde bir teknolojidir. TinyML, makine öğrenmesi ve gömülü sistemlerin ortak kümesi olarak düşünülebilir. TinyML yeni bir alan olmasına rağmen, büyüyen, gelişen ve potansiyeli olan bir alandır. Literatür incelendiğinde, TinyML'in potansiyelinden tam olarak faydalanabilmek için özellikle karmaşık modellerin doğruluğunu artırmaya yönelik optimizasyon yöntemlerinin geliştirilmesine devam edilmesi gerektiği düşünülmektedir. Bu geliştirme süreçlerinin, TinyML'in hem akademik hem de endüstriyel düzeyde daha geniş alanlarda kullanılmasını sağlayacaktır.

Modelin geliştirilmesi sürecinde gürültüden arındırılmış MIT-BIH Aritmi Veri Tabanı kullanılmıştır. Bu sebeple çalışma sürecinde gürültü gidermeye yönelik bir yöntem kullanılmamıştır. MIT-BIH Aritmi Veri Tabanı, PhysioNet tarafından herkese açık olarak paylaşılmaktadır (Moody ve Mark 2001). MIT-BIH veri seti, literatürde kabul görmesi nedeniyle tercih edilmiştir, aritmi üzerine yapılan çalışmalarda bir standart olmakla birlikte bu veri seti Dünya'da 500'in üzerindeki araştırma merkezinde kardiyak ritimlerin araştırılmasında kullanılmaktadır (Moody ve Mark 2001).

Şekil 1.3'te bu çalışmanın iş akışındaki ana başlıklar belirtmiştir. Çalışma, bir dizi ön işleme, özellik çıkarımı ve optimizasyon süreçlerinin birlikte kullanılmasıyla, MATLAB programı üzerinde eğitilmiş bir modelin STM32H750B geliştirme kartı üzerinde çalıştırılmasının sağlanması olarak özetlenebilir. Şekil 1.3'te gösterildiği üzere, segmentasyonu ve alt örnekleme tamamlanan veri setinin özellik çıkarımı işlemi Morlet Dalgacık Fonksiyonu kullanılarak yapılmıştır. Dalgacık fonksiyonu sonucunda elde edilen özellikler geliştirilerek modelin girdisi olarak kullanılmıştır. Model, dalgacık fonksiyon çıktısı ile eğitildiği için, gömülü cihaz üzerinde yapılarak sınıflandırma tahmininin de dalgacık dönüşümünün çıktıları üzerinden yapılması gerekmektedir. Bu sebeple, MATLAB kütüphanesinde bulunan dalgacık dönüşümü fonksiyonu, MATLAB Coder eklentisi kullanılarak, STM32H750B geliştirme kartı için optimize bir şekilde yeniden oluşturulmuştur. Çalışma sonucunda, MIT-BIH Aritmi Veri Tabanı'ndan alınmış ve üzerinde herhangi bir işlem yapılmamış ham veriler ile STM32H750B geliştirme kartı üzerinde %88 ile doğruluğunda sınıflandırma işlemi yapılmıştır.



Şekil 1.3. Çalışmanın İş Akışı

Bilindiği kadarıyla bu çalışma, dalgacık dönüşümü ve derin öğrenme algoritmalarını kaynak kısıtlı bir cihaz üzerinde birlikte çalıştıran ilk çalışmadır. Literatürde çeşitli aritmi veri setleri kullanılarak gerçekleştirilen bir çok yapay zeka çalışması olsa da, bu çalışma gömülü sistem üzerinde, geliştirilen modelin 11 farklı alt sınıfta sınıflandırma yapabildiği ilk çalışmadır. Modelin bu derece verimli olmasının zaman serilerinde kullanılan LSTM katmanı ve sinyalin zaman ve frekans bilgisini aynı anda içeren Morlet dalgacık fonksiyonunun birlikte kullanımından kaynaklanmaktadır.

Çalışma, başarıyla tamamlanmış olup, TinyML teknolojisinin gömülü sistemlerde, özellikle karmaşık sinyaller ile, uygulanabilirliğini göstermiştir. Çalışma teorik bilgiler sunmanın ötesinde, düşük maliyetli ve kısıtlı bir cihaz üzerinde TinyML'in ek optimizasyon yöntemleri kullanılarak ne denli etkili olabileceğini uygulamalı olarak göstermiştir, bu yönü ile gelecek bu alanda yapılacak çalışmalara referans oluşturacaktır.

2. KAYNAK TARAMASI

Literatürde ve endüstride, yapay zeka algoritmaları sağlık teknolojileri alanında sıklıkla kullanılmaktadır. Bu kapsamda, yapay zeka fizyolojik sinyallerin analizi ve sınıflandırılması gibi karmaşık görevleri üstlenmektedir. Literatürde çeşitli makine öğrenimi algoritmalarının kullanılmasıyla EKG verileri üzerinde oldukça başarılı sonuçlar elde edilmiş bir çok çalışma bulunmaktadır. Ancak literatürde TinyML ile kaynak kısıtlı sistemler üzerinde makine öğrenmesi algoritmalarının çalıştırılmasına ve özellikle EKG sinyalinin TinyML ile kaynak kısıtlı sistemler üzerinde analiz edilmesine yönelik oldukça sınırlı sayıda çalışma bulunmaktadır. Bu durum, kısıtlı sistemler üzerinde çalıştırılan yapay zeka algoritmalarının potansiyelinin tam olarak anlaşılamamış olmasına ya da gömülü sistemlerin limitlerinden kaynaklanıyor olabilir. Bu çalışmanın amaçlarından biri, literatürde sözü edilen boşluğu doldurarak, TinyML'in EKG sinyali analizinde etkin bir araç olduğunu göstermektir.

Kaynak taraması, bu çalışmanın temellini oluşturan dört alt başlık altında incelenmiştir.

2.1 EKG Sinyalinin Ön İşlenmesi

Safdar vd. (2024), çalışmasında 2012-2022 yılları arasında yapay zeka yöntemleri ile EKG analizi üzerine yapılan çalışmaları incelemiştir. İncelenen çalışmalar yazılım, veri seti kaynağı, kullanılan ön işleme tekniklerine göre ayrılmışlardır. İncelemeler sonucunda MIT-BIH veri setinin literatürde en çok kullanılan veri seti olduğu vurgulanmıştır. İncelenen çalışmaların 69 tanesinde evrişimsel sinir ağı (CNN), 14 tanesinde uzun kısa süreli bellek (LSTM) mimarisi kullanılmıştır. Veri artırma yöntemi EKG analizi çalışmalarında en çok kullanılan yöntem olmuştur. Dalgacık dönüşümünü de içeren spektogram yöntemi, veri artırma yönteminden sonra en çok kullanılan ikinci yöntem olmuştur.

Salimi vd. (2023), ön işlemeyi, verilen üzerinde yapılan, performansı artırmak ya da eğitim yükünü azaltmak için yapılan her türlü basitleştirme ya da dönüştürme işlemi olarak tanımlanmıştır. EKG verilerine en çok uygulanan ön işleme yöntemlerini belirlemek için PhysioNet Challenge 2020 incelenmiştir. Bu yöntemler, bant geçiren filtreleme, yeniden örnekleme (downsampling), yeniden boyutlandırma ve normalizasyondur. Çalışma kapsamında, farklı ön işleme tekniklerinin model performansı üzerindeki etkisinin incelenmesi için çeşitli veri setleri bu ön işleme yöntemleri uygulanmış ve performans sonuçları değerlendirilmiştir. Performansların değerlendirilmesinde Wilcoxon İşaretli Sıra Testi ve Kruskal-Wallis testi kullanılmıştır. Testler sonucunda, örnekleme frekansı 500 Hz olan bir sinyalin yeniden örnekleme ile 250 Hz'e düşürülmesi sonucunda önemli bir performans kaybının yaşanmadığı belirtilmiştir.

Pan ve Tompkins (1985), Pan-Tompkins algoritmasını tanıtmıştır. Algoritma, EKG sinyallerindeki QRS kompleksini gerçek zamanlı olarak tespit etmek için kullanılan, etkili ve kabul görmüş bir algoritmadır. Algoritmanın geliştirilmesinde MIT-BIH veri seti kullanılmıştır. QRS kompleksi, kalbin atış hızının tayin edilmesinde ve ritminin analiz edilmesinde kullanılan temel bir parametredir. Algoritma, düşük hesaplama gücü gereksiniminden dolayı kaynak kısıtlı mikrodenetleyici cihazlarda kullanmak için uygundur, bu sebeple günümüzdeki kullanılan bir çok EKG ön işleme tekniğinin temelini oluşturur.

2.2 Makine Öğrenmesi ve EKG

EKG aritmileri, kalp ritim bozukluğu olarak tanımlanır ve erken teşhisi hayati önem taşımaktadır. Son yıllarda makine öğrenmesi algoritmaları, bu aritmilerin sınıflandırılmasında yüksek doğruluk oranları elde etmektedir. Ebrahimi vd. (2020), çalışmasında makine öğrenmesi yöntemlerinin aritmi tespiti ve sınıflandırmasındaki rolünü kapsamlı bir şekilde ele almıştır. Bu çalışmanın amacı, literatürde yer alan aritmi tespit ve sınıflandırmasında makine öğrenmesi algoritmalarının rolünü incelemektedir. Makale, farklı derin öğrenme algoritmalarının aritmi sınıflandırmasına nasıl uygulanması gerektiğini ve performanslarını ele almıştır. Ayrıca aritmi sınıflandırma modellerinin performans metrikleri de çalışma kapsamında tartışılmıştır.

Kwon vd. (2021), çalışmasında son yıllarda kalp yetmezliği ile ilgili hastalıkların erken tanısının makine öğrenmesi algoritmaları tarafından yapılması üzerine birçok araştırma yapıldığını belirtmektedir. Bu bağlamda, korunmuş ejeksiyon fraksiyonlu kalp yetmezliği (HFpEF) gibi hastalıkların erken teşhisi tedavi şansını artırır. Çalışma, EKG verilerinin yanı sıra hastanın yaş, cinsiyet, boy ve kilo bilgilerini de işleyerek daha güvenilir sonuçlar elde etmeyi hedeflemektedir. Çalışma, birden fazla türdeki veri türünü ağırlıklı girişinde kullanarak ensemble sinir ağlarının kalp yetmezliği gibi bir çok değişkene bağlı olan bir durumda etkili olduğunu sonuçlar verdiğini göstermiştir.

Wu vd. (2021) çalışmalarında, MIT-BIH aritmi veri setinde bulunan 5 farklı aritmi tipini sınıflandırmak için makine öğrenmesi algoritması önermektedir. Yazarlar 12 katmanlı bir derin öğrenme algoritması kullanarak dalgacık dönüşümü ve gürültü azaltma teknikleri uyguladıkları EKG sinyallerini sınıflandırmıştır. BP sinir ağı kullanılarak elde edilen sonuç, Random Forest ve diğer CNN ağlarına kıyasla daha yüksek doğruluğa sahiptir. Çalışma, makine öğrenmesi algoritmalarının hastalık teşhisinde kullanılmasının, tıbbi kaynakların daha verimli kullanılması konusunda oldukça etkili olacağını vurgulamıştır.

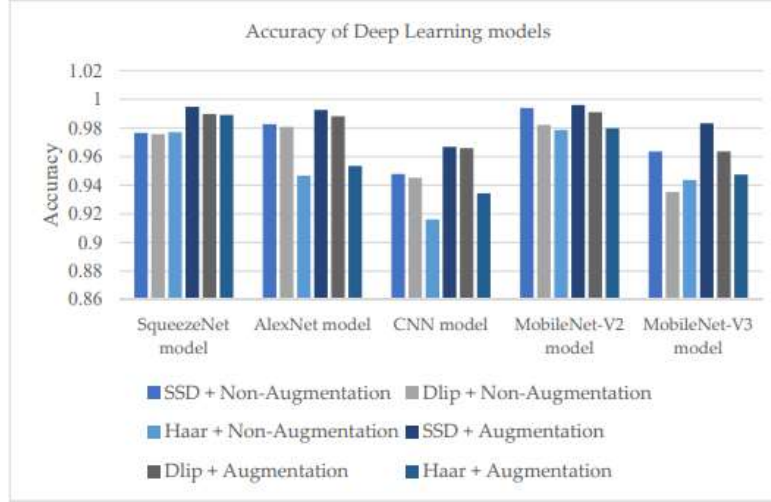
Muzammil vd. (2024), çalışmalarında yapay zekanın kardiyovasküler hastalıkların tanı ve tedavisi için bir teknik haline geldiğini belirtmektedir. Çalışma, kendisinden önce yapılan bir çok çalışmayı incelemiştir. İncelenen bazı yapay zeka ile kardiyovasküler hastalıkların tanı ve tedavisi çalışmalarında %98 üzerinde doğruluğa

ulaşılmıştır. Bir diğer başlığın altında, uzmanlar ile yapay zeka karşılaştırılmıştır. Çalışmalar hipertrofik kardiyomiyopati ölçümlerinde makine öğrenmesi tahminlerinin, insan tahminlerinin doğruluğundan belirtilmiştir. Yapay zekanın ilgili alanda kullanılmasının önünde en önemli engel, siber saldırı tehdidinin bulunması olduğu belirtilmiştir. Çeşitli çalışmalarda yapay zeka destekli EKG yorumlamanın, geleneksel EKG yorumlamaya kıyasla daha yüksek doğruluk oranına sahip olduğu belirtilmiştir. Çalışma, bu tür sonuçların yapay zeka destekli EKG yorumlama sistemlerinin kullanımının daha yaygın hale gelmesi için önemli bir adım olduğunu vurgulamaktadır.

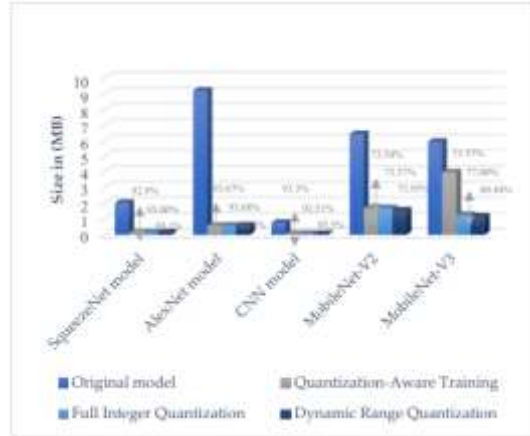
2.3 TinyML ve Gömülü Sistemlerde Yapay Zeka Çalışmaları

Atanane vd. (2023), çalışmalarında binalarda su yönetimini geliştirmek için TinyML tabanlı su kaçak tespit yöntemi önermişlerdir. Çalışmada PVC borulardaki sızıntıların tespitini, laboratuvar ortamında oluşturulan akustik veri seti ile eğitilen model ile gerçekleştirmişlerdir. Toplanan akustik veriler öncelikle skalogram verisine dönüştürülmüş, sonrasında ise transfer öğrenme yöntemi ile 5 farklı CNN (EfficientNet, ResNet, AlexNet, MobileNet V1 ve MobileNet V2) uygulanmıştır. En yüksek doğruluk oranı EfficientNet ile %97.45 olarak elde edilmiştir. Çalışma, TinyML'in yerel mekanizmalarda karar almayı sağlayarak merkezi sistemlere bağıllığı azaltmaktaki potansiyelini vurgulamıştır.

Alajlan ve Ibrahim (2023), TinyML teknolojisini araç sürücülerinde uyuklama tespiti için uygulamışlardır. Çalışma, literatürde bulunan uyuklama tespiti modellerinin 10 MB ile 54 MB arasında değiştiğini ve yüksek boyutundan dolayı bu algoritmaların IoT temelli sistemlerin bir parçası olabildiğini vurgulamıştır. Ancak bulut tabanlı sistemlerdeki gecikme etkili bir tespit sisteminin geliştirilmesinin önünde engel oluşturulmaktadır. Çalışma, sürüş esnasında toplanan kapalı göz ve esmene fotoğraflarını uyuklama olarak kabul etmiştir. Birden fazla veri işleme, model ve niceleme algoritmalarının birlikte kullanılması ile çalışma sonucunda parametre seçiminin TinyML algoritmasının doğruluğu ve boyutu üzerindeki etkisi kapsamlı biçimde belirtilmiştir. Şekil 2.1 ve Şekil 2.2 Alajlan ve Ibrahim (2023)'nın çalışmasının sonuçlarını göstermektedir. Şekil 2.1'de farklı ön işleme teknikleri ve modeller performans yönünden birbirleriyle karşılaştırılmışlardır. Şekil 2.2'de ise niceleme yöntemlerinin model boyutu yönünden modeller üzerindeki etkisi paylaşılmıştır.



Şekil 2.1. Ön işleme teknikleri ve modellerin performans yönünden karşılaştırması (Alajlan ve Ibrahim 2023)



Şekil 2.2. Modellerin ve niceleme yöntemlerinin boyut yönünden karşılaştırması (Alajlan ve Ibrahim 2023)

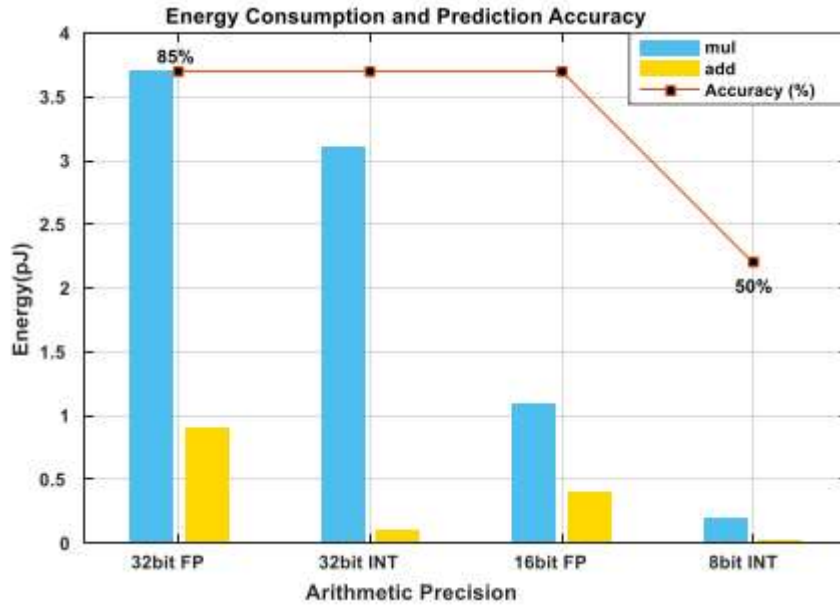
Abu Adla vd. (2023), Arapça dilindeki rakamların gerçek zamanlı olarak algılanması için TinyML ile optimize edilmiş bir evrişimli sinir ağı kullanılmıştır. Bu çalışma, uç cihazlarda, sunucu bağlantısı olmadan, ses ile etkinleşen akıllı uygulamaların TinyML ile geliştirilebileceğini göstermektedir. Çalışmanın doğruluğu, 3 farklı platformda 2 farklı niceleme yöntemi ile karşılaştırılmıştır. Tam sayı niceleme metodu(int8) kullanılan model ile Arduino Nano 33 BLE platformunda ~%90 doğruluk elde edilmiştir.

Srinivasagan vd. (2023), çalışmalarında TinyML'i, hurma meyvesinin raf ömrünü tahmin etmek için kullanmışlardır. Veri seti hurmanın saklama süresi boyunca

fiziksel ve kimyasal özelliklerinin takip ve kayıt edilmesi ile etiketlemiştir. Etiketler 18 kanallı spektrometre verisi ile eşleştirilerek veri seti oluşturulmuştur. Modelin eğitimi Edge Impulse platformu üzerinde yapılmıştır. Nano 33 BLE kullanılarak yapılan testlerde ~%95 doğruluk elde edilmiştir.

Sabry vd. (2022), çalışmalarında giyilebilir sağlık teknolojilerinde yapay zekanın yeri incelenmiştir. İnsan vücudundan toplanabilen EEG, EKG, EDA, EMG, BVP gibi bir çok veri bulunmaktadır. Bu verilerin toplanıp yapay zeka algoritmaları ile işlenmesi, insan aktivitesi izlenmesi, sağlık durumu değerlendirmesi ve hastalık teşhisi alanlarında kullanılmaktadır. Ancak çalışma, verilerin boyutlarından dolayı özellikle bant genişliği kaynaklı problemlere ve gizlilik problemlerine değinmektedir. Bu sınırlama bir çözüm olarak TinyML önerilmiştir. Hafıza mimarilerinde ve mikrodenetleyici cihazlarda yapılacak geliştirmeler sayesinde, karmaşık makine öğrenmesi algoritmalarının kaynak kısıtlı cihazlar üzerinde çalıştırılması sağlayacak ve bu sayede TinyML'in bant genişliği, gecikme ve gizlilik problemlerini ortadan kaldırılabileceğine değinilmiştir.

Ajani vd. (2021), çalışmalarında TinyML'i, IoT cihazlar üzerinde makine öğrenimi algoritmalarının çalışmasına uygun bir ortam oluşturma fikri olarak tanımlamışlardır. TinyML'in güç tüketimi, enerji verimliliği ve düşük maliyeti vurgulanmıştır. TinyML'in ortalama 0.1-0.3 W gibi oldukça düşük bir enerji tüketimine sahip olduğu yer almıştır. Çalışma ayrıca günümüzde yaygın kullanılan TinyML çerçevelerini paylaşmıştır. Çalışmadaki bir diğer başlık altında 32 bit floating point, 32 bit integer, 16 bit floating point ve 8 bit integer veriler üzerinde yapılan işlemler karşılaştırılmıştır. Özellikle 8 bit 16 bit verilerle çalışılırken düşük enerji tüketildiği, ancak doğruluğun buna bağlı olarak düştüğü belirtilmiştir. Bu sonuçlar TinyML uygulamalarında optimizasyon ve budama işlemleri yapılırken uygun tekniğin seçilmesinin önemini vurgulamaktadır. Şekil 2.3, nicelemenin matematiksel işlemlerin enerji tüketimleri üzerindeki etkilerini göstermektedir.



Şekil 2.3. Niceleme yöntemlerinin toplama ve çıkarma işlemleri esnasında harcadığı enerji ve model doğruluklarının karşılaştırılması (Ajani vd. 2021)

2.4 Gömülü Sistemlerde Makine Öğrenmesi ile EKG Sinyali Sınıflandırması

Kim vd (2023), yaptıkları çalışmada IoT teknolojilerindeki gelişmelerin, kişisel ve sürekli EKG izleme sistemlerini mümkün hale getirdiğini belirtmiştir. Yüksek hesaplama gücü gereksiniminden dolayı toplanan EKG verisinin bir sunucuya aktarılması gerektiği, ancak batarya ve sunucuya ulaşım kaynaklarının sınırlı olmasının, sürekli EKG izlemenin önünde önemli bir engel olduğu belirtilmiştir. Bu duruma çözüm olarak TinyML önerilmiştir. Yapılan çalışmada MIT-BIH ve PTB veri seti kullanılmış olup %97 başarı elde edilmiştir. Ancak geliştirilen model, ilgili veri setinde bulunan sinyalleri normal – anomal olarak ikili sınıflandırılmış olup, anormal olarak sınıflandırılan EKG segmentleri için çoklu sınıflandırma yapılmamıştır.

Falaschetti vd. (2022), çalışmalarında EKG verileri üzerinde doğrudan çalışan RNN varyasyonlarının etkinliğini incelemişlerdir. MIT-BIH veri seti AAMI protokolü kullanarak eğitilen modellerin doğruluk oranı test edilmiştir. Çalışmada EKG segmentleri ikili ve çoklu olarak sınıflandırılmıştır. Ancak, MIT-BIH veri setinde bulunan etiketler, çalışma kapsamında AAMI protokolüne tanımlanmış etiketler altında sınıflandırılmış olup, çalışmada yapılan çıkarımlar MIT-BIH veri setinde tanımlı tüm etiketleri kapsamamaktadır. Elde edilen model herhangi bir niceleme algoritmasında geçirilmemiştir. Çalışmada 4 farklı model denenmiştir ve elde edilen en yüksek doğruluk oranı %90'dır. Çizelge 2.1'de AAMI standartlarına göre aritmi sınıfları ve MIT-BIH veri setinde bulunan aritmi sınıflarının bir karşılaştırması bulunmaktadır.

Çizelge 2.1. MIT-BIH veri Seti Sınıfları ve AAMI Sınıflarının Karşılaştırılması
(Falaschetti vd. 2022)

AAMI	Classes	MIT-BIH Labels
Normal	N	N, L, R
Supraventricular Ectopic Beat (SVEB)	SVEB	e, j, A, a, J, S
Ventricular Ectopic Beat (VEB)	VEB	V, E
Fusion	F	F
Unknown Beat	Q	/, f, Q

3. MATERYAL VE METOT

3.1 Kalbin Çalışması ve EKG Sinyali

Kalp, sağ ve sol kalp olmak üzere iki pompa olarak tanımlanır. Sağ kalp, kanı pulmoner damar aracılığıyla akciğere pompalar, sol kalp ise kanı aort damarı aracılığıyla sistemik dolaşıma pompalar. Kalp, atriyum ve ventrikül isimli odacıklardan oluşur. Atriyum, ventrikülü dolduran öncü bir pompa olarak çalışır. Kan ile dolan ventrikül, ana pompalama gücünü sağlar ve kan, dolaşım sistemine pompalanır. Kalbin ritmik kasılmaları, aksiyon potansiyelleri tarafından uyarılan kalp kası ile sağlanır. Kalp kaslarındaki ardışık ve sürekli kasılmalar kalp ritmini oluşturur. İnsan kalbi her gün 100.000 kez atar. Bu kasılmalar, kalp kasının kasılması için ritmik sinyaller üreten ve bu sinyalleri kalp boyunca hızlı bir şekilde ileten bir sistem tarafından gerçekleştirilir (Hall ve Guyton 2016).

Vücudun doğal pacemaker'ı olan sinoatrial düğüm, her kalp atışını başlatmaktan sorumludur. Sağ atriyum ve üst ana toplardamarın bitişiğinde bulunan sinoatrial düğüm, aksiyon potansiyellerini üretir, ileti sistemi aracılığıyla artiyum ve ventrikül miyokardına ileterek sinyal tüm kalbe yayılır (MacDonald vd. 2020).

Teknolojik gelişmeler ile birlikte, EKG'nin önemi son yıllarda artmıştır. EKG, kardiyak aritmilerin veya miyokard infarktüsü ve kalp yetmezliği gibi bazı kalp hastalıklarının tespiti ve takibinde kullanılan bir tetkiktir. EKG, fizyolojik ve teknolojik bir sürecin sonucu olarak karşımıza çıkmaktadır. Hücre zarları boyunca iyon akışının bir sonucu olarak transmembran iyonik akımları oluşturulur ve akımların bir sonucu olarak kardiyak elektrik alan oluşur. Bu elektrik alan, kalp ve çevresinde meydana gelir; vücudun diğer bölümlerinin içinden geçerken bozulmaya uğrar. Bu elektrik akımları, gövdenin belirli noktalara yerleştirilen elektrotlar tarafından tespit edilir ve kaydedilir (Libby vd. 2012).

Bir algoritmanın EKG sinyali üzerinden çıkarım yapabilmesi için öncelikle EKG sinyalinin matematiksel modelinin algoritmaya kısmen ya da tamamen tanıtılması gerekmektedir. EKG sinyalinin matematiksel modeli, teknolojik uygulamalar için oldukça güçlü bir temel sağlar. Bu çalışmada kullanılan matematiksel yöntemler harmonik analiz, dalga analizi ve segmentasyon analizidir. Literatürde yaygın olarak kullanılan matematiksel yöntemler Özellik Çıkarımı ve Dönüşüm Teknikleri başlığı altında ayrıntılı olarak incelenecektir.

3.2 Özellik Çıkarımı ve Dönüşüm Teknikleri

EKG sinyali doğrudan bir algoritmanın girdisi olarak kullanılabilir. Fakat bu durumda bazı zorluklar ile karşılaşılacaktır. Ham verilerin kullanılması, gürültü, fizyolojik ya da fizyolojik olmayan artefakt ya da düşük kaliteli sinyallerin sebep olabileceği problemlerle karşılaşılmasına sebep olur. Fizyolojik artefakt, vücudun diğer organlarından kaynaklanan biyoelektriksel aktivitelerdir (Kodal Sevindir vd. 2018). İlgili özellik çıkarımı ve dönüşüm tekniklerinin kullanılmasındaki amaç, ham verilerin

daha anlamlı ve işlenebilir hale dönüştürülmesini sağlamaktır. Bu teknikler genellikle gürültü seviyesini azaltmak, veri boyutunu küçültmek, doğruluğu artırmak için kullanılır.

Bununla birlikte, bu tekniklerin bazı dezavantajları da bulunmaktadır. Kullanılan teknikler verinin karmaşıklığının azaltılmasında kullanıldığı için sinyalin önemli bileşenlerinin kaybolmasına neden olabilir. Ayrıca doğru tekniğin seçilmesi uzmanlık ve deneyim gerektiren bir süreçtir. Veri setinin ön işlenmesi aşamasında yapılan yanlış seçimler ham olarak anlamlı olan bir verinin dönüşüm sonucunda anlamsız olmasına ya da yanlış yorumlanmasına sebep olabilir. Bir tekniğin çıkarım yapma yeteneği ile hesaplama yükü genellikle doğru orantılıdır. Bu durum, bir başka problem olarak karşımıza gelmektedir.

3.2.1 Segmentasyon analizi

Veri segmentasyonu, verinin daha küçük, anlamlı parçalara ayrılması işlemidir. Segmentasyon işlemi büyük ve karmaşık veri setlerini daha yönetilebilir ve analiz edilebilir hale getirmeyi hedefler. Veri segmentasyonu, özellikle kaynak kısıtlı sistemler üzerinde çalışan makine öğrenmesi algoritmaları güçlü bir uyum sağlar. Oluşturulacak modelin doğruluğu, çıkarım süresi ve boyutunun segmentasyona göre nasıl değiştiği bulgular başlığı altında detaylı olarak incelenecektir.

Literatürde kullanılan en yaygın EKG segmentasyon tekniği R dalga tepesi tespitidir. EKG sinyallerinin daha ileri düzeyde yorumlanabilmesi için ST segmentinin uzunluğu yada EKG sinyalini oluşturan diğer segmentlerin de analiz edilmesi gerekebilir (Beraza ve Romero 2017). Bu çalışmada, her iki R dalgası tepesi arasında kalan sinyal bölümü makine öğrenmesi modelinin girdisi olarak kullanılacaktır. Ancak çalışmada herhangi bir R dalga tepesi tespit algoritması kullanılmamıştır. Kullanılan veri seti, R dalgası tepelerinin hangi örnekleme bulundugu bilgisini sunmaktadır. Veri seti ve özellikleri, MIT-BIH Aritmi Veri Seti başlığı altında detaylı şekilde incelenecektir.

3.2.2 Dalgacık dönüşümü ile öznitelik çıkarımı

Fourier dönüşümü ve dalgacık dönüşümü bir sinyalin özellik çıkarımında kullanılabilecek tekniklerdir. Dalgacık dönüşümü, başlangıçta sismik sinyallerin analizinde Fourier dönüşümünden daha iyi bir araç olarak ortaya çıkmıştır (Sifuzzaman vd. 2009). Bir sinyalin karmaşıklık seviyesi, sinyal karakteristiğini hem zaman hem de frekans alanında temsil eden bir dönüşüm kullanmayı gerektirebilir. Fourier dönüşümü sinyali sürekli ve tekrar eden farklı frekanslardaki alt sinyallere ayırırken, bu frekans bileşenlerinin zaman içerisinde ne zaman meydana geldiği hakkında bilgi vermez (Najmi ve Sadowsky 2009). Günümüz teknolojisi ile toplanan sinyallerin karmaşıklığı düşünüldüğünde Fourier dönüşümü ham sinyalden anlamlı veri çıkarmak konusunda yetersiz kalabilir.

1982 yılında Fransız Jeofizikçi Jean Morlet ‘Wavelet’ kavramını ortaya atmıştır. Wavelet, küçük dalga anlamına gelir (Sifuzzaman vd. 2009). Dalgacık dönüşümü, sinyalin zaman ve frekans bilgisini aynı anda içermesi nedeniyle, araştırmacılar tarafından sıkça tercih edilmektedir. Dalgacık dönüşümü sürekli ya da ayrık olabilir. Sürekli dalgacık dönüşümü tekniği, faz bilgisini daha iyi koruduğu için daha sık tercih edilmektedir (Pham ve Wong 1999).

Çizelge 3.1, Fourier dönüşümü ve dalgacık dönüşümünün karşılaştırılarak teknikler arasındaki fark ayrıntılı olarak incelenmiştir.

Çizelge 3.1. Fourier Dönüşümü ve Dalgacık Dönüşümünün Karşılaştırılması

	Fourier Dönüşümü	Wavelet Dönüşümü
Amaç	Bir sinyali farklı frekanslardaki bileşenlerine ayırır.	Bir sinyali zaman ve frekans bileşenleri ayırır.
Zaman – Frekans Bilgisi	Zaman bilgisi vermez. Sadece frekans içeriği sağlar.	Zaman ve frekans bilgisini aynı anda taşır.
Çözünürlük	Tüm frekanslar için sabit frekans sunar.	Değişken çözünürlük sağlar.(yüksek frekansta zaman, düşük frekansta frekans alanında daha iyidir.)
Uygulama	Sabit frekans içeriğine sahip sinyaller için idealdir.	Zamanla değişen frekansa sahip olan sinyaller için daha uygundur.
Temsil	Sinyalleri sonsuz adet sinüs toplamı olarak ifade eder.	Sinyaller, hem zaman hem frekans cinsinden tanımlanmış küçük dalga benzeri fonksiyonlar dizisi olarak temsil eder.
Dezavantaj	Geçici ya da zamana bağlı değişen frekans varyasyonları temsil edemez.	Yüksek hesaplama yükü getirir.
Matematiksel Temel	Sinüs sinyallerine dayalıdır.	Küçük dalgalara dayalıdır.
Örnek	Görüntü sıkıştırma, spektral analiz	Sinyal sıkıştırma, gürültü azaltma, veri madenciliği, zaman-frekans analizi
Veri Cinsi	Sinyalin frekans spektrumu	Zaman frekans çözümlemesi, çoklu çözünürlük analizi

Dalgacık dönüşümü ve makine öğrenmesi, sinyal işleme ve veri analizi alanlarında sıkça birlikte kullanılan güçlü araçlardır (Guo vd. 2022). Dalgacık dönüşümü özellikle biyomedikal sinyaller, ses analizi, görüntü işleme gibi değişken frekansa sahip sinyallerde tercih edilir. Dalgacık dönüşümü kullanılarak sinyalden özellikler çıkarılabilir ve sonuçlar yapay zekaya girdi olarak sunulabilir. Bu yol ile yapay zeka tahminlerindeki doğruluk oranı artırılabilir. Bu sinerji sadece doğruluğu artırmak ile kalmaz, aynı zamanda veri küçültme, gürültü giderme gibi işlemleri de sağlayarak makine öğrenmesi modellerinin verimliliğini artırır. Karmaşık ve büyük verilerle çalışırken ya da kaynak kısıtlı platformlarda yapılan çıkarımlarda bu iki kavramın birlikte kullanılması önemli avantajlar sağlar. Bulgular kısmında ham verinin girdi olarak kullanıldığında karşılaşılan problemlerde detaylı olarak bahsedilecektir.

Bu çalışmada, derin öğrenme modelinin girdileri, ham EKG sinyaline dalgacık dönüşümü uygulanması ile elde edilmiştir. Dalgacık dönüşümü kullanılmasına karar verildikten sonraki adım ise çalışmaya uygun bir dalgacık fonksiyonunun seçilmesi olacaktır. Literatürde bir çok farklı dalgacık fonksiyonu bulunmaktadır. Bu çalışmada Haar, Mexican Hat, Daubechies, Gaussian, Meyer ve Morlet dalgacık dönüşümü fonksiyonları incelenmiştir, sonuçları bulgular başlığı altında paylaşılacaktır. Çalışmada görece olarak düşük veri boyutu, yüksek makine öğrenmesi modeli doğruluğu sağlamasında dolayı Morlet dalgacık fonksiyonu kullanılacaktır. Morlet dalgacık fonksiyonu Morlet Dalgacık Dönüşümü başlığı altında incelenecektir.

3.3 MIT-BIH Aritmi Veri Seti

Kardiyak ritmin korunması, miyokardiyumda impulsların düzenli yayılmasını sağlayan iyon kanalları ve taşıyıcıların koordinasyon içerisinde çalışmasını gerektirir. Bu sürecin aksaması kardiyak aritmilerin oluşmasına sebep olur (Kingma vd. 2023). Aritmi, kalbin çok hızlı, çok yavaş ya da düzensiz atmasına sebep olabilir. Kardiyak aritmi, kalp kasının sistematik kasılmasının bozar ve ciddi sağlık sorunlarına sebep olur. Bu sebepler, aritminin yakalanması ve doğru bir şekilde sınıflandırılması hayati önem taşımaktadır.

Bu çalışmada bir dizi ön işleme ve makine öğrenmesi algoritmasından geçirilerek tahmini yapılan aritmi tipleri ve tanımları MIT-BIH Aritmi Veri Seti başlığı altında incelenecektir.

Bu çalışmada kullanılacak makine öğrenmesi modelinin eğitimi ve çalışma boyunca yapılacak tüm testler için MIT-BIH Aritmi Veri Seti kullanılmıştır. MIT-BIH Aritmi Veri Seti, Beth Israel Deaconess Medical Center ve Massachusetts Institute of Technology tarafından oluşturulmuş, literatürde çok yaygın kullanılan bir kardiyak aritmi veri setidir. Veri Seti, 1980 yılından itibaren yaklaşık olarak 500 farklı merkezde aritmi dedektörlerinin değerlendirilmesinde ve EKG temelli araştırmalarda kullanılan ilk standart veri setidir (Moody ve Mark 2001). Bu çalışmada literatürdeki yeri,

güvenirliliği, veri çeşitliliği gibi makine öğrenmesi modelinin doğruluk oranını etkileyecek sebeplerden dolayı MIT-BIH aritmi veri seti seçilmiştir.

MIT-BIH Aritmi Veritabanı, 47 gönüllüden elde edilen, EKG kayıtlarının 48 adet yarım saatlik kaydını içermektedir. Bu kayıtlar, 1975 – 1979 yılları arasında BIH Aritmi Laboratuvarlarında incelenen insanlardan alınmışlardır. 23 adet kayıt, Boston'daki Beth Israel Hastanesi'ndeki 4000 adet hasta kaydı arasında rastgele seçilerek elde edilmiştir. Diğer 25 kayıt ise veri tabanı içerisinde nadir bulunan ancak klinik açıdan önemli kabul edilen aritmileri içerdiği için özel olarak seçilmiştir. Bu veri seti, “reference annotation” dosyaları ile birlikte 2005 yılından itibaren PhysioNet sitesi üzerinden erişilebilir durumdadır (Moody ve Mark 2001). Reference annotation, bir verinin üzerine yapılan, belirli ya da önemli olayları işaretleme işlemidir.

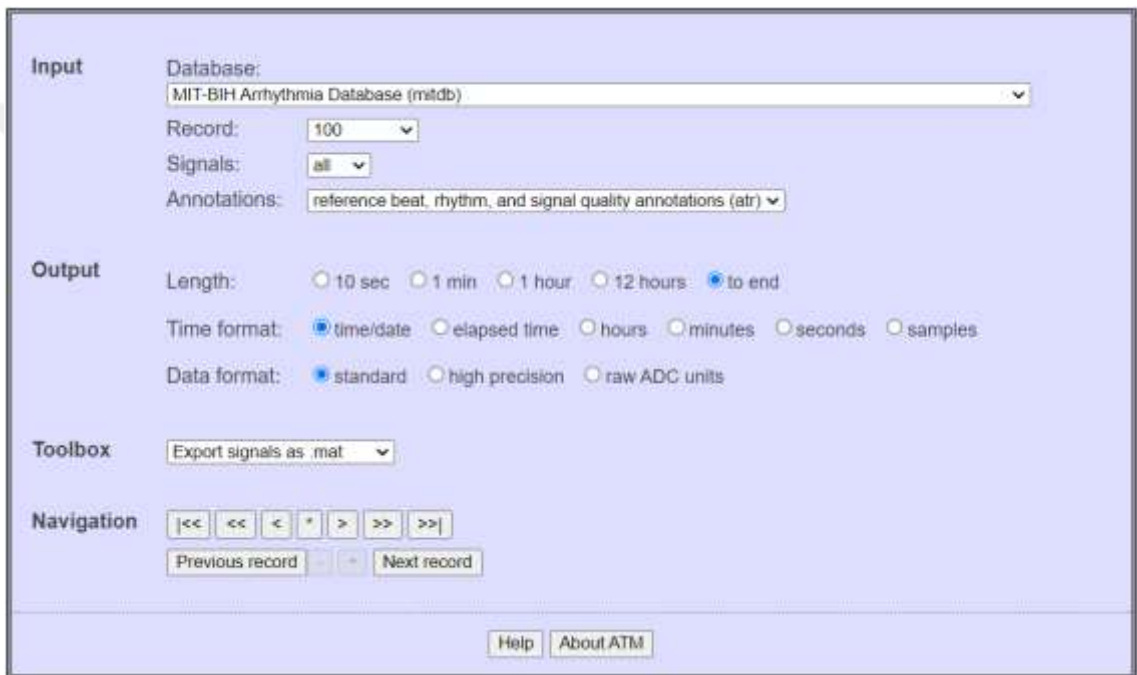
Veri seti 48 adet yaklaşık 30 dakikalık EKG kaydından oluşmaktadır. Örnekleme hızı 360 Hz, çözünürlüğü 11 bit'tir. Ölçümlerde 0.1-100 Hz bant geçiren filtre kullanılmıştır (Moody ve Mark, 2001).

Çizelge 3.2'de veri setinde bulunan etiketler ve anlamları paylaşılmıştır. Bu verilere veri seti ile birlikte indirilen Annotation dosyasından ulaşılabilir. Çizelgede bulunan etiketlerin tamamı bu çalışmaya dahil edilmemiştir. Çalışmaya dahil edilen aritmi tipleri, açıklamaları ile birlikte Veri setinde bulunan aritmi sınıfları başlığı altında incelenecektir.

Çizelge 3.2. MIT-BIH Aritmi Verisetinde Bulunan Etiketler ve Açıklamaları

Sembol	Anlamı
“.” ya da “N”	Normal Atım
“L”	Sol dal bloğu atımı
“R”	Sağ dal bloğu atımı
“A”	Atrial erken atım
“a”	Bozulmuş atrial erken atım
“J”	Nodal (kesişimsel) erken atım
“S”	Supraventriküler erken atım
“V”	Prematür ventriküler kontraksiyon
“F”	Ventriküler ve normal atımın birleşimi
“l”	Ventriküler flutter/fibrilasyon başlangıcı
“!”	Ventriküler flutter dalgası
“J”	Ventriküler flutter/fibrilasyon sonu
“e”	Atrial kaçış atımı
“j”	Nodal (kesişimsel) kaçış atımı
“E”	Ventriküler kaçış atımı
“/”	Paced (takip edilen) atım
“f”	Paced ve normal atımın birleşimi
“x”	İletilmeyen P-dalgası (bloklanmış APB)
“q”	Sınıflandırılmayan atım
“ ”	İzole QRS-benzeri artefakt

Bu çalışmada kullanılan veri seti Physiobank internet adresinden indirilmiştir (Anonymous 3). Şekil 3.1’de yer alan arayüz, MIT-BIH Aritmi Veri Seti’nin indirilmesi öncesinde uygun parametreleri göstermektedir. Bu arayüzde, kayıt numarası seçilebilir, sinyal ve anotasyon (etiket) dosyasının türü belirlenebilir. Kullanıcı, belirli bir kaydı için parametreleri belirleyerek veriyi farklı dosya formatlarında indirebilir. Bu çalışmada MATLAB programı kullanıldığından dolayı kayıtları içeren dosya .mat formatında indirilmiştir. Veri seti uzunluğu 10 saniye, 1 dakika, 1 saat, 12 saat ya da tümü olarak seçilebilir. MIT-BIH veri seti içerisinde tüm kayıtlar yaklaşık olarak 30 dakikadır. Ayrıca zaman formatı kullanıcı tercihinine göre özelleştirilebilir.



Şekil 3.1. Physiobank Arayüzü

3.4 Verinin Ön İşlenmesi

Bu çalışmada oluşturulan makine öğrenmesi modeli STM32H750B-DK geliştirme kartı üzerinde çalışacaktır. Çalışmanın kaynak kısıtlı sistemlerde aritmi sınıflandırılması gerçekleştirmesi itibarıyla ön işleme ve model geliştirme adımlarında hafıza optimizasyonu ön planda tutulacaktır. Geliştirme kartı üzerinde makine öğrenmesi modelinin yanı sıra verilen EKG sinyalinin dalgacık dönüşümü hesaplamaları da gerçekleştirilecektir. Ön işleme ve model geliştirme süreci boyunca gerekli RAM ve FLASH hafızasını düşüren ancak doğruluk oranını azaltan birçok işlem yapılmıştır. Ön işleme adımları aşağıdaki gibidir. Verinin ön işlenmesinde MATLAB programı ile birlikte MATLAB Wavelet Toolbox kullanılmıştır.

3.4.1 Verilerin segmentasyonu ve etiketlenmesi

Veri setinde bulunan 48 kaydın her biri için, EKG örneklem ve etiket dosyası olmak üzere, 2 adet dosya bulunmaktadır. Veri segmentasyonu, annotation dosyası içerisinde bulunan R-Peak'in kaç numaralı örneklemde yakalandığı bilgisini kullanarak, kayıt dosyası içerisinde bulunan verilerin indekslenmesini kapsamaktadır. Çizelge 3.3'te "100" isimli kaydın ilk 10 satırı bulunmaktadır. İlk kolon R-Peak'in tespit edilme zamanını(saniye cinsinden), ikinci kolon R-Peak'in örneklem numarasını, 3. kolon ise ilgili R-R aralığının etiketini belirtmektedir. Örneğin 9 numaralı satırdaki R-R intervali 0:05.678 saniyesinde ve 2044. örneklemde başlamış. 10. satırdaki R-R intervali ise 0:06.672 ve 2402. örneklemde başlamıştır. Bu sebeple 9. satırdaki R-R intervali 994 milisaniye ve 358 örneklem boyunca sürmüştür ve "A" ile etiketlenmiştir. "A" etiketi, atrial erken atımı ifade eder. Çizelge 3.3'teki diğer tüm atımlar Normal atım olarak etiketlenmiştir.

Çizelge 3.3. MIT-BIH Veri Seti Anotasyon Dosyası Örneği (Moody ve Mark 2001).

0:00.050	18	+
0:00.214	77	N
0:01.028	370	N
0:01.839	662	N
0:02.628	946	N
0:03.419	1231	N
0:04.208	1515	N
0:05.025	1809	N
0:05.678	2044	A
0:06.672	2402	N

Segmentasyon işlemi tüm veri seti için yapılır. Segmente edilen R-R aralıkları, etiketlerine göre bir matris üzerinde toplanır. Veri setinde 20 adet olan etiket, bu işlem sonunda bazı etiketlerin çalışmaya dahil edilmemesi ile 10'a düşer.

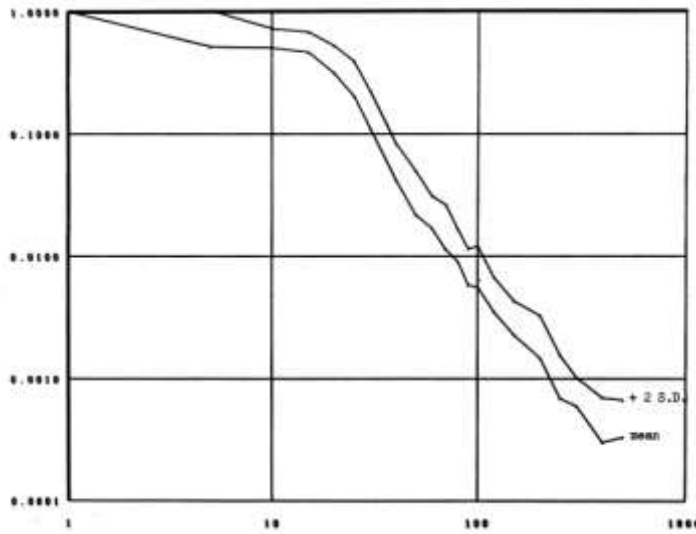
Sonuç olarak bu adımda, Her bir satırında bir R-R intervali ölçüm değerlerini içeren, 10 adet matris(çalışmaya dahil edilen her bir aritmi tipi için 1 matris) elde edilmiştir. Matrislerde bulunan satır sayıları, veri seti içerisindeki etiketlerin sayısına bağlı olarak değişiklik göstermektedir.

3.4.2 Verilerin alt örnekleme

Aşağı örnekleme (downsampling), bir sinyalin örnekleme oranının bilinçli olarak azaltılması işlemidir. Bu işlem ölçeklemeden kaynaklanan doğruluk kaybını göz önünde bulundurarak verilerin boyutlarının küçültülmesi amacıyla kullanılır (Zhou 2020).

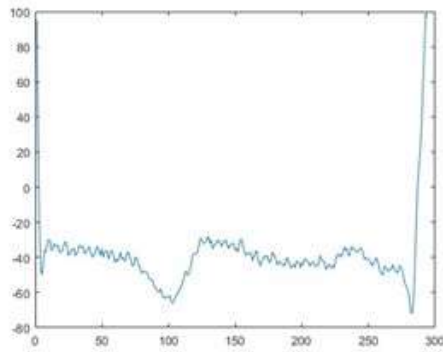
R-R intervali segmentasyonu ve sinyallerin etiketlerine göre ait oldukları matrise dağıtılması tamamlandıktan sonra dalgacık dönüşümü hesaplamasında gerekli olan RAM ihtiyacını azaltmak için alt örnekleme işleminin yapılması gerekir. Daha önce belirtildiği gibi, bu çalışmada geliştirilen makine öğrenmesi modeli, STM32H750B-DK geliştirme kartı üzerinde çalışacaktır. Dalgacık dönüşümü sonuçları makine öğrenmesi modelinin girdisi olmasından dolayı, MATLAB programında yapılan dalgacık dönüşümü hesaplamalarının aynısının STM32H750B-DK geliştirme kartı ile de yapılması gerekmektedir. Bu nedenle, STM32H750B-DK geliştirme kartının sınırlı RAM kapasitesi dikkate alınarak, 360 Hz olan ham sinyalin 200 Hz'e dönüştürecek şekilde veri boyutu alt örnekleme yöntemi ile seyreltilir.

Alt örnekleme işleminde dikkat edilmesi gereken önemli bir nokta, sinyalin bilgi içeriğinin korunmasıdır. Bu sebeple sinyal spektrumu göz önünde bulundurularak Nyquist kriterine uygun bir şekilde alt örnekleme oranı belirlenmiştir. Nyquist teoremine göre, eğer bir sinyalin en yüksek frekans bileşeninin B 'den büyük olmadığı biliyor bu sinyali bilgi kaybı olmadan örnekleme için sinyalin $2B$ 'den daha büyük bir frekansta örnekleme gerekir (Arie vd. 2020). Bu oran, dalgacık dönüşümden elde edilen özelliklerin, makine öğrenmesi modelinde yeterli doğruluğu sağlamasını doğrudan etkileyecektir. İlgili veri setinin Örnekleme hızı 360 Hz'tir ve ölçümlerde 0.1-100Hz bant geçiren filtre kullanılmıştır (Moody ve Mark 2001). Standart EKG'nin frekans bileşenleri, 0.05-150Hz aralığındadır ancak yine de EKG sinyalinde daha yüksek frekans bileşenleri de bulmak mümkündür (Golden 1973). Bu bilgiler göz önünde bulundurulduğunda sinyal frekansının, yarı yarıya düşürülmesi mümkündür. Alt örnekleme işlemi, veri setinde ardışık olan her iki değerin ortalaması alınarak yeni bir değer elde edilmesi sonucunda tamamlanmış olur. Şekil 3.2'de bir EKG verisinin sinyal bileşenlerinin frekanslarına ait dağılımı görülmektedir. Görüldüğü gibi, 100 Hz üzerindeki bileşenler oldukça zayıf bir şekilde temsil edilmektedir.

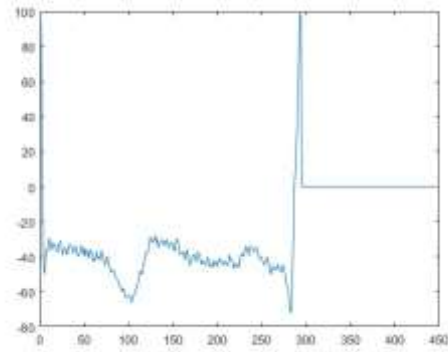


Şekil 3.2. Bir EKG sinyalinin frekans bileşenleri (Golden 1973)

Segmentasyonu ve alt örnekleme yapılan tüm R-R intervallerinin veri uzunluklarının eşit olması, dalgacık dönüşümünün uygulanabilmesi için gereklidir. Ancak veri setindeki R-R intervallerinin uzunluğu farklılık göstermektedir. Bu uzunlukların eşitlenmesi için en uzun R-R intervali dışındaki tüm verilerin sonuna sıfır eklenerek tüm R-R intervalleri en uzun R-R intervaline eşitlenmiştir. Veri setinde normal olarak etiketlenen bir R-R intervalinin veri uzunluğu eşitlenmeden önceki ve sonraki hali aşağıdaki gibidir. Uzunluğu 300 olan bir R-R intervalinin uzunluğu, veri sonuna 0 eklenerek 450'ye çıkarılmıştır. Şekil 3.3'te (a) ve (b) numaralı alt başlıklar altında, uzunluk eşitleme işleminden önce ve sonra elde edilen R-R intervallerinin görselleri yer almaktadır. Bu işlem, özellikle kaynak kısıtlı sistemlerde, veri analizi için önemli bir adımdır.



(a)



(b)

Şekil 3.3. a) Uzunluk Eşitleme Öncesi R-R İntervali; b) Uzunluk eşitleme sonrası R-R intervali

3.4.3 Morlet dalgacık dönüşümü

Morlet dalgacık dönüşümü fonksiyonu, zaman - frekans analizinde yaygın olarak kullanılan bir tekniktir. Bu dönüşüm temelde bir Gauss fonksiyonu ile çarpılan bir sinüs dalgasında oluşur (Zhang ve Cottrell 2004). Morlet dalgacık fonksiyonu, hem yerel hem de genel düzeyde sinyal özelliklerinin çıkarılmasını sağlayarak, sinyaldeki hızlı frekans değişimlerini tespit etme konusunda etkilidir (Daubechies 1990).

Denklem 3.1, Morlet dalgacığının (ω) kompleks formda, bir sinüs dalgası ile Gauss pencereleme fonksiyonunun çarpımı olarak ifade eder (Cohen 2018).

$$\omega = e^{2i\pi ft} \cdot e^{\frac{-t^2}{2\sigma^2}} \quad (3.1)$$

e : Euler Sayısı, ~2.71828

i : İmajiner operatör

2π : Sinüs sinyalinin açısal frekansını standart frekansa dönüştürmek için kullanılır.

F : Sinüs dalgasının frekansını ifade eder. Hertz (Hz) cinsinden ölçülür.

T : Zaman, saniye cinsinden ölçülür.

σ : Sigma, Morlet dalgacık fonksiyonunda karmaşık sinüs dalgası ile çarpılan Gauss fonksiyonunun standart sapmasını ifade eder. Sigma değeri büyüdükçe, Gauss penceresi genişler ve frekans çözünürlüğü düşer. Yüksek frekans bileşeni içeren sinyaller, düşük sigma değeri ile hesaplanmalıdır. Sigma, frekans çözünürlüğü ve işlem yükünün dengelenmesi için kullanılır. Denklem 3.2, σ 'nın frekans (f) ve bir ölçek faktörü olan n ile ilişkili olarak tanımlandığını göstermektedir.

$$\sigma = \frac{n}{2\pi f} \quad (3.2)$$

Morlet dalgacık fonksiyonu, bir sinüs dalgasının Gauss fonksiyonu ile pencerelenmesi sonucu oluşur. Zaman frekans analizinde yaygın olarak kullanılan Morlet dalgacık fonksiyonu, zaman serisi sinyallerinde her bir noktadaki güç ve faz bilgilerini bulabilir. Morlet dalgacığı, zaman ve frekans bilgisini aynı anda sağlar. Çünkü sinyali kısa zaman dilimlerinde frekans yönünden analiz etmektedir. Bu özellik,

sinyal içerisinde zamanla değişen frekans bileşenlerinin analiz etmesinde kullanılır. Dalgacık dönüşümü hesaplamalar esnasında Hızlı Fourier Dönüşümü kullansa da, Fourier Dönüşümü ile arasındaki en temel farktır (Cohen 2018).

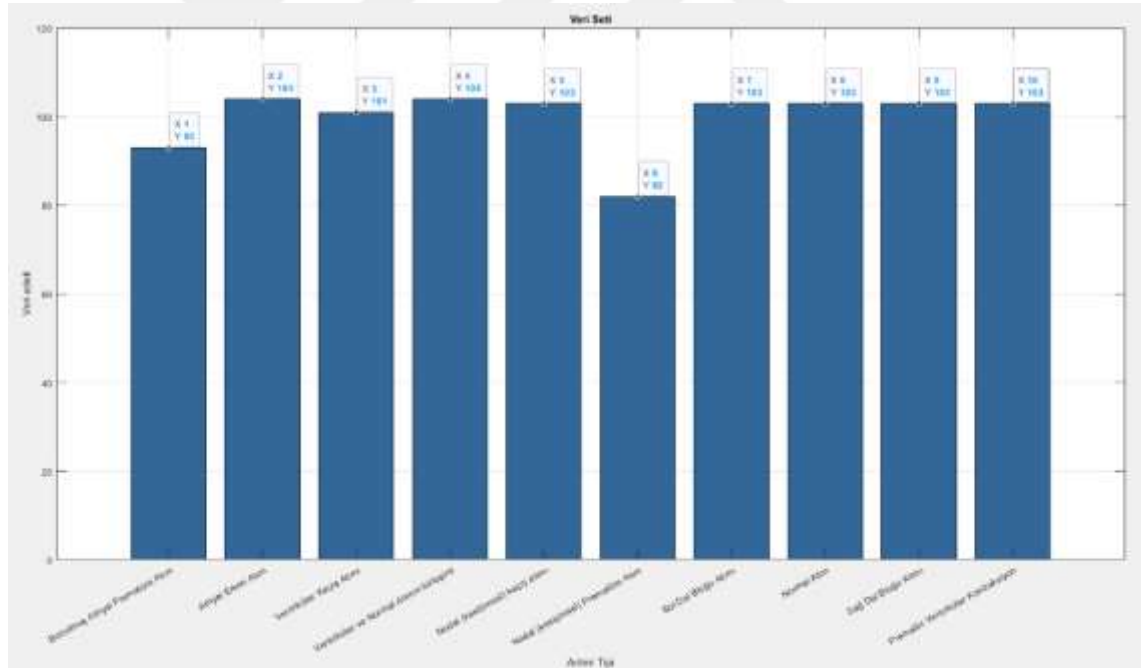
3.4.4 Veri setinde bulunan aritmi sınıfları

MIT-BIH veri seti içerisinde bir çok işaretleme bulunmaktadır. Bu işaretlemelemlerinin bir kısmı kalp atışlarını sınıflandırmak için kullanılırken bir kısmı ise bazı kardiyak durumları tanımlamak için kullanılmıştır (Moody ve Mark 2001). MIT-BIH veri setinde bulunan etiketlerin tamamı çalışmadaki makine öğrenmesi modelinin eğitiminde kullanılacak olan veri setine dahil edilmemiştir. Özellikle aritmi dışındaki işaretlemelemler ve bazı aritmi etiketleri model performansının artırılması, çalışmanın kapsamı ve veri dengesizliğinin önlenmesi amacı ile oluşturulan veri setine dahil edilmemiştir. Bu yaklaşım, modelin aritmi türlerine odaklanmasını sağlamış ve model eğitimi sürecini optimize etmiştir. Modele dahil edilen 11 farklı tipteki artimi ve açıklamaları aşağıda belirtildiği gibidir.

- **Normal Beat (Normal Atım):** Kalbin doğal ritmine uygun bir şekilde ilerleyen elektriksel kalp aktivitesidir. Atım, SA düğümünden başlayarak atriyum ve ventrikülleri uyarır ve kalp uygun şekilde kasılır.
- **Atrial Premature Beat (Atriyal Erken Atım):** Atriyumlardan kaynaklanan ve normalden önce oluşan elektriksel aktivite olarak tanımlanır. Atriyal premature atımlar sağlıklı insanlarda da görülebilir ancak sıklığı atriyal fibrilasyon habercisi olabilir (Leng vd. 2020).
- **Aberrated Atrial Premature Beat (Bozulmuş Atriyal Erken Atım):** Dr. Thomas Lewis aberrated kavramını ventrikülde bir impulsun anormal dağılımı olarak tanımlamıştır (Gouaux ve Ashman 1947). Aberrated Atrial Premature Beat ise atriyumdan kaynaklanan, normalden önce gelen sinyaldir. Ancak aberasyon, bu atım sırasında iletim sistemi de anormal çalışır ve ventriküllere iletilen sinyal iletim esnasında bozulmaya uğradığını gösterir.
- **Nodal Junctional Escape Beat (Nodal Kaçış Atımı):** Kaçış atımı, pacing durduktan sonra, kalbin ritmini sürdürebilmesi için başka noktalardan kaynaklanan ve normal zamanından geç gelen atımdır (Viamonte ve Rosen 1990). Junctional ise bu atımın atriyumlar ile ventriküller arasındaki geçiş bölgesinden kaynaklandığını ifade eder.
- **Nodal Premature Beat (Nodal Erken Atım):** Nodal, düğümle ilgili anlamına gelir ve sinüs düğümü ya da atriyoventriküler düğümü ifade eder. Nodal premature beat, düğüm kaynaklı erken atım anlamına gelir.
- **Ventricular Escape Beat (Ventriküler Kaçış Atımı):** Ventricular, karıncıkla ilgili anlamına gelir. Ventricular escape beat, karıncık kaynaklı geç gelen (sinüs atımına göre) atım anlamına gelir.
- **Premature Ventricular Contraction (Erken Ventriküler Kasılma):** Premature Ventricular Contraction, ventriküllerde oluşan ve normal kalp ritmini bozan erken elektriksel aktivitedir (Takahashi vd. 1998).

- **Left Bundle Branch Block Beat (Sol Dal Bloğu Atımı):** Kalbin sol bloğunda oluşan iletim problemleri nedeniyle oluşur. Klinik öneme sahip, sık görülen bir anomalidir (Littmann ve Symanski 2000).
- **Right Bundle Branch Block Beat (Sağ Dal Bloğu Atımı):** Sol dal bloğuna benzer olarak, dallardaki iletimin tamamen kesilmediği, iletim derecesinin diğer daldan farklı olduğu ve kalbin sağ dalını etkileyen aritmidir (Rosenimm 1954).
- **Fusion of Ventricular and Normal Beat (Ventriküler ve Normal Atımın Birleşimi):** Fusion birden fazla pacemaker'ın kalp atımlarını aynı anda kontrol etmesidir (Henry vd. 1962). Fusion of Ventricular and Normal Beat ise sinüs düğümü ve ventriküler pacemaker'dan gelen atımların birleşmesi sonucu oluşur.
- **Fusion of Paced and Normal Beat (Kalp Pili ve Normal Atımın Birleşimi):** Kalp pili ve sinüs düğümünden gelen atımların birleşmesi sonucu oluşur.

Şekil 3.4'te bulunan histogram grafiğinde, çalışmaya dahil edilen aritmi tipleri ve her birinden kaç adet sinyalin makine öğrenmesi modelinin eğitimi için kullanıldığı bilgisine yer verilmiştir.



Şekil 3.4. Veri setinin aritmi tiplerine göre dağılımı

3.4.5 Veri kümesinin ayrılması

Veri seti MIT-BIH aritmi veri setinden rastgele olarak seçilen 1000 adet R-R intervalinden oluşmaktadır. Modelin eğitiminde kullanılacak veri seti oluşturulduktan sonra veri seti karıştırma işleminin yapılması önemlidir. Bu, veri setinde sıralamaya

bağlı olarak ortaya çıkacak olan yanlılığı önler. Eğitim için kullanılacak Veri setinin uygun bir şekilde ayrılması, modelin genelleştirme yeteneği ve performansı açısından büyük önem taşımaktadır. Bu çalışmada veri seti eğitim, doğrulama ve test olarak üç alt kümeye ayrılmıştır. Bu ayrım, modelin yalnızca eğitimi süresince değil, hiperparametre optimizasyonu ve eğitim sonrasında da doğruluğun doğru bir şekilde ölçülmesini sağlamaktadır. Şekil 3.5 verilerin Eğitim (80%), Test (10%) ve Doğrulama (10%) olmak üzere üç bölüme ayrıldığını göstermektedir.

Veri seti aşağıda belirtilen oranlarda ayrılmıştır.

- **Eğitim Veri Seti(%80):** Modelin öğrenme süreci için kullanılır. Bu veri seti, modelin özellikleri ile tanımla etiketler arasında ilişkinin modele öğretilmesini sağlar.
- **Doğrulama Veri Seti(%10):** Hiperparametreleme sürecinde ve aşırı öğrenmesinin önlenmesinde kullanılır. Doğrulama veri seti, modelin eğitim esnasında karşılaşmadığı verilerden oluşur ve bu durum modelin genelleme performansının değerlendirilmesinde kritik rol oynar.
- **Test Veri Seti(%10):** Modelin eğitim sonrasındaki performansının değerlendirilmesinde kullanılır. Test veri seti, modelin daha önce karşılaşmadığı verilerden oluşur.

3.5 TinyML

TinyML, kaynak sınırlı sistemler üzerinde yapay zeka modellerinin çalıştırılmasını mümkün kılan bir teknoloji alanıdır. İsmi, “Tiny”(küçük) ve “ML” (makine öğrenimi) kelimelerinin birleşiminden alır. TinyML özellikle düşük güç tüketimi, düşük hafıza ve düşük hesaplama gücüne sahip olan mikrodenetleyici cihazlarda makine öğrenmesi işlemlerinin yapılması için kullanılır. TinyML, büyük ve karmaşık modellerin niceleme yöntemi ile küçültülmesi ile bir çok yeniliği beraberinde getiren yeni bir alandır. TinyML, karmaşık modelleri miliwatt’ın altında güç tüketen sistemlere entegre eder. Bu sayede, verilerin gerçek zamanlı olarak yorumlanmasını sağlar, bu durum gizlilik, gecikme ve maliyet açısından büyük avantajlar sunar (Alajlan ve Ibrahim 2023b).

3.5.1 TinyML tanımı

TinyML, büyük ve karmaşık makine öğrenimi algoritmalarını kaynak sınırlı sistemler üzerinde çalıştırmayı mümkün kılan teknolojidir. Büyük modelleri bir dizi optimizasyon ve niceleme algoritmasından geçirdikten sonra bu modeller mikrodenetleyiciler üzerinde çalışabilir boyuta dönüştürülür. TinyML, gömülü sistemler ile makine öğrenmesinin kesiştiği bir alandır (David vd. 2021). Pete Warden TinyML kitabında “bir çok çevresel etkinin cihaz üzerinde analizini yapabilen, makine öğrenimi algoritmalarına sahip tüm teknikler, araçlar ve yöntemler” olarak tanımlar. Ayrıca bu

sistemlerin genellikle 1 miliwatt altında çalıştığını ve IoT cihazlar ile kullanıma çok uygun olduğunu belirtir.

Sonuç olarak TinyML, düşük güç kapasitesine sahip olan mikrodenetleyiciler ve benzeri cihazlar üzerinde bir sinir ağı modelinin uygulanması olarak tanımlanır.

Bu çalışmanın ana odağı, düşük güçlü gömülü cihazlar üzerinde makine öğrenmesi algoritmaları çalıştırılmasına imkan tanıyan TinyML'dir. Çalışmada TinyML kullanarak optimize edilen ve nicelenen makine öğrenmesi modeli, kaynak sınırlı bir cihaz üzerinde EKG sinyalinin aritmi tespitini yapacaktır. Çalışma, TinyML'in temel prensiplerini, avantaj ve dezavantajlarını ve uygulama alanlarını detaylı bir şekilde ele alacaktır.

3.5.2 TinyML'in tarihçesi

21. yüzyılda IoT'nin yaygınlaşmasıyla birlikte makine öğrenmesi algoritmaları, bulut sunuculara bağımlıydı ve bu durum gecikme, gizlilik sorunları ve yüksek enerji tüketimine sebep olmaktaydı (Xue vd. 2020). Sonraki yıllarda ise endüstride ve akademik makalelerde TinyML terimi bir paradigma olarak ortaya çıktı (Elhanashi vd. 2024). Bu durum, özellikle enerji tüketimi ve hafıza gereksinimini azaltarak, gömülü sistemlerde makine öğrenmesi algoritmalarının koşturulmasını mümkün kılan yeni bir sürece işaret ediyordu.

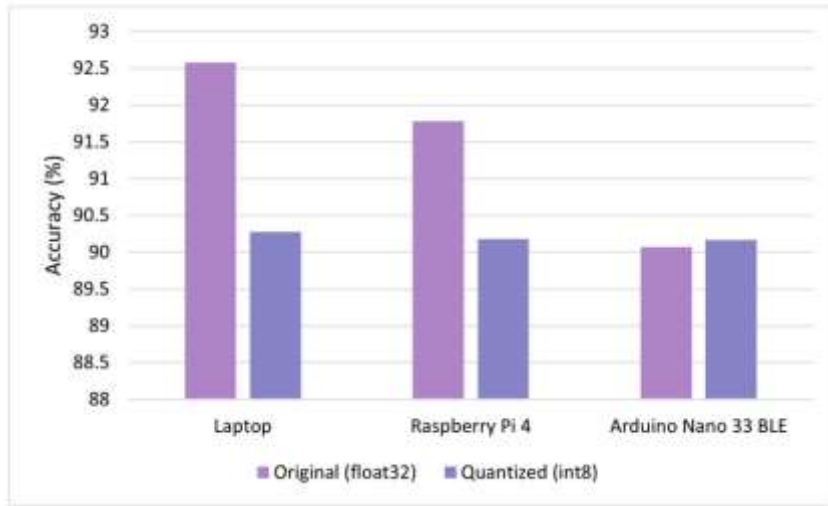
Son yıllarda Google, Arm, Arduino gibi şirketlerin kaynak sınırlı sistemler ve makine öğrenmesi algoritmalarının entegrasyonuna yaptıkları yatırım ile birlikte TinyML alanındaki gelişmeler hız kazandı. Günümüzde Tensorflow Lite for Microcontrollers Edge Impulse gibi açık kaynak projeler ve TinyML Foundation gibi girişimler geliştiricilerin TinyML'e erişimini hızlandırdı. Teknoloji araştırmalar şirketi ABI Research'e göre 2030 yılına kadar TynymL destekli 2,5 milyar adet yapay zeka yonga seti üretilecektir. Bu cihazların ana özellikleri düşük güç tüketimi, düşük gecikme ve düşük üretim maliyeti olacaktır (Alajlan ve Ibrahim 2022).

TinyML'in teknolojinin en yaygın kullanıldığı alanlardan biri anahtar kelime algılama uygulamalarıdır. Günümüzde bir çok şirket TinyML'i sistemlerinde uyandırma kelimelerinin yakalanması amacı ile kullanılmaktadır. Amazon, Apple, Google ve diğer bir çok şirket kullanıcılarının sesli asistana erişimini sağlayan "Hey Siri", "Alexa", "Hey Google" gibi uyandırma kelimelerini (wakewords) TinyML yaklaşımları ile yakalar (Chen vd. 2014). Bu sayede cihazlar, sunucu gereksinimi olmadan, düşük enerji tüketimiyle sürekli aktif olarak kalabilir. TinyML bu uygulamada sunucu yükünün azaltılması ve kullanıcı gizliliği konularında önemli rol oynar. Bu yararlar, IoT ve makine öğrenmesi alanlarında verinin uç cihazlarda işlenmesini sağlayarak gizlilik ve verimlilik odaklı sistemlerin oluşturulmasını sağlar.

3.5.3 TinyML'nin temel özellikleri

Daha önce belirtildiği gibi, TinyML, yapısı gereği yüksek enerji ve hesaplama gücü gerektiren makine öğrenmesi algoritmaları ile düşük güç gerektiren ve düşük maliyetli mikrodenetleyici tabanlı cihazları birbirine entegre ederek yeni bir dönemin başlamasına sebep olmuştur.

- **Düşük Güç Tüketimi:** TinyML, işlenmesi gereken veriyi azaltarak, sistemin daha az enerji tüketmesini sağlar. TinyML algoritmaları hesaplama yükünü model performansı ile dengeler, bu üzerinde çalıştığı cihazın enerji verimliliğini artırır.(Abadade vd. 2023)
- **Düşük Maliyet:** TinyML, makine öğrenmesi modellerinin düşük enerji ihtiyacına sahip, düşük maliyetli mikrodenetleyicilerde çalışmasına olanak sağlar. Bu durum makine öğrenmesi çözümlerinin daha büyük kitlelere ulaşmasını olanaklı kılar. Yasmine vd (2023), aynı modeli 1000 dolarlık bir bilgisayar, 30 dolarlık Raspberry Pi 4 Model B ve 32 dolarlık Arduino Nano 33 BLE Sense cihazları üzerinde çalıştırmış ve performans karşılaştırması yapmışlardır. Çalışmada cihaz maliyetindeki düşüşe rağmen model doğruluğunda ciddi bir düşüş ile karşılaşılmamıştır (Abu Adla vd. 2023b). Şekil 3.5'da nicelme yöntemlerinin farklı platformlar üzerinde model performansına olan etkisi gösterilmiştir.



Şekil 3.5. Nicelme seviyesinin model performansına etkisi (Abu Adla vd. 2023b)

- **Hızlı ve Gerçek Zamanlı İşleme:** TinyML, bulut sunuculara bağımlılığı ortadan kaldırarak hızlı ve gerçek zamanlı olarak işlem yapma kapasitesine sahiptir. TinyML'in uç cihazlar üzerinde işlem yapması, bant genişliği ve

bağlantı sorunlarına bağlı problemleri ortadan kaldırır. Bu durum, özellikle sağlık uygulamaları gibi yanıt süresinin kritik olduğu durumlarda büyük avantaj sağlar.

- **Gizlilik ve Güvenlik:** TinyML, verilerin işlenmesi için sunucu gereksinimini ortadan kaldırdığı için tüm verileri ilgili cihaz içerisinde tutar, bu nedenle hassas bilgileri buluta göndermek zorunda kalmaz. Bu özellik, TinyML'in veri koruma yönetmelikleri açısından uyumluluk zorunluğu olan sistemlere entegrasyonunu mümkün kılar (Abadade vd. 2023).
- **Büyük Ölçekli Uygulamalar:** TinyML, endüstriyel IoT ve mobil uygulamalar benzeri görece olarak büyük sistemlerin bir parçasını oluşturmak için uygundur. Bu, gerektiğinde cihaz üzerinde hesaplama yapılmasına sağlayarak sisteme bir çok konuda destek olabilir.

3.5.4 TinyML'nin yazılım araçları

TinyML'deki gelişmelere paralel olarak, günümüzde TinyML uygulamalarının geliştirilebileceği bir çokaraç geliştirilmiştir. Bu araçlar, mikrodenetleyiciler üzerinde çalışacak makine öğrenmesi modellerini geliştirmek, optimize etmek ve dağıtımının sağlanması için kullanılır. Günümüzde sık kullanılan bazı araçlar aşağıdaki gibidir.

- **Tensorflow Lite For Microcontrollers(TFLM):** Google tarafında gerçekleştirilen bu araç, makine öğrenmesi modellerinin mikrodenetleyiciler üzerinde çalışmasını sağlar. Tensorflow Lite For Microcontrollers(TFLM), gömülü sistemler üzerinde makine öğrenmesi modellerinin çalıştırılmasını mümkün kılan bir çerçevedir (David vd. 2021). Bu çerçeve, tensorflow türündeki modellerin TFLM dönüşümünü yapabilir. Pete Warden, TinyML kitabında TFLM'nin temel yeteneklerinden birinin modeller üzerinde niceleme(quantization) ve budama(pruning) gibi işlemleri destekleyerek, hafif ve hızlı çalışan modeller yaratabilmesi olduğunu vurgulamıştır.
- **X-CUBE-AI:** X-CUBE-AI, STM tarafından oluşturulan yazılım kütüphanesidir. Gömülü sistemler üzerinde yapay zeka modellerinin koşturulmasını mümkün kılar. Bu kütüphane, daha önceden eğitilmiş modellerin optimizasyon ve niceleme işlemleri ile birlikte simüle edilmesini de mümkün kılar. Yapay zekamodelinin cihaza entegre edilmesi esnasında "Application Template" seçeneğinin seçilmesi, kullanıcıya temel bir proje yapısı sunar. Bu, algoritmanın oluşturulması aşamasını hızlandırır ve kullanıcı kaynaklı hataları azaltır.
- **Edge Impulse:** Edge Impulse, gömülü cihazlarda yapay zeka modellerinin geliştirilmesini ve dağıtılmasını sağlayan bir platformdur. Kullanıcılarına eğitim, optimizasyon ve kod üretimi gibi bir çok araç sunar. Ayrıca veri toplama ve modelin cihaza yerleştirilmesi aşamalarında da kullanıcılarına yardımcı olur.
- **Arduino TinyML:** Arduino TinyML kütüphanesi, gömülü sistemler üzerinde yapay zeka algoritmalarının koşturulmasını sağlayan bir araçtır. "Arduino TensorFlowLite" ve "Harvard TinyMLx" kütüphaneleri açık kaynaklı olması sebebi ile geliştirici kitlesi tarafından kolaylıkla erişilebilir. Literatürde bulunan bir çok çalışmada kullanılmış olup, büyük çoğunlukla Arduino Nano 33 BLE Sense cihazı üzerinde kullanılmıştır.

3.5.5 TinyML'in zorlukları ve sınırlamaları

TinyML'in bir çok avantajının yanı sıra bazı sınırlama ve dezavantajları bulunmaktadır. Bu zorlukların bir çoğu, kullanılan cihazın kaynaklarının sınırlı olmasından kaynaklanmaktadır. Bu çalışmada, kullanılan cihazın sınırlarından dolayı ortaya çıkan zorluklar, veri setinin ve modelin bir çok kez küçültülmesi, nicelenmesi ve budanması ile aşılmıştır. Modellerin küçültülmesi ve optimize edilmesi hafıza ve enerji tüketimini azaltırken, model doğruluğunun düşmesine sebep olmaktadır (Yelchuri ve Rashmi 2020). TinyML kullanımına bağlı olarak ortaya çıkabilecek zorluklar ve sınırlamalar aşağıdaki gibidir.

- **Donanım Kısıtlamaları:** Donanım kısıtlamaları, hesaplama gücü ve depolama olarak 2 alt başlık altında incelenebilir. Bu iki alt başlık için, uygulamalarda kullanılan cihazların Flash hafızası, RAM hafızası ve işlemci hızı parametreleri incelenmelidir. Alajlan ve Ibrahim (2022) çalışmasında, literatürde bulunan TinyML çalışmalarında kullanılmış cihazların büyük çoğunluğunun Flash hafızasının 256KB - 2MB arasında, RAM hafızasının 32KB - 2MB arasında ve işlemci hızlarının 10MHz ile 400MHz arasında olduğunu belirtmiştir. Bu makine öğrenmesi modellerinin bilgisayar ortamında eğitildiği ve TinyML kullanımı ile gömülü sistemler üzerine aktarıldığı göz önünde bulundurularak, Alajlan'ın çalışmasında belirttiği cihaz özellikleri bir bilgisayar ile karşılaştırıldığında kaynak farkı göze çarpmaktadır.
- **Model Performansı:** Model performansı, donanım kısıtlamalarına bağlı olarak yapılan niceleme, budama ve küçültme işlemlerine bağlı olarak düşer. Burada yapılan küçültme işlemleri, veri setine uygulananlar ve modele uygulanan teknikler olarak iki alt başlıkta incelenebilir. Bu çalışmada mikrodenetleyici cihaz üzerinde yapay zeka algoritmasının yanı sıra dalgacık dönüşümü hesaplamalarının da yapıyor olmasından dolayı, veri setine uygulanan küçültme işlemleri kritiktir. Veri setinin küçültülmesi ön işleme sürecinde hesaplama yükü ve depolama gereksinimini azaltır. Ayrıca modelin giriş ve çıkış katmanlarının büyüklüğünü ve tahmin için gerekli olan hesaplama gücü ihtiyacını azaltır. Ancak bunların en önemli sonucu model performansındaki düşüştür. Bir diğer adım ise TinyML'in ana fikri olan model niceleme ve budama işlemleridir. Yapılan işlemler Model Dönüşümü ve Niceleme başlığı altında ayrıntılı olarak işlenecektir. Ancak bu işlemin sonucu model performansının düşmesidir. Sonuç olarak, TinyML çalışmalarında performans ve sınırlar arasında dengesi sağlamak, kullanılan cihazın sınırları düşürüldüğünde büyük bir zorluk oluşturur. Model küçültme, niceleme ve budama işlemlerinin yapılma süreci dikkatli bir çalışma gerektirir.
- **Batarya Süresi:** TinyML uygulamaları genellikle taşınabilir ve sınırlı enerji kaynağına sahip batarya ile çalışan cihazlar üzerinde bulunur. Ön işleme ve modelin çalışması esnasındaki hesaplama yükü cihazın enerji tüketimini doğrudan etkiler. Modelin daha az enerji tüketecek şekilde optimize edilmesi batarya süresini artırabilir ancak bu genellikle model performansını düşürür.

3.5.6 TinyML'in çalışmamızdaki rolü ve önemi

Bu çalışmanın ana odağı, MATLAB'da oluşturulan makine öğrenmesi ile aritmi sınıflandırma algoritmasının, TinyML kullanılarak mikrodenetleyici üzerinde çalıştırılmasıdır. Bu yaklaşım, geliştirilen sistemin sunucu bağımsız olarak çalıştırılmasını sağladığı için kritik avantajlar sunar, çünkü EKG verileri buluta gönderilmeden uç cihaz üzerinde işlenir.

3.6 Derin Öğrenme Modelinin Oluşturulması

Çalışmada LSTM ve CNN tabanlı 2 farklı model geliştirilmiştir ancak CNN ağının boyutundan dolayı LSTM ağı tercih edilmiştir. Her iki ağın sonuçları bulgular ve tartışma Bu çalışmada makine öğrenmesi modelinin oluşturulması için MATLAB programı kullanılmıştır. MATLAB programında kullanılan derin öğrenme katmanları ve eğitim seçenekleri açıklanırken MATLAB kaynakları kullanılmıştır (Anonymous 4).

Kullanılan katmanlar ve açıklamaları aşağıdaki gibidir.

- **sequenceInputLayer:** MATLAB programında bulunan bir derin öğrenme katmanıdır ve sıralı veri işleme görevlerinde kullanılır. Genellikle doğal dil işleme, fizyolojik sinyal analizi gibi zaman bağımlı sinyaller için kullanılır. Bu katman, zaman serilerinin doğru modellenmesi için genellikle LSTM katmanı ile birlikte giriş katmanı olarak kullanılır. Bu çalışmada sequenceInputLayer ve dalgacık dönüşümünün birlikte kullanılması istatistiksel bir sinyal olmayan EKG sinyalinin işlenmesinde oldukça etkili bir yaklaşım olabilir.
- **flattenLayer:** Bu katman, çok boyutlu verileri bir vektöre dönüştürür. Katmanın asıl rolü, girdi verisinin şeklinde değişiklik yaparak sonraki katmanlara uyumlu hale getirmektedir.
- **lstmLayer:** LSTM, sıralı verilerin öğrenilmesi için kullanılan bir derin öğrenme katmanıdır. LSTM katmanları RNN katmanlarına göre daha gelişmiş katmanlardır. LSTM'ler geçmiş bilgileri aklında tutarak daha uzun zaman alan bağımlılıkları öğrenebilir. Bu sebeple LSTM katmanları, istatistiksel ya da birikimli olmayan, konuşma fizyolojik sinyal gibi anlık olmayan ve uzun süren sinyallerin öğrenilmesinde kullanılır.
- **dropoutLayer:** Dropout katmanı, derin öğrenme ağlarında aşırı uyum sorununu engellemek için kullanılır (Y. Zhou vd. 2023). Bu katman, eğitim sırasında rastgele olarak bazı nöronları geçici olarak devre dışı bırakarak, eğitilen modelin daha genel hale gelmesini sağlar.
- **fullyConnectedLayer:** Yaygın olarak kullanılan bir katmandır ve genellikle derin öğrenme ağının son katmanlarında yer alır. Modelin sınıflandırma işlemlerini gerçekleştirebilmesi için kullanılır.
- **softmaxLayer:** Softmax katmanı genellikle modelin son katmanında ve sınıflandırma algoritmalarında kullanılır. Softmax katmanı, girdisine softmax fonksiyonu uygular. Bu fonksiyon, çıktı katsayıları $0 \leq c_i \leq 1$ ve $c_1 + c_2 + \dots = 1$

olmak üzere düzenler (Carlini ve Wagner 2017). C kaysayısı, her bir sınıfa ait olasılığı temsil eder ve tüm olasılıkların toplamı her zaman 1'e eşittir.

- **classificationLayer:** ClassificationLayer katmanı ağıın son katmanı olarak bulunur. Bu katman, bir ağıın sınıflandırma görevinin gerçekleştiği katmandır. softmaxLayer katmanında olasılıklara dönüştürülmüş sonuçlar, bu katmanda olasılıklarına göre sınıflara atanır.

Kullanılan eğitim seçenekleri ve açıklamaları aşağıdaki gibidir.

- **'adam':** Adam (Adaptive Moment Estimation) optimizier, öğrenme modellerinde eğitim sürecini kısaltmak için kullanılan bir algoritmadır. 2015 yılında Kingma ve Ba tarafından tanıtılmıştır. Adam, moment tabanlı ve adaptif öğrenme tabanlı yöntemleri birleştirerek kararlılığı artırmayı hedefler (Kingma ve Ba 2017).
- **'MaxEpochs':** Epoch derin öğrenme modellerinde eğitim veri seti içerisinde bulunan tüm örneklerin model eğitimi için 1 kez kullanılması anlamına gelir. Bu çalışmada epoch oranı hiperparametre ile belirlenmiştir. Epoch'un gerekenden az olması yetersiz öğrenmeye sebep olurken yüksek epoch ise aşırı uyuma sebep olur.
- **'MiniBatchSize':** Batch size, makine öğrenmesi algoritmalarında kullanılan bir terimdir ve bir döngüde modele verilecek örnek sayısını ifade eder. Model eğitiminde Batch size, öğrenme süresini ve aşırı öğrenme miktarı üzerinde etkilidir (Radiuk 2017). Radiuk çalışmasında yüksek batch size seçiminin model doğruluğunu hesaplama yükü ile birlikte artıracağını belirtmiştir.
- **'InitialLearnRate':** Öğrenme oranı, bir model ağırlığının, eğitim sürecinde karşılaşılan bir hata sonrasında ne kadar değiştirileceğini belirten katsayı olarak tanımlanabilir (Wilson ve Martinez 2001). Öğrenme hızının model doğruluğu ve eğitim süresi üzerinde önemli etkileri bulunmaktadır.
- **'SequenceLength':** Sequence length, sıralı veriler ile çalışan Recurrent Neural Networks (RNNs) veya Long Short-Term Memory (LSTM) gibi modellerde kullanılan bir parametredir. Sequence length, ağıın geçmişten ne kadar veri tutacağını belirler. Daha büyük sequence length, veri seti içerisinde bulunan daha uzun vadeli ve daha karmaşık özelliklerin yakalanmasını sağlar ancak hesaplama maliyetini artırır (Safwan Mahmood Al-Selwi vd. 2023).
- **'Shuffle':** Shuffle, veri setinin her eğitim döngüsünden (epoch) sonra rastgele olarak karıştırılmasını ifade eder. Veri setinin karıştırılması verinin sıralı bir şekilde öğrenilmesinden dolayı ortaya çıkabilecek problemleri engeller.
- **'Plots':** Eğitim süresinde kayıp (loss) ve doğruluk (accuracy) parametrelerini grafik üzerinde görselleştirmek için kullanılan seçenektir.
- **'ValidationData':** ValidationData parametresi, bir makine öğrenmesi modeli eğitilirken model performansının ölçülmesinde ve aşırı öğrenmenin (overfitting) önlenmesi amacıyla kullanılır. Bu parametrenin kullanılması eğitim esnasında modelin doğrulama veri kümesi üzerindeki performansını izler. Model sadece eğitim veri seti üzerinden optimize edilirse aşırı öğrenme riski artar.

3.7 Model Dönüşümü ve Niceleme

Bir makine öğrenmesi modelinin eğitim aşaması genellikle güçlü ve kaynakları yeterli bir bilgisayar üzerinde yapılır (Cantero vd. 2022). Ancak eğitilen model her zaman bir bilgisayar üzerinde çalışmaz. Bu çalışmada olduğu gibi, gömülü sistemlerde yapay zeka algoritmaları söz konusu olduğunda bilgisayarda yaratılan bir modelin gereksinimlerinin azaltılması gerekir. Bu çalışmada kullanılan gereksinim azaltmak işlemleri 3 alt grupta toplanabilir. Bunlar dönüşüm, niceleme ve budamadır. TinyML çalışmalarında, modelin kaynak sınırlı sistemler üzerinde çalışabilmesini sağlamak için modelin dönüştürülmesi (transformation), nicelenmesi (quantization) ve budanması (pruning) büyük öneme sahiptir.

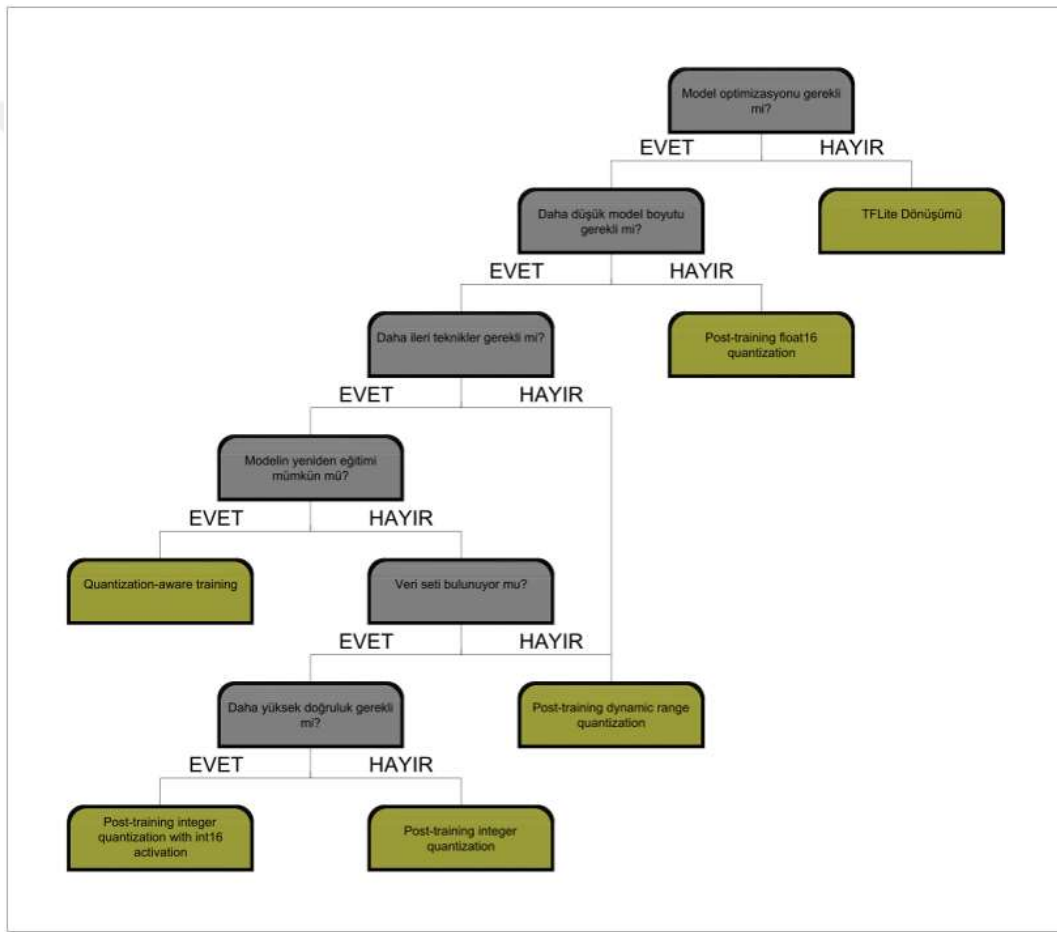
Bir modelin hesaplama gücü ve gerekli hafıza alanı bakımından küçültülmesi çıkarım süresi azaltmaktadır ancak daha küçük modeller bazı dezavantajlar getirir. Bu çalışmada karşılaşılan başlıca dezavantaj model performansındaki düşüştür. Bu başlık altında bir modelin cihaz üzerinde çalıştırılması için gerekli olan hesaplama gücü hafızanın düşürülmesini amaçlayan işlemler incelenecektir. Ayrıca bu işlemlerin model performansı üzerindeki etkileri Bulgular başlığı altında ayrıntılı olarak incelenecektir.

Google tarafından geliştirilen Tensorflow, en popüler derin öğrenme kütüphanelerinden biridir. Tensorflow, Tensorboard eklentisi ile görselleştirme yeteneği olan, esnek bir kütüphanedir. Tensorflow, NVIDIA'nın bir platformu olan CUDA'da ileri düzeyde optimize edilmiş olan C++ kodu ile yazılmıştır (Pang vd. 2020). CUDA, GPU'lar için algoritmalar oluşturulan, C/C++ programlama dili benzeri bir programlama ortamıdır (Chakrabarti vd. 2012). Tensorflow, bir kütüphaneden öte bir platform olarak tanımlanabilir. Çünkü yalnızca makine öğrenmesi algoritmalarının eğitimi için değil, bir modelin geliştirilmesi aşamasındaki bir çok aşama için kullanılabilir (Abadi vd. 2016). Tensorflow, modeller için sunduğu çeşitli API'ler ve özellikle TPU desteği ile özellikle büyük veriler için idealdir (Abadi vd. 2016). Bir diğer avantajı ise geniş bir kullanıcı kitlesine sahip olmasıdır, bu sebeple bilgi kaynağına erişim kolaydır.

Çalışmada TensorFlow Model Optimization Toolkit kullanılmıştır. Bu araç, makine öğrenmesi modellerini optimize etmek için kullanılır. Yapılan işlem, bulut ve kenar cihazlardaki tahmin süresini ve maliyetini azaltma amacıyla işlem yükü, bellek kullanımı ve güç tüketiminin azaltılması olarak özetlenebilir. Niceleme yöntemleri eğitim sırasında niceleme ve eğitim sonrasında niceleme olarak 2 sınıf altında incelenebilir. Bir çok çalışmada genellikle eğitim sonrası niceleme kullanılsa da, eğitim sırasında niceleme genellikle daha iyi sonuç vermektedir.

Şekil 3.6, yapay zeka modellerinin dönüşüm ve niceleme süreçlerinde kullanılabilecek bir akış diyagramı göstermektedir. TinyML ile modelin optimize

edilmesi esnasında kullanılabilecek farklı seçenekler ve bu seçeneklerin koşulları paylaşılmıştır. Diyagram, “model optimizasyonu gerekli mi?” sorusu ile başlar. Eğer model boyutu küçültülmek istenmiyorsa sadece TFLite dönüşümü yapılabilir. Sonraki adımda ise model boyutunun ne kadar küçültülmesi gerektiğine karar verilir. Bu çalışmada LSTM ağı kullanıldığı için float16 niceleme yapılmamıştır. LSTM ağlarının yapısal karmaşıklığından dolayı float16 nicelemesi LSTM ağlar için kullanılmamaktadır (Crocioni vd. 2020). Bu çalışmada integer quantization tekniği kullanılmıştır. Kullanılan niceleme yönteminin model üzerindeki etkileri bulgular başlığı altında paylaşılmıştır.



Şekil 3.6. Model niceleme akış diyagramı (Anonymous 5)

3.8 STM32H750B-DK Tabanlı Platform

Bu çalışmada STM32H750B Discovery Kit kullanılmıştır. STM32H750B-DK, üzerinde yüksek performanslı Arm Cortex M7 çekirdeğine sahip STM32H750 mikrodenetleyicisi bulundurmaktadır. Kart bir çok çevresel birimin yanı sıra üzerinde

SDRAM ve Quad-SPI Flash hafıza bulundurmaktadır. Mikrodenetleyicinin frekansı 480 MHz olup, üstün performans sağlar. Şekil 3.7’de, STMicroelectronics tarafından üretilen bir STM32H750B geliştirme kartı bulunmaktadır.



Şekil 3.7. STM32H750B-DK Geliştirme Kartı (Anonymous 6)

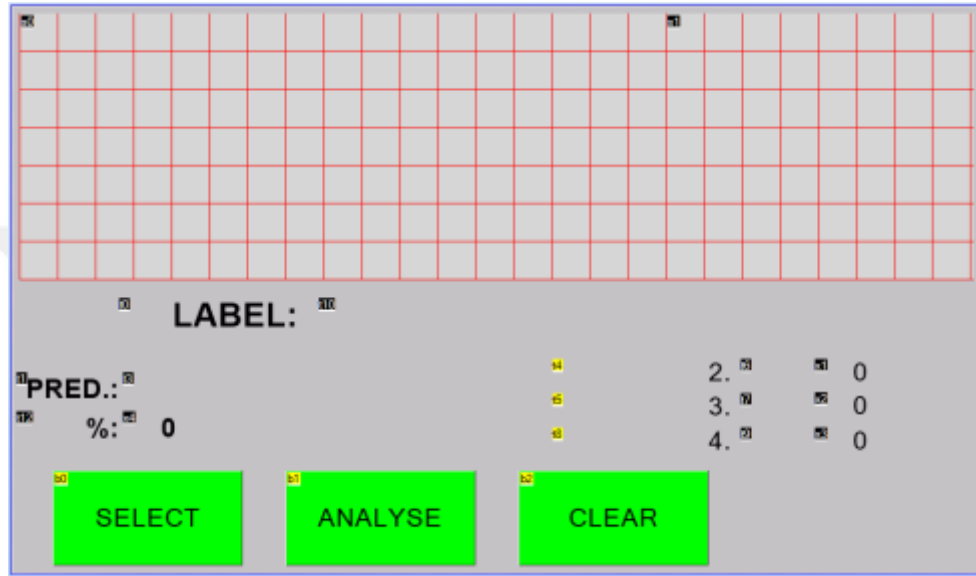
Dahili olan 128 kB flash hafıza ve 1 MB RAM hafızasının yanı sıra kart üzerinde 2×512 -Mbit Quad-SPI flash hafıza ve 128-Mbit SDRAM bulundurmaktadır. Ek bellekler bulundurması özellikle TinyML uygulamaları için avantaj sağlamaktadır. İki adet 512 Mbit (128 MB) Quad-SPI flash hafıza, veri setlerinin ve makine öğrenmesi model ağırlıklarının depolanması için idealdir. 128 Mbit büyüklüğündeki SDRAM ise özellikle büyük matrislerin kullanıldığı hesaplamalarda avantaj sağlar. Bu çalışmada mikrodenetleyici üzerinde dalgacık dönüşümü hesaplamaları yapılmaktadır. SDRAM’ın harici olarak sağladığı kapasite, hesaplamalar esnasında oluşturulan matrislerin işlenmesi açısından önemli bir avantajdır.

STM32H750B-DK geliştirme kartının ön yüzünde 4.3 inch kapasitif ekran bulunmaktadır. Ancak ekran boyutunun yeterli büyüklükte olmadığı düşünüldüğü için cihaz üzerinde bulunan dahili ekran yerine 10.1 inch NEXTION NX1060P101-011C-I ekran kullanılmıştır. Kullanılan ekranın ekran görüntüleri ise Bulgular başlığı altında detaylı olarak incelenecektir.

Çalışma sonucunun gerçek zamanlı olarak görselleştirilmesi için Nextion NX1060P101-011C-I ekran kullanılmıştır. Kullanılan ekran, 10.1 inch boyutunda dokunmatik bir ekrana sahiptir. Nextion ekranın bağımsız bir işlemci ve hafızaya sahip olması mikrodenetleyicinin yükünü azaltır. Arayüz tasarımı Nextion Editor programı kullanılarak tasarlanmıştır. Ekran ve geliştirme kartı UART protokolü ile haberleşmektedir.

Şekil 3.8’de Nextion Editor’den bir ekran görüntüsü bulunmaktadır. Üst kısımda MIT-BIH aritmi veri seti içerisinde bulunan herhangi bir kayıt çizdirilir. Alt kısımda “LABEL:” yanında anlık olarak çizdirilmekte olan EKG sinyalinin gerçek etiketi yer

alır. Alt kısımdaki “PRED.:



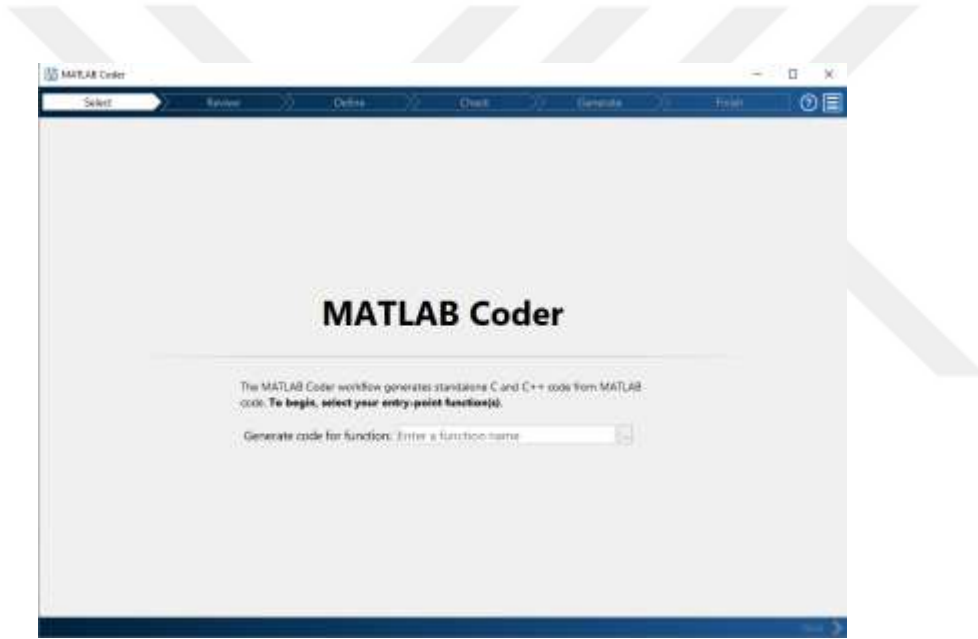
Şekil 3.8. Nextion Editor Arayüzü

Daha öncesinde belirtildiği üzere bu çalışmada verilerin işlenmesi, modelini eğitimi ve doğrulaması için MATLAB kullanılmıştır. Veri ön işlemenin bir parçası olan dalgacık dönüşümü, sinyalin özellik çıkarımının gerçekleştirilmesi için kullanılmıştır. Elde edilen bu özellikler makine öğrenmesi modelinin daha doğru ve hızlı eğitilebilmesi için model girişine gönderilmiştir. Geliştirilen model bir mikrodenetleyici üzerinde çalıştığı ve model girişinin bir dalgacık dönüşümü fonksiyonunun çıktısı olmasından dolayı, MATLAB programında kullanılan dalgacık dönüşümü algoritmasının aynısının geliştirme kartı üzerinde de yapılması gerekmektedir. Ancak Morlet dalgacık dönüşümünün karmaşıklığı ve optimizasyonu için gerekli olan iş yükünden dolayı algoritmanın geliştirme kartı üzerinde doğrudan geliştirilmesi zorlu ve uzun zaman alacak bir işlemdir.

MATLAB programındaki dalgacık dönüşümü algoritmasının aynısının gömülü sistem geliştirme kartı için geliştirilmesinde MATLAB Coder kullanılmıştır. MATLAB Coder, MATLAB ortamında geliştirilmiş bir algoritma ya da fonksiyonu hedef cihaza uygun bir şekilde C/C++ dillerinde yazabilir. Bu dönüşüm algoritma ve ürün geliştirme

sürecini hızlandırır, özellikle gömülü sistemler için yüksek performansta çalışan uygulamalar geliştirebilir.

MATLAB ile kod üretmek için öncelikle MATLAB Coder eklentisinin yüklenmesi gerekmektedir. MATLAB kütüphanelerinde tanımlı bir fonksiyonun MATLAB Coder ile yeniden kodlanması için fonksiyonun C/C++ Code Generation desteğinin bulunması gerekir. Her MATLAB fonksiyonu MATLAB Coder kullanılarak yeniden oluşturulamaz, bir fonksiyonun MATLAB Coder desteğine sahip olup olmadığını tayin etmek için MATLAB Help Center’da ilgili fonksiyona ait sayfa açılır ve Extended Capabilities bölümü kontrol edilir. Eğer Extended Capabilities bölümünde “C/C++ Code Generation” yazıyor ise ilgili fonksiyon MATLAB Coder ile yeniden kodlanabilir. Şekil 3.9’da bulunan MATLAB Coder arayüzü, hedef cihaz için yeniden oluşturacak fonksiyonunun seçildiği ekrandır.



Şekil 3.9. MATLAB Coder Arayüzü

MATLAB Coder açıldıktan sonra öncelikle yeniden yazılacak projenin dosya yolu belirtilir. Dosya yolu belirlendikten sonra fonksiyon parametrelerinin belirlenmesi gerekmektedir. Burada belirtilen parametrelerin değişken tipinin “single” olarak belirlenmesi daha optimize bir algoritma oluşturulmasını sağlar. Bir sonraki adım ise ayarlar ve hedef cihaz seçim ekranıdır. Hedef cihaz, AMD, ARM, Atmel, Microchip, NXP, Texas Instruments gibi bir çok üreticinin cihazlarını kapsamaktadır. Oluşturma tipi kaynak kodu yada .lib, .dll, .exe, .mex uzantılı olarak oluşturulabilir. Bu çalışmada kaynak kodu olarak oluşturulmuştur.

Ayarlar ekranında hız ve hafıza optimizasyonları gerçekleştirilebilir, özellikle gömülü sistemlerde çalışacak algoritmalar için ayarlar kısmındaki seçimler büyük önem taşımaktadır. Örneğin, sabit veri boyutu kullanımı, gereksiz hafıza kullanımının önüne geçebilir ve işlem hızını artırabilir.

Kod oluşturma işlemi tamamlandıktan sonra üretilen kodun doğruluğunun test edilmesi için .mex dosyasının çalıştırılması önerilir, bu yükleme öncesinde kodun kontrol edilmesini sağlar. Eğer bir uygunsuzluk tespit edilirse kod, gerekli düzenlemeler yeniden yapılarak baştan yaratılabilir. Hazırlanan kaynak kodu paket olarak çıktı alınabilir. MATLAB tarafında oluşturulan örnekler dosyasında ise ilgili algoritmanın ana döngü içerisinde nasıl çalıştırılacağına dair bilgiler bulunmaktadır.

Son olarak, Materyal ve Metod bölümündeki tüm adımlar özetlenecektir. Verinin ön işlemeden geçirilmesinden algoritmanın STM platformu üzerinde çalıştırılmasına kadar geçen süreç kullanılan araç ve yöntemler bağlamında özetlenecektir. Sonrasında ise bu adımların STM32H750B geliştirme kartı üzerinde nasıl birleştirildiği ayrıntılı olarak açıklanacaktır. Model performansı ve program çıktıları başlık altında incelenmeyecek olup, Bulgular başlığı altında incelenecektir.

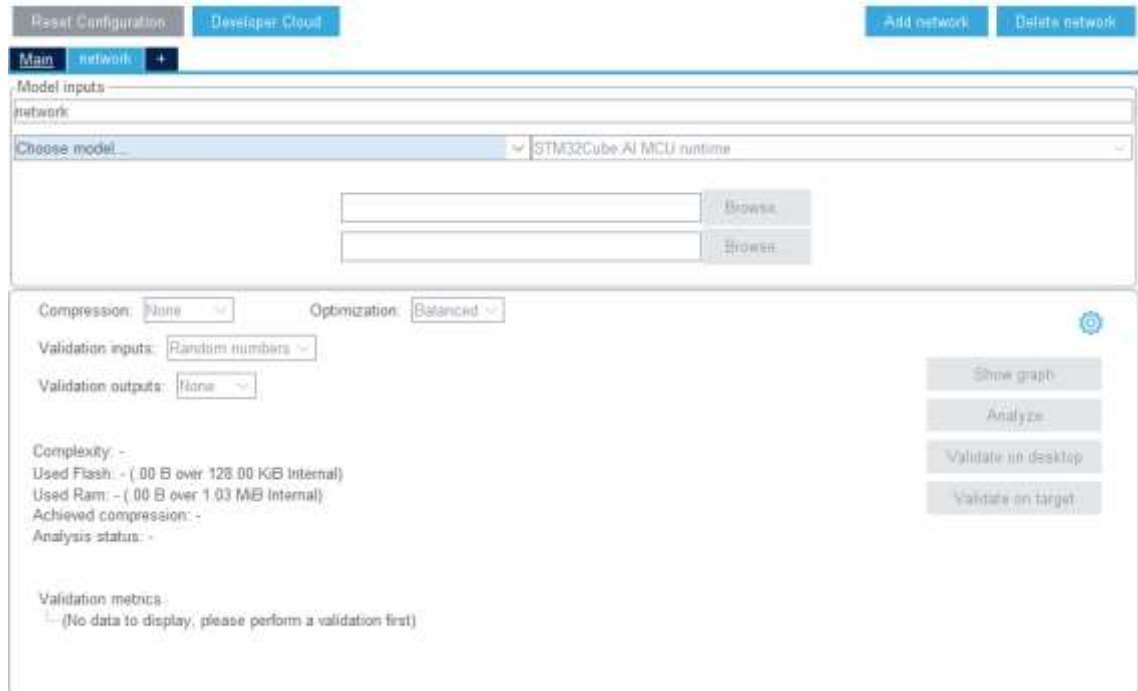
Çalışmada, verilerin ön işlenmesinden, oluşturulan modelin geliştirme kartı üzerinde çalıştırılmasına kadar olan tüm süreç ele alınmıştır. Başlangıçta, veri ön işleme sürecinin adımları olarak, veriler segmente edilmiş, alt örneklenmiş ve dalgacık dönüşümü uygulanarak öznitelik çıkarımı yapılmıştır. Öznitelik çıkarımı sonucunda elde edilen veriler makine öğrenmesi modelinin eğitiminde kullanılmıştır. Bu modelin eğitimi için MATLAB programı kullanılmıştır. Bu süreçte, LSTM (Uzun – Kısa Süreli Bellek) katmanları içeren bir derin öğrenme algoritması kullanılmıştır. Eğitim işlemi tamamlandıktan sonra, model ağırlıkları STM32H750B geliştirme kartına uygun olacak şekilde TinyML optimizasyonları ile nicelenmiştir.

Bir diğer yandan dalgacık dönüşümünü işleminin STM32H750B geliştirme kartı üzerinde de gerçekleştirilebilmesi için MATLAB Programının Coder eklentisi kullanılarak, dönüşüm fonksiyonu hedef cihaz için yeniden oluşturulmuştur. Hedef cihazın seçimi için Code Generation arayüzünde “ARM Compatible” ve “ARM Cortex-M” seçenekleri seçilmelidir. Kod oluşturma işlemi tamamlandıktan sonra dosya çıktısı paket (.pkg) olarak alınmıştır. Oluşturan oluşturulan paket dosyası içerisinde başlık dosyası, kaynak dosyası ve ana fonksiyon örnekleri bulunur. Daha sonrasında ise MATLAB ile oluşturulan dosyanın STM32H750B geliştirme kartına dahil edilir.

Geliştirilen derin öğrenme modelinin STM32H750B geliştirme kartı üzerine entegre edilmesi için öncelikle X-CUBE-AI paketi indirilmelidir. X-CUBE-AI, STM32 mikrodenetleyicileri ile yapay zeka algoritmalarının optimize bir şekilde geliştirilmesi sağlayan kütüphanedir. Keras, TFLite ve ONNX çerçevelerinde oluşturulan modeller kütüphane tarafından desteklenir. Bir derin öğrenme modelinin entegrasyonu için öncelikle X-CUBE-AI indirilir ve STM32CubeIDE yazılıma entegre edilir. Eğitilmiş

model, X-CUBE-AI ile içe aktarılır ve model üzerinde optimizasyon işlemi gerçekleştirilir. Gerekli optimizasyon işlemlerinin tamamlanması ise proje dosyası oluşturulur. Proje dosyası, modelin çalışması için gerekli olan kütüphaneleri ve modelin ağırlıklarını içerir. X-CUBE-AI arayüzündeki test ve doğrulama seçenekleri ile modelin kart üzerindeki performansı test edilir. Gerekirse parametrelerde yeniden düzenleme yapılır.

Şekil 3.10'da eğitilmiş modelin X-CUBE-AI platformuna dahil edilmesinde kullanılan arayüz bulunmaktadır. Kullanıcı, bu ekran ile model dosyasını yükler. Sıkıştırma (**Compression**) ve optimizasyon (**Optimization**) seçenekleriyle performans ve bellek kullanımında ihtiyaca bağlı değişiklikler yapılabilir. Ayrıca model karmaşıklığı, Flash ve RAM hafıza kullanımı gibi detaylara ulaşılabilir. Sağ tarafta yer alan Show Graph, Show graph, Analyze, Validate on desktop ve Validate on target butonları aracılığıyla doğrulama ve performans testleri gerçekleştirilebilir.



Şekil 3.10. X-CUBE-AI Arayüzü

Optimize edilmiş makine öğrenmesi modeli ve MATLAB Coder tarafından oluşturulmuş dalgacık dönüşümü paketinin STM32H750B geliştirme kartına dahil edilmesinden sonra kaynakları farklı olan bu iki yazılımın birleştirilmesi gerekmektedir. Birleştirme işlemi dalgacık dönüşümü sonucunun, makine öğrenmesi modelinin girişi olarak düzenlenmesinden ibarettir. Ancak, bu işlemin oldukça optimize bir şekilde

yapılması gerekmektedir. Bu çalışmada, dalgacık dönüşümü algoritmasının sonuç değişkenleri, makine öğrenmesi algoritmasının giriş değişkenlerine bellek bloklarının kopyalanması yöntemiyle eşitlenmiştir. Süreç ile ilgili çıktılar ve sonuçlar, Bulgular başlığı altında incelenecektir.



4. BULGULAR VE TARTIŞMA

Bu başlık altında çalışmanın bulguları ve her adımda elde edilen sonuçlar tartışılacaktır. Araştırmanın her adımın elde edilen sonuçların incelenmesi yapılacak, sonuçlar çalışmanın hipotezi ile karşılaştırılacak ve gereksinimlerin karşılanıp karşılanmadığı konusunda eleştiriler yapılacaktır. Ayrıca çalışma sırasında karşılaşılmış zorlukların alternatif çözüm yöntemleri incelenecek olup, uygulanan yöntemler güçlü ve zayıf yönleri ile incelenecektir. Elde edilen sonuçları değerlendirilmesi ile gelecekte yapılacak çalışmalara öneriler sunulacaktır.

4.1 Alt Örneklem Sonuçları

Çalışmada alt örneklemenin kullanılması, modelin geliştirme kartı üzerinde çalıştırılmasında büyük avantajlar sağlamıştır. Alt örneklem kullanılmasıdaki amaç, dalgacık dönüşümü için gerekli olan RAM miktarının düşürülmesidir. Alt örneklem sayesinde dalgacık dönüşümünün gerçekleştirilmesi esnasında gerekli olan RAM azaltılmış ve işlemin STM32H750B geliştirme kartı üzerinde de yapılması mümkün kılınmıştır. Alt örneklem işlemi sırasında, sinyalin özelliklerinin korunması oldukça önemlidir. Bu çalışmada, alt örneklem oranının alt sınırı Nyquist Teoremine göre, üst sınırı ise geliştirme kartının RAM kapasitesine göre seçilmiştir. Çalışmada bir EKG sinyalinde 100 Hz üzerindeki frekans bileşenlerinin oldukça az olduğu kabul edildiği düşünüldüğünde, uygulanacak alt örneklemenin frekansının en az 200 olması gerekmektedir. Üst sınır ise daha önce belirtildiği gibi cihaz hafızası ile ilişkilidir. Yüksek olması model doğruluğunu artıran bir etkidir.

$1 \times n$ boyutundaki bir matris ile ifade edilen EKG sinyali, dalgacık dönüşümü uygulandıktan sonra $x \times y$ boyutunda bir matrise dönüşür. x ve y değerlerinin, kaynak kısıtlı bir cihaz olan geliştirme kartı hafızasına uygun olması gerekmektedir. Çizelge 4.1’de farklı uzunluklardaki EKG sinyallerinin satır sayısı, kolon sayısı ve veri boyutu paylaşılmıştır. Bir EKG sinyaline Morlet dalgacık fonksiyonu uygulanması esnasında birçok ara matris oluşturulmaktadır. Oluşturulan ara matrislerin boyutu, EKG sinyali boyutu ile doğrusal olarak artmamaktadır. Örneğin, uzunluğu 9000 olan bir EKG sinyaline, STM32H750B geliştirme kartı üzerinde dalgacık dönüşümü uygulandığında “Cihaz hafızası 3.65 GB aşıldı uyarısı” alınmaktadır. Veri uzunluğu 150 olduğunda ise dalgacık dönüşümü işlemi cihaz üzerinde herhangi bir aksaklık olmadan yapılabilmektedir. Bu durum, hesaplama için gerekli ara matrislerin boyutlarının doğrusal olarak değil, hem satır hem de kolon sayısındaki aynı artıştan dolayı çok daha hızlı artmaktadır.

Çizelge 4.1. EKG sinyali uzunluğu ve dalgacık dönüşümü sonucunun karşılaştırılması

Sinyali Uzunluğu (n)	Satır Sayısı (x)	Kolon Sayısı (y)	Boyut (byte)
100	15	7	840
150	20	10	1600
225	35	8	2240
256	36	8	2304
500	65	8	4160
1000	102	8	6528

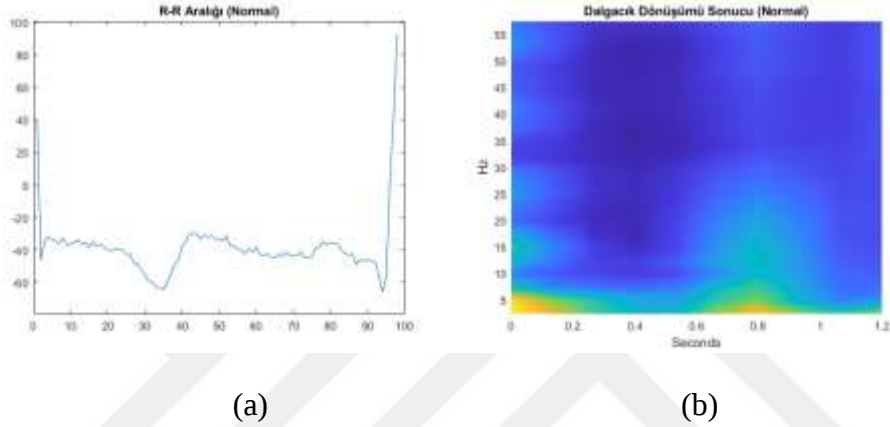
Çizelge 4.1’de gerçek frekansı 360 olan EKG sinyallerinin, farklı frekanslarda alt örneklendikten sonra oluşturulan modellerin TinyML optimizasyonu öncesinde model performansları görülmektedir. Çizelge 4.1, Nyquist teoremine aykırı olacak şekilde örnekleme frekansının düşürülmesinin, model doğruluğunu belirgin bir şekilde düşürdüğünü göstermektedir. Elde edilen sonuçlar, Nyquist teoreminin alt örnekleme işlemi sırasında dikkate alınmasının, hem veri ön işleme adımının kalitesini korumak hem de modelin sınıflandırma performansını optimize etmek açısından kritik öneme sahip olduğunu göstermektedir. Alt örnekleme frekansı, donanım sınırları ile Nyquist teoreminin arasında bir yerde, model performansı en yüksek olacak şekilde seçilmelidir.

4.2 MATLAB Programında Dalgacık Dönüşümü Sonuçları

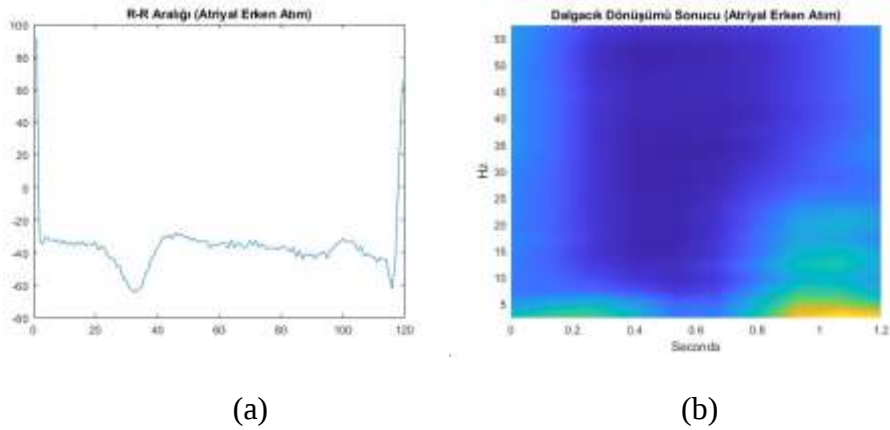
Morlet Dalgacık Fonksiyonu dışında MATLAB Programında bulunan diğer dalgacık dönüşümleri incelenmiş olup, bu çalışmada EKG sinyallere Morlet dalgacık fonksiyonu uygulanarak sinyallerin öz nitelik çıkarımı yapılmıştır. MATLAB programında Morlet dalgacık dönüşümü kullanılarak sinyalin frekans bileşenleri analiz edilmiş, zaman – frekans düzleminde özellikleri elde edilmiştir. Dalgacık dönüşümü sonucunda sinyalin farkı frekans bileşenlerindeki enerji yoğunlukları her bir sınıfın farklılıklarını ortaya koymak için yeterli olmuştur..

Haar, Mexican Hat, Daubechies, Gaussian, Meyer fonksiyonları çalışma Morlet Fonksiyonu seçilmeden önce denenmiştir. Ancak, MATLAB’da yalnızca haar ve Morlet dalgacık fonksiyonları için MATLAB CODEGEN desteği bulunmaktadır. Yapılan denemelerde Haar dalgacık fonksiyonu ile anlamlı bir sonuç elde edilememiştir. Bu nedenle, yüksek performanslı ve iyi zaman-frekans çözünürlüğü ile öne çıkar Morlet dalgacık fonksiyonu tercih edilmiştir.

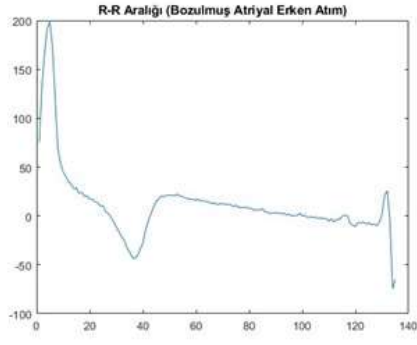
Şekil 4.1 ve Şekil 4.11 arasında yer alan görsellerde, bu çalışmada veri setine dahil edilmiş her bir aritmi sınıfına ait R-R aralıkları ve dalgacık dönüşümü sonuçları yer almaktadır. Her bir şekil, belirli bir ritim sınıfı ile birlikte, o ritim türüne ait dalgacık dönüşümünün skalogram verisini içermektedir. Dalgacık dönüşümü sayesinde sinyalin frekans bileşenlerinin zaman içerisindeki enerji yoğunluklarının frekans bileşenlerine göre dağılımı gözlemlenebilir. Bu veriler farklı sınıflardaki kalp ritimlerinin sınıflandırılmalarında önemli bir rol oynamaktadır. Skalogram verileri özellikle yapay zeka algoritmalarının eğitiminde sıklıkla kullanılmaktadır.



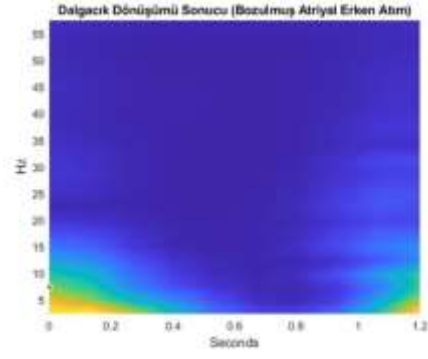
Şekil 4.1. a) “Normal Atım” etiketli R-R aralığı; b) Skalogram



Şekil 4.2. a) “Atrial Erken Atım” etiketli R-R aralığı; b) Skalogram

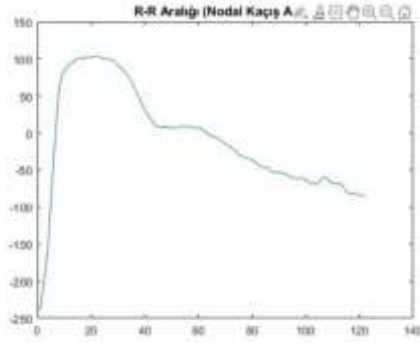


(a)

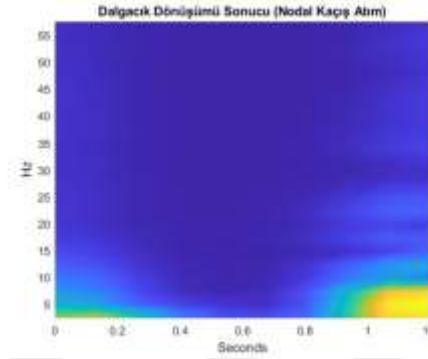


(b)

Şekil 4.3. a) “Bozulmuş Atriyal Erken Atım” etiketli R-R aralığı; b) Skalogram

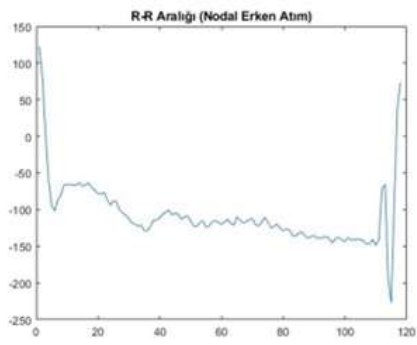


(a)

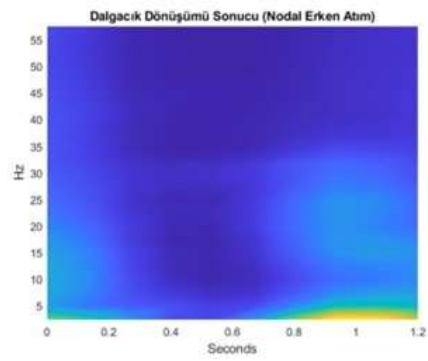


(b)

Şekil 4.4. a) “Nodal Kaçış Atımı” etiketli R-R aralığı; b) Skalogram

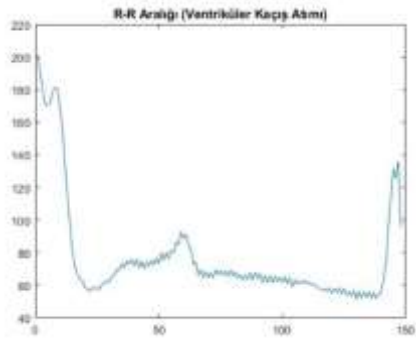


(a)

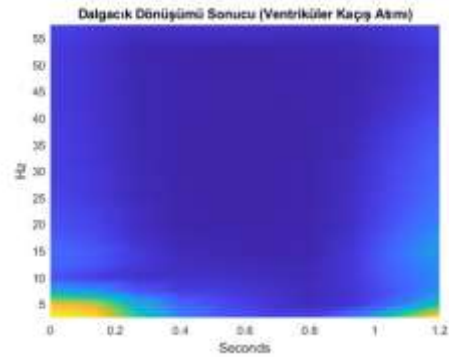


(b)

Şekil 4.5. a) “Nodal Erken Atım” etiketli R-R aralığı; b) Skalogram

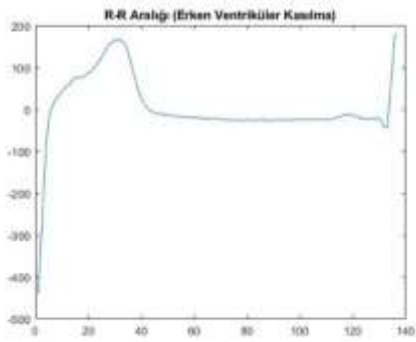


(a)

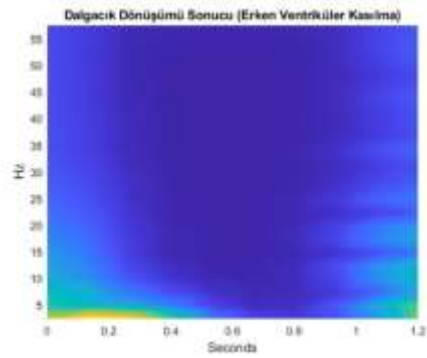


(b)

Şekil 4.6. a) “Ventriküler Kaçış Atımı” etiketli R-R aralığı; b) Skalogram

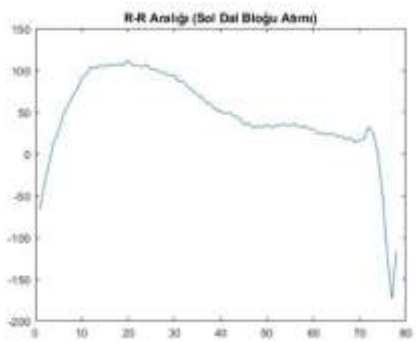


(a)

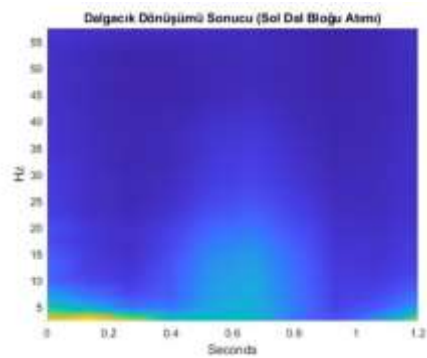


(b)

Şekil 4.7. a) “Erken Ventriküler Kasılma” etiketli R-R aralığı; b) Skalogram

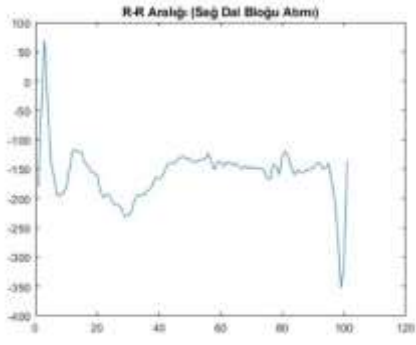


(a)

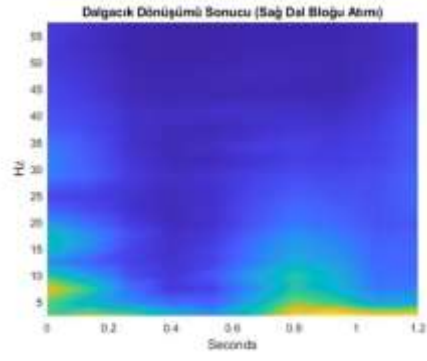


(b)

Şekil 4.8. a) “Sol Dal Bloğu Atımı” etiketli R-R aralığı; b) Skalogram

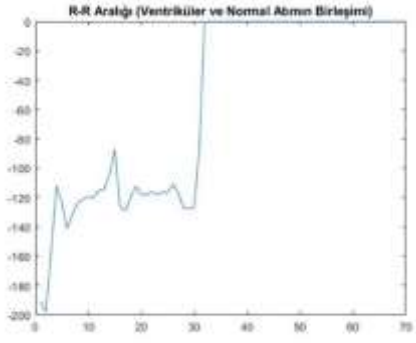


(a)

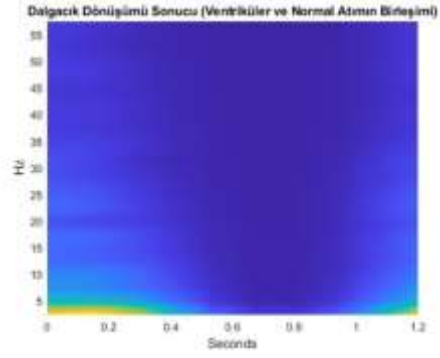


(b)

Şekil 4.9. a) “Sağ Dal Bloğu Atımı” etiketli R-R aralığı; **b)** Skalogram

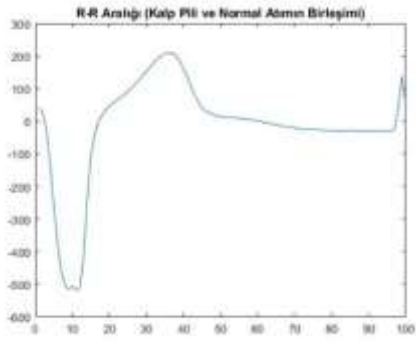


(a)

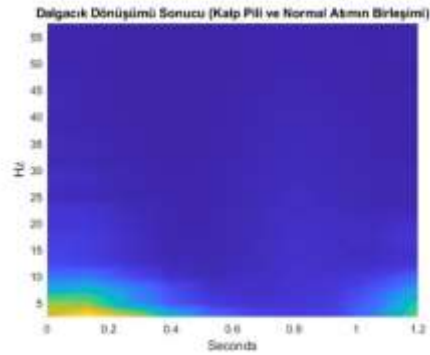


(b)

Şekil 4.10. a) “Ventriküler ve Normal Atımın Birleşimi” etiketli R-R aralığı; **b)** Skalogram



(a)



(b)

Şekil 4.11. a) “Kalp Pili ve Normal Atımın Birleşimi” etiketli R-R aralığı; **b)** Skalogram

Çizelge 4.2’de “Normal Atım” olarak etikelenmiş bir EKG segmentine Morlet Dalgacık fonksiyonu uygulandığında elde edilen sonuç matrisi bulunmaktadır. Sonuç matrisleri ile skalogram verileri aynı bilgiyi farklı şekillerde ifade etmektedir. Başka bir ifade ile skalogram, matrisin görsel karşılığıdır. Matris üzerinde açık renk ile ifade edilen kısımlar ilgili zaman ve frekansta yüksek enerjiyi temsil etmektedir. Skalogramda açık renkler gösterilen bölgeler matristeki büyük değerlere karşılık gelir.

Çizelge 4.2. MATLAB programında dalgacık dönüşümü sonuç matrisi

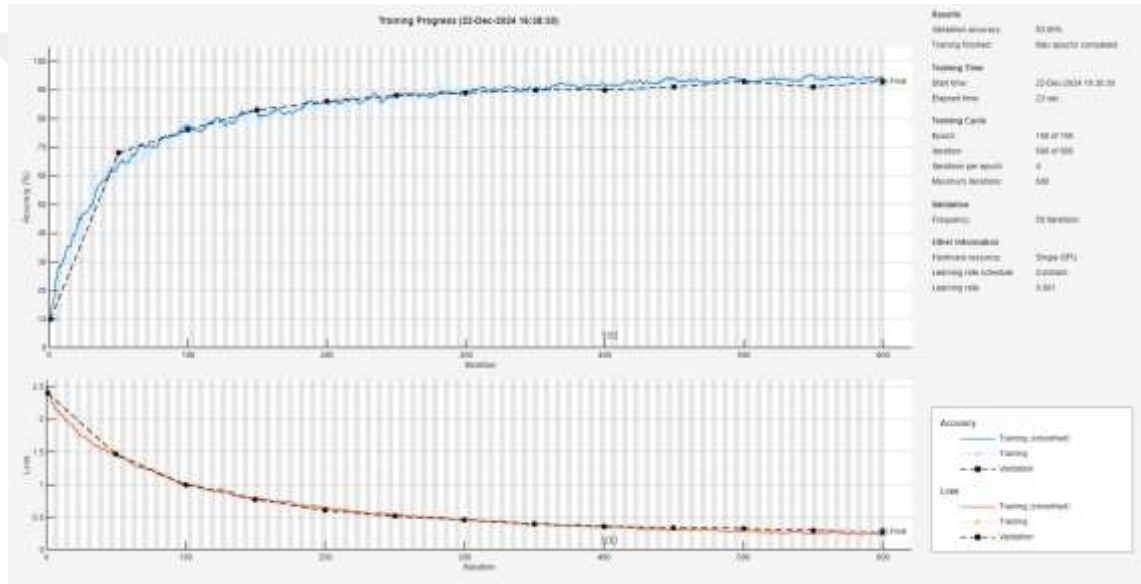
35x8 double								
	1	2	3	4	5	6	7	8
1	105.5075	105.6762	102.3941	96.5306	100.8750	101.5222	93.5965	97.3524
2	8.5491	8.2589	6.9985	6.8071	6.9374	7.5584	4.7294	5.1719
3	6.1620	5.7100	4.7397	4.5309	7.6073	9.8243	7.5425	6.0457
4	4.5202	3.4366	2.7285	2.8627	2.4935	2.6434	2.6430	3.7010
5	3.0307	2.7888	2.6590	2.9761	2.1341	1.5053	1.5872	1.9776
6	3.5751	2.4402	1.5485	1.8893	1.9006	2.0319	2.7701	3.6592
7	4.5237	2.4615	1.8436	2.8661	3.0220	3.3533	4.0411	5.0260
8	6.6434	6.0698	5.4356	6.6439	6.7330	5.6637	4.6933	4.8614
9	6.7875	10.9783	11.6411	11.1674	10.5676	6.7891	6.0430	5.5835
10	3.0955	3.2717	2.1295	2.5386	3.2871	2.8563	2.8644	2.8560
11	3.6250	2.2188	2.6190	2.8466	2.3445	2.2937	3.9124	5.5326
12	4.1878	2.0453	1.2313	0.9855	1.2069	1.4120	3.2463	5.5979
13	3.5757	1.7480	0.7855	0.9090	2.0531	4.0788	6.4794	6.0671
14	2.6095	1.8033	1.6753	1.5080	1.7567	3.1603	3.5973	2.8236
15	1.9581	2.1933	2.0557	1.1460	1.4344	2.3029	1.9848	1.8594
16	1.6509	0.4949	0.9698	2.2521	2.2974	2.3181	4.3495	4.2698
17	1.4782	0.9150	1.8443	2.3843	1.6345	0.9615	1.2496	2.9330
18	1.5162	1.2748	2.5384	2.6699	2.4974	4.0387	5.1500	3.7222
19	1.4659	0.9064	1.4710	2.9553	3.1811	4.4298	5.9040	3.6744
20	2.6100	2.4709	3.2966	3.2521	3.7762	5.0580	6.7388	5.3425
21	1.7035	1.2263	1.8788	1.6194	1.0080	1.1309	1.8173	2.9411
22	2.0500	1.2209	1.5563	1.3741	1.7945	3.2080	4.5675	3.9991
23	2.1754	1.7004	2.2280	3.2200	3.1014	3.6125	4.1393	3.0307
24	2.7654	2.7072	2.7987	3.0566	1.5332	1.4562	3.3622	3.9474
25	3.3333	2.2529	3.7791	6.1923	6.1307	5.3449	6.0676	6.3310
26	1.2707	1.1173	1.0442	1.5572	1.6736	2.0929	1.4196	1.0029
27	1.4792	1.1566	1.1595	1.7968	1.7461	1.8051	1.5095	1.5988
28	1.9517	1.6391	1.1207	0.7599	0.6436	0.7299	0.7572	1.4193
29	1.5307	1.1922	1.2605	1.3513	1.1839	0.9113	0.9827	1.3863
30	0.6333	0.7696	0.6539	0.5884	0.4193	0.4565	0.6224	0.5047
31	0.6033	0.5460	0.4072	0.6106	0.7239	0.6494	0.5661	0.5245
32	0.3945	0.4692	0.4204	0.5211	0.4845	0.3536	0.3861	0.2819
33	0.4632	0.3941	0.5046	0.7388	0.5898	0.4106	0.4358	0.4652
34	0.8615	0.5840	0.3421	0.5136	0.5838	0.4851	0.4361	0.5673
35	0.9069	0.6588	0.4785	0.6547	0.6621	0.6426	0.5475	0.6939

4.3 Model Sonuçları

Bu çalışmada gömülü sistemlerde EKG sinyallerinin sınıflandırılmasına yönelik CNN ve LSTM tabanlı 2 farklı ağ kullanılmıştır. CNN modeli karmaşık özelliklerin çıkarımında güçlü performans sergilemiştir ancak model boyunun büyük olmasından dolayı gömülü sistemlerde kullanıma uygun değildir. Diğer yandan LSTM modeli,

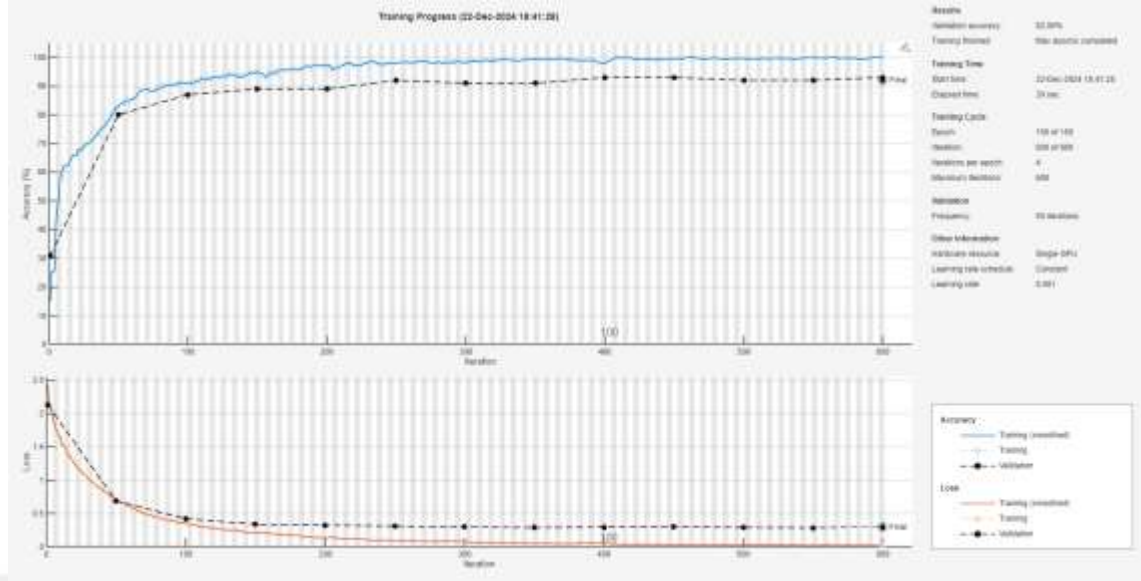
zaman serilerinin işlenmesinde olup küçük bir performans farkı ile oldukça küçük boyutta bir model oluşturmayı başarmıştır.

Şekil 4.12’de Uzun-kısa süreli bellek (Long Short-Term Memory - LSTM) modelinin eğitim ve doğrulama grafiği bulunmaktadır. Grafikte, modelin doğruluk oranı iterasyonlar boyunca artış göstererek %93 oranına ulaşmıştır. Alt grafikte ise kayıp değerinin düzenli olarak azaldığı görülmektedir. Eğitim ve validasyon değerlerinin birbirine yakın seyretmesi modelin veri setinin genelleyebilme kabiliyetinin yüksek olduğunu göstermektedir. Aşırı öğrenme problemi önlenmiştir. Bu bulgular, LSTM modelinin literatürdeki çalışmaların doğruluk oranlarına benzer olduğunu göstermektedir.



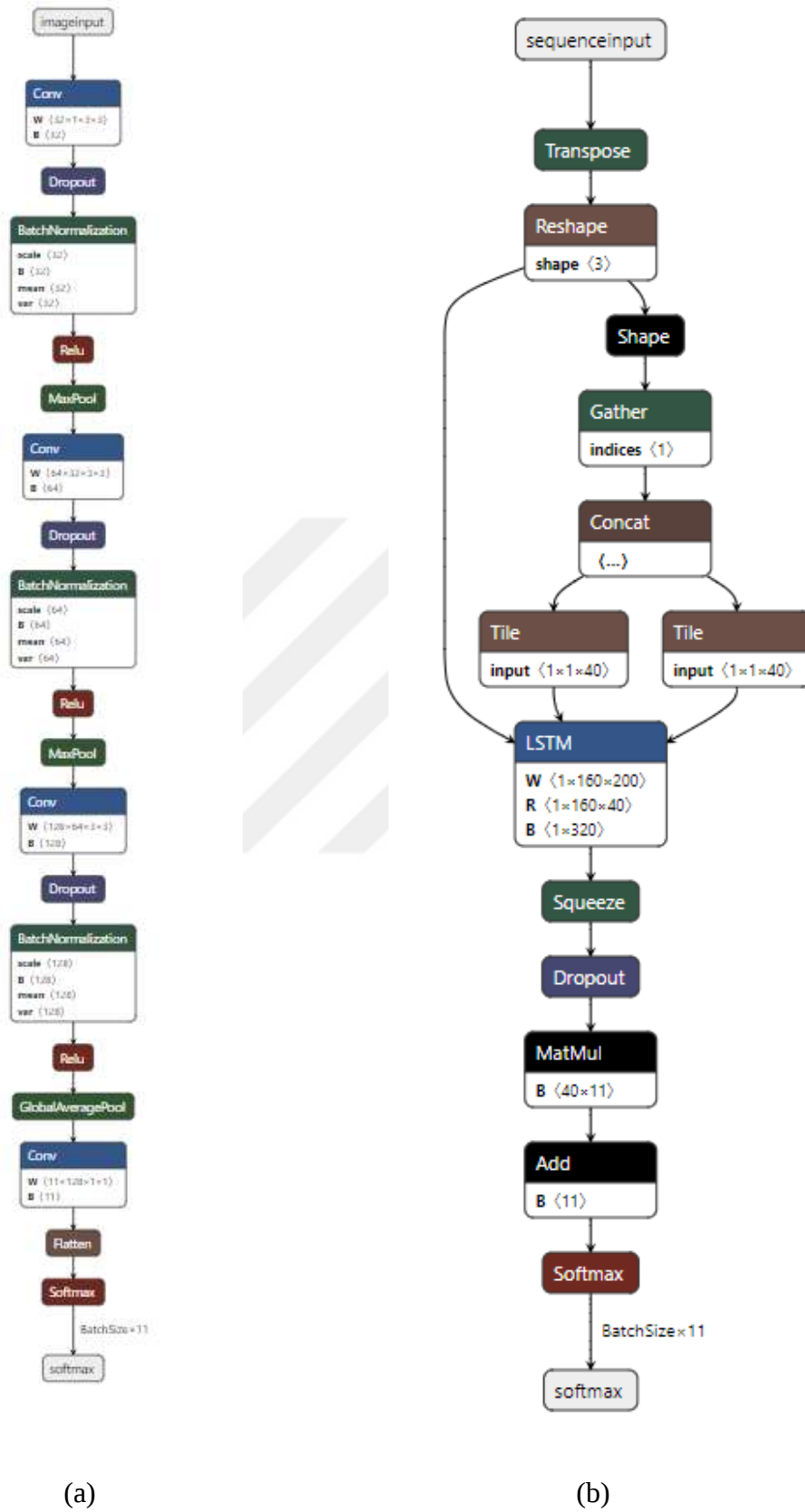
Şekil 4.12. LSTM Modeli Eğitim Süreci ve Performans Grafikleri

Şekil 4.13 ise Evrimsel Sinir Ağı (Convolutional Neural Network - CNN) modelinin eğitim ve doğrulama sürecine ilişkin metrikleri sunmaktadır. Eğitim grafiği, modelin doğruluğunun yaklaşık olarak %97, validasyon doğruluğunun ise %92 olduğunu göstermektedir. Doğruluklar arasında bir miktar fark bulunsa da, bu aşırı öğrenme açısından kabul edilebilir seviyededir.



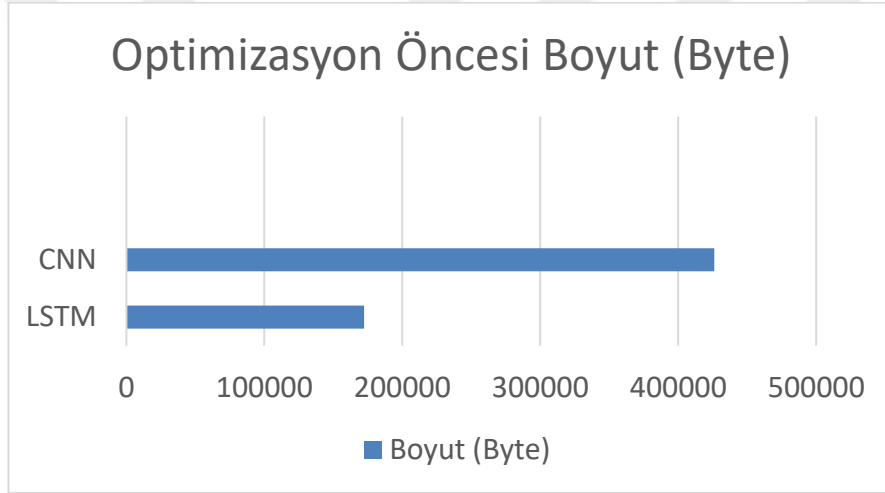
Şekil 4.13. CNN Modeli Eğitim Süreci ve Performans Grafikleri

Bu çalışmada kullanılan derin öğrenme modelleri, EKG verileri gibi sıralı verilerin analizi için uygun yapılar sunmaktadır. Şekil 4.14 b’de gösterildiği gibi LSTM tabanlı model, sıralı verilerin işlenmesi için tasarlanmış olup ilk aşamalarında veri uyumlama katmanları bulunmaktadır. Sonrasında gelen LSTM katmanı ise zaman serilerinin özelliklerini öğrenmektedir. Modelde bulunan dropout katmanı aşırı öğrenmeyi engellemek için eklenmiştir. CNN tabanlı modelin gösterildiği Şekil 4.14 a’da da benzer işlemler yer almaktadır. Ancak CNN ağı zaman serileri için özelleşmiş bir ağ değildir. Modelin başlangıç aşamasında giriş verileri konvolüsyon katmanlarına yönlendirilerek özellik çıkarımların gerçekleştirilmektedir. Her konvolüsyon adımından sonra bulunan dropout ve batch normalization katmanları, modelin aşırı öğrenmesini engellemekte ve katmanlar arasında veri transferinden önce verinin normalize edilmesini sağlamaktadır. Bu teknik farklılıklar modellerin boyutu ve performansını etkilemektedir. Çalışmanın ilerleyen kısımların her iki modele ait performans parametreleri incelenecektir.

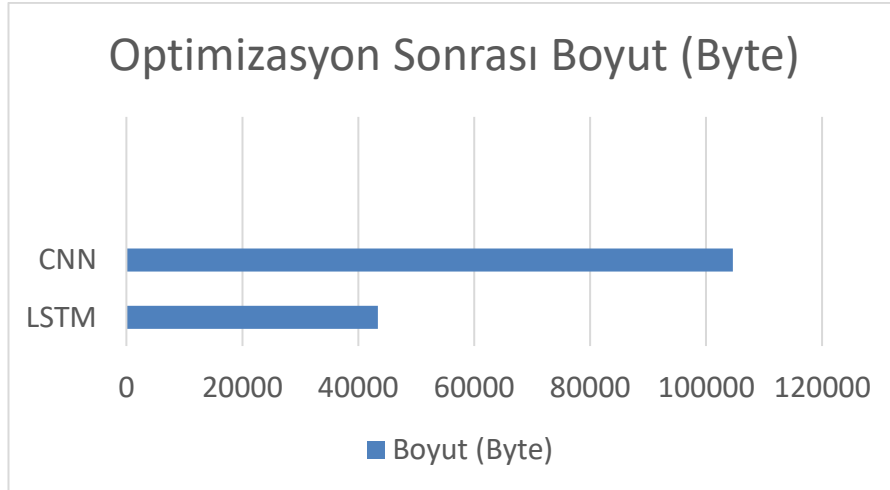


Şekil 4.14. a) CNN Modeli Katman Yapısı b) LSTM Modeli Katman Yapısı

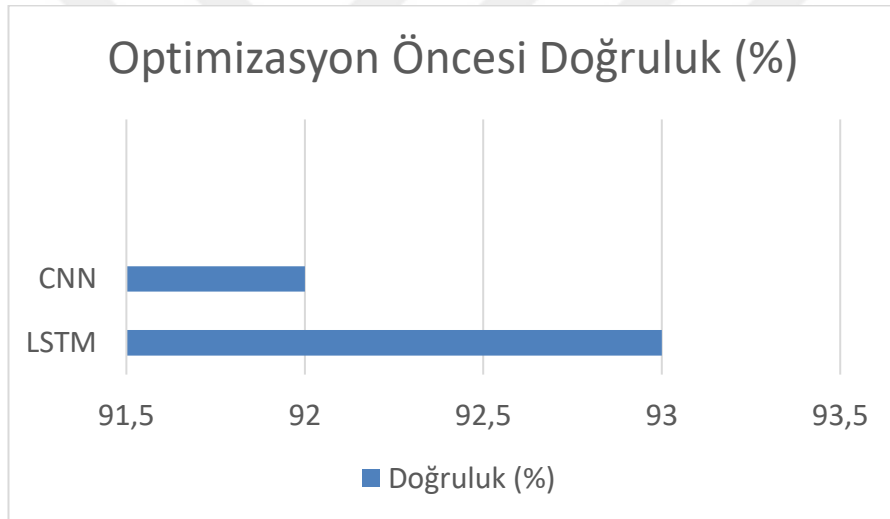
Çalışma kapsamında CNN ve LSTM tabanlı olmak üzere 2 farklı model geliştirilmiştir. CNN ağının doğruluğu %1 kadar düşüktür. Ancak LSTM ağının kullanılmasının asıl sebebi, LSTM ağının model boyutunun CNN ağına göre oldukça düşük olmasıdır. CNN ağının kullanılması, boyutundan ötürü mümkün değildir. CNN modeli optimizasyon öncesindeki boyutu 426098 byte olup, doğruluk oranı %92'dir. TinyML optimizasyonu sonrasında ise 104622 byte'a inmiştir. Doğruluk oranı LSTM'den yüksek olsa da kaynak kısıtlı bir sistemde çalışmaya uygun değildir. TinyML optimizasyonu öncesinde 172246 byte olan LSTM model boyutu, optimizasyon sonrasında 43376 byte'a düşmüştür. Kullanım geliştirme kartının 128 kB flash hafızası olduğu düşünüldüğünde uygundur. LSTM tabanlı derin öğrenme modelinin doğruluğu optimizasyon öncesinde %93, sonrasında ise %88 olarak kaydedilmiştir. Şekil 4.15'te ilgili değerler grafik halinde paylaşılmıştır.



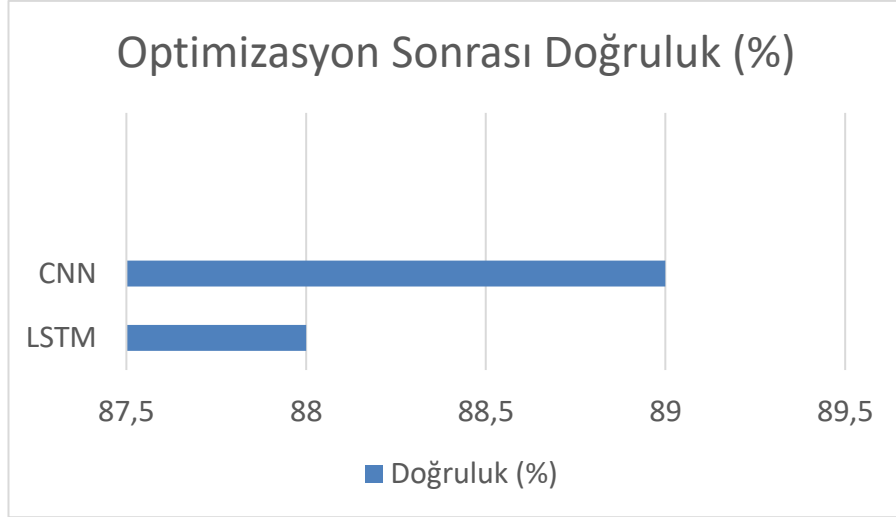
(a)



(b)



(c)



(d)

Şekil 4.15. a) Optimizasyon önce modellerin boyut karşılaştırması; b) Optimizasyon sonrası modellerin boyut karşılaştırması; c) Optimizasyon önce modellerin doğruluk karşılaştırması; d) Optimizasyon sonrası modellerin doğruluk karşılaştırması

Çizelge 4.3'te yer alan metrikler, geliştirilen LSTM modelinin TinyML optimizasyonu öncesinde ve sonrasındaki performanslarını karşılaştırmalı olarak sunmaktadır. Optimizasyon öncesinde precision, Recall ve F1 Score değerleri sırasıyla 0.91, 0.91 ve 0.90 olarak ölçülmüştür. Bu değerler modelin optimizasyon öncesinde yüksek doğrulukta çalıştığını göstermektedir. Optimizasyon sonrasında ise metrikler sırasıyla 0.87, 0.86 ve 0.85'e düşmüştür. Bu düşüş, TinyML optimizasyonu sürecinde karmaşıklık ve boyut azaltma sürecinin ve modelin kaynak kısıtlı sistemler üzerinde çalıştırılmasının gerekliliklerinin bir sonucudur.

Çizelge 4.3. Optimizasyon öncesi ve sonrasında performans parametrelerinin karşılaştırılması

Performans Metrikleri	Optimizasyon Öncesi	Optimizasyon Sonrası
Precision	0.91	0.87
Recall	0.91	0.86
F1 Score	0.90	0.85

Şekil 4.16'da geliştirilen LSTM modelinin MATLAB Programında ve STM32H750B geliştirme kartı üzerindeki bulunan tahmin sonuçları paylaşılmıştır. Tahmin çıktıları, girdiye ait olasılık dağılımlarını vektör olarak temsil eden formatta paylaşılmıştır. Şekil 4.16 a'da, modelin MATLAB tahminleri paylaşılmıştır. Şekil 4.16 b'de ise output_matrix[] değişkeni altında modelin gömülü sistem üzerinde yaptığı tahminler bulunmaktadır. Burada görüldüğü gibi, tahmin değerleri her iki platformda da aynıdır. STM32CubeIDE'nin debug sürecinde yaşanan bir hatadan dolayı Şekil 4.16a'da output_matrix[0] için beklenmedik bir şekilde 29.81 değeri alınmaktadır. Ancak olasılıklar UART ile HMI ekrana yazdırıldığında sonucun aslında MATLAB programında olduğu gibi 0.0087 olduğu görülmektedir.

ans =

0.0087 0.0686 0.0055 0.0122 0.0276 0.0150 0.6996 0.1248 0.0060 0.0074 0.0246

(a)

Expression	Type	Value	Address
input_matrix	float [10][20]	[10]	0x24000c70
input_matrix[0]	float [20]	[20]	0x24000c70
input_matrix[1]	float [20]	[20]	0x24000cc8
input_matrix[2]	float [20]	[20]	0x24000d18
input_matrix[3]	float [20]	[20]	0x24000d88
input_matrix[4]	float [20]	[20]	0x24000db8
input_matrix[5]	float [20]	[20]	0x24000e08
input_matrix[6]	float [20]	[20]	0x24000e90
input_matrix[7]	float [20]	[20]	0x24000ea8
input_matrix[8]	float [20]	[20]	0x24000f08
input_matrix[9]	float [20]	[20]	0x24000f48
output_matrix	float [11]	[11]	0x24003d0c
output_matrix[0]	float	29.8160152	0x24003d0c
output_matrix[1]	float	0.06860625	0x24003d40
output_matrix[2]	float	0.00553863257	0x24003d84
output_matrix[3]	float	0.0122006389	0x24003dc8
output_matrix[4]	float	0.0276481327	0x24003e0c
output_matrix[5]	float	0.0150756456	0x24003e50
output_matrix[6]	float	0.699633089	0x24003e94
output_matrix[7]	float	0.124804888	0x24003ed8
output_matrix[8]	float	0.0060386836	0x24003f0c
output_matrix[9]	float	0.0074154375	0x24003f50
output_matrix[10]	float	0.0246741709	0x24003f94
output_data	Failed to evaluate e...	Failed to evaluate e...	
eg_values[0]	short [140]	[140]	0x24000000
order	int	1007292853	0x24003738
index1	int	8	0x2400373c
index2	int	3	0x24003740
index3	int	6	0x24003744
index4	int	0	0x24003748

(b)

Şekil 4.16. a) Yapılan çıkarımın MATLAB Programındaki Sonuçları (Yüzde olarak); **b)** Yapılan çıkarımın STM32H750B Geliştirme Kartı üzerindeki Sonuçları (Yüzde olarak)

Çalışma her iki platform üzerinde de aynı değerlerin elde edilmesi ile tamamlanmıştır. Sonuçlar, elıştırılan sistemin hem MATLAB ortamında hem de STM32 gömülü platformunda tutarlı bir şekilde çalıştığını ortaya koymaktadır. MATLAB'da tasarlanan bir modelin TinyML formatına dönüştürülmesi ve sunucu bağımsız olarak

çalıştırılması sistemin taşınabilirliği ve gerçek zamanlı uygulamalara uygunluğu açısından önemlidir.

Şekil 4.17’deki görselde, çalışmanın bir parçası olarak geliştirilmiş HMI ekran tasarımı paylaşılmıştır. Sistem, EKG sinyallerinin ve tahmin sonuçlarının görselleştirilmesi amaçlayan bir donanım – yazılım çözümüdür. Bu sistemde STM32H750B geliştirme kartı ile UART protokolü vasıtasıyla haberleşen bir dokunmatik ekran kullanılmıştır. Ekran, tahmin edilen aritminin gerçek etiketini, tahmin edilen etiketini, tahmin olasılığı ve en tahmin edilen sınıftan sonraki 3 en yüksek olasılığı yazdırmak için kullanılır. Sistem, “SELECT” butonu ile veri tabanında rastgele bir R-R aralığı seçer. Seçilen R-R aralığının grafiği çizilir ve sinyale ait etiket “LABEL:” bölümünün sağ tarafından gösterilir. “ANALYZE” butonu ile analiz tamamlanan sinyalin tahmin edilen etiketi ve yüzdesi ekrana yazdırılır.

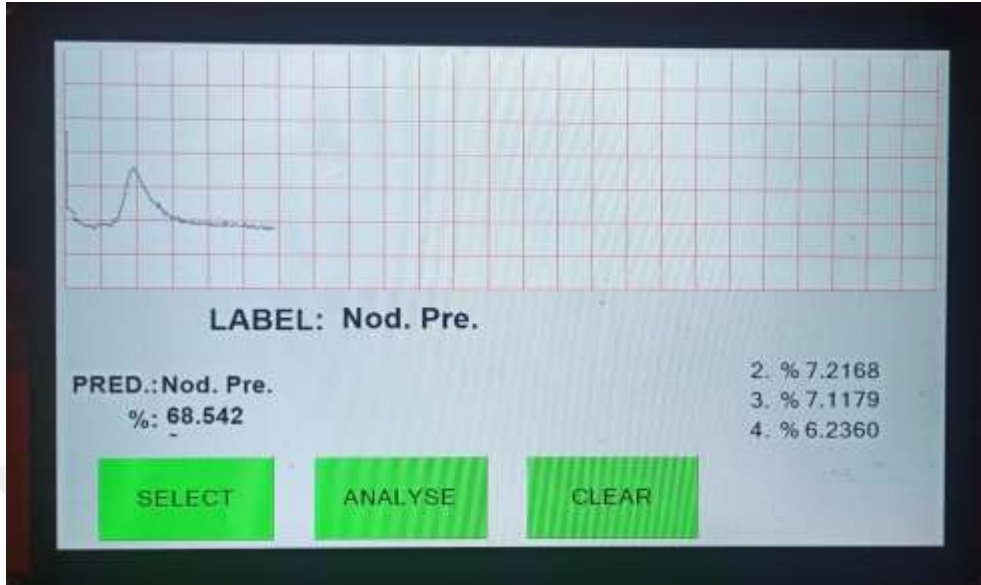


Şekil 4.17. HMI Ekran Tasarımı

Tahmin sonuçlarına eşlik eden olasılık değerleri, sistemin güvenliğini artırmak amacıyla ve kullanıcıya tahminin doğruluğu hakkında bilgi vermek amacıyla kullanılabilir. Bu da, sağlık profesyonelleri ve araştırmacıların karar alma süreçlerini etkileyebilir.

Şekil 4.18’de örnek olarak bir tahminin sonucu verilmiştir. STM32H750B geliştirme kartı üzerinde çalışan algoritma, rastgele seçilen sinyalin grafiğini çizdirmiş, ve doğru olarak tahmin etmiştir. Örnekteki aritmi tipi Nodal erken atımdır (Nodal

Premature Beat). Tahmin olasılığını ise %68.54'tür. Sonraki olasılıklar ise %7.22, %7.12 ve %6.24'tür.



Şekil 4.18. HMI ekran üzerinde örnek analiz sonucu

6. SONUÇLAR

Bu çalışma, EKG aritmilerinin makine öğrenmesi ile gömülü sistemler üzerinde sınıflandırılması hedefiyle yola çıkmıştır. TinyML teknolojisi kullanılarak STM32H750B geliştirme kartı üzerinde çalışan LSTM modeli başarıyla uygulanmıştır. MATLAB programında oluşturulan model, TinyML optimizasyonu öncesinde %93 doğrulukta çalışmakta olup, optimizasyon sonrasında ise %88 doğrulukta çalışmaktadır. Doğruluk oranı büyük ölçüde korunmuştur. Elde edilen model verimliliği, gelecekteki tıbbi cihaz çalışmalarına önemli katkılar sunabilecek potansiyeldedir.

Çalışma, özellikle gömülü sistemler gibi düşük güçlü ve sınırlı hesaplama yeteneği olan sistemlerin getirdiği zorluklara odaklanmakla birlikte, gerçek zamanlı sinyallerin TinyML ile kısıtlı sistemler üzerinde analiz edilmesine yönelik bir yol haritası sunmaktadır. Fizyolojik ya da fizyolojik olmayan sinyaller, TinyML yaklaşımı sayesinde optimize edilen modeller ile analiz edilebilir. Elde edilen modeller, doğruluk oranındaki bir miktar düşüşe rağmen verimli bir sistem oluşturmaya yeterlidir.

Daha önce belirtildiği gibi çalışma, TinyML teknolojisinin verimli modeller geliştirmekteki performansını göstermeyi hedeflemektedir. TinyML ile optimize edilmiş derin öğrenme modelleri, alt örnekleme, dalgacık dönüşümü, uygun geliştirme kartı ve uygun derin öğrenme modeli gibi tekniklerle birlikte desteklendiğinde kısıtlı cihazlar üzerinde literatürdeki diğer çalışmaların doğruluk oranlarına yakın bir doğruluk oranı ile çalışmaktadır. Ancak doğruluk oranı, giriş verilerinin kalitesi, optimizasyon düzeyi, ön işleme tekniklerinin seçimi gibi bir çok parametreye bağlıdır. Başlangıçta belirlenen hedefler gerçekleştirilmiş olup çalışmanın hipotezi doğrulanmıştır.

STM32H750B geliştirme kartı üzerinde yapılan testler, çalışmanın sağlık alanına kolayca adapte olabileceğini göstermektedir. TinyML teknolojisi, düşük maliyetli, taşınabilir ve enerji verimli giyilebilir sağlık teknolojilerinin geliştirilmesinde temel taşı olabilir. Model, kalp ritim bozukluklarının erken teşhisinde kullanılabilir. Giyilebilir EKG cihazlarına entegre edilmesiyle anormal durumlar gerçek zamanlı olarak tespit edilebilir ve müdahale imkanı sağlayabilir. Model, sadece anormal durumların merkeze iletilmesi ile büyük bant genişliği tasarrufu sağlayabilir ve kullanıcıyı uyabilir. Geliştirilen çözüm geleneksel tıbbi cihazlara kıyasla oldukça düşük maliyetlidir. Bu, düşük gelirli toplumların sağlık hizmetine erişimini artırır. TinyML, hastalıkların doğrudan tedavi edilmesinde kullanılmasıyla birlikte, sağlık profesyonellerinin iş yükünü azaltarak sağlık hizmetlerinin daha verimli ve etkin sunumuna katkı sunabilir. Uzun EKG kayıtlarının hasta üzerinde gerçekleştirilmesi, anormal olan durumların işaretlenmesi gibi birçok yöntem ile ulusal sağlık sisteminin ve profesyonellerin iş yükünü azaltılmasında kullanılabilir.

Bu çalışmada donanım kısıtlamaları nedeniyle birçok zorluk ve kısıtlama ile karşılaşmıştır. Bu zorluklar, gelecekte yapılacak donanım ve optimizasyon çalışmaları ile aşılabacaktır. Hibrit algoritmalar, veriyi daha iyi özetleyen veri setleri gibi diğer olanaklar doğruluğun artırılmasını sağlayabilir.

Gelecekteki çalışmalar, daha geniş ve çeşitli veri setleri ile eğitilmiş, daha yüksek performanslı modellerin oluşturulmasına yönelik olmalıdır. Aritmi yönünden daha zengin olan, birden fazla R-R aralığı içeren yada içerisinde gürültü bulunduran

verilerin kullanımı sonraki çalışmaların konusu olabilir. Yüksek performansı ve hesaplama ihtiyacı ile bilinen Morlet dalgacık fonksiyonu yerine diğer dalgacık dönüşümü fonksiyonlarının kullanıldığı bir model geliştirilebilir.

Son olarak, cihaz maliyetini ve enerji tüketimini minimize eden ön işleme yöntemlerinin TinyML ile birlikte kullanılması gelecekte taşınabilir cihazlardaki TinyML kullanımını artıracaktır. TinyML, bu çalışma ile birlikte potansiyelini ortaya koymuştur.



7. KAYNAKLAR

- Abadade, Y., Temouden, A., Bamoumen, H., Benamar, N., Chtouki, Y., Hafid, A. S. 2023. "A Comprehensive Survey on TinyML". IEEE Access, 11, 96892-96922.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Zheng, X. 2016. "TensorFlow: A system for large-scale machine learning" (Versiyon 2). arXiv.
- Abu Adla, Y., Saghir, M. A. R., Awad, M. 2023a. "Real-Time Arabic Digit Spotting with TinyML-Optimized CNNs on Edge Devices". İçinde I. Maglogiannis, L. Iliadis, J. MacIntyre, & M. Dominguez (Ed.), Artificial Intelligence Applications and Innovations (C. 675, ss. 527-538). Cham: Springer Nature Switzerland.
- Ajani, T. S., Imoize, A. L., Atayero, A. A. 2021. "An Overview of Machine Learning within Embedded and Mobile Devices—Optimizations and Applications". Sensors, 21(13), 4412.
- Alajlan, N. N., Ibrahim, D. M. 2022. "TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications". Micromachines, 13(6), 851.
- Alajlan, N. N., Ibrahim, D. M. 2023a. "DDD TinyML: A TinyML-Based Driver Drowsiness Detection Model Using Deep Learning". Sensors, 23(12), 5696.
- Anonymous 1: World Health Organization, Cardiovascular Diseases. <https://www.who.int/health-topics/cardiovascular-diseases> [Son erişim tarihi: 11.11.2024].
- Anonymous 2: World Heart Federation, World Heart Report 2023. <https://world-heart-federation.org/news/deaths-from-cardiovascular-disease-surged-60-globally-over-the-last-30-years-report/> [16.12.2024].
- Anonymous 3: Physionet, PhysioBank ATM <https://archive.physionet.org/cgi-bin/atm/ATM> [18.12.2024].
- Anonymous 4: MATLAB Help Center. <https://www.mathworks.com/help/> [13.12.2024].
- Anonymous 5: Google AI for Developers, Model Optimization https://ai.google.dev/edge/litert/models/model_optimization
- Anonymous 6: ST, Discovery kit with STM32H750XB MCU <https://www.st.com/en/evaluation-tools/stm32h750b-dk.html> [09.12.2024].
- Arie, R., Brand, A., Engelberg, S. 2020. "Compressive sensing and sub-Nyquist sampling". IEEE Instrumentation & Measurement Magazine, 23(2), 94-101.
- Atanane, O., Mourhir, A., Benamar, N., Zennaro, M. 2023. "Smart Buildings: Water Leakage Detection Using TinyML". Sensors, 23(22), 9210.
- Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., ... Yadav, P. 2020. "Benchmarking TinyML Systems: Challenges and Direction" (Versiyon 4). arXiv.
- Beraza, I., Romero, I. 2017. "Comparative study of algorithms for ECG segmentation".

- Biomedical Signal Processing and Control, 34, 166-173.
- Cantero, D., Esnaola-Gonzalez, I., Miguel-Alonso, J., Jauregi, E. 2022. "Benchmarking Object Detection Deep Learning Models in Embedded Devices". *Sensors*, 22(11), 4205.
- Capogrosso, L., Cunico, F., Cheng, D. S., Fummi, F., Cristani, M. 2023, Eylül 26. "A Machine Learning-oriented Survey on Tiny Machine Learning". *arXiv*.
- Carlini, N., Wagner, D. 2017. "Towards Evaluating the Robustness of Neural Networks". İçinde 2017 IEEE Symposium on Security and Privacy (SP) (ss. 39-57). San Jose, CA, USA: IEEE.
- Chakrabarti, G., Grover, V., Aarts, B., Kong, X., Kudlur, M., Lin, Y., ... Wang, J.-Z. 2012. "CUDA: Compiling and optimizing for a GPU platform". *Procedia Computer Science*, 9, 1910-1919.
- Chen, G., Parada, C., Heigold, G. 2014. "Small-footprint keyword spotting using deep neural networks". İçinde 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (ss. 4087-4091). Florence, Italy: IEEE.
- Chutchavong, V. 2019. "Mathematical Modeling ECG Waveform Using Abel's Bernstein Polynomials". İçinde 2019 4th International Conference on Communication and Information Systems (ICCIS) (ss. 141-146). Wuhan, China: IEEE.
- Cohen, M. X. 2018 "A better way to define and describe Morlet wavelets for time-frequency analysis".
- Crocioni, G., Pau, D., Delorme, J.-M., Gruosso, G. 2020. "Li-Ion Batteries Parameter Estimation With Tiny Neural Networks Embedded on Intelligent IoT Microcontrollers". *IEEE Access*, 8, 122135-122146.
- David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., ... Warden, P. 2021, Mart 13. "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems". *arXiv*.
- Daubechies I. 1990, Ten Lectures of Wavelet
- Ebrahimi, Z., Loni, M., Daneshtalab, M., Gharehbaghi, A. 2020. "A review on deep learning methods for ECG arrhythmia classification". *Expert Systems with Applications*: X, 7, 100033.
- Elhanashi, A., Dini, P., Saponara, S., Zheng, Q. 2024. "Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices". *Electronics*, 13(17), 3562.
- Falaschetti, L., Alessandrini, M., Biagetti, G., Crippa, P., Turchetti, C. 2022. "ECG-Based Arrhythmia Classification using Recurrent Neural Networks in Embedded Systems". *Procedia Computer Science*, 207, 3479-3487.
- Gaziano, T. A., Gaziano, J. M. 2012. "Global Burden of Cardiovascular Disease". İçinde Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine (ss. 1-20). Elsevier.
- Gouaux, J. L., Ashman, R. 1947. "Auricular fibrillation with aberration simulating ventricular paroxysmal tachycardia". *American Heart Journal*, 34(3), 366-373.

- Guo, T., Zhang, T., Lim, E., Lopez-Benitez, M., Ma, F., Yu, L. 2022. "A Review of Wavelet Analysis and Its Applications: Challenges and Opportunities". IEEE Access, 10, 58869-58903.
- Hall, J. E., Guyton, A. C. 2016. "Guyton and Hall textbook of medical physiology" (13th edition). Philadelphia, Pa: Elsevier.
- HENRY J. L. MARRIOTT, M.D., NEIL L. SCHWARTZ, 1962. " Ventricular Fusion Beats "
- Humphreys, M., Warlow, C., McGowan, J. 2011. "Arrhythmias and their Management". İçinde M. Humphreys (Ed.), Nursing the Cardiac Patient (1. bs, ss. 132-155). Wiley.
- IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, SEPTEMBER 1973.
- Joung, B., Chen, P.-S. 2015. "Function and Dysfunction of Human Sinoatrial Node". Korean Circulation Journal, 45(3), 184.
- Kim, E., Kim, J., Park, J., Ko, H., Kyung, Y. 2023. "TinyML-Based Classification in an ECG Monitoring Embedded System". Computers, Materials & Continua, 75(1), 1751-1764.
- Kingma, D. P., Ba, J. 2017, Ocak 30. "Adam: A Method for Stochastic Optimization". arXiv.
- Kingma, J., Simard, C., Drolet, B. 2023. "Overview of Cardiac Arrhythmias and Treatment Strategies". Pharmaceuticals, 16(6), 844.
- Kodal SeviNdiR, H., ÇetiNkaya, S., Yazici, C. 2018. "Makine öğrenmesi algoritmaları ve dalgacık dönüşümü ile EKG sinyalinde özellik çıkarımı". Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 20(1), 94-109.
- Kwon, J., Kim, K.-H., Eisen, H. J., Cho, Y., Jeon, K.-H., Lee, S. Y., ... Oh, B.-H. 2021. "Artificial intelligence assessment for early detection of heart failure with preserved ejection fraction based on electrocardiographic features". European Heart Journal - Digital Health, 2(1), 106-116.
- Lane, N. D., Bhattacharya, S., Mathur, A., Georgiev, P., Forlivesi, C., Kawsar, F. 2017. "Squeezing Deep Learning into Mobile and Embedded Devices". IEEE Pervasive Computing, 16(3), 82-88.
- Leng, L., Zheng, J., Zhang, J. 2020. "Characteristics of Atrial Premature Beat ECG Signals on Variant Maps". İçinde U. Lakshmanadoss (Ed.), Practical Applications of Electrocardiogram. IntechOpen.
- Littmann, L., Symanski, J. D. 2000. "Hemodynamic implications of left bundle branch block". Journal of Electrocardiology, 33, 115-121.
- MacDonald, E. A., Rose, R. A., Quinn, T. A. 2020. "Neurohumoral Control of Sinoatrial Node Activity and Heart Rate: Insight From Experimental Models and Findings From Humans". Frontiers in Physiology, 11, 170.
- Moody, G. B., Mark, R. G. 2001. "The impact of the MIT-BIH Arrhythmia Database". IEEE Engineering in Medicine and Biology Magazine, 20(3), 45-50.
- Muzammil, M. A., Javid, S., Afridi, A. K., Siddineni, R., Shahabi, M., Haseeb, M., ...

- Nashwan, A. J. 2024. "Artificial intelligence-enhanced electrocardiography for accurate diagnosis and management of cardiovascular diseases". *Journal of Electrocardiology*, 83, 30-40.
- Najmi, A.-H., Sadowsky, J. 1997 "The Continuous Wavelet Transform and Variable Resolution Time-Frequency Analysis".
- Pan, J., Tompkins, W. J. 1985. "A Real-Time QRS Detection Algorithm". *IEEE Transactions on Biomedical Engineering*, BME-32(3), 230-236.
- Pang, B., Nijkamp, E., Wu, Y. N. 2020. "Deep Learning With TensorFlow: A Review". *Journal of Educational and Behavioral Statistics*, 45(2), 227-248.
- Park, D. S., Fishman, G. I. 2011. "The Cardiac Conduction System". *Circulation*, 123(8), 904-915.
- Pham, V. L., Wong, K. P. 1999. "Wavelet-transform-based algorithm for harmonic analysis of power system waveforms". *IEE Proceedings - Generation, Transmission and Distribution*, 146(3), 249.
- Radiuk, P. M. 2017. "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets". *Information Technology and Management Science*, 20(1).
- Rafie, N., Kashou, A. H., Noseworthy, P. A. 2021. "ECG Interpretation: Clinical Relevance, Challenges, and Advances". *Hearts*, 2(4), 505-513.
- Reichlin, T., Abächerli, R., Twerenbold, R., Kühne, M., Schaer, B., Müller, C., ... Osswald, S. 2016. "Advanced ECG in 2016: is there more than just a tracing?". *Swiss Medical Weekly*.
- Rosenimm, M. B. 1954. "BILATERAL BLINDLE BRANCH BLOCK".
- Sabry, F., Eltaras, T., Labda, W., Alzoubi, K., Malluhi, Q. 2022. "Machine Learning for Healthcare Wearable Devices: The Big Picture". *Journal of Healthcare Engineering*, 2022, 1-25.
- Safdar, M. F., Nowak, R. M., Pałka, P. 2024. "Pre-Processing techniques and artificial intelligence algorithms for electrocardiogram (ECG) signals analysis: A comprehensive review". *Computers in Biology and Medicine*, 170, 107908.
- Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, Amgad Muneer. 2023. "LSTM Inefficiency in Long-Term Dependencies Regression Problems". *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 30(3), 16-31.
- Saha, S. S., Sandha, S. S., Srivastava, M. 2022. "Machine Learning for Microcontroller-Class Hardware: A Review". *IEEE Sensors Journal*, 22(22), 21362-21390.
- Salimi, A., Kalmady, S. V., Hindle, A., Zaiane, O., Kaul, P. 2023, Kasım 2. "Exploring Best Practices for ECG Signal Processing in Machine Learning". *arXiv*.
- Sifuzzaman, M., Islam, M. R., Ali, M. Z. 2009 "Application of Wavelet Transform and its Advantages Compared to Fourier Transform".
- Sörnmo, L., Laguna, P. 2006. "Electrocardiogram (ECG) Signal Processing". İçinde M. Akay (Ed.), *Wiley Encyclopedia of Biomedical Engineering* (1. bs). Wiley.

- Srinivasagan, R., Mohammed, M., Alzahrani, A. 2023. "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits". *Sensors*, 23(16), 7081.
- Takahashi, N., Ito, M., Saikawa, T., Arita, M., Fujino, T., Kagiya, H., ... Sakata, T. 1998. "Clinical suppression of bradycardia dependent premature ventricular contractions by the potassium channel opener nicorandil". *Heart*, 79(1), 64-68.
- Viamonte, V. A., Rosen, M. R. 1990. "Premature escape beats induced by overdrive pacing in canine Purkinje fibers. Evidence for the role of normal automaticity as an underlying cellular mechanism.". *Circulation*, 82(1), 234-243.
- Wilson, D. R., Martinez, T. R. 2001. "The need for small learning rates on large problems". İçinde *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222) (C. 1, ss. 115-119)*. Washington, DC, USA: IEEE.
- Wu, M., Lu, Y., Yang, W., Wong, S. Y. 2021. "A Study on Arrhythmia via ECG Signal Classification Using the Convolutional Neural Network". *Frontiers in Computational Neuroscience*, 14, 564015.
- Xue, H., Huang, B., Qin, M., Zhou, H., Yang, H. 2020. "Edge Computing for Internet of Things: A Survey". 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData).
- Yelchuri, H., & R, R. 2020. "A Review of TinyML". *RV College of Engineering*.
- Zhang, L., Cottrell, G. W. "When Holistic Processing is Not Enough: Local Features Save the Day".
- Zhou, D.-X. 2020. "Theory of deep convolutional neural networks: Downsampling". *Neural Networks*, 124, 319-327.
- Zhou, Y., Ren, C., Zhang, S., Xue, X., Liu, Y., Lu, J., Ding, C. 2023. "A Second-Order Method for Removing Mixed Noise from Remote Sensing Images". *Sensors*, 23(17), 7543.

ÖZGEÇMİŞ

Doğan Can ÖZBEY

ÖĞRENİM BİLGİLERİ

Yüksek Lisans	Akdeniz Üniversitesi
2023-2025	Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Antalya
Lisans	Akdeniz Üniversitesi
2016-2022	Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Antalya

MESLEKİ VE İDARİ GÖREVLER

Ar-Ge Mühendisi	Fresenius Medikal Hizmetler A.Ş.
2022-Devam Ediyor	Antalya