

T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
FEN FAKÜLTESİ  
İSTATİSTİK BÖLÜMÜ

---

GİYİLEBİLİR CİHAZ VE AKILLI TELEFON VERİLERİ İLE AKTİVİTE  
TANIMA

---

Bitirme Projesi Raporu

---

Berke SEVİM  
Furkan PAŞAHAN  
Mert KAVAS  
Furkan ERGÜNEŞ

May 2023



# Rapor Deęerlendirme

“GİYİLEBİLİR CİHAZ VE AKILLI TELEFON VERİLERİ İLE AKTİVİTE TAN-  
IMA” başlıklı bitirme projesi raporu tarafımdan okunmuş, kapsamı ve nitelięi açısn-  
dan bir Bitirme Projesi raporu olarak kabul edilmiştir.

Dr. Engin YILDIZTEPE



# Teşekkür

Tüm çalışma süresince yönlendiriciliği, katkıları ve yardımları ile yanımızda olan danışmanımız Dr. Engin YILDIZTEPE 'ye ve böyle bir çalışmayı yapmamız için bize fırsat tanıyan Dokuz Eylül Üniversitesi Fen Fakültesi İstatistik Bölümüne teşekkür ederiz.

Berke SEVİM  
Furkan PAŞAHAN  
Mert KAVAS  
Furkan ERGÜNEŞ



# Özet

Akıllı telefonların ve giyilebilir teknolojilerin günlük hayata girmesi farklı uygulamaların yaygın olarak kullanılmasını sağlamıştır. Bu cihazlarda bulunan sensörlerden elde edilen veriler ile aktivitenin tanınması da bu uygulamalardan biridir. Aktiviteler temel olarak yürüme, koşma gibi basit aktiviteler ve yemek yeme, uyuma, dış fırçalama gibi karışık aktiviteler olmak üzere ikiye ayrılmaktadır. Bu çalışmada, ivmeölçer ve jireskop sensörleri ile elde edilmiş veriler kullanılarak aktivitelerin “Yürüme”, “Merdiven Çıkma”, “Merdiven İnme”, “Yatma”, “Ayakta Durma”, “Oturma” olarak sınıflandırılması problemi incelenmiştir. Uygulamada üç farklı yaklaşımda gruplanabilecek k-en yakın komşu, destek vektör makineleri, rassal ormanlar, XGBoost ve yapay sinir ağları yöntemleri kullanılmıştır. Sınıflandırma yöntemlerinin performansı tek sensörden elde edilen değişkenler ve her iki sensörden de elde edilen tüm değişkenler kullanılarak ayrı ayrı değerlendirilmiştir. Sonuçlar, farklı performans ölçütleri ile karşılaştırılmıştır. Aktivitelerin sınıflandırılmasında en iyi model her iki sensörden elde edilen değişkenlerin yapay sinir ağları yöntemiyle kullanılmasıyla elde edilmiştir. Uygulamada Python programlama dili kullanılmıştır.

**Anahtar Kelimeler:** aktivite tanıma, makine öğrenimi, sensör verileri





# Abstract

The introduction of smartphones and wearable technologies into daily life has led to the widespread use of different applications. Recognizing activity with the data obtained from the sensors in these devices is one of these applications. Activities are basically divided into simple activities such as walking, running, and complex activities such as eating, sleeping, brushing teeth, etc. In this study, the problem of classifying activities as “Walking”, “Climbing Stairs”, “Descending Stairs”, “Lying Down”, “Standing”, “Sitting” using data obtained with accelerometer and gyroscope sensors is investigated. In the application, k-nearest neighbor, support vector machines, random forests, XGBoost and artificial neural networks, which can be grouped into three different approaches, were used. The performance of the classification methods is evaluated separately using variables obtained from a single sensor and all variables obtained from both sensors. The results are compared with various performance metrics. The best model for the classification of activities was obtained by using the variables obtained from both sensors with the artificial neural network method. Python programming language was used in the implementation.

**Keywords:** activity recognition, machine learning, sensor data



# Contents

<b>Giriş</b>	<b>1</b>
<b>Chapter 1: Aktivite Tanıma</b>	<b>3</b>
1.1 Aktivite	3
1.1.1 Basit Aktivite	4
1.1.2 Karışık Aktivite	4
1.1.3 Aktivite Tanıma Uygulamaları	4
1.2 Sensör	4
1.2.1 Aktivite Tanımda Kullanılan Sensörler	4
1.2.2 Aktivite Tanımda Sensör Kullanımı	5
<b>Chapter 2: YÖNTEM</b>	<b>7</b>
2.1 Geleneksel Yöntemler	7
2.1.1 K En Yakın Komşular Algoritması	7
2.1.2 Destek Vektör Makinesi	8
2.2 Ağaç Yöntemleri	9
2.2.1 Rassal Ormanlar (Random Forest)	9
2.2.2 XGBoost	10
2.3 Sinir Ağları Yöntemleri	10
2.3.1 Tekrarlayan Sinir Ağları (Recurrent Neural Network)	14
2.3.2 Evrişimli Sinir Ağları (Convolutional Neural Network)	15
2.4 Makine öğrenmesi değerlendirme ölçütleri	17
<b>Chapter 3: UYGULAMA</b>	<b>19</b>
3.1 Uygulama	19
3.2 Çok Sınıflı (Multiclass) Sınıflama Problemi	23
3.3 Jiroskop Sensörüne Ait Değişkenler İle Modelleme	25
3.3.1 KNN Modeli	25
3.3.2 Rassal Ormanlar Modeli	27
3.3.3 eXtreme Gradient Boosting (XGBoost)	29
3.3.4 Destek Vektör Makinesi	30
3.3.5 Yapay Sinir Ağları (Neural Networks)	32
3.4 İvme Ölçer Sensörüne Ait Değişkenler İle Modelleme	34
3.4.1 K-En Yakın Komşuluk Modeli	34
3.4.2 Rassal Ormanlar Modeli	36

3.4.3	eXtreme Gradient Boosting (XGBoost) . . . . .	37
3.4.4	Destek Vektör Makinesi . . . . .	39
3.4.5	Yapay Sinir Ağları (Neural Networks) . . . . .	40
3.5	Jiroskop ve İvme Ölçer Sensörlerine Ait Değişkenler İle Modelleme . .	42
3.5.1	K-En Yakın Komşuluk Modeli . . . . .	42
3.5.2	Rassal Ormanlar Modeli . . . . .	44
3.5.3	eXtreme Gradient Boosting (XGBoost) . . . . .	46
3.5.4	Destek Vektör Makinesi . . . . .	47
3.5.5	Yapay Sinir Ağları (Neural Networks) . . . . .	49
<b>Chapter 4: Sonuç . . . . .</b>		<b>53</b>
<b>Kaynaklar . . . . .</b>		<b>55</b>

# List of Tables

2.1	Destek Vektör Makinesinin Aktivite Tanımadaki Yeri . . . . .	9
3.1	Jiroskop ve İvme Ölçer Sensörlerinden elde edilen değişkenler . . . . .	24
3.2	Sadece Jiroskop Sensöründen elde edilen Değişkenler . . . . .	25
3.3	Sadece İvmeölçer Sensöründen elde edilen Değişkenler . . . . .	25
3.4	Jiroskop Sensöründen elde edilen değişkenler ile kurulan k en yakın komşular modelinin başarı sonuçları . . . . .	26
3.5	Jiroskop Sensöründen elde edilen değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları . . . . .	27
3.6	Jiroskop Sensöründen elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları . . . . .	29
3.7	Jiroskop Sensöründen elde edilen değişkenler ile kurulan destek vektör makinası modelinin başarı sonuçları . . . . .	31
3.8	Jiroskop Sensöründen elde edilen değişkenler ile kurulan yapay sinir ağları algoritmasının başarı sonuçları . . . . .	33
3.9	İvmeölçer Sensöründen elde edilen değişkenler ile kurulan k en yakın komşular algoritmasının başarı sonuçları . . . . .	34
3.10	İvmeölçer Sensöründen elde edilen değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları . . . . .	36
3.11	İvmeölçer Sensöründen elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları . . . . .	38
3.12	İvmeölçer Sensöründen elde edilen değişkenler ile kurulan destek vektör makinası algoritmasının başarı sonuçları . . . . .	39
3.13	İvmeölçer Sensöründen elde edilen değişkenler ile kurulan yapay sinir ağları modelinin başarı sonuçları . . . . .	41
3.14	Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan k en yakın komşular modelinin başarı sonuçları . . . . .	43
3.15	Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları . . . . .	45
3.16	Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları . . . . .	46
3.17	Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan destek vektör makinesi modelinin başarı sonuçları . . . . .	48
3.18	Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan yapay sinir ağları modelinin başarı sonuçları . . . . .	50

4.1	Yöntemlere ve Sensörlere göre F1 skorları . . . . .	53
-----	---	----

# List of Figures

2.1	Biyolojik sinir hücresi ve yapay sinir ağı . . . . .	11
2.2	Yapay sinir hücresi . . . . .	11
2.3	İleri beslemeli sinir ağı yapısı . . . . .	12
2.4	TSA'nın genel yapısı . . . . .	14
2.5	Evrişimli Sinir Ağları Yapısı . . . . .	16
2.6	Karışıklık matrisi örneği . . . . .	17
3.1	Hedef değişkene göre gözlem sayıları . . . . .	20
3.2	Sabit ve hareketli aktiviteler için tBodyAccMagmean değişkeninin yoğunluk grafiği . . . . .	21
3.3	Sabit ve hareketli aktivitelerin yoğunluk grafiklerinin karşılaştırılması	22
3.4	tBodyAccMagmean Değişkeninin Aktivitelere Göre BoxPlot Grafiği .	23
3.5	K-NN Modeli Karmaşıklık Matrisi . . . . .	26
3.6	K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları . .	27
3.7	Rassal Ormanlar Modeli Karmaşıklık Matrisi . . . . .	28
3.8	Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Performansları . . . . .	28
3.9	XGBoost Modeli Karmaşıklık Matrisi . . . . .	30
3.10	XGBoost Modelinin Eğitim ve Test Verilerinin Başarı Performansları	30
3.11	Destek Vektör Makinesi Modeli Karmaşıklık Matrisi . . . . .	31
3.12	Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları	32
3.13	Yapay Sinir Ağları Modeli Karmaşıklık Matrisi . . . . .	33
3.14	Jiroskop Sensörü İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları . . . . .	34
3.15	K-NN Modeli Karmaşıklık Matrisi . . . . .	35
3.16	K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları . .	35
3.17	Rassal Ormanlar Modeli Karmaşıklık Matrisi . . . . .	37
3.18	Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Performansları . . . . .	37
3.19	Xgboost modeli Karmaşıklık Matrisi . . . . .	38
3.20	Xgboost Modelinin Eğitim ve Test Verilerinin Başarı Performansları .	39
3.21	Destek Vektör Makinesi Modeli Karmaşıklık Matrisi . . . . .	40
3.22	Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları	40
3.23	Yapay Sinir Ağları Modeli Karmaşıklık Matrisi . . . . .	42
3.24	İvme Ölçer Sensörü İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları . . . . .	42

3.25 K-NN Modeli Karmaşıklık Matrisi . . . . .	44
3.26 K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları . .	44
3.27 Rassal Ormanlar Modeli Karmaşıklık Matrisi . . . . .	45
3.28 Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Perfor- mansları . . . . .	46
3.29 XGBoost Modeli Karmaşıklık Matrisi . . . . .	47
3.30 XGBoost Modelinin Eğitim ve Test Verilerinin Başarı Performansları	47
3.31 Destek Vektör Makinesi Modeli Karmaşıklık Matrisi . . . . .	49
3.32 Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları	49
3.33 Yapay Sinir Ağları Modeli Karmaşıklık Matrisi . . . . .	51
3.34 Jiroskop ve İvme Ölçer Sensörleri İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları	51



# Giriş

Günümüzde, insan aktivitelerinin otomatik olarak tanınması ve sınıflandırılması, çeşitli uygulama alanlarında büyük ilgi gören bir araştırma konusudur. Aktivite tanıma, bir kişinin yürüyüş, koşu, oturma, yatma gibi farklı aktivitelerini doğru bir şekilde tanımlayabilen ve sınıflandırabilen bir sistem oluşturmayı amaçlar. Bu alandaki gelişmeler, taşınabilir cihazların yaygınlaşması, sağlık takibi, yaşlı bakımı, spor analizi, akıllı ev sistemleri ve robotik gibi birçok uygulama alanında büyük potansiyel sunmaktadır.

Aktivite tanıma, öncelikle sensör verilerinin (jiroskop, ivmeölçer) toplanması ve işlenmesiyle gerçekleştirilir. Bu sensör verileri, bir kişinin beden hareketlerini, konumunu ve mekanda gerçekleştirdiği değişiklikleri temsil eder. Makine öğrenimi algoritmaları, bu sensör verilerini analiz ederek farklı aktiviteleri sınıflandırabilir ve tanımlayabilir. Aktivite tanıma sistemlerinin geliştirilmesiyle birlikte, sağlık takibini kolaylaştırmak ve günlük yaşam aktivitelerini daha verimli bir şekilde yöneterek yaşam kalitesini artırmak için yeni olanaklar sunulmaktadır.

Bu çalışma, aktivite tanıma konusunda yapılan araştırmaları incelemeyi ve sensör verilerini kullanarak aktivite sınıflama uygulaması geliştirmeyi amaçlamaktadır. İlk olarak Bölüm 1’de, aktivite tanımanın önemi, kullanılan sensörler ve uygulama alanları üzerinde durulmuştur. İkinci bölümde, yaygın olarak kullanılan sınıflandırma yöntemleri incelenmiştir. Bölüm 3’de sunulan uygulamada üç farklı yaklaşımda gruplanabilecek k-en yakın komşu, destek vektör makineleri, rassal ormanlar, XGBoost ve yapay sinir ağları yöntemleri kullanılmıştır. Sınıflandırma yöntemlerinin performansı tek sensörden elde edilen değişkenler ve her iki sensörden de elde edilen tüm değişkenler kullanılarak ayrı ayrı değerlendirilmiştir. Bölüm 4’de sonuçlar, farklı performans ölçütleri ile karşılaştırılmıştır.



# Chapter 1

## Aktivite Tanıma

Teknolojinin gelişimi ve elektronik devrelerin küçülmesi ile sadece bilgisayarlar ve akıllı telefonlar değil aynı zamanda günlük hayatta kullanılan kıyafet ve aksesuarlara sensör vb. donanımlar eklenerek akıllı cihaza dönüştürülebilmekte ve bunlar da kendi aralarında haberleşebilmektedir. Bu yüzden, kullanıcı veri alışverişi ve hesaplama gibi işlemlerini, daha büyük bilgisayarlara ihtiyaç duymadan vücuduna monte durumda bulunan elektronik devreler veya üzerine giydiği kıyafetlerinin hesaplama ve haberleşme yeteneği kazanması sonucunda kolayca sağlayabilmektedir.

Son yıllarda özellikle akıllı telefonların ve giyilebilir cihazların gelişmesiyle birlikte aktivite tanıma alanındaki çalışmalar hız kazanmıştır. Giyilebilir teknoloji, aksesuar olarak giyilebilen, giysiye gömülü, kullanıcının vücuduna yerleştirilmiş veya hatta cilde dövülmüş bir elektronik cihaz kategorisidir. Cihazlar pratik kullanımlı, mikro işlemciler tarafından desteklenen ve İnternet üzerinden veri gönderme ve alma özelliğine sahip cihazlardır. Nesnelerin internetinin (IoT) arkasındaki amaç, gerçek zamanlı olarak kendini raporlayan, verimliliği artıran ve önemli bilgileri yüzeye insan müdahalesine bağlı bir sistemden daha hızlı bir şekilde getiren cihazlara sahip olmaktır.

Giyilebilir teknoloji, bir aksesuar olarak veya giyside kullanılan malzemenin bir parçası olarak, vücut üzerinde takılabilen elektronikler bir alettir. Çok sayıda giyilebilir teknoloji vardır, ancak en popüler cihazlardan bazıları etkinlik izleyicileri ve akıllı saatlerdir.

### 1.1 Aktivite

İnsanların günlük yaşantılarında gerçekleştirdikleri aktiviteler; basit (basic) ve karmaşık (complex) olarak iki kategoriye ayrılmaktadır. Basit aktivitelere; oturma, merdiven çıkma, merdiven inme, koşma, yürüme gibi aktiviteler örnek verilebilir. Birden fazla aktivitenin birlikte gerçekleştirilmesiyle oluşan karmaşık aktivitelere; ilaç kullanmak, araba kullanmak, evi temizlemek ve bilgisayar kullanmak gibi aktiviteler olarak literatürdeki çalışmalarda da yer verilmektedir.

Basit ve karmaşık aktiviteleri tanıma işlemleri bazı cihaz ve sensörler yardımıyla yapılmaktadır. Aktivite tanıma çalışmaları 1980'li yıllarda başlamıştır (Paul ve George,

2015). Teknolojinin gelişmesi ile birlikte bu çalışmalar hızlı bir ivme kazanmıştır. İnsan aktivitelerini tanımak hayatımıza katma değer katmak ile beraber işlerimizi kolaylaştırmaktadır.

### 1.1.1 Basit Aktivite

Basit aktiviteleri tahminlemek ve çalışmalar yapmak daha kolaydır. En çok tercih edilen aktivite çalışmalarıdır. Çalışmalar genel olarak yürüme, koşma, oturma, merdiven çıkma ve inme vb. gibi basit aktiviteleri tahminleme üzerine yapılmaktadır. Bu aktiviteler için en iyi elde edilen doğruluk oranı ise ivmeölçer, jiroskop ve mıknatıs ölçer kullanarak %97'dir (Ustev, Durmaz Incel ve Ersoy, 2013).

### 1.1.2 Karışık Aktivite

Karışık aktiviteler aslında basit aktivitelerden oluşmaktadır. Bu aktiviteleri tahminlemek daha zordur. Örnek vermek gerekir ise düş almak, ders çalışmak, diş fırçalama vb. aktiviteler karmaşık aktivite olarak örneklendirilebilir. Bu aktiviteleri tanımlayabilmek için birden fazla sensör ve teknoloji aynı anda kullanılması gerekmektedir. Karmaşık Aktivitelerde en iyi doğruluk oranı ivmeölçer ve GPS kullanarak %93.44 olarak kayıtlara geçmiştir (Riboni ve Bettini, 2011).

### 1.1.3 Aktivite Tanıma Uygulamaları

Aktivite tanıma uygulamalarının kullanılabilecek alanları başta sağlık hizmetleri olmak üzere akıllı ortam, güvenlik ve benzerleridir (Chen, Hoey, Nugent, Cook ve Yu, 2012). Akıllı ortamlar ile ilgili verilebilecek örnekler arasında evden çıktığımızda otomatik olarak kombi, klima ve ışıkların kapatılması gibi örnekler verilebilir. Sağlık alanında ise kişisel aktivite asistanı, yaşlılar ve hastalar için uzun süre sabit kalınması durumunda uyarı sistemi söylenebilir. Karmaşık aktivitelerin kullanım alanlarında örnek vermek gerekir ise insanların çalışma durumları ve verimlilikleri tanımlanıp çalışma tempoları ve performans ölçümleri yapılabilir.

## 1.2 Sensör

Sensör amacı ortamdaki değişiklikleri farketmek ve diğer elektronik cihazlara bilgi göndermek olan bir cihaz alt sistemlerdir.

### 1.2.1 Aktivite Tanımadaki Kullanılan Sensörler

- İvmeölçer : Cihazın 3 ekseninde (x,y,z) ivmesini ölçer.
- Jiroskop : Cihazın (x,y,z) 3 ekseninde rotasyonunu (dönüşünü) ölçer.
- Gps (Global positioning system) : Cihazın konumunu belirler.
- Mıknatıs Ölçer : Cihazın bulunduğu ortamdaki manyetik alanın ölçümünü yapar.

### 1.2.2 Aktivite Tanımda Sensör Kullanımı

Cihazlardaki dahili sensörler ise aktivite tanımayı daha kolay hale getirmiştir. Akıllı telefon ve akıllı saatler kolay ulaşılabilir ve sürekli üzerimizde çalışır halde bulunuyor olduğundan çoğunlukla basit aktiviteleri tanımda akıllı cihazlarda bulunan sensörler kullanılır. Yürüme, koşma, merdiven inme, merdiven çıkma, uzanma, ayakta durma vs. basit aktiviteleri tanımda çoğunlukla ivmeölçer, jiroskop ve gps sensörleri kullanılmıştır.

Aktivite tanımda hem dahili hem harici sensörler kullanılmaktadır. Akıllı ev, akıllı ofis gibi vb. yerlerde karışık aktivitelerin tespitinde harici sensörler kullanılmaktadır. Örneğin akıllı evlerde sensörler ortamda kimseyi tespit etmediğinde klima ve ışıkların otomatik kapatılmasını sağlıyor. Kamera, mikrofön vb. sensörler harici sensörlerdir (Khan, Lee ve Kim, 2008; Kwapisz, Weiss ve Moore, 2011).

Bu çalışmalar temelinde her bir insan aktivitesini farklı sınıf etiketleri olarak algılarlar ve sensör verilerinden elde ettikleri öz nitelikler ile bu sınıf etiketlerinin tespiti üzerine kurulum sağlanır. Bu aktivitelerin tanınmasında farklı makine öğrenmesi algoritmaları kullanılmıştır.

Akıllı cihazlar ile yapılan bazı çalışmaları inceleyecek olursak;

- Fan ve arkadaşlarının yaptığı çalışmada sabit durma, yürüme, koşma, merdiven inme ve çıkma aktivitelerinin karar ağacı algoritması kullanılarak tanınması amaçlanmıştır ve algoritmanın çalışma süresi sonunda %80 civarında doğrulukla aktiviteler tanınmıştır (Fan, Wang ve Wang, 2013).
- Yang ve arkadaşlarının yaptığı çalışmada basit aktivitelerin yanı sıra dış fırçalama, temizlik ve bilgisayarda çalışma gibi aktiviteler de izlenmiş, sinir ağları ve geleneksel makine öğrenmesi algoritmaları performans açısından kıyaslanmıştır ve elde edilen sonuçlar sinir ağlarının bu aktivitelerini %95’lik genel doğruluk oranında tanıyabildiğini ortaya koymuştur (J.-Y. Yang, Wang ve Chen, 2008).
- Uzun Kısa Süreli Bellek (UKSB) hücreleri ile insan aktivitelerinin sınıflandırılmasında %92,1 test başarımlı doğruluğunda bir sonuca, halkın kullanımına açık bir veri kümesi üzerinde yapılan bir çalışmada ulaşılmıştır (Du, Fu ve Wang, 2015).



# Chapter 2

## YÖNTEM

Bu bölümde uygulama kısmında kullanılmış olan makine öğrenmesi algoritmalarına değinilmiştir. Bu yöntemler 3 sınıfta kategorize edilmiştir; Geleneksel yöntemler, Ağaç yöntemleri ve Sinir Ağları yöntemleri.

### 2.1 Geleneksel Yöntemler

#### 2.1.1 K En Yakın Komşular Algoritması

K En Yakın Komşular algoritması makine öğrenim algoritmaları içerisinde en çok bilinen ve kullanılan algoritmalarından biridir. Seçilen bir özelliğin kendine en yakın olan özellikler arasındaki yakınlığı kullanarak sınıflandırma yapılır. Burada bulunan K değeri örnek olarak 3 veya 5 gibi bir sayı ile ifade edilebilir. Çalışma şekline baktığımızda, tanımlanan verilere göre yeni bir tanımlanması gereken nesne geldiğinde öncelikle K değerine bakılır. Burada eşitlik olmaması için genellikle K sayısı tek sayı olarak seçilir. Yeni gelen veri ile diğer veriler arasındaki mesafeler hesaplanırken Kosinüs, Öklid ya da Manhattan uzaklığı gibi yöntemler kullanılır (Dolgun, Özdemir ve Oğuz, 2009).

**K En Yakın Komşular algoritmasında temel olarak aşağıdaki adımlar gerçekleştirilir:**

- K değerinin belirlenmesi.
- Tüm öğrenme örnekleri ile olan uzaklığının hesaplanması.
- Minimum uzaklığa göre sıralama işleminin yapılması.
- Ait oldukları sınıf değerlerinin bulunması.
- Değeri baskın olan sınıfın seçilmesi

K En Yakın Komşular algoritması tanımlı gözlemlerin sınıflandırılmasına ve regresyon yapılabilmesine olanak sağlamaktadır. K En Yakın Komşular algoritması basitliği ve kolay uygulanabilirliği nedeniyle yaygın kullanılan bir algoritma.

madır. Bu özelliğinden ötürü sıkça kullanılmaktadır. Birçok algoritma verilerin istatistiksel dağılımı ile ilgili olarak bazı varsayımları gerektirmesine rağmen K En Yakın Komşular kullanılacağı durumlarda böyle bir varsayıma gereksinim olmamaktadır. K En Yakın Komşular algoritmasının tahmin yapabilmesi için eğitimi hızlı olmaktadır (Lantz, 2019).

İnsan aktivitesi tanıma konusunda şimdiye kadar pek çok çalışma yapılmıştır ve yapılmaya da devam edilmektedir. Tanınacak aktiviteye bağlı olarak aktiviteler basit veya karmaşık aktiviteler olarak iki sınıfa ayrılabilir. Basit aktiviteler, küçük bir frekans aralığında bileşenler içeren sinyallerle temsil edilebilirken karmaşık aktiviteler, farklı frekans bileşenlerine sahip olan ve kendini tekrar eden sinyaller ile temsil edilmesi zor olan aktiviteler olarak tanımlanabilir. Bu aktivitelerin tanınmasında farklı makine öğrenmesi algoritmaları kullanılmıştır. Paul ve George, akıllı telefonlardaki sensörlerden elde edilen verileri kullanarak K En Yakın Komşular algoritmasında insan aktivite tanımlama çalışması yapmışlardır (Paul ve George, 2015). Sağbaş ve Ballı kullanıcıların yazma davranışlarını; yazma hızı, silme sıklığı gibi bilgilere ek olarak ivmeölçer ve jiroskop gibi hareket algılayıcılarından elde edilen veriler ile birlikte incelemiştir (Sağbaş ve Ballı, 2017). Algılayıcı verileri telefonun duruşu yazma anında sergilenen durum hakkında önemli bilgiler vermektedir. 2 sınıflı stres tespitinde K En Yakın Komşular ile %87.56 sınıflandırma doğruluğu elde edilmiştir. Çalışmaya 35 erkek, 11 kadın olmak üzere 46 gönüllü katılmıştır. Gerçek referans değer için ise 5 seviyeli öz değerlendirme anketi kullanılmıştır (Sağbaş, Korukoğlu ve Ballı, 2021).

### 2.1.2 Destek Vektör Makinesi

Destek Vektör Makineleri genellikle sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden biridir. Bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun iki sınıf noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur.

Destek vektör makinesi, sınıflandırma ve regresyon analizi için kullanılan verileri analiz eden ilişkili öğrenme algoritmalarına sahip denetimli öğrenme modelleridir. Eldeki verilerden yeni kalıpları tanımlamak için büyük miktarda veriyi analiz ederler. Destek Vektör Makinesi, aşağıdaki resimde gösterildiği gibi bir veri kümesini en iyi iki sınıfa ayıran bir hiper düzlem bulma amacı ile oluşturulur.

#### Hiper düzlem nedir

Basit bir örnek olarak, yalnızca iki özelliğe sahip bir sınıflandırma görevi için, bir hiper düzlemi bir veri kümesini doğrusal olarak ayıran ve sınıflandıran bir çizgi olarak düşünebiliriz. Şöyle de düşünebiliriz, veri noktalarımız hiper düzlemde ne kadar uzak olursa, doğru sınıflandırıldıklarından o kadar emin oluruz. Bu nedenle, veri noktalarımızın doğrulara hiper düzlemde olabildiğince uzakta olmasını istiyoruz. Dolayısıyla, yeni test verileri eklendiğinde, iniş yapan hiper düzlemin herhangi bir tarafı, ona atadığımız sınıfa karar verecektir.

Destek Vektör Makinesi algoritmasının aktivite tanımadaki yeri;



Table 2.1: Destek Vektör Makinesinin Aktivite Tanımadaki Yeri

yazar_yıl	veri_kaynagi	aktivite	basari_metrigi	basari_degeri	yöntem
Kim ve Choi,2012	Dokunmatik Panel,İvmeölçer,Jiroskop	Evet(Oturuyormu?,Ayaktami?)	F ölçütü	%82	Karar Ağacı
Ciman ve Wac,2018	Dokunma İstatistiği,Uygulama kullanımı,Aktivite,Işık,Ekran açma/kapama	Evet	F ölçütü	%88	K En Yakın Komşular
Sağbaş vd.,2020	Yazma Karakteristiği,İvmeölçer,Jiroskop	Hayır	Doğruluk oranı	%87.56	K En Yakın Komşular
Lee vd.,2012	Yazma karakteristiği,Konum,Hava durumu,Işık	Hayır	Doğruluk oranı	%67.52	Bayes Ağları

## 2.2 Ağaç Yöntemleri

### 2.2.1 Rassal Ormanlar (Random Forest)

2000’li yılların başında Breiman tarafından tasarlanan rassal ormanlar algoritması verileri işlemek için şu anda mevcut olan en başarılı yöntemler arasında yer alır.Bu denetimli öğrenme yöntemi basit ama etkili olan “böl ve fethet” ilkesine dayanır.Verinin her bir örneklem fraksiyonu üzerinde ağaç tahmincisi kurup daha sonra bu tahmincilerin ortalamalarını kullanarak tahminde bulunur.Rassal ormanlar sınıflandırma ve regresyon problemleri için kullanılır.Rassal ormanları popüler bir makine öğrenmesi algoritması yapan özelliklerinden bir diğeri ise ayarlanması gereken az sayıda parametreye sahip olması ve tahmin problemlerinde çok geniş kullanım alanına sahip olmasıdır.Değişken sayısının gözlem sayısından fazla olduğu verilerde çok iyi performans göstermesiyle bilinen bir algoritmadır(Biau ve Scornet, 2016).

#### Algoritma

Algoritmanın genel çerçevesi verilen P boyutlu bir X vektörü ile P boyutlu Y vektörünün nonparametrik regresyon tahminidir. Regresyon fonksiyonu  $= m(x) = E[Y|X = x]$   $D_n = ((X_1, Y_1), \dots, (X_n, Y_n))$  birbirinden bağımsız dağılan eşleştirilmiş gözlemler olsun.Amacımız  $D_n$ ’i kullanarak  $M$ ’in fonksiyonu olan  $M_n : X \rightarrow R$ ’i tahmin etmektir.  $M_n$  regresyon fonksiyonu eğer  $E[M_n(X) - M(X)]^2 \rightarrow 0, n \rightarrow \infty$  tutarlıdır.Rassal orman rassallaştırılmış M tane regresyon ağacını içeren bir tahmin edicidir.Verilen bir X noktası için J’nci ağaçtaki tahmin değeri matematiksel olarak aşağıdaki gibi ifade edilir.

$m_n(x; \Theta_j, D_n), \Theta_1, \dots, \Theta_M$  rassal değişkenleri  $D_n$ ’den bağımsızdır. $\Theta$  Dallandırılacak her bir ağaç için yeniden örnekleme yapmak için ve bölünme için ardışık yönler belirlemek için kullanılır.

$$m_n(x; \Theta_j, D_n) = \sum_{x \in D_n^*(\Theta_j)} \frac{X_i \in A_n(x; \Theta_j, D_n) Y_i}{N_n(x; \Theta_j, D_n)}$$

$D_n^*(\Theta_j)$  ağaç yapımından önce seçilen verideki noktaları ifade eder.

$A_n(x; \Theta_j, D_n)$  x’i içeren hücre ve  $N_n(x; \Theta_j, D_n)$  ise  $A_n(x; \Theta_j, D_n)$  hücresinin içine düşen nokta sayısını ifade eder.

Sonlu boyutlu rassal ağaç tahmini:

$$m_{M,n}(x; \Theta_1, \dots, \Theta_M, D_n) = \frac{1}{M} \sum_{j=1}^M m_n(x; \Theta_j, D_n)$$

### 2.2.2 XGBoost

Rassal orman algoritmasına benzeyen bu algoritma karar ağaçları ile çalışmaktadır. XGBoost algoritması hız ve performans için tasarlanmış gradyan artırılmış karar ağaçlarının bir uygulamasıdır. Boosting (artırma), mevcut ağaçların yaptığı hataları düzeltmek için yeni ağaçların eklendiği bir grup tekniktir. Bu yöntemin çalışma mantığına göre; verideki her bir değeri analiz etmek yerine onları parçalara ayırma işlemi uygulanmaktadır. Bu şekilde küçük parçalar üzerinden daha iyi tahminleme yapması beklenmektedir (Yangın, 2019).

Gradyan Artırma Algoritması (Gradient Boosting): Bu algoritma, makine öğrenmesinde tahminleri bağımsız yapmak yerine sırayla yapan bir çeşit topluluk algoritmasıdır. Amaç zayıf tahminleri iterasyonlarla kademeli olarak güçlü tahminlere dönüştürmektir. Gradyan Artırma Algoritmasında ilk iterasyonda tahminleri üreten bir  $y_i^p$  ve "Loss" fonksiyonları birleştirilir ve tekrar tahminler ile hedefler arasındaki fark hesaplanır. Böylece sürekli üstüne ekleyerek  $y_i^p$  fonksiyonunun başarısı arttırılmaya ve tahminler ile hedefler arasındaki fark sıfıra indirilmeye çalışılır.

$$LOSS = MSE = \sum (y_i - y_i^p)^2$$

Burada  $y_i$  hedef değer  $y_i^p$  i. tahmin değeri  $L(y_i, y_i^p)$  kayıp fonksiyonu ve  $\alpha$  ise öğrenme oranıdır. Tahminleri, kayıp fonksiyonu minimum olacak şekilde elde etmek gereklidir. Gradyan iniş kullanarak tahminler öğrenme oranına göre iterasyonlarda güncellenir ve kayıp fonksiyonunun minimum değerleri bulunur.

$$y_i^p = y_i^p + \alpha * \delta \sum \frac{(y_i - y_i^p)}{\delta y_i^p}$$

$$y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p)$$

Böylece artıkların toplamı sıfıra yakın olması yani tahmin edilen değerler gerçek değerlere olabildiğince yaklaşması arzu edilen durumdur.

## 2.3 Sinir Ağları Yöntemleri

İnsan beyninin üstün özelliklerini ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Yapay sinir ağları, insan beyninin öğrenme yolunu taklit ederek beynin öğrenme, hatırlama, genelleme yapma yolu ile topladığı verilerden yeni veri üretebilme gibi temel işlevlerin gerçekleştirildiği bilgisayar yazılımlarıdır (Öztürk ve Şahin, 2018).

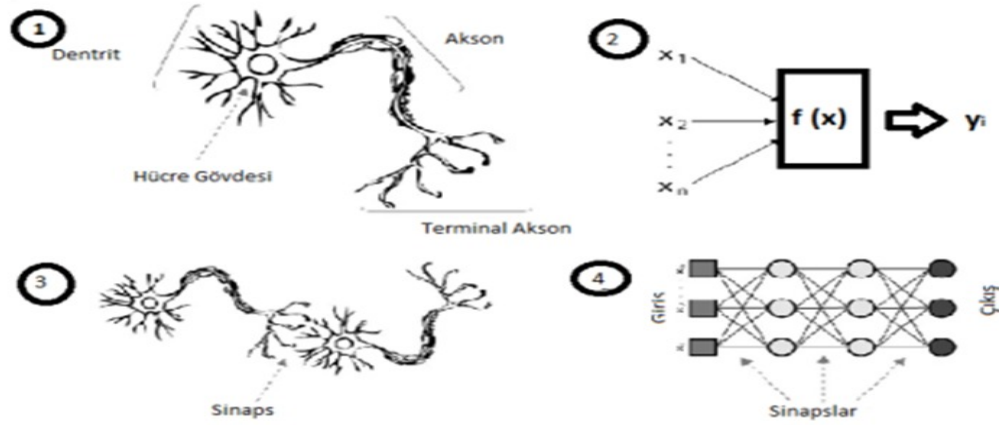


Figure 2.1: Biyolojik sinir hücresi ve yapay sinir ağı

(Maltarollo, Honório ve Silva, 2013)

Biyolojik sinir hücresi ve yapay sinir ağı benzetimleri 2.1.'de verilmiştir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar halinde düzenlenir. İnsan beynine iki şekilde benzerlik göstermektedir:

- Bilgi, öğrenme süreci yoluyla ağ tarafından elde edilir.
- Sinaptik ağırlıklar olarak bilinen nöronlar arası bağlantı kuvvetlerini, bilgiyi depolamak için kullanır

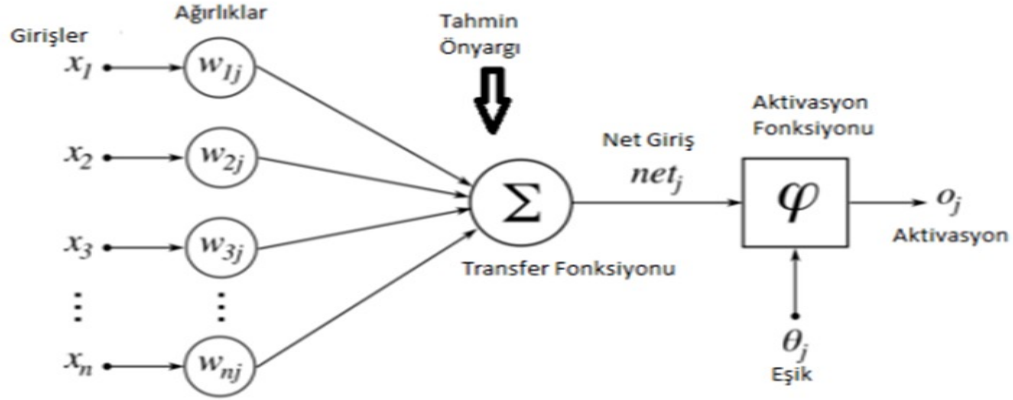


Figure 2.2: Yapay sinir hücresi

(Öztürk ve Şahin, 2018)

2.2. 'de görüldüğü gibi bir hücreye  $n$  adet veri girişi yapılmıştır. Bu girdilere ait ağırlıklar ile girilen veriler çarpılıp toplanması ile elde edilen transfer fonksiyonu, transfer fonksiyonundan gelen değer belirli bir aralığa normalize edildiği aktivasyon fonksiyonu ve aktivasyondan oluşur. Çok katmanlı yapay sinir ağlarına bakıldığında temel olarak 3 katmandan meydana gelir;

- Girdi Katmanı (Input Layer)
- Gizli, Ara Katmanlar (Hidden Layers)
- Çıktı Katmanı (Output Layer)

Yapay Sinir Ağları uygulamaları en çok tahmin, sınıflandırma, veri ilişkilendirme, veri yorumlama ve veri filtreleme işlemlerinde kullanılmaktadır(Ağyar, 2015).

Bu prensipte çalışan yapay sinir ağları girdi değerinden çıktıları tahmin etme üzerine çalışır, örneğin altın ons fiyatının tahmini.

Bu doğrultuda kodlanan yapay ağlar toplanan veriler arasından en işe yarayan verileri kullanır.

- Sınıflandırma Girdi değerlerini sınıflandırarak sistemin daha hızlı sonuca varmasına etkide bulunur.

Önceden eğitilen ağ girdilerini analiz eder, bir olay hakkında bu girdiler sayesinde yeni yorumlamalar yapabilmektedir.

Öğrendiği bilgilerle konuları ilişkilendirir ve bunun sonucunda ortaya çıkan eksik bilgileri tamamlar(Öztürk ve Şahin, 2018).

Bir yapay sinir ağının yapısı ve sinir hücrelerinin sayısı değişiklik göstermelerine rağmen, yapay sinir ağının oluşumu için kabul görmüş herhangi bir kural bulunmamaktadır. Gerekli gizli katman sayısından az gizli katmana sahip yapay sinir ağları komplike fonksiyonların çözümünde yetersiz kalırken, çok fazla gizli katmana sahip yapay sinir ağları ise istenmeyen kararsızlıklarla karşılaşmaktadır. Gizli katman sayısı belirlendikten sonra karşılaşılan problem ise her bir tabakada kaç tane nöronun yer alacağına karar vermede karşımıza çıkmaktadır. Girdi katmanı için bir sorun bulunmamaktadır. Bu sayı sistem içerisindeki girdilerin sayısına eşittir. Aynı şekilde, çıktı katmanı da istenilen çıktı sayısı ile belirlenebilmektedir. Esas sorun, gizli katmanlarda nöron sayısını belirlemektir. Geleneksel matris algoritması, matris boyutlarının ya girdi sayısına ya da çıktı sayısına eşit olması gerektiğini söylemektedir. Ne yazık ki, gizli katmanda en verimli şekilde kaç tane nöronun bulunacağı konusunda herhangi bir matematiksel test bulunmamaktadır. Deneme ve yanılma yöntemi uygulanarak karar verilmelidir(Bell, Detienne ve Joshi, 2003).

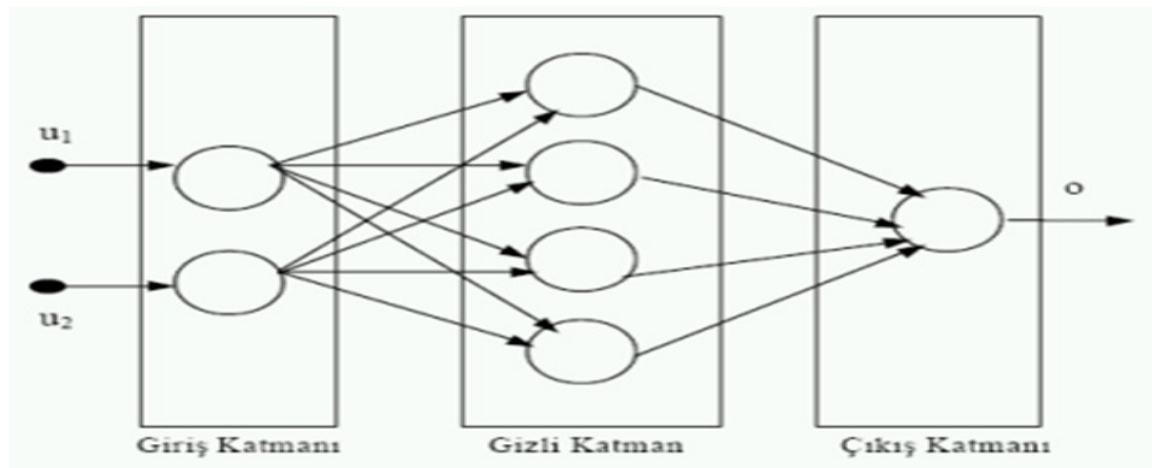


Figure 2.3: İleri beslemeli sinir ağı yapısı

(Fırat ve Güngör, 2004)

YSA, hücrelerin bağlanma biçimlerine göre “ileri beslemeli” ve “geri beslemeli” olmak üzere iki mimari yapı altında sınıflandırılabilir.

### İleri beslemeli ağlar

Verilerin sadece girdi birimlerinden çıktı birimlerine ileri doğru aktığı ağ yapısıdır. Bu yapıda nöronlar katmanlar şeklinde düzenlenir. Bir katmandaki nöronların çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Aynı katmandaki nöronlar arasında veya bir önceki katmana bağlantı yani geri besleme çevrimi yoktur. Uygulamalarda genellikle ileri beslemeli ağların tercih edildiği görülmektedir (Asilkan ve Irmak, 2009).

### Geri beslemeli ağlar

Veri akışının sadece ileriye doğru değil geriye doğru da olabileceği ağ yapısıdır. Bu yapıda en az bir tane geri besleme çevrimi bulunur. Geri besleme, aynı katmandaki hücreler arasında olabileceği gibi farklı katmanlardaki nöronlar arasında da olabilir (Asilkan ve Irmak, 2009).

İşlem sürecinde toplama ve aktivasyon adlarında iki fonksiyon yer almaktadır. Toplama (Birleştirme) Fonksiyonu: Yapay sinir hücresine gelen girdilerin kendilerine ait ağırlıklarla çarpıldıktan sonra birleştirilmesi işlemini gerçekleştiren fonksiyondur. Bu fonksiyon, adından da anlaşılacağı gibi, genelde toplama işlemini kullanmakla birlikte farklı işlemleri de kullanabilir. Hatta araştırmacının kendi kurduğu işlemi de kullanması mümkündür. Toplama fonksiyonunda kullanılan işlem, genellikle seçilen ağ mimarisine de bağlıdır. En sık kullanılan fonksiyonlar şunlardır; Toplama, çarpım, maksimum, minimum, çoğunluk, kümülatif toplamdır (Asilkan ve Irmak, 2009).

Yapay sinir hücresinin çıktısının büyüklüğünü sınırlandıran fonksiyondur. Bazı kaynaklarda transfer, eşik veya sıkıştırma fonksiyonu olarak da isimlendirilmektedir (Mandic ve Chambers, 2001). Bir ağdaki tüm hücrelerin aktivasyon fonksiyonu birbirinden farklı olabilir. Aktivasyon fonksiyonunda doğrusal fonksiyonlar genelde tercih edilmez. Zaman serileri için “Sigmoid”, ikili (binary) değişkenler için “Adım” fonksiyonu önerilmektedir (Tebelskis, 1995). En çok kullanılan aktivasyon fonksiyonları şunlardır: Doğrusal, adım, eşik değer, hiperbolik tanjant, sigmoidtir (Asilkan ve Irmak, 2009).

Derin ağlar, görüntü tanıma, nesne tanıma ve doğal dil işleme gibi birçok alanda kullanılmaktadır. Aktivite tanıma alanında derin öğrenme yöntemleri ile son yıllarda çalışmalar yapılmıştır. Bu alanda derin öğrenme yöntemleri kullanmanın en büyük avantajı ham sensör verilerinden saklı kalmış önemli özelliklerin otomatik çıkartılabilmesidir. Aktivite tanıma için sensör verilerinden hangi özelliklerin kullanılacağını belirlemek için bilirkişiye ihtiyaç kalmamaktadır. Derin ağlarla cihaz konumundan ve oryantasyonundan bağımsız özellikler de çıkartılabilmektedir.

Derin ağların eğitim süresi çok uzun süreler alabildiği için süreyi kısaltmak adına güçlü donanımların kullanımına ihtiyaç duyulmaktadır. Dolayısıyla derin öğrenme ile eğitimin şu anki mobil cihaz donanımları üzerinde yapılması mümkün değildir. Sadece

bilgisayar ortamında eğitilmiş hazır modeller telefon üzerinde gerçek zamanlı aktivite sınıflandırmak için kullanılabilir. Derin öğrenme yöntemlerinin her birisinin farklı çalışma prensipleri vardır. İncelenmiş makaleler içerisinde daha çok tercih edilen ağ Konvolüsyonel Sinir Ağları (Convolutional Neural Network) olmuştur. Kullanılan derin öğrenme yöntemleri ve sonuçlar karşılaştırmalı (Iskenderov ve Güvensan, 2019).

### 2.3.1 Tekrarlayan Sinir Ağları (Recurrent Neural Network)

TSA, zaman serilerini kullanarak işlemleri gerçekleştiren bir YSA türüdür. İleri beslemeli ağların çalışma mantığından farklı olarak ağın girdi verileri TSA eğitimlerinde kullanılabilir. Geri yayılım (backpropagation) ise YSA'lara benzer olarak gerçekleştirebilmektedir. TSA ağları, aynı yapıya sahip TSA hücrelerinin birbirinden farklı gizli durumlarını işleyen kopyalarının bir araya gelmesi ile oluşmaktadır. Bu kopyalar TSA hücrelerinin gizli olmasına neden olmaktadır. Bu derin mimari modelinin tekrarlayan olarak isimlendirilmesinin sebebi çıktı verilerinin tekrardan girdi verisi olarak kullanılmasıdır (Metin ve Karasulu, 2021a; Metin ve Karasulu, 2019).

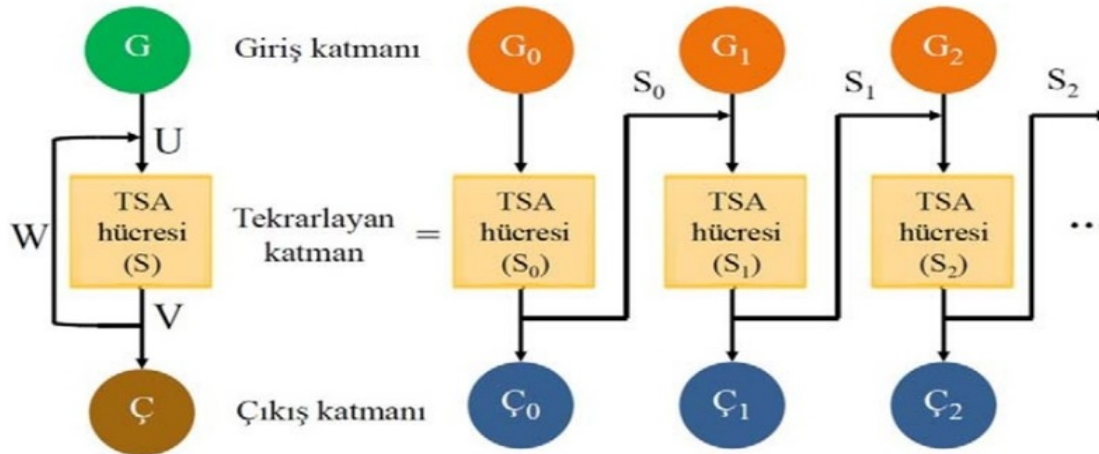


Figure 2.4: TSA'nın genel yapısı

2.4 'de TSA'nın genel yapısı gösterilmektedir (Eşref, 2019).

Burada,  $S_t$  değeri ile TSA hücrelerinin  $t$  anındaki gizli durumları ifade edilmektedir. Bu yöntem yardımı ile o anki adıma kadar ağa giren tüm girdi verileri üzerinden buradaki  $t$  anında işleme alınan  $G_t$  durumundaki girdi verisi hesaplanmaktadır. Ayrıca,  $(t - 1)$  anındaki mevcut gizli durum ise  $S_{t-1}$  değeriyle ifade edilmektedir. Aşağıdaki  $S_t$  eşitliğinde görülebileceği gibi uygun model oluşturularak buna ait  $U$  ve  $W$  ağırlık parametreleri kullanılmasıyla  $S_t$  değeri hesaplanmaktadır.

Bu eşitlikteki  $f_a$  aktivasyon fonksiyonunu ifade etmektedir. Genellikle burada hiperbolik tanjant ( $\tanh$ ) aktivasyon fonksiyonu seçilerek  $[-1,1]$  aralığında sonuç üretmektedir (Z. Yang, Salakhutdinov ve Cohen, 2016).

$$S_t = f_a(UG_t + WS_{t-1})$$

belirtilen hesaplamada normalize işlemi gerçekleştirilmiş softmax fonksiyonu kullanılarak  $[0,1]$  değerleri arasında ağıın çıktısı  $\hat{C}_t$  elde edilmektedir.

$$\hat{C}_t = \text{softmax}(V * S_t)$$

YSA'da girdi verileri tüm katmanlarda kullanıldığından gradyan değeri önceki katmanlara bağımlı biçimde değişmektedir. Geri yayılımın sürekli yenilenmesi sonrasında, hesaplamalardaki değerlerde azalma yaşanması sonucu kaybolan gradyan (vanishing gradient) problemi oluşabilmekte, buna benzer diğer bir durumdaysa 1'den büyük değer alan gradyanlar hesaplanan sonucun büyümesine yol açmaktadırlar. Bu durumda da patlayan gradyan (exploiting gradient) problemi ortaya çıkmaktadır. Oluşan bu problemlerin en aza indirilmesi için, uzun süreli bağımlılıkları da öğrenebilen yapıdaki TSA mimarisinin özel bir tipi olan UKSB hücreleri kullanılmaktadır. Uzun vadeli bağımlılık ve kaybolan gradyan gibi problemlere çözüm için geliştirilmiş olan ve TSA mimarisi göz önüne alındığında daha hızlı ve görece daha basit yapıda olan Kapılı Tekrarlayan Birim (KTB) modelleri de literatürde mevcuttur. Bu mimari bakış açısıyla, unutma kapısı (forget gate), girdi kapısı (input gate) ve çıktı kapıları (output gate) UKSB hücrelerinde bulunmaktadır (Z. Yang ve diğerleri, 2016). KTB hücreleri reset kapısı ve güncelleme kapısı (update gate) içermekte, böylece UKSB hücrelerine göre hafızada bir verinin ne kadar uzun süre tutulup, güncel veri ile hafızadaki verinin hangi zaman adımında birleştirileceği gibi olgular belirlenebilmektedir (Z. Yang ve diğerleri, 2016). Veri dizileri içerisinde belirli bir önceki  $(t - 1)$  ve buna ilişkin sonraki  $(t + 1)$  zaman adımlarını işleyebilen çift yönlü işlevsel yapılar da literatürdeki çeşitli çalışmalarda mevcuttur. Çift yönlü işlevsel bir yapıya sahip olarak oluşturulan KTB ve UKSB hücrelerini içeren çeşitli modellerin aynı zamanda ardışık verileri işlemede bazı avantajları da vardır (Schuster ve Paliwal, 1997).

### 2.3.2 Evrişimli Sinir Ağları (Convolutional Neural Network)

ESA, görme duyusunun işlevinden faydalanan, görüntüleri alt bölümlere parçalayarak bütün görüntü üzerinde işlem yapan Çok Katmanlı Algılayıcı (ÇKA) türüdür. 1988 yılında Yann LeCun'ın öne sürdüğü ve 1998 yılına kadar da üzerinde iyileştirmelerin gerçekleştirildiği ilk ESA ağlarından birisi de LeNet ağıdır (Metin ve Karasulu, 2021b). LeNet ağında; ÇKA yapısına benzeyen tam bağlı katman (fully connected layer), biriktirme katmanı (pooling layer) ve evrişim katmanı (convolution layer) yer almaktadır (LeCun, Bottou, Bengio ve Haffner, 1998). ESA, veriyi işlerken çeşitli filtreler kullanmakta, bu filtrelerin içerikleri veriden elde edilen öznitelikler olarak otomatik bir biçimde ağ tarafından öğrenilebilmektedir. 2.5'te ESA ağının iki boyutlu (2B) yapıda oluşturulan modeline dair altyapı şematik olarak görülmektedir (Bengio, Courville ve Vincent, 2013). ESA modellerinin katmanları incelendiğinde; evrişim katmanı ESA'nın temelini oluşturan dönüşüm katmanıdır. Bu katmanda, belirli ölçeklerdeki filtreler görüntüler üzerinde kaydırılarak öznitelik haritasının (feature map) oluşturulması sağlanır. Biriktirme katmanı; evrişim katmanları arasına eklenen katmandır. Model içerisinde parametre değerlerinin hesaplama miktarını azaltarak aşırı öğrenme (overfitting) nedeniyle ağ'da oluşan ezberlemeyi önler. Bu sayede

ağ üzerinde uyumsuz durumların da engellenmesi sağlanmış olur. Düzleştirilmiş (flatten) veriler, tam bağlı katmanda alınarak sinir ağı vasıtasıyla öğrenme işlemi gerçekleştirilmektedir. İşlemler sonrasında belirlenen sınıflara göre dönüşümler gerçekleştirilmektedir. Bazı katmanlardaki sinir düğümlerinin eğitim süresi içerisinde pasif bir duruma getirilmesi ise iletim sönümü (dropout) katmanında gerçekleştirilir. Sadece ve sadece eğitim süresi içerisinde iletim sönümü gerçekleştirilmekte, tüm sinir düğümleri ise test işlemi süresince aktif durumda olmaktadır.

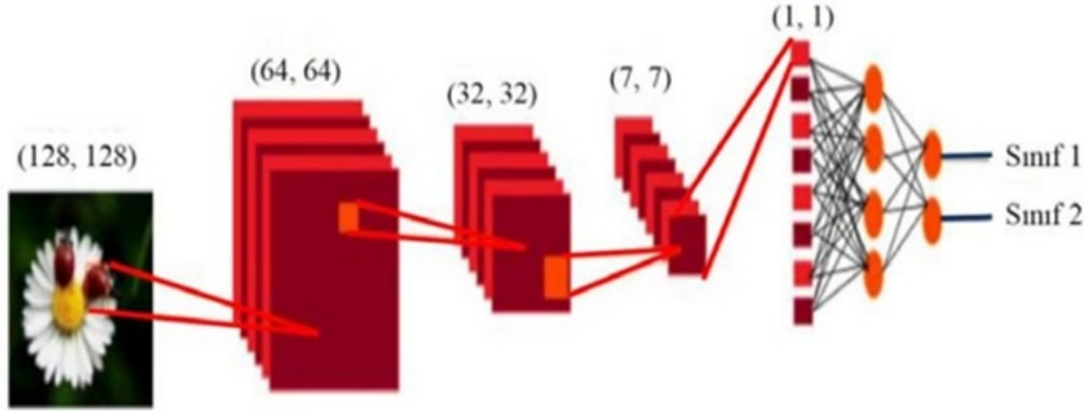


Figure 2.5: Evrişimli Sinir Ağları Yapısı

(Metin ve Karasulu, 2019)

Böylece iletim sönümü kullanımıyla; çok fazla ayrıntıya ağ modelinin odaklanması da engellenerek başarımlar iyileştirilmektedir. Her bir döngü süresince belirli bir oranda farklı sinir düğümleri böylece pasif hale getirilmekte ve aktif haldeki sinir düğümleri sürekli değiştirilmektedir (LeCun ve diğerleri, 1998). Ara katmanların doğrusal olmayan biçimde veriyi işlemede aktivasyon fonksiyonu olarak doğrultulmuş biçimli doğrusal birim olarak bilinen ReLU (Rectified Linear Unit) içeren bir yapı sıklıkla literatürdeki çalışmalarda kullanılmaktadır (Buduma, 2017). Bu fonksiyon sayesinde öznelilikler üzerinde belirli bir filtreleme de yapılmaktadır.

Günümüzde çeşitli duyargalardan elde edilen sinyal verilerinin işlenmesine dayanan çalışmalar literatürde oldukça popüler bir araştırma alanı olmuştur. Özellikle bu tip sinyal verilerinden elde edilen öznelilikler literatürdeki ilgili çalışmalarda klasik makine öğrenmesi veya derin öğrenme teknikleri ile kullanılmakta, insan aktivitelerinin tespiti ve sınıflandırılması, bu aktiviteler sayesinde cinsiyet belirleme gibi çeşitli konulardaki birçok problemin çözümü adına uygulamaya yönelik çalışmalar yapılmaktadır (Kuncan ve Kaya, 2019). Derin öğrenme ağ mimari modelleri gerçek hayattaki problemlere çözüm üreten birçok çalışmada sıklıkla kullanılmaktadır. Özellikle ham veriden otomatik olarak öznelilik öğrenebilme yeteneği ve yüksek doğrulukta başarımlar sayesinde derin öğrenme modelleri gün geçtikçe çalışmalarda daha çok tercih edilir olmaktadır. Derin öğrenme için belli başlı ağ mimari modelleri çeşitli araştırma alanlarındaki çalışmalarda kullanılmaktadır. Literatürdeki örnek teşkil eden son yıllardaki güncel bazı çalışmalara baktığımızda; görüntü işleme



tabanlı olarak köpeklerin davranışlarının incelenmesiyle bunlara dair verilerin daha hızlı bölgesel ESA ağ modeli ile sınıflandırılmasında (Dandıl ve Polattimur, 2020), uydu duyarları ile elde edilen hiperspektral görüntülerin ESA ağ modeli ile sınıflandırılmasında (Hanbay, 2020) ve ses sinyallerinin işlenmesi tabanlı olarak UKSB ağ modelinin kullanıldığı prozodik açıdan Türkçe ağız tanımada (Isık ve Artuner, 2020) derin öğrenme sayesinde yüksek doğrulukta başarımlar elde edildiği görülmektedir. Bu bakış açısıyla çalışmamızda duyarlar sinyallerinden elde edilen öznitelikler kullanılarak derin öğrenme mimari modelleriyle sınıflandırma yapılmasına dayanan uygulamaya yönelik bir çalışma yapılmıştır.

## 2.4 Makine öğrenmesi değerlendirme ölçütleri

Veriler kullanılarak, birçok farklı makine öğrenmesi algoritması ile farklı modeller oluşturmak mümkündür. Oluşturulan modellerden hangisinin daha iyi sonuç vereceğini ölçmek için değerlendirme metriklerine ihtiyaç duyulmaktadır. Değerlendirme metrikleri modelden elde edilen tahminler ile gerçek sonuçları karşılaştırarak bize rakamsal sonuçlar vermektedirler. Makine öğrenmesi modellerinin doğruluğu, 4 farklı değerlendirme metriği ile ölçülür. Karışıklık matrisi, sınıflandırma yapan uygulamalarda, gerçek ve tahmin edilen değerleri bir tablo üzerinden kolayca kıyaslayabilmek için kullanılmaktadır (Kelle ve Hüseyin, 2022).

		Tahmin	
		Pozitif	Negatif
Gerçek	Pozitif	Gerçek Pozitif	Yanlış Negatif
	Negatif	Yanlış Pozitif	Gerçek Negatif

Figure 2.6: Karışıklık matrisi örneği

(Kelle ve Hüseyin, 2022)

2.6'de yer alan matriste;

- Gerçek Pozitif (GP): Doğru olarak tahmin edilen, gerçekte de doğru olan değerler,
- Gerçek Negatif (GN): Yanlış olarak tahmin edilen, gerçekte de yanlış olan değerler,
- Yanlış Pozitif (YP): Doğru olarak tahmin edilen, gerçekte yanlış olan değerler,
- Yanlış Negatif (YN): Yanlış olarak tahmin edilen, gerçekte doğru olan değerleri ifade etmektedir.

**Doğruluk (Accuracy)**

Doğru tahmin edilen değerlerin, tüm değerlere bölünmesi ile elde edilmektedir. Doğruluk değeri, 0 ile 1 arasındadır. Doğruluk değeri 1'e yaklaştıkça başarı artmaktadır(Kelle ve Hüseyin, 2022).

$$Doğruluk = \frac{(GP + GN)}{(GP + GN + FP + FN)}$$

**Duyarlılık (Recall)**

Doğru olarak tahmin etmemiz gereken değerlerin, ne kadarını doğru tahmin ettiğimizi belirtmektedir. Duyarlılık değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, tüm doğru değerlere bölünmesi ile elde edilmektedir(Kelle ve Hüseyin, 2022).

$$Duyarlilik = \frac{GP}{(GP + FN)}$$

**Kesinlik (Precision)**

Doğru olarak tahmin ettiğimiz değerlerin, ne kadarının gerçekte doğru olduğunu göstermektedir. Kesinlik değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, doğru olarak tahmin edilen tüm değerlere bölünmesi ile elde edilmektedir(Kelle ve Hüseyin, 2022).

$$Kesinlik = \frac{GP}{(GP + FP)}$$

**F1 Skoru (F1 Score)**

F1 skoru, duyarlılık ve kesinlik değerlerinin harmonik ortalamasının hesaplanması ile elde edilmektedir. Her iki değerinde hesaplamaya katılarak dengeli bir değer elde edilmesi amaçlanmaktadır. Eşit dağılıma sahip olmayan veri setlerinde başarılı sonuçlar elde etmek için kullanılmaktadır(Kelle ve Hüseyin, 2022).

$$F1Skoru = \frac{2 * Kesinlik * Duyarlilik}{(Kesinlik + Duyarlilik)}$$

# Chapter 3

## UYGULAMA

### 3.1 Uygulama

Bu bölümde akıllı telefon sensörleri ile elde edilen insan aktivitesi tanıma verileri kullanılmıştır(Anguita ve diğerleri, 2013). Gözlemler, jiroskop ve ivmeölçer sensörlerinden 100hz frekans, 1.28 saniyelik pencere aralığı ile elde edilmiştir. Sınıflandırma yöntemleri olarak K En Yakın Komşular, Rassal Ormanlar, XGBoost, Destek Vektör Makinesi, Yapay Sinir Ağları yöntemleri kullanılmıştır.Takip eden bölümde uygulamada kullanılan veriler tanıtılmış ve keşifsel veri analizi sonuçlarına yer verilmiştir. Kullanılan verilerde, Jiroskop ve ivmeölçerden elde edilen X,Y,Z eksenlerindeki ölçümler üzerinde aşağıdaki istatistiksel fonksiyonlar kullanılarak 561 değişken türetilmiştir.Veriler -1,1 aralığında olacak şekilde normalize edilmiştir.

- mean(): Ortalama
- std(): Standart sapma
- mad(): Medyan mutlak sapma
- max(): Dizideki en büyük değer
- min(): Dizideki en küçük değer
- sma(): Sinyal büyüklük alanı
- energy(): Karelerin toplamı bölü değer sayısı (enerji ölçüsü)
- iqr(): Çeyrekler arası aralık
- entropy(): Sinyal entropisi
- arCoeff(): Burg mertebesi 4'e eşit olan otoregresyon katsayıları
- correlation(): İki sinyal arasındaki korelasyon katsayısı
- maxInds(): En büyük genliğe sahip frekans bileşeninin indeksi
- meanFreq(): Ortalama bir frekans elde etmek için frekans bileşenlerinin ağırlıklı ortalaması
- skewness(): Frekans alanı sinyalinin çarpıklığı
- kurtosis(): Frekans alanı sinyalinin basıklığı
- bandsEnergy(): Her pencerenin FFT'sinin 64 kutusu içindeki bir frekans aralığının enerjisi
- angle(): Vektör arasındaki açı

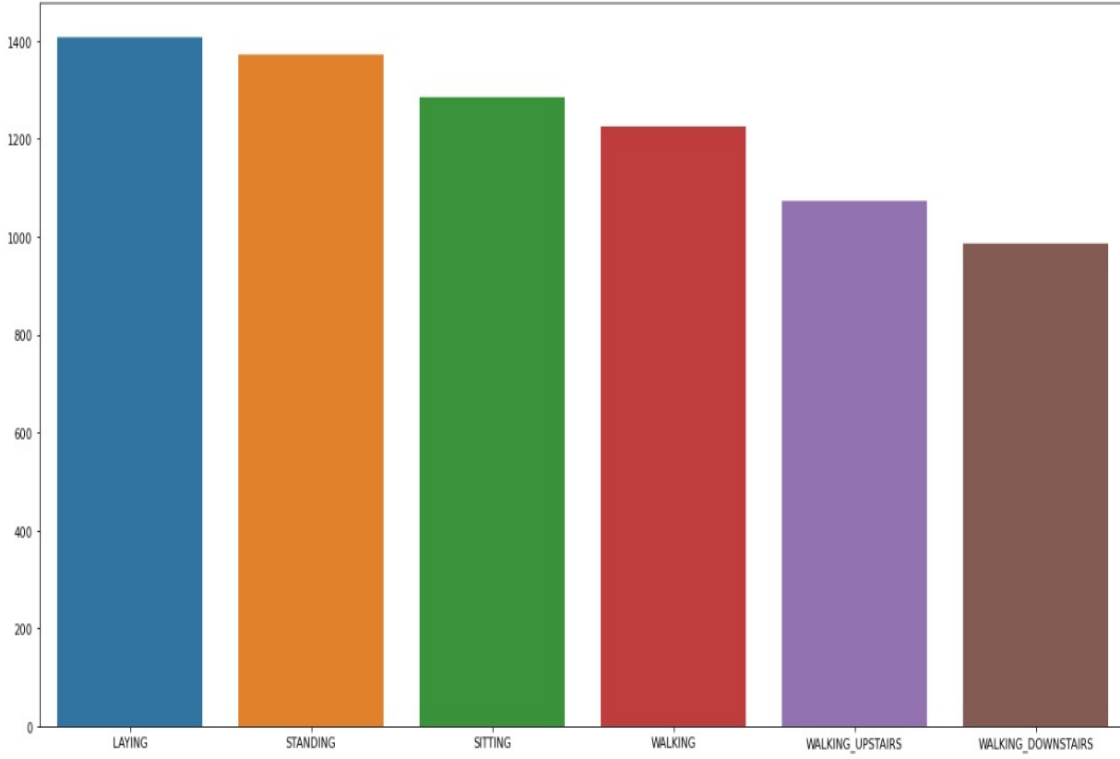


Figure 3.1: Hedef deęiřkene gre gzlem sayıları

Hedef deęiřken; Yatma(0), Ayakta Durma (1), Oturma(2), Yrme(3), Merdiven ıkma(4), Merdiven İnme(5) olmak zere 6 farklı kategoriye sahiptir. 3.1.' de hedef deęiřkene gre gzlem sayıları grselleřtirilmiřtir.

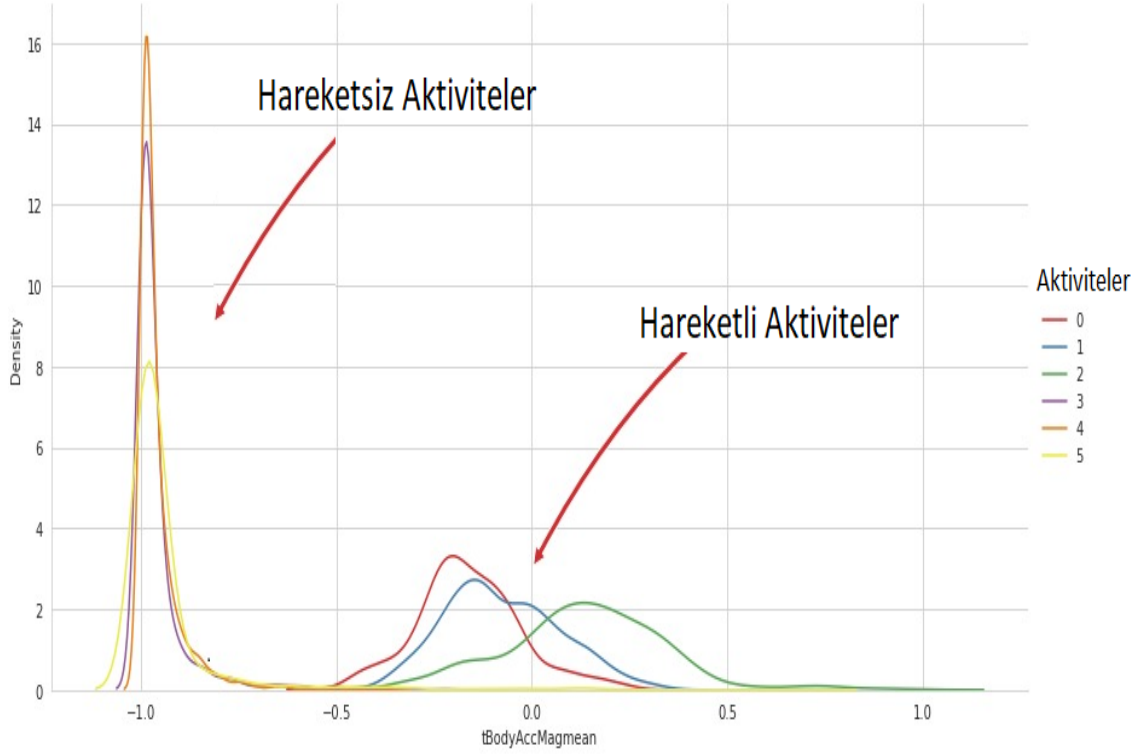


Figure 3.2: Sabit ve hareketli aktiviteler için  $tBodyAccMagmean$  değişkeninin yoğunluk grafiği

3.2. Veri setindeki ' $tBodyAccMagmean$ ' değişkeninin yoğunluk grafiğidir. Aktiviteler, farklı renklerde gösterilmiş. Grafiğin sol tarafında "Sabit Aktiviteler" olarak adlandırılan Uzanma, Ayakta durma, Oturma, sağ tarafındaki ise "Hareketli Aktiviteler" olarak adlandırılan Yürüme, Merdiven inme, Merdiven çıkma aktivitelerini temsil etmektedir.

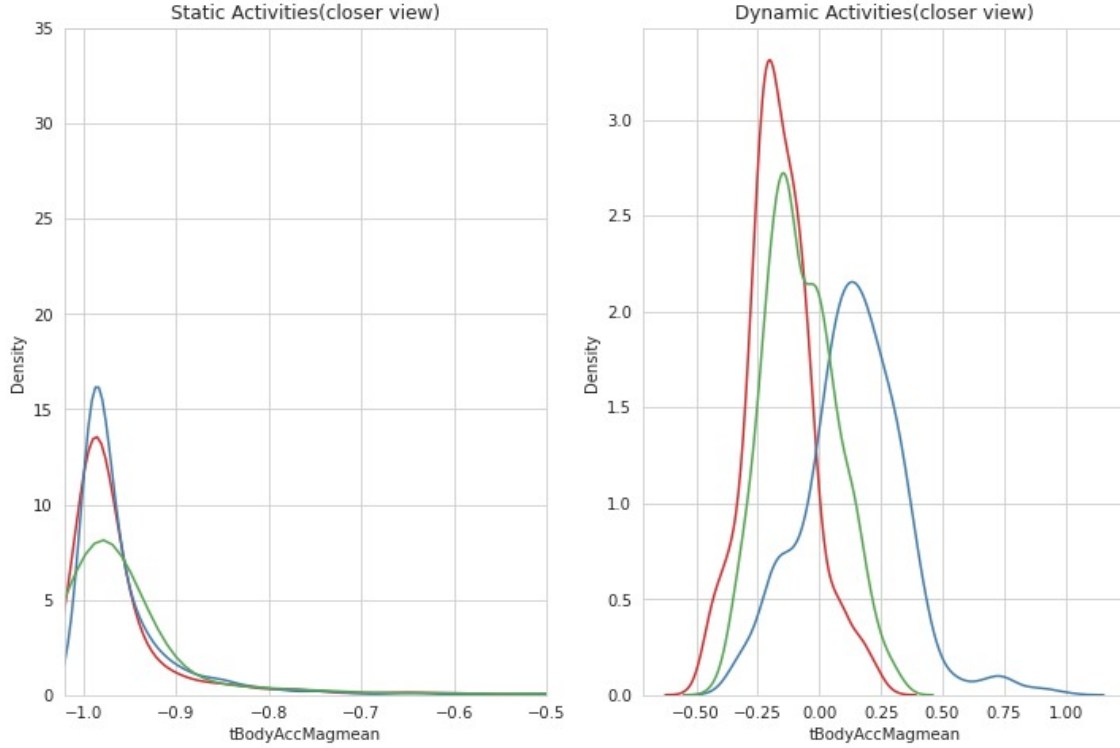


Figure 3.3: Sabit ve hareketli aktivitelerin yoğunluk grafiklerinin karşılaştırılması

3.3. ‘tBodyAccMagmean’ değişkeninin aktivitelere göre ayrılmış ve yakınlaştırılmış yoğunluk grafikleridir. İlk grafik, “Oturma”, “Ayakta durma” ve “Yatma” aktivitelerini içerir ve yatay eksen, “-1.02” ile “-0.5” arasında bir aralığı kapsar. İkinci grafik ise “Yürüme”, “Aşağı inme” ve “Yukarı çıkma” aktivitelerini içerir ve yine yatay eksen aynı aralıkta kısıtlanmıştır. Her iki grafik de, aktivitelerin ayrı ayrı dağılımlarını farklı renklerle gösterir ve dağılımların şekli hakkında bilgi sağlar. Bu grafiği oluşturmak için seaborn kütüphanesi kullanılmıştır.

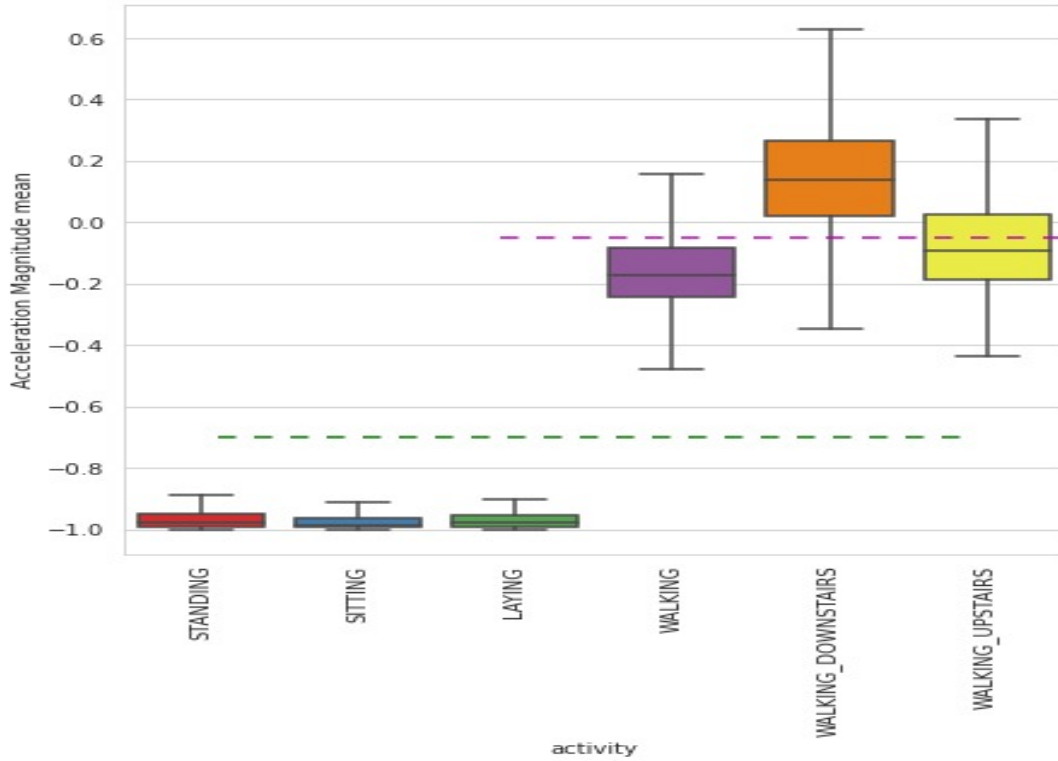


Figure 3.4: tBodyAccMagmean Değişkeninin Aktivitelere Göre BoxPlot Grafiği

3.4. Her bir aktivitenin ortalama ivme büyüklüğünün dağılımını gösterir. Kesik çizgiler, değişkenin değerinin belirli bir eşğin altında veya üstünde olma durumlarını gösterir. ‘y=- 0.7’ çizgisi, sabit haldeki aktiviteler için eşği temsil ederken, ‘y=- 0.05’ çizgisi, hareket halindeki aktiviteler için eşği temsil eder. `plt.axhline()` fonksiyonu içerisindeki ‘dashes’ parametresi, çizgilerin kesikli görüntüsünü sağlar. Bu grafiği oluşturmak için `seaborn` kütüphanesindeki fonksiyonlar kullanılmıştır. Aşağıdaki kurallar verideki tBodyAccMagmean değişkeninin aldığı değerlere göre aktivitelerin gruplanmasını göstermektedir;

- Eğer  $-0.8 > \text{tBodyAccMagmean} > -1$  ise , sabit aktivite (ayakta, oturuyor veya uzanıyor) olarak değerlendirilebilir.
- Eğer  $-0.6 < \text{tBodyAccMagmean} < 0.7$  ise, hareketli aktivite (yürüyor, merdivenlerden aşağı iniyor ya da merdivenlerden yukarı çıkıyor) olarak değerlendirilebilir.

Araştırma çok sınıflı sınıflama problemi olarak ele alındığında makine öğrenmesi yöntemleri ile sınıflandırma modelleri geliştirilmesi gereklidir.

## 3.2 Çok Sınıflı (Multiclass) Sınıflama Problemi

Bu çalışmada akıllı telefon sensörleri ile elde edilen insan aktivitelerinin sınıflandırması Rassal Ormanlar , XGBoost, Destek Vektör Makinesi (SVM), K-En Yakın

Komşuluk Modeli (KNN), Yapay Sinir Ağları yöntemleri kullanılmıştır. İlk olarak orijinal verilerdeki altı aktivite (Yürüme, Yukarı çıkma, Aşağı inme, Oturma, Ayakta, Yatma) için sınıflandırma hedeflenmiştir. Bölüm 3.1’de bu problem için farklı modeller ile elde edilen sonuçlara yer verilmiştir. Uygulamada Python programlama dili kullanılmış ve Scikit-Learn (Buitinck ve diğerleri, 2013), Pandas, Numpy, Matplotlib, LazyClassifier, SVM, XGBoost, KNeighborsClassifier Tensorflow, Keras kütüphanelerinden yararlanılmıştır. Modellerin performanslarını değerlendirmek üzere kesinlik, duyarlılık, F1-skoru, doğruluk oranı, dengelenmiş doğruluk oranı hesaplanmıştır.

Üç ayrı değişken grubu kullanılarak modellemeler yapılmıştır:

- Jiroskop Sensörüne Ait Değişkenler İle Modelleme
- İvme Ölçer Sensörüne Ait Değişkenler İle Modelleme
- Jiroskop ve İvme Ölçer Sensörlerine Ait Değişkenler İle Modelleme

Sensörlerin ayrı ayrı ve birlikte kullanılmasının modellerin performansına etkisi araştırılmıştır.

Table 3.1: Jiroskop ve İvme Ölçer Sensörlerinden elde edilen değişkenler

Değişkenler	Elde edilen değişken sayısı
fBodyGyro	79
fBodyAcc	79
fBodyAccJerk	79
tBodyAcc	40
tBodyAccJerk	40
tBodyGyro	40
tBodyGyroJerk	40
tGravityAcc	40
fBodyGyroJerkMag	13
fBodyGyroMag	13
fBodyAccJerkMag	13
fBodyAccMag	13
tBodyGyroJerkMag	13
tBodyGyroMag	13
tBodyAccJerkMag	13
tGravityAccMag	13
tBodyAccMag	13
angle	7



Table 3.2: Sadece Jiroskop Sensöründen elde edilen Değişkenler

Değişkenler	Elde edilen değişken sayısı
fBodyGryo	79
tBodyGyro	40
tBodyGyroJerk	40
tBodyGyroMag	13
tBodyGyroJerkMag	13
fBodyGyroMag	13
fBodyGyroJerkMag	13
angle	5

Table 3.3: Sadece İvmeölçer Sensöründen elde edilen Değişkenler

Değişkenler	Elde edilen değişken sayısı
fBodyAcc	79
fBodyAccJerk	79
tBodyAcc	40
tGravityAcc	40
tBodyAccJerk	40
tBodyAccMag	13
tGravityAccMag	13
tBodyAccJerkMag	13
fBodyAccMag	13
fBodyAccJerkMag	13
angle	5

### 3.3 Jiroskop Sensörüne Ait Değişkenler İle Modelleme

#### 3.3.1 KNN Modeli

Bu bölümde jiroskop sensöründen elde edilen veriler kullanılmıştır. K-En yakın komşuluk modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.2’ te belirtilen değişkenler kullanılarak modelleme gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır. K-En yakın komşuluk modeli için en yüksek doğruluk oranı aşağıdaki parametreler ile bulunmuştur;

- ‘metric’: ‘manhattan’
- ‘n\_neighbors’: 9
- ‘weights’: ‘distance’

## En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.4: Jiroskop Sensöründen elde edilen değişkenler ile kurulan k en yakın komşular modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.70	0.84	0.77	496
1	0.86	0.75	0.80	471
2	0.76	0.71	0.74	420
3	0.82	0.75	0.78	491
4	0.79	0.89	0.84	531
5	1.00	0.95	0.97	537
Accuracy			0.82	2946
macro avg	0.82	0.81	0.82	2946
weighted avg	0.83	0.82	0.82	2946

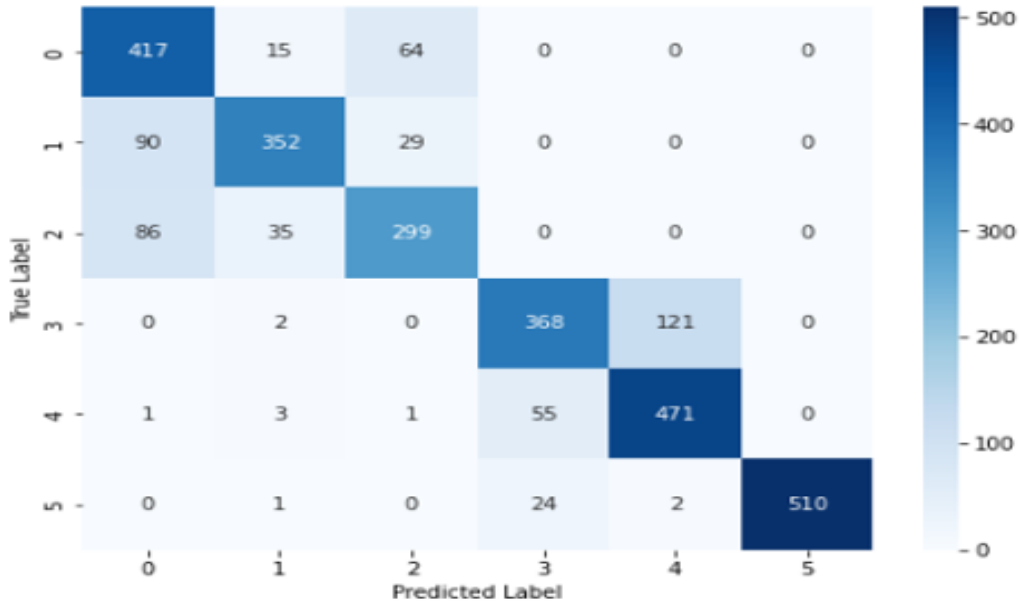


Figure 3.5: K-NN Modeli Karmaşıklık Matrisi

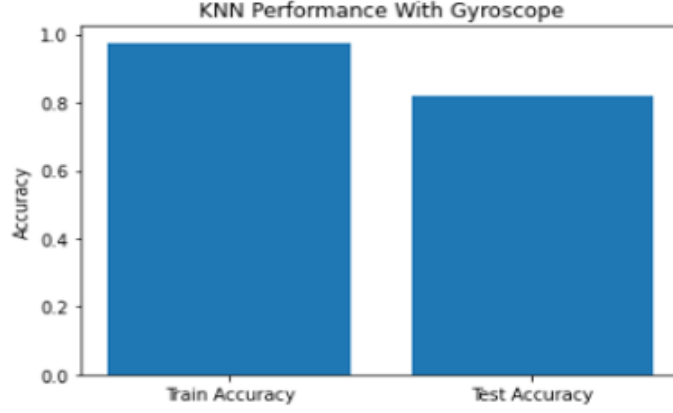


Figure 3.6: K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.3.2 Rassal Ormanlar Modeli

Bu bölümde jiroskop sensöründen elde edilen veriler kullanılmıştır. Rassal ormanlar modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.2' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 20
- min\_samples\_leaf: 4
- min\_samples\_split: 10
- n\_estimators: 500

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.5: Jiroskop Sensöründen elde edilen değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.89	0.85	0.87	496
1	0.89	0.94	0.92	471
2	0.81	0.81	0.81	420
3	0.97	0.88	0.92	491
4	0.90	0.97	0.93	531

	precision	recall	F1-score	support
5	1.00	1.00	1.00	537
Accuracy			0.91	2946
macro avg	0.91	0.81	0.91	2946
weighted avg	0.91	0.82	0.91	2946

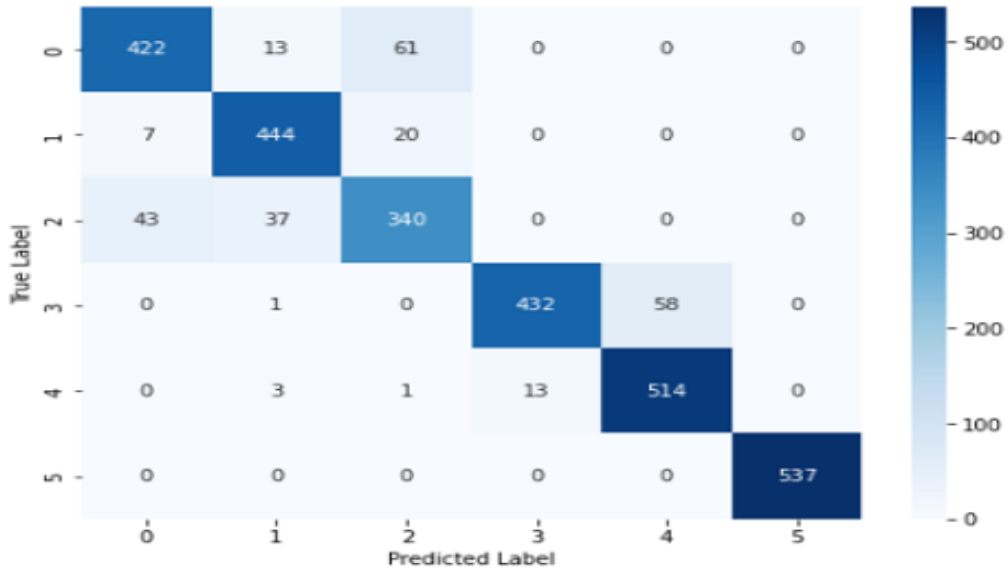


Figure 3.7: Rassal Ormanlar Modeli Karmaşıklık Matrisi

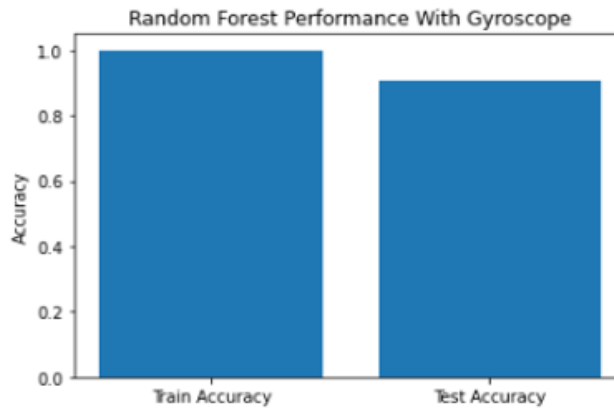


Figure 3.8: Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.3.3 eXtreme Gradient Boosting (XGBoost)

Bu bölümde jiroskop sensöründen elde edilen veriler kullanılmıştır. XGBoost modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.2.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 3
- learning\_rate: 0.01
- n\_estimators: 100

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.6: Jiroskop Sensöründen elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.91	0.84	0.87	496
1	0.84	0.90	0.87	471
2	0.81	0.83	0.82	420
3	0.94	0.86	0.90	491
4	0.88	0.94	0.91	531
5	1.00	1.00	1.00	537
Accuracy			0.90	2946
macro avg	0.90	0.89	0.89	2946
weighted avg	0.90	0.90	0.90	2946

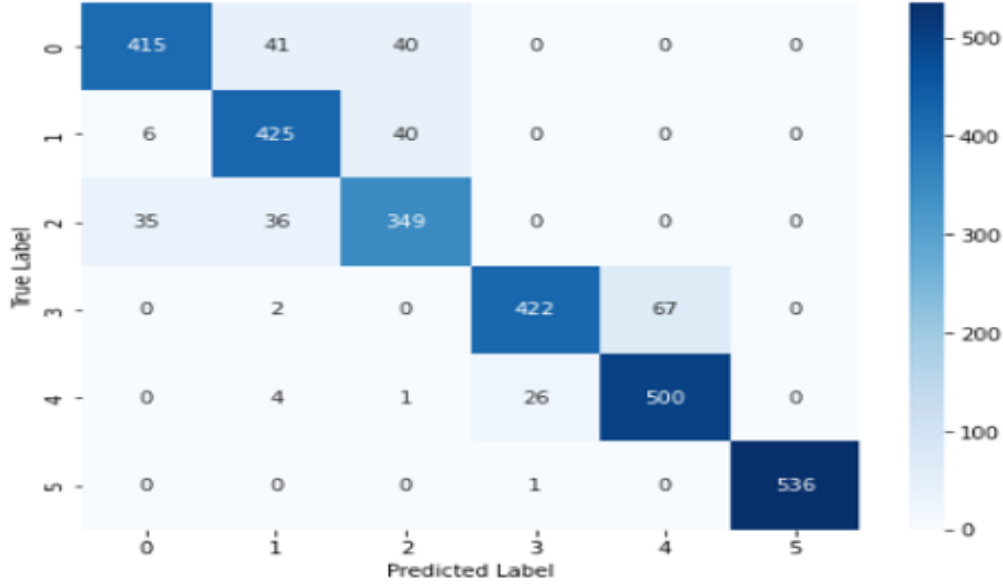


Figure 3.9: XGBoost Modeli Karmaşıklık Matrisi

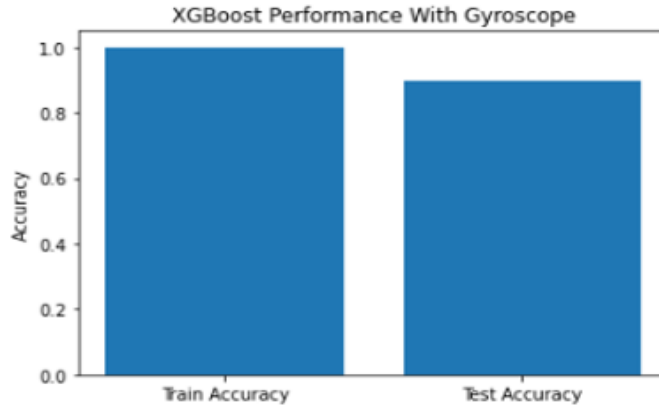


Figure 3.10: XGBoost Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.3.4 Destek Vektör Makinesi

Bu bölümde jiroskop sensöründen elde edilen veriler kullanılmıştır. Destek Vektör Makinesi (SVM) modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.2.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- C: 0.1
- gamma: 0.01
- kernel: linear

### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.7: Jiroskop Sensöründen elde edilen değişkenler ile kurulan destek vektör makinası modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.86	0.89	0.88	496
1	0.94	0.94	0.94	471
2	0.83	0.83	0.83	420
3	0.92	0.88	0.90	491
4	0.90	0.92	0.91	531
5	1.00	1.00	1.00	537
Accuracy			0.91	2946
macro avg	0.91	0.91	0.91	2946
weighted avg	0.91	0.91	0.91	2946

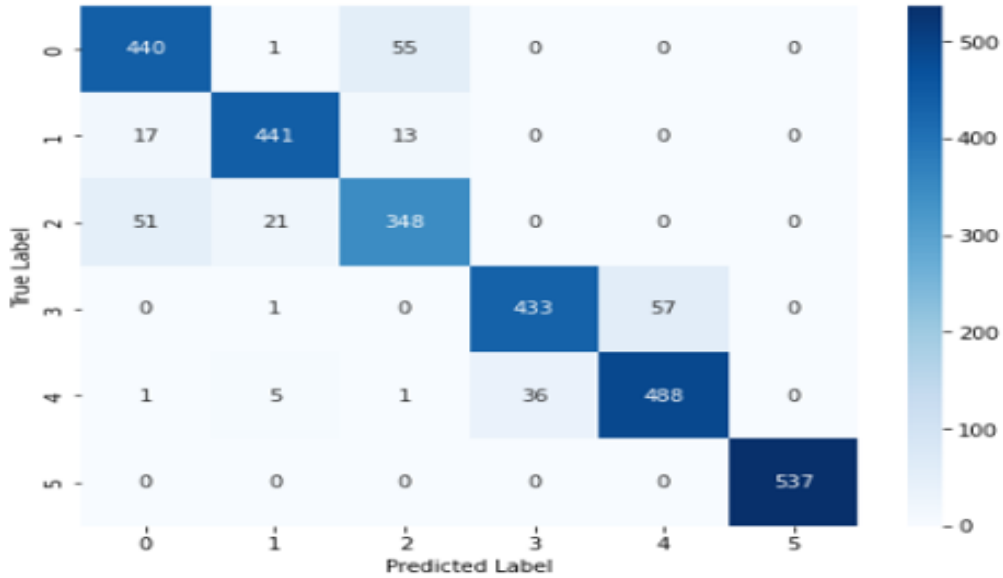


Figure 3.11: Destek Vektör Makinesi Modeli Karmaşıklık Matrisi

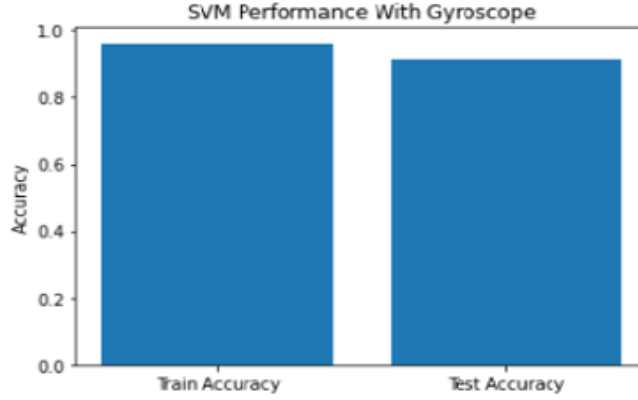


Figure 3.12: Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.3.5 Yapay Sinir Ağları (Neural Networks)

Bu bölümde jiroskop sensörü üzerinde yapay sinir ağları modeli kullanılmış ve çıktıları değerlendirilmiştir. Tensorflow ve Keras kütüphanelerindeki Sequential() fonksiyonu kullanılmıştır.

#### Kullanılan Katmanlar

İlk katman, tamamen bağlı (Dense) katmandır. Bu katman 64 nörona sahiptir ve girdi boyutu, train veri kümesinin özellik sayısına eşittir. kernel\_initializer parametresi “normal” olarak ayarlanmıştır, yani rastgele normal bir dağılım kullanılarak başlangıç ağırlıkları oluşturulacaktır. Bu katmanın aktivasyon fonksiyonu “sigmoid” olarak ayarlanmıştır. İkinci katman, bir Dropout katmanıdır. Dropout, ağıdaki aşırı uyumu azaltmak için kullanılan bir düzenleme tekniğidir. Bu katmanın Dropout oranı %20 olarak ayarlanmıştır. Üçüncü ve son katman, yine bir tamamen bağlı (Dense) katmandır. Bu katman 6 nörona sahiptir ve softmax aktivasyon fonksiyonu kullanılarak çoklu sınıflandırma işlemi gerçekleştirilir. kernel\_initializer parametresi “normal” olarak ayarlanmıştır. Modelin derlenmesi, “adam” optimizasyon algoritması kullanılarak gerçekleştirilir. Kayıp fonksiyonu “sparse\_categorical\_crossentropy” olarak ayarlanmıştır, çünkü etiketler doğrudan kategorik olarak kodlanmamıştır. Ölçülen metrik, doğruluk (accuracy) olarak ayarlanmıştır. 3.2’ te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### En iyi Model

Kullanılan parametrelerle ve belirtilen epochs değeri ile kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.



Table 3.8: Jiroskop Sensöründen elde edilen değişkenler ile kurulan yapay sinir ağları algoritmasının başarı sonuçları

	precision	recall	F1-score	support
0	0.87	0.83	0.85	496
1	0.95	0.96	0.96	471
2	0.80	0.84	0.82	420
3	0.96	0.86	0.91	491
4	0.87	0.96	0.92	531
5	1.00	0.99	0.99	537
Accuracy			0.91	2946
macro avg	0.91	0.91	0.91	2946
weighted avg	0.91	0.91	0.91	2946

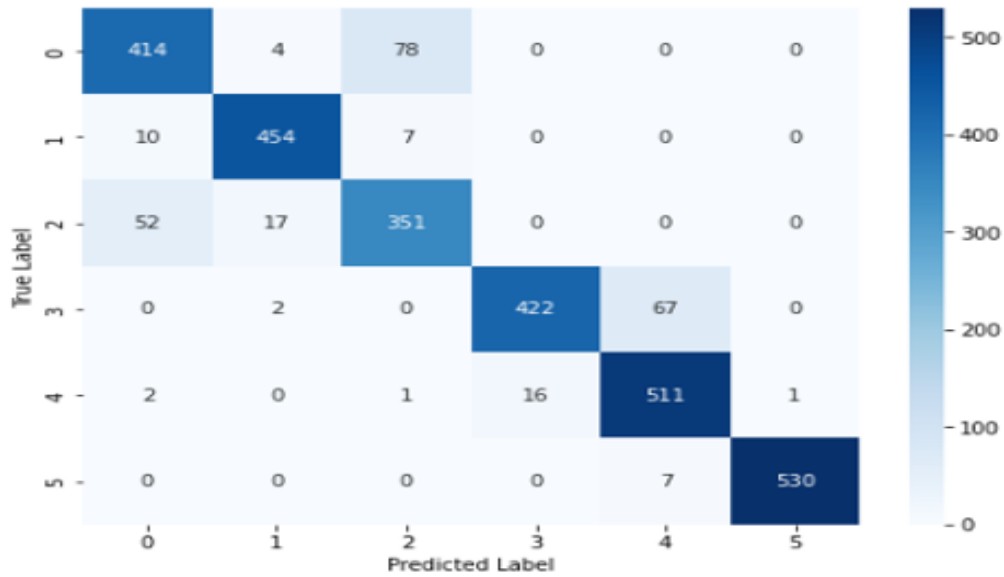


Figure 3.13: Yapay Sinir Ağları Modeli Karmaşıklık Matrisi

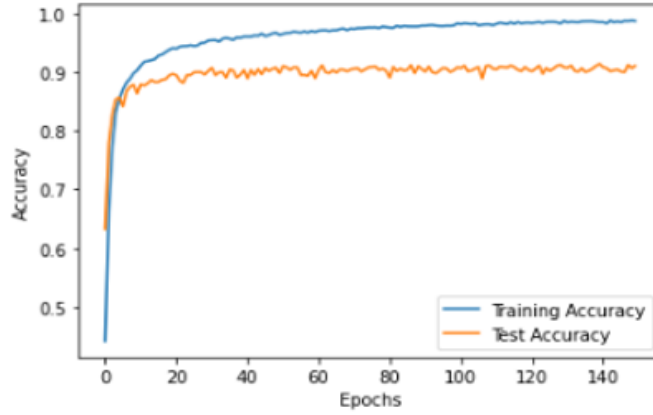


Figure 3.14: Jiroskop Sensörü İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları

## 3.4 İvme Ölçer Sensörüne Ait Değişkenler İle Modelleme

### 3.4.1 K-En Yakın Komşuluk Modeli

Bu bölümde ivme ölçer sensöründen elde edilen veriler kullanılmıştır. K-En yakın komşuluk modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.3.' te belirtilen değişkenler kullanılarak modelleme gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır. K-En yakın komşuluk modeli için en yüksek doğruluk oranı aşağıdaki parametreler ile bulunmuştur;

- metric: manhattan
- n\_neighbors: 9
- weights: distance

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.9: İvmeölçer Sensöründen elde edilen değişkenler ile kurulan k en yakın komşular algoritmasının başarı sonuçları

	precision	recall	F1-score	support
0	0.89	0.85	0.87	496

	precision	recall	F1-score	support
1	0.89	0.94	0.92	471
2	0.81	0.81	0.81	420
3	0.97	0.88	0.92	491
4	0.90	0.97	0.93	531
5	1.00	1.00	1.00	537
Accuracy			0.91	2946
macro avg	0.91	0.91	0.91	2946
weighted avg	0.91	0.91	0.91	2946

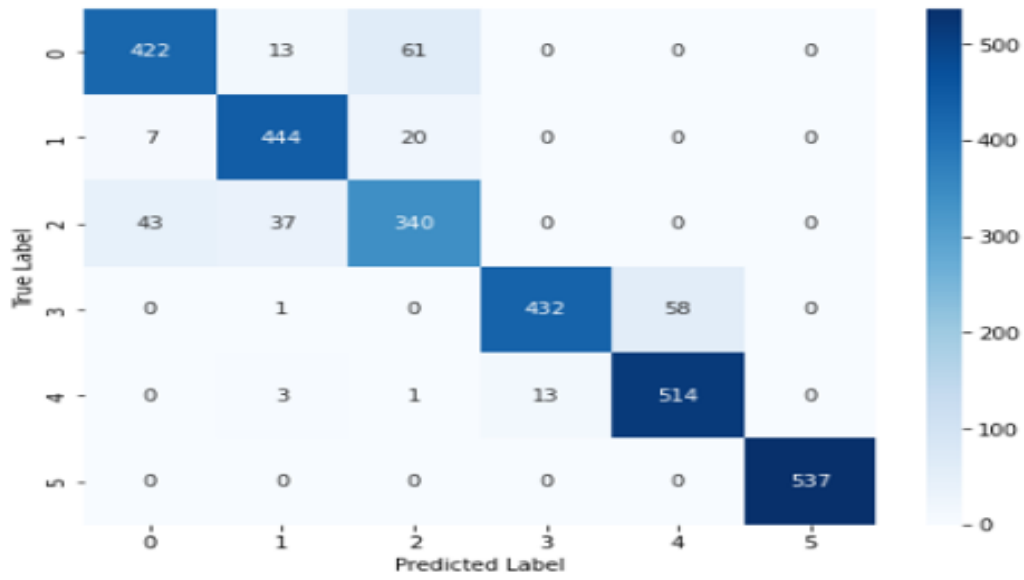


Figure 3.15: K-NN Modeli Karmaşıklık Matrisi

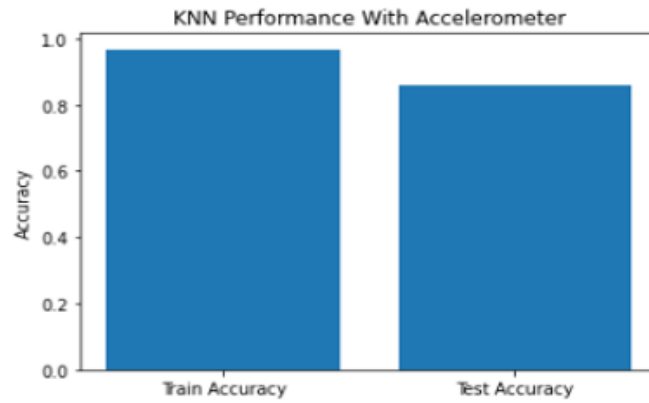


Figure 3.16: K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.4.2 Rassal Ormanlar Modeli

Bu bölümde ivme ölçer sensöründen elde edilen veriler kullanılmıştır. Rassal ormanlar modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.3.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 20
- min\_samples\_leaf: 4
- min\_samples\_split: 10
- n\_estimators: 500

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.10: İvmeölçer Sensöründen elde edilen değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.85	0.96	0.90	496
1	0.88	0.86	0.87	471
2	0.95	0.94	0.89	420
3	0.81	0.81	0.81	491
4	0.82	0.83	0.83	531
5	1.00	1.00	1.00	537
Accuracy			0.88	2946
macro avg	0.89	0.88	0.88	2946
weighted avg	0.89	0.88	0.88	2946

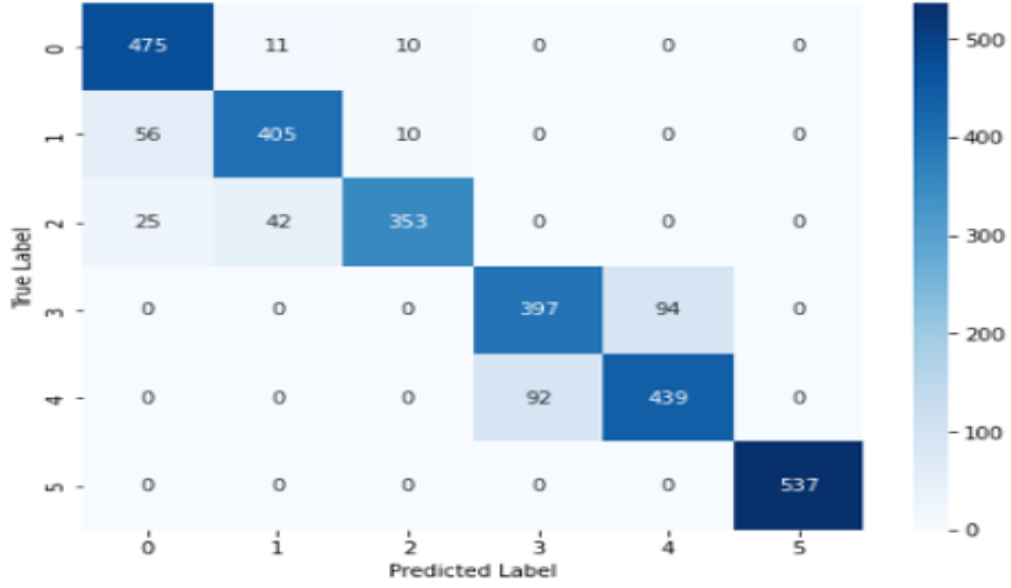


Figure 3.17: Rassal Ormanlar Modeli Karmaşıklık Matrisi

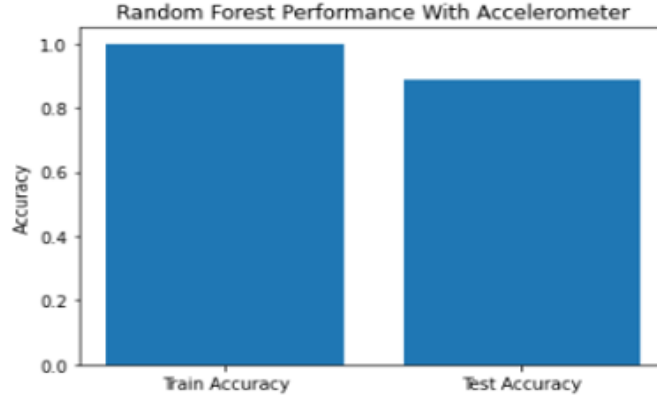


Figure 3.18: Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.4.3 eXtreme Gradient Boosting (XGBoost)

Bu bölümde ivme ölçer sensöründen elde edilen veriler kullanılmıştır. XGBoost modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.3.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 3
- learning\_rate: 0.01
- n\_estimators: 100

## En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.11: İvmeölçer Sensöründen elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.88	0.95	0.91	496
1	0.87	0.89	0.88	471
2	0.98	0.89	0.93	420
3	0.82	0.80	0.81	491
4	0.82	0.83	0.83	531
5	1.00	1.00	1.00	537
Accuracy			0.89	2946
macro avg	0.90	0.89	0.89	2946
weighted avg	0.89	0.89	0.89	2946

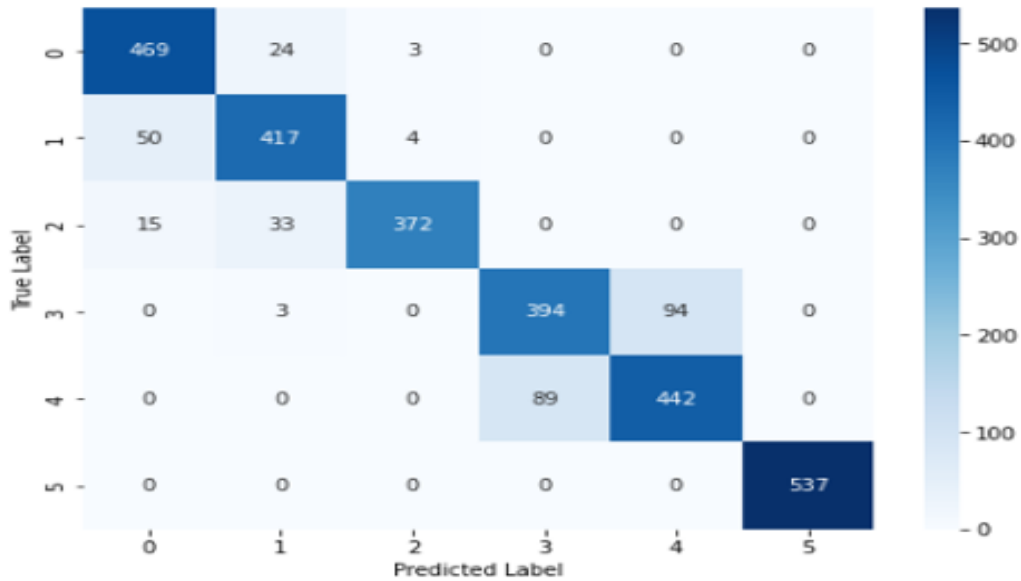


Figure 3.19: Xgboost modeli Karmaşıklık Matrisi

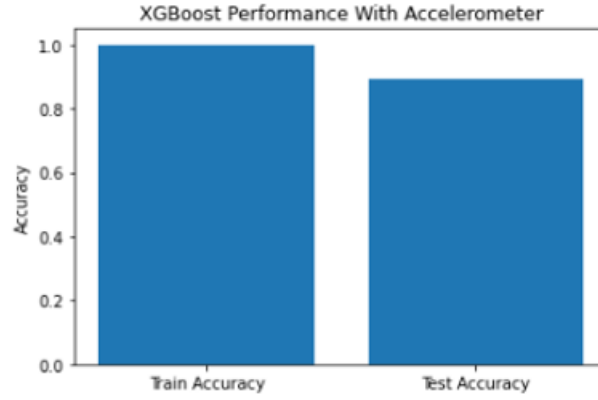


Figure 3.20: Xgboost Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.4.4 Destek Vektör Makinesi

Bu bölümde ivme ölçer sensöründen elde edilen veriler kullanılmıştır. Destek Vektör Makinesi (SVM) modeli kurulmuş ve çıktıları değerlendirilmiştir 3.3' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- C: 0.1
- gamma: 0.01
- kernel: linear

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.12: İvmeölçer Sensöründen elde edilen değişkenler ile kurulan destek vektör makinası algoritmasının başarı sonuçları

	precision	recall	F1-score	support
0	0.89	0.96	0.92	496
1	0.90	0.90	0.90	471
2	0.98	0.90	0.94	420
3	0.86	0.77	0.81	491
4	0.81	0.88	0.84	531
5	1.00	1.00	1.00	537

	precision	recall	F1-score	support
Accuracy			0.90	2946
macro avg	0.91	0.90	0.90	2946
weighted avg	0.91	0.90	0.90	2946

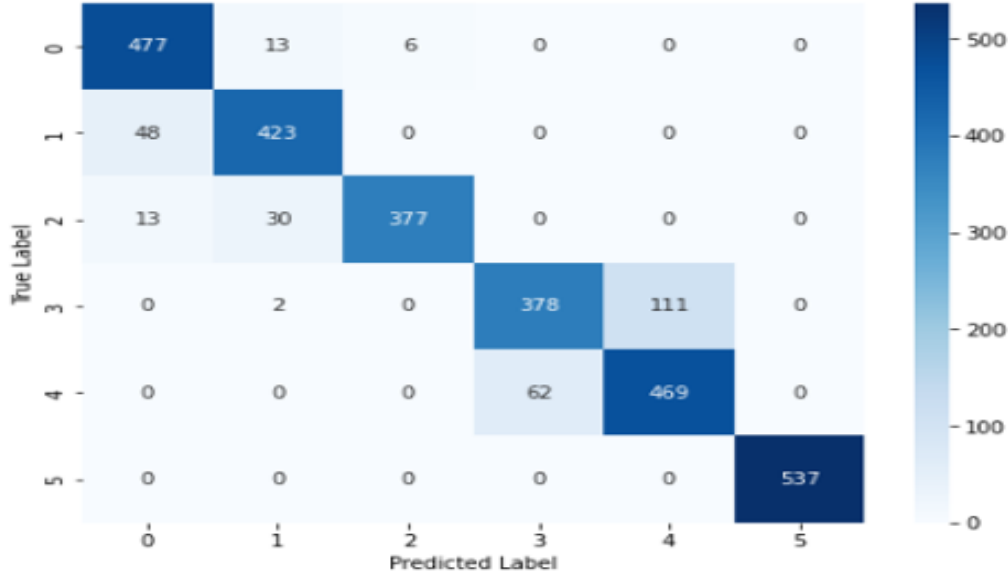


Figure 3.21: Destek Vektör Makinesi Modeli Karmaşıklık Matrisi

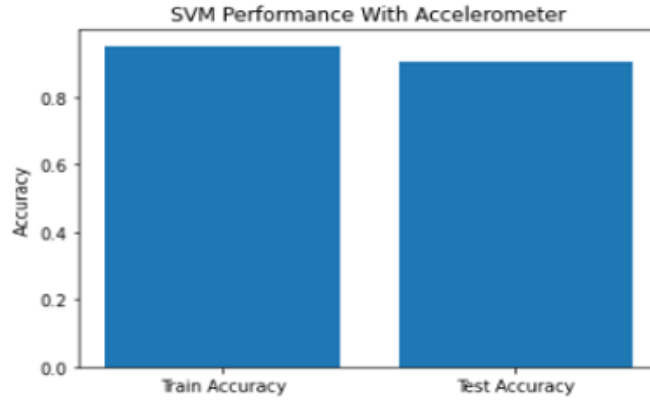


Figure 3.22: Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.4.5 Yapay Sinir Ağları (Neural Networks)

Bu bölümde ivme ölçer sensörü üzerinde yapay sinir ağları modeli kullanılmış ve çıktıları değerlendirilmiştir. Tensorflow ve Keras kütüphanelerindeki Sequential() fonksiyonu kullanılmıştır.



### Kullanılan Katmanlar

İlk katman, tamamen bağlı (Dense) katmandır. Bu katman 64 nörona sahiptir ve girdi boyutu, train veri kümesinin özellik sayısına eşittir. `kernel_initializer` parametresi “normal” olarak ayarlanmıştır, yani rastgele normal bir dağılım kullanılarak başlangıç ağırlıkları oluşturulacaktır. Bu katmanın aktivasyon fonksiyonu “sigmoid” olarak ayarlanmıştır. İkinci katman, bir Dropout katmanıdır. Dropout, ağdaki aşırı uyumu azaltmak için kullanılan bir düzenleme tekniğidir. Bu katmanın Dropout oranı %20 olarak ayarlanmıştır. Üçüncü ve son katman, yine bir tamamen bağlı (Dense) katmandır. Bu katman 6 nörona sahiptir ve softmax aktivasyon fonksiyonu kullanılarak çoklu sınıflandırma işlemi gerçekleştirilir. `kernel_initializer` parametresi “normal” olarak ayarlanmıştır. Modelin derlenmesi, “adam” optimizasyon algoritması kullanılarak gerçekleştirilir. Kayıp fonksiyonu “sparse\_categorical\_crossentropy” olarak ayarlanmıştır, çünkü etiketler doğrudan kategorik olarak kodlanmamıştır. Ölçülen metrik, doğruluk (accuracy) olarak ayarlanmıştır. 3.3.’ te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

### En iyi Model

Kullanılan parametrelerle ve belirtilen epochs değeri ile kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.13: İvmeölçer Sensöründen elde edilen değişkenler ile kurulan yapay sinir ağları modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.89	0.98	0.93	496
1	0.95	0.87	0.91	471
2	0.98	0.95	0.97	420
3	0.85	0.81	0.83	491
4	0.82	0.86	0.84	531
5	1.00	0.97	0.99	537
Accuracy			0.91	2946
macro avg	0.91	0.91	0.91	2946
weighted avg	0.91	0.91	0.91	2946

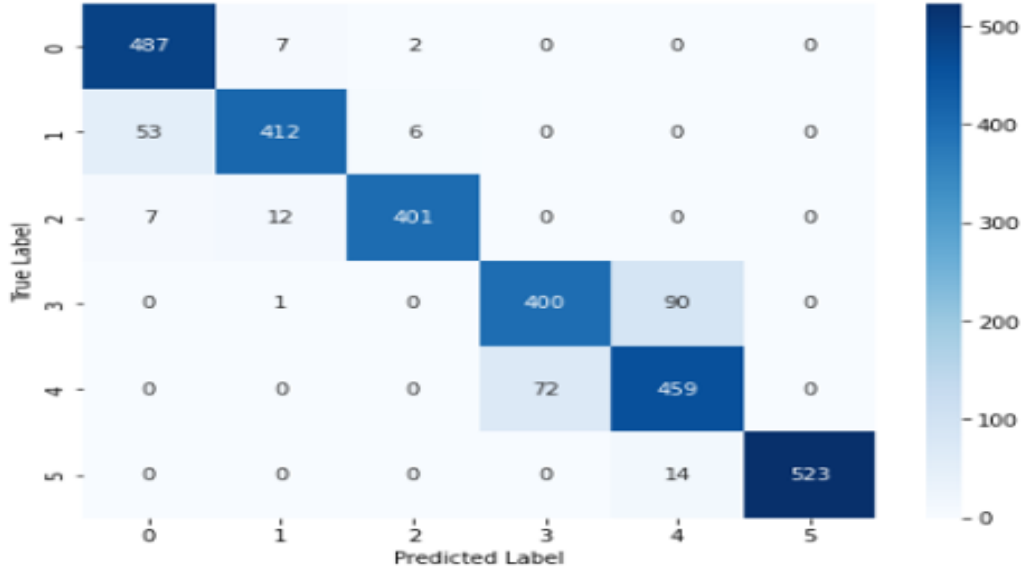


Figure 3.23: Yapay Sinir Ağları Modeli Karmaşıklık Matrisi

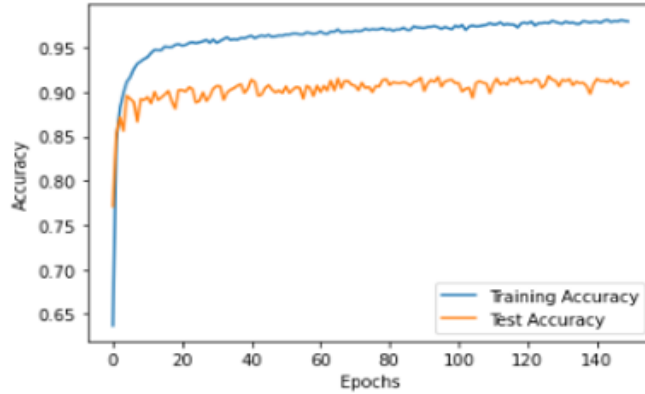


Figure 3.24: İvme Ölçer Sensörü İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları

## 3.5 Jiroskop ve İvme Ölçer Sensörlerine Ait Değişkenler İle Modelleme

### 3.5.1 K-En Yakın Komşuluk Modeli

Bu bölümde jiroskop ve ivme ölçer sensörlerinden elde edilen veriler kullanılmıştır. K-En yakın komşuluk modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.1.' te belirtilen değişkenler kullanılarak modelleme gerçekleştirilmiştir.

### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır. K-En yakın komşuluk modeli için en yüksek doğruluk oranı aşağıdaki parametreler ile bulunmuştur;

- ‘metric’: ‘manhattan’
- ‘n\_neighbors’: 9
- ‘weights’: ‘distance’

### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.14: Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan k en yakın komşular modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.87	0.98	0.92	496
1	0.88	0.92	0.90	471
2	0.98	0.78	0.87	420
3	0.96	0.82	0.89	491
4	0.86	0.97	0.91	531
5	1.00	1.00	1.00	537
Accuracy			0.92	2946
macro avg	0.92	0.91	0.91	2946
weighted avg	0.92	0.92	0.92	2946

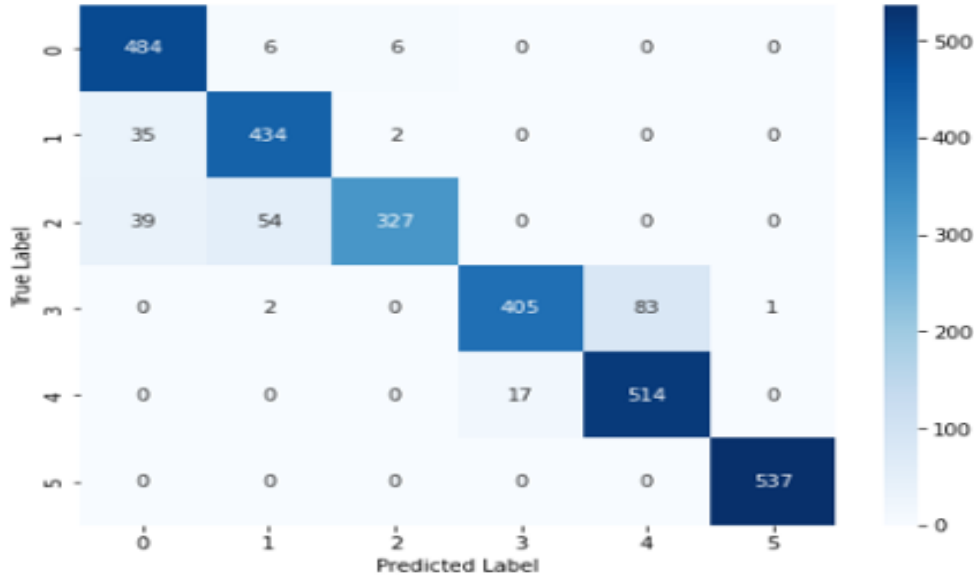


Figure 3.25: K-NN Modeli Karmaşıklık Matrisi

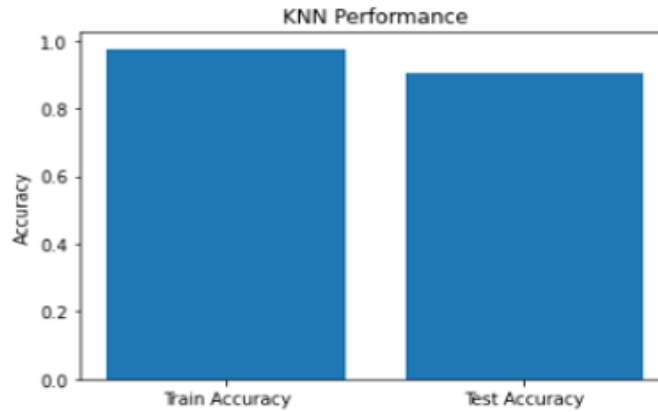


Figure 3.26: K-NN Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.5.2 Rastal Ormanlar Modeli

Bu bölümde jiroskop ve ivme ölçer sensörlerinden elde edilen veriler kullanılmıştır. Rastal ormanlar modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.1.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 20
- min\_samples\_leaf: 4
- min\_samples\_split: 10
- n\_estimators: 500

### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.15: Jiroskop ve İvmeölçer Sensörlerinden elde değişkenler ile kurulan rassal ormanlar modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.90	0.97	0.93	496
1	0.89	0.92	0.91	471
2	0.96	0.84	0.89	420
3	0.91	0.88	0.90	491
4	0.90	0.92	0.91	531
5	1.00	1.00	1.00	537
Accuracy			0.93	2946
macro avg	0.93	0.92	0.92	2946
weighted avg	0.93	0.93	0.93	2946

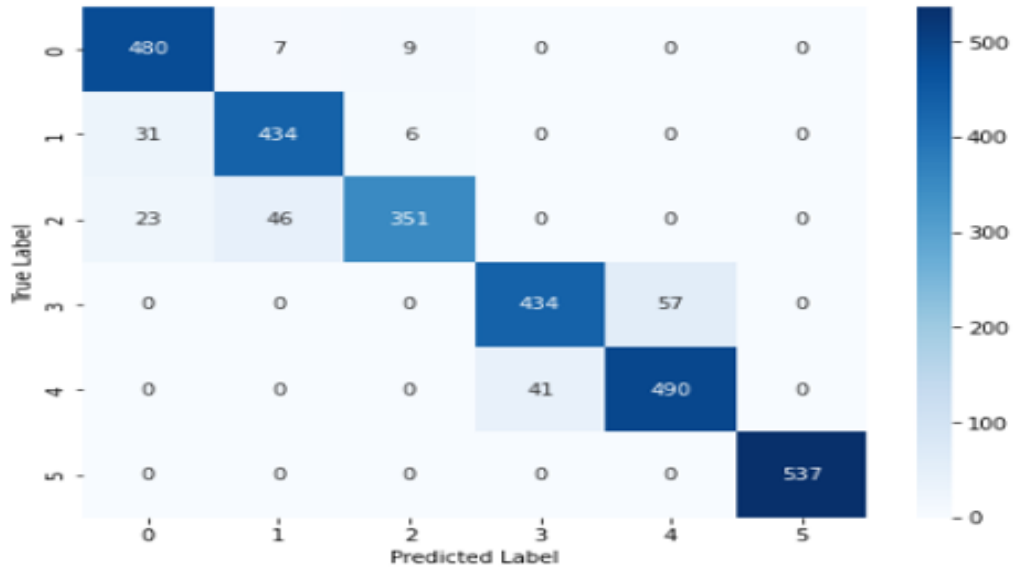


Figure 3.27: Rassal Ormanlar Modeli Karmaşıklık Matrisi

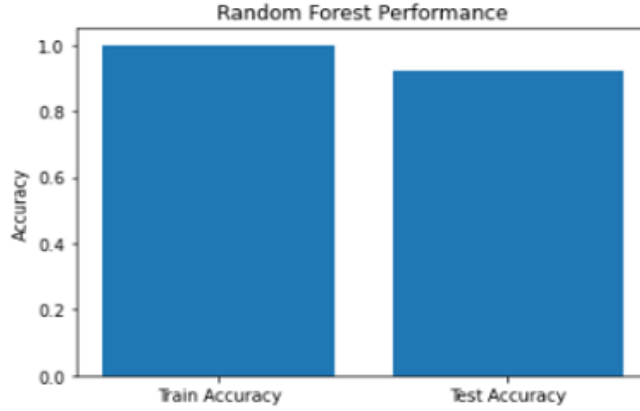


Figure 3.28: Rassal Ormanlar Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.5.3 eXtreme Gradient Boosting (XGBoost)

Bu bölümde jiroskop ve ivme ölçer sensörlerinden elde edilen veriler kullanılmıştır. XGBoost modeli kurulmuş ve çıktıları değerlendirilmiştir. 3.1.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

#### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- max\_depth: 3
- learning\_rate: 0.01
- n\_estimators: 100

#### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.16: Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan xgboost modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.85	0.90	0.88	496
1	0.85	0.76	0.80	471
2	0.88	0.88	0.88	420
3	0.88	0.82	0.85	491
4	0.80	0.92	0.85	531
5	1.00	0.97	0.99	537

	precision	recall	F1-score	support
Accuracy			0.87	2946
macro avg	0.88	0.87	0.87	2946
weighted avg	0.88	0.87	0.87	2946

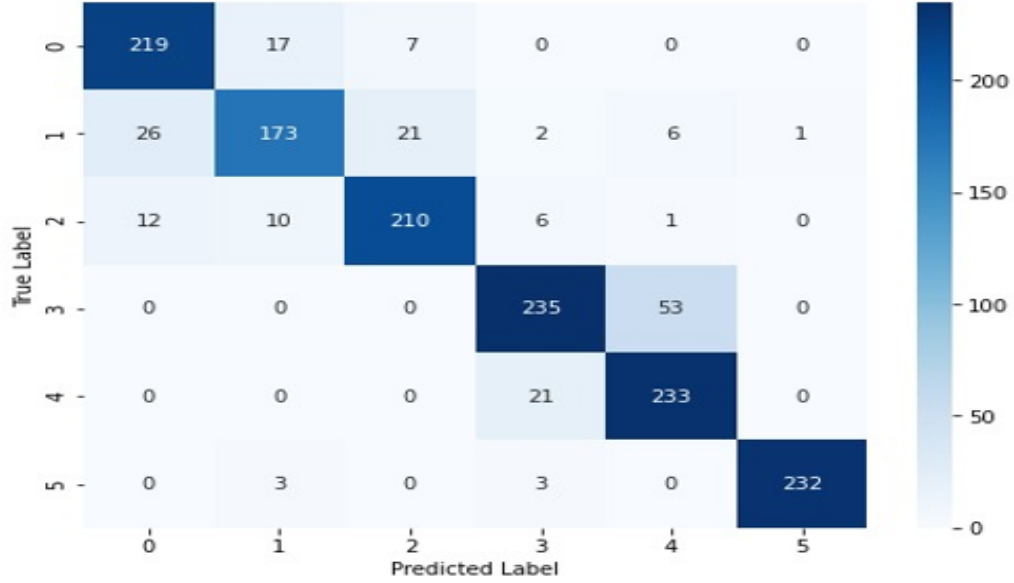


Figure 3.29: XGBoost Modeli Karmaşıklık Matrisi

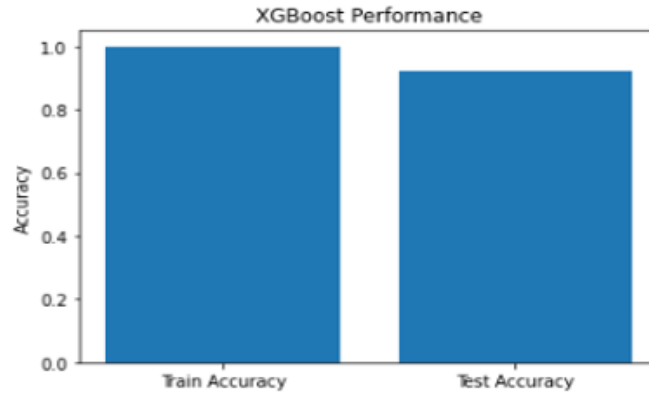


Figure 3.30: XGBoost Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.5.4 Destek Vektör Makinesi

Bu bölümde jiroskop ve ivme ölçer sensörlerinden elde edilen veriler kullanılmıştır. Destek Vektör Makinesi (SVM) modeli kurulmuş ve çıktıları değer-

lendirilmiştir 3.1.' te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

### Hiper Parametre Seçimi

Daha önce belirlenen parametre uzayını ve Scikit-Learn kütüphanesinde bulunan GridSearchCV algoritması ile en yüksek doğruluk oranı yakalanana kadar çalışması sağlanmıştır.

- C: 0.1
- gamma: 0.01
- kernel: linear

### En İyi Model

Bulunan parametrelerle kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.17: Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan destek vektör makinesi modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.85	0.87	0.86	496
1	0.86	0.75	0.80	471
2	0.83	0.85	0.84	420
3	0.86	0.82	0.84	491
4	0.78	0.87	0.82	531
5	0.96	0.97	0.97	537
Accuracy			0.85	2946
macro avg	0.86	0.85	0.85	2946
weighted avg	0.86	0.85	0.85	2946



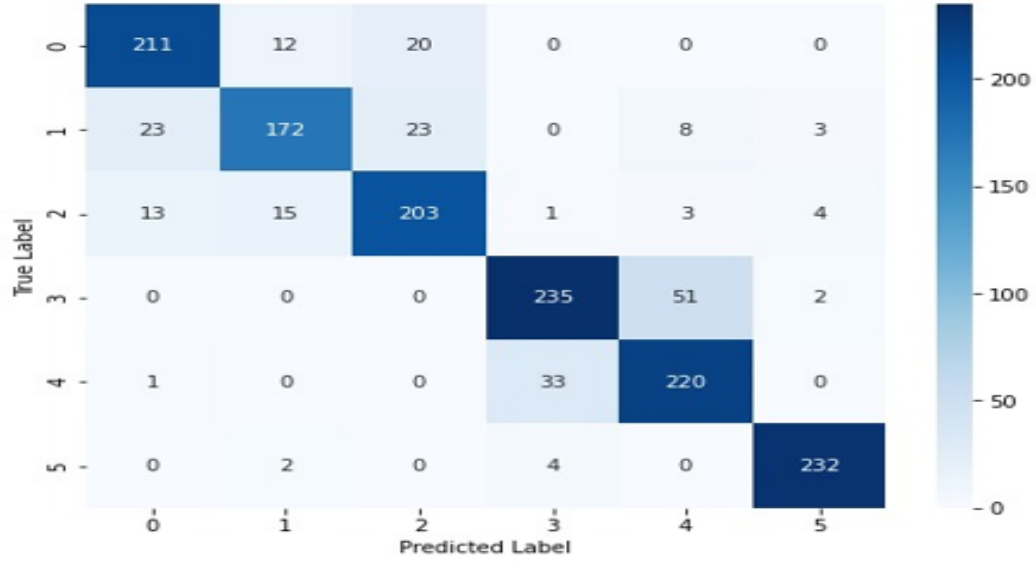


Figure 3.31: Destek Vektör Makinesi Modeli Karmaşıklık Matrisi

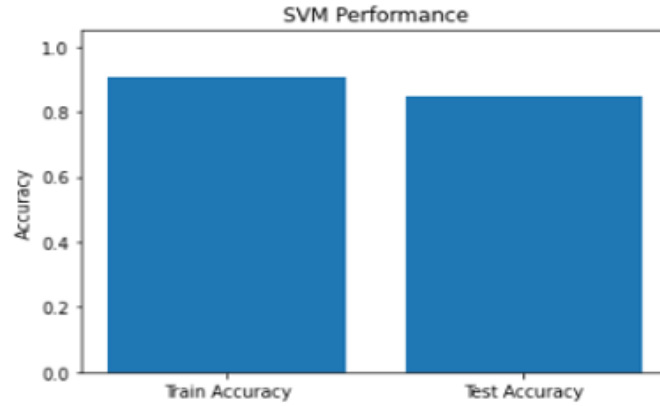


Figure 3.32: Destek Vektör Modelinin Eğitim ve Test Verilerinin Başarı Performansları

### 3.5.5 Yapay Sinir Ağları (Neural Networks)

Bu bölümde jiroskop ve ivme ölçer sensörleri üzerinde yapay sinir ağları modeli kullanılmış ve çıktıları değerlendirilmiştir. Tensorflow ve Keras kütüphanelerindeki Sequential() fonksiyonu kullanılmıştır.

#### Kullanılan Katmanlar

İlk katman, tamamen bağlı (Dense) katmandır. Bu katman 64 nörona sahiptir ve girdi boyutu, train veri kümesinin özellik sayısına eşittir. kernel\_initializer parametresi “normal” olarak ayarlanmıştır, yani rastgele normal bir dağılım kullanılarak başlangıç

ağırlıkları oluşturulacaktır. Bu katmanın aktivasyon fonksiyonu “sigmoid” olarak ayarlanmıştır. İkinci katman, bir Dropout katmanıdır. Dropout, ağıdaki aşırı uyumu azaltmak için kullanılan bir düzenleme tekniğidir. Bu katmanın Dropout oranı %20 olarak ayarlanmıştır. Üçüncü ve son katman, yine bir tamamen bağlı (Dense) katmandır. Bu katman 6 nörona sahiptir ve softmax aktivasyon fonksiyonu kullanılarak çoklu sınıflandırma işlemi gerçekleştirilir. `kernel_initializer` parametresi “normal” olarak ayarlanmıştır. Modelin derlenmesi, “adam” optimizasyon algoritması kullanılarak gerçekleştirilir. Kayıp fonksiyonu “`sparse_categorical_crossentropy`” olarak ayarlanmıştır, çünkü etiketler doğrudan kategorik olarak kodlanmamıştır. Ölçülen metrik, doğruluk (accuracy) olarak ayarlanmıştır. 3.1.’ te belirtilen değişkenler kullanılarak modelleme işlemi gerçekleştirilmiştir.

## En iyi Model

Kullanılan parametrelerle ve belirtilen epochs değeri ile kurulan modelin sınıflandırma metrikleri aşağıdaki gibidir.

Table 3.18: Jiroskop ve İvmeölçer Sensörlerinden elde edilen değişkenler ile kurulan yapay sinir ağları modelinin başarı sonuçları

	precision	recall	F1-score	support
0	0.93	1.00	0.96	496
1	0.95	0.92	0.93	471
2	0.99	0.94	0.96	420
3	0.95	0.91	0.93	491
4	0.92	0.95	0.94	531
5	1.00	1.00	1.00	537
Accuracy			0.95	2946
macro avg	0.96	0.95	0.95	2946
weighted avg	0.96	0.95	0.95	2946

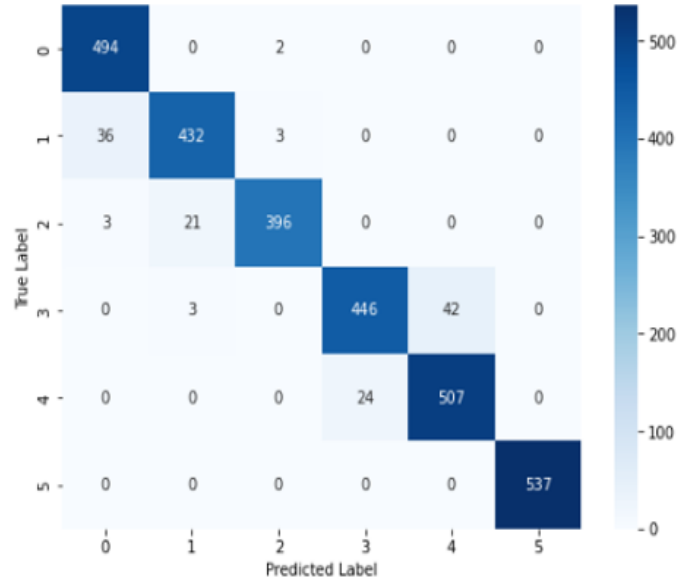


Figure 3.33: Yapay Sinir Ağları Modeli Karmaşıklık Matrisi

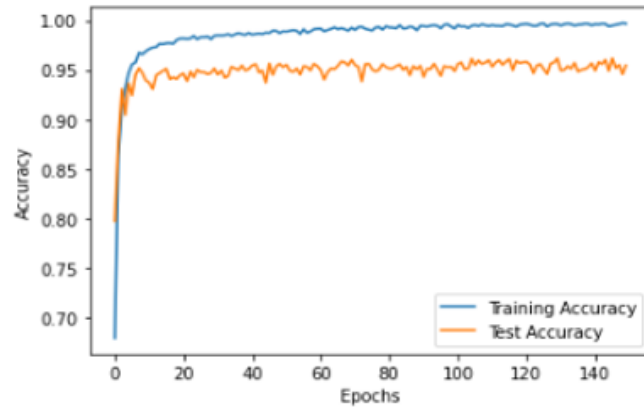


Figure 3.34: Jiroskop ve İvme Ölçer Sensörleri İçin Yapay Sinir Ağları Modelinin Epochs Değerlerine Göre Eğitim ve Test Verilerinin Başarı Performansları



# Chapter 4

## Sonuç

Bu çalışmada, insan aktivitelerini tanımak için makine öğrenmesi yöntemleri kullanarak modeller geliştirilmiştir. Farklı modeller arasında karşılaştırma yaparak en iyi sonucu elde etmek için çeşitli deneyler gerçekleştirilmiştir. Tablo 4.1'a göre, sadece jiroskop sensöründen elde edilen değişkenler ile geliştirilen modeller arasında en başarılı sonuç Rassal Ormanlar, Destek Vektör Makinesi ve Yapay Sinir Ağları modelleri ile elde edilmiştir. Sadece ivmeölçer sensöründen elde edilen değişkenler ile geliştirilen modeller arasında en başarılı sonuç K en yakın komşular ve Yapay sinir ağları modelleri ile elde edilmiştir. Her iki sensörle elde edilen değişkenler ile geliştirilen modeller arasında en başarılı sonuç Yapay Sinir Ağları Modeli ile elde edilmiştir Yapay sinir ağları modeli, karmaşık ilişkileri yakalama yeteneğiyle öne çıkmıştır. Derin öğrenme tabanlı modelin eğitimi için geniş bir veri kümesi kullanılmıştır ve ardından modeli optimize etmek için çeşitli hiperparametre ayarlamaları yapılmıştır. Bu ayarlamaları GridSearchCV algoritması yardımı ile yapılmıştır. Modeli eğitirken, aşırı öğrenmeyi önlemek için Droupout tekniği uygulanmıştır. Sonuçlar, yapay sinir ağları modelinin, yüksek bir doğruluk oranı ve genel performans ile insan aktivitelerini tanımada diğer modellere göre daha başarılı olduğunu göstermiştir. Bu model, farklı aktiviteler arasında hassas bir şekilde ayırım yapabilmış ve gerçek zamanlı uygulamalarda kullanıma uygun olduğu düşünülmüştür. Bu proje, insan aktivitelerini otomatik olarak tanımlayarak sağlık izleme sistemleri, yaşlı bakımı, spor performans analizi ve güvenlik uygulamaları gibi birçok alanda kullanılabilir. Yapay sinir ağları tabanlı modelimizin geliştirilmesi, insan aktivitelerini anlama ve analiz etme konusunda yeni fırsatlar sunmaktadır.

Table 4.1: Yöntemlere ve Sensörlere göre F1 skorları

Yöntemler	Jiroskop	İvmeölçer	Jiroskop ve İvmeölçer
KNN	0.82	0.91	0.91
Rassal Ormanlar	0.91	0.88	0.92
XGBoost	0.89	0.89	0.87
Destek Vektör Makinesi	0.91	0.90	0.85
Yapay Sinir Ağları	0.91	0.91	0.95



# Kaynaklar

- 10 Ağyar, Z. (2015). Yapay sinir ağlarının kullanım alanları ve bir uygulama. *Mühendis ve Makine*, 56(662), 22-23.
- Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., ve diğerleri. (2013). A public domain dataset for human activity recognition using smartphones. *Esann* içinde (C. 3, s. 3).
- Asilkan, Ö. ve Irmak, A. G. S. (2009). İkinci el otomobillerin gelecekteki fiyatlarının yapay sinir ağları ile tahmin edilmesi. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 14(2), 375-391.
- Bell, K., Detienne, D. H. ve Joshi, S. A. (2003). Neural networks as statistical tools for business researchers. *Organizational Research Methods*, 6(2), 236-265.
- Bengio, Y., Courville, A. ve Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.
- Biau, G. ve Scornet, E. (2016). A random forest guided tour. *Test*, 25, 197-227.
- Buduma, N. (2017). Fundamentals of Deep Learning, O'Reilly Media. Inc.
- Chen, L., Hoey, J., Nugent, C. D., Cook, D. J. ve Yu, Z. (2012). Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 790-808.
- Dandıl, E. ve Polattimur, R. (2020). Dog behavior recognition and tracking based on faster R-CNN. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35(2), 819-834.
- Dolgun, M. Ö., Özdemir, T. G. ve Oğuz, D. (2009). Veri madenciliğiâ nde yapısal olmayan verinin analizi: Metin ve web madenciliği. *İstatistikçiler Dergisi: İstatistik ve Aktüerya*, 2(2), 48-58.
- Du, Y., Fu, Y. ve Wang, L. (2015). Skeleton based action recognition with convolutional neural network. *2015 3rd IAPR Asian conference on pattern recognition (ACPR)* içinde (ss. 579-583). IEEE.
- Eşref, Y. (2019). Türkçe Dizi Etiketleme İçin Sinir Ağ Modelleri.
- Fan, L., Wang, Z. ve Wang, H. (2013). Human activity recognition model based on decision tree. *2013 International Conference on Advanced Cloud and Big Data* içinde (ss. 64-68). IEEE.
- Fırat, M. ve Güngör, M. (2004). Askı madde konsantrasyonu ve miktarının yapay sinir ağları ile belirlenmesi. *Teknik Dergi*, 15(73).
- Hanbay, K. (2020). Hyperspectral image classification using convolutional neural network and two-dimensional complex Gabor transform. *Journal of the Faculty of*

*Engineering and Architecture of Gazi University*, 35(1), 443-456.

Isık, G. ve Artuner, H. (2020). Turkish dialect recognition in terms of prosodic by long short-term memory neural networks. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 35(1).

Iskanderov, J. ve Güvensan, M. A. (2019). Akıllı telefon ve giyilebilir cihazlarla aktivite tanıma: Klasik yaklaşımlar, yeni çözümler. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 25(2), 223-239.

Kelle, A. C. ve Hüseyin, Y. (2022). MQTT Trafiğinde DoS Saldırıların Makine Öğrenmesi ile Sınıflandırılması ve Modelin SHAP ile Yorumlanması. *Journal of Materials and Mechatronics: A*, 3(1), 50-62.

Khan, A. M., Lee, Y.-K. ve Kim, T.-S. (2008). Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets. *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* içinde (ss. 5172-5175). IEEE.

Kuncan, F. ve Kaya, M., Yılmaz. (2019). New approaches based on local binary patterns for gender identification from sensor signals. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34(4), 2173-2185.

Kwapisz, J. R., Weiss, G. M. ve Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2), 74-82.

Lantz, B. (2019). *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd.

LeCun, Y., Bottou, L., Bengio, Y. ve Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Lim, T. W. (2019). The Internet of Things (IoT). *Industrial Revolution 4.0, Tech Giants, and Digitized Societies* içinde (ss. 33-49). Springer.

Maltarollo, V. G., Honório, K. M. ve Silva, A. B. F. da. (2013). Applications of artificial neural networks in chemical problems. *Artificial neural networks-architectures and applications*, 203-223.

Mandic, D. ve Chambers, J. (2001). *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley.

Metin, İ. A. ve Karasulu, B. (2021a). İnsanın günlük aktivitelerinin yeni bir veri kümesi: Derin öğrenme tekniklerini kullanarak sınıflandırma performansı için kıyaslama sonuçları. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 36(2), 759-778.

Metin, İ. A. ve Karasulu, B. (2021b). A novel dataset of human daily activities: Its benchmarking results for classification performance via using deep learning techniques. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 36(2), 759-777.

Metin, İ. A. ve Karasulu, B. (2019). İnsan aktivitelerinin sınıflandırılmasında tekrarlayan sinir ağı kullanan derin öğrenme tabanlı yaklaşım. *Veri Bilimi*, 2(2), 1-10.

Öztürk, K. ve Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, 6(2), 25-36.

Paul, P. ve George, T. (2015). An effective approach for human activity recognition on smartphone. *2015 IEEE International Conference on Engineering and Tech-*



nology (ICETECH) içinde (ss. 1-3). IEEE.

Riboni, D. ve Bettini, C. (2011). COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3), 271-289.

Sağbaş, E. A. ve Ballı, S. (2017). Akıllı saat algılayıcıları ile insan hareketlerinin sınıflandırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 21(3), 980-990.

Sağbaş, E. A., Korukoğlu, S. ve Ballı, S. (2021). AKILLI TELEFON VERİLERİ VE MAKİNE "ÖĞRENME" YÖNTEMLERİ KULLANILARAK STRES TESTİ ÇALIŞMALARI ÜZERİNE BİR LİTERATÜR ARAŞTIRMASI. *Mühendislik Bilimleri ve Tasarım Dergisi*, 9(3), 1030-1038.

Schuster, M. ve Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11), 2673-2681.

Tebelskis, J. M. (1995). *Speech recognition using neural networks*. Carnegie Mellon University.

Ustev, Y. E., Durmaz Incel, O. ve Ersoy, C. (2013). User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* içinde (ss. 1427-1436).

Yang, J.-Y., Wang, J.-S. ve Chen, Y.-P. (2008). Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern recognition letters*, 29(16), 2213-2220.

Yang, Z., Salakhutdinov, R. ve Cohen, W. (2016). Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Yangın, G. (2019). *XGboost ve Karar Ağacı tabanlı algoritmaların diyabet veri setleri üzerine uygulaması*. (Yayımlanmamış mathesis). Mimar Sinan Güzel Sanatlar Üniversitesi.