

---

# Makine Öğrenme Algoritmalarını Kullanarak İnsan Aktivitesi Tanıma

Hanife ŞAHİN

---

## Anahtar Kelimeler

İnsan aktivitesi tanıma,  
Makine öğrenimi,  
Nesnelerin interneti

**Özet:** Nesnelerin interneti hayatımızda birçok kolaylık sağlamaktadır. Buna hastalık teşhisi, akıllı ev projeleri örnek olarak verilebilir. Nesnelerin interneti ile hayatımıza giren giyilebilir cihazlar insan aktivitelerini tanımak için veri toplama işlemini kolaylaştırmıştır. Çalışmada UCI veri deposundan alınan "akıllı telefon kullanarak insan aktivitesini tanıma" veri kümesi kullanılmıştır. 6 insan aktivitesini(yürüme, merdiven çıkma, merdiven inme, oturma, ayakta durma, uzanma) sınıflandırmak için makine öğrenimi algoritmaları olan karar ağaçları, naive bayes, rastgele orman, destek vektör makinesi, lojistik regresyon ve yapay sinir ağları(YSA) kullanılmıştır. Eğitim verisi üzerinde %97,5 sınıflandırma doğruluğu ile YSA diğer algoritmalara göre daha iyi performans göstermiştir. Test veri kümesi için YSA %97,9 sınıflandırma doğruluğu göstermiştir.

---

## Human Activity Recognition Using Machine Learning Algorithms

---

## Keywords

Human activity recognition,  
Machine learning,  
Internet of things

**Abstract:** The Internet of Things provides many conveniences in our lives. Disease diagnosis, smart home projects can be given as examples. Wearable devices that have entered our lives with the Internet of Things have facilitated the data collection process to recognize human activities. In the study, the dataset "recognition of human activity using a smartphone" from the UCI data store was used. Machine learning algorithms, decision trees, naive bayes, random forest, support vector machine, logistic regression and artificial neural networks (ANN) were used to classify 6 human activities (walking, walking upstairs, walking downstairs, sitting, standing, laying). ANN performed better than other algorithms with 97.5% classification accuracy on the training data. ANN showed 97.9% classification accuracy for the test dataset.

---

## 1. Giriş

Akıllı telefon ilk defa 1994 yılında dünya literatürüne girmiş, zaman içinde gelişerek şu an dünyada neredeyse herkesin kullandığı bir mobil iletişim cihazı haline gelmiştir. Teknolojinin gelişmesiyle hayatımıza 1999 yılında giren ve son zamanlarda popülerleşen nesnelerin interneti kavramı ise fiziksel nesnelerin birbirleriyle bağlantıda olduğu bir iletişim şeklidir. Akıllı telefonlar, akıllı saatler ve akıllı kameralar nesnelerin internetine örnek olarak verilebilir. Bu tür nesneler ivmeölçer, jiroskop gibi çeşitli algılayıcıları bulundurması ile hayatımıza birçok kolaylık sağlamaktadır. Sağlık ve spor gibi birçok alanda kullanılmakta ve hastalık teşhisi, akıllı ev projeleri gibi günümüzde birçok uygulaması bulunmaktadır.

Aktivite tanıma insanların yürüme, koşma, uzanma gibi birçok aktivitesini algılayıcılar yardımı ile tanımlama işlemidir. Aktivite tanıma çalışmaları yıllar önce hayatımıza girmiş olmasına rağmen nesnelerin interneti ile önemli bir konu haline gelmiştir.

Nesnelerin interneti ile akıllı nesnelerin bazılarının giyilebilir olması veri toplama aşamasında önemlidir.

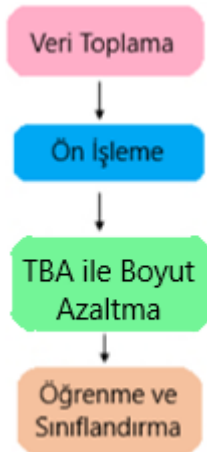
Aktivite tanıma çalışmalarında giyilebilir cihazlarda(akıllı bileklik, akıllı saat) bulunan algılayıcılar yardımı ile veriler daha kolay elde edilmektedir. Gerekli ön işleme aşamalarından sonra çeşitli makine öğrenme ve derin öğrenme algoritmaları ile aktivite tanıma işlemi gerçekleştirilmektedir.

Literatürde bu alanda yapılan birçok çalışma bulunmaktadır. Makine öğrenme algoritmaları (rastgele orman, destek vektör makinesi, k-en yakın komşu gibi) ile yapılan çalışmalarda genelde %90'ın üzerinde doğruluk oranına ulaşılmıştır [1], [2], [3], [4], [5], [6], [7]. Derin öğrenme algoritmaları(tekerrarlayan sinir ağı, evrişimli sinir ağı, uzun kısa süreli bellek gibi) kullanarak yapılan çalışmalarda da bu durum değişmemiş ve yüksek doğruluk oranları elde edilmiştir [8], [9], [10], [11], [12].

Çalışmada makine öğrenimi algoritmaları kullanılarak insan aktiviteleri sınıflandırılmaya çalışılmıştır. Kullanılan veri ve yöntemler ikinci bölümde açıklanmıştır. Üçüncü bölümde uygulama sonuçlarına yer verilmiştir. Dördüncü bölümde ise sonuçlar tartışılmıştır.

## 2. Materyal ve Metot

Çalışmada kullanılan veri "A Public Domain Dataset for Human Activity Recognition using Smartphones" başlıklı makale ile 2013 yılında ESANN etkinliğinde sunulmuştur[13]. UCI veri deposundan alınmıştır[14]. Verinin elde edilebilmesi için deneylerde 19-48 yaş aralığındaki 30 gönüllü kişinin beline bir akıllı telefon takarak(Samsung Galaxy S II) 6 aktivite(Yürüme, merdiven çıkma, merdiven inme, oturma, ayakta durma, uzanma) gerçekleştirmeleri sağlanmıştır. Akıllı telefonda bulunan ivmeölçer ve jiroskop algılayıcıları ile 3 eksenli doğrusal ivme ve 3 eksenli açısal hız sinyalleri elde edilmiştir. Daha sonra bu sinyallere gürültü filtreleri uygulanarak ön işleme yapılmıştır ve ardından 2.56 saniyelik sabit genişlikli pencerelerde ve %50 örtüşme ile örneklenmiştir. Her pencereden zaman(ortalama, standart sapma, varyans, minimum, maksimum, entropi gibi) ve frekans(ortalama ve medyan gibi.) alanından değişkenler hesaplanarak 561 öznitelik elde edilmiştir. Bu öznitelikler  $[-1,1]$  aralığında normleştirilmiştir. Deney sonucunda elde edilen 10299 veri %70 eğitim ve %30 test verisi olmak üzere ikiye ayrılmıştır.



Şekil 1. Uygulama adımlarının şeması

Çalışmada veriye uygulanan adımların kısa özeti Şekil 1'de gösterilmektedir. Analiz aşamasında kullanılan algoritmalar hakkında bilgiler aşağıda yer almaktadır.

### 2.1. Temel Bileşen Analizi(TBA)

TBA, çok boyutlu veriyi bilgi kaybı olmayacak şekilde az boyutlu veriye indüsmünü, varyansı maksimize edecek şekilde bulan doğrusal bir analizdir [15]. İlişki yapısını yok etmek veya boyut indirgemek için kullanılmaktadır. Birbirleriyle ilişkili değişkenler TBA sonucunda birbirleriyle ilişkisi olmayan bileşenler ile ifade edilmektedir.

Veri  $n$  gözlem ve  $p$  değişken ile  $X_{pxn}$  matrisi şeklinde ifade edilsin.

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix} \quad (1)$$

Çok değişkenli verilerde ölçü birimleri farklı olduğu için veri standartlaştırılmalıdır. Standartlaştırılmış veri  $Z_{pxn}$  ile ifade edilsin. Veri standartlaştırıldığı için  $R$  ilişki matrisi ele alınmalıdır. Ham veri üzerinden ilerlenirse kovaryans matrisi ele alınmalıdır.  $R$  ilişki matrisi üzerinden özdeğerler ve özvektörler cebirsel işlemler ile hesaplanır.

$$Y_{pxn} = T'_{pxp} Z_{pxn} \quad (2)$$

Eşitlik 2'de ifade edilen  $T'_{pxp}$  özvektörler matrisini,  $Y_{pxn}$  ise dönüştürülmüş veriyi ifade etmektedir.

### 2.2. Karar Ağaçları

Verileri, veri noktaları arasındaki en önemli fark yaratan öğeyi temel alarak ağaç dalları şeklinde oluşturan hem sınıflandırma hem de regresyon yöntemidir. Kullanımı ve yorumlanması kolay olduğu için yaygın kullanılan bir yöntemdir.

Ağaç yapısında en üst yapı kök, en son yapı yaprak ve bunların arasında kalan yapılar dal olarak adlandırılmaktadır. Ağaç yapısı oluşturulurken gini ya da entropi kullanılmaktadır. Entropi daha dengeli bir ağaç çıkarmaya meyilli iken gini frekans fazla olan sınıfı ayırtmaya meyillidir [16].

Karar ağaçlarını oluşturmak için birçok algoritma geliştirilmiştir. ID3, C4.5, C5.0, Sınıflandırma ve Regresyon Ağaçları(CART), CHAID ve QUEST örnek olarak verilebilir.

### 2.3. Naive Bayes

Değişkenler arasındaki bağımsızlık varsayımı ile Bayes teoremine dayanır ve ilgili bir olayın oluşumunu temel olarak bir olayın oluşma olasılığını hesaplar.

$$P(A \setminus B) = \frac{P(B \setminus A)P(A)}{P(B)} \quad (3)$$

Algoritma bir eleman için her durumun olasılığını hesaplar ve olasılık değeri en yüksek olana göre sınıflandırma işlemi gerçekleştirir [17]. Modeli oluşturmak kolaydır ve büyük veri kümeleri için kullanışlıdır.

Test kümesinde, eğitim kümesinde gözlenmeyen bir durum ile karşılaşıldığında olasılık değeri olarak 0 verir. Bu durum genellik sıfır frekans olarak bilinen bir durumdur. Bu durum yumuşatma teknikleri ile çözülebilir [17].

Naive bayes, normal(gaussian) naive bayes, bernoulli naive bayes ve çok değişkenli(multinomial) naive bayes olmak üzere üç çeşittir.

## 2.4. Rastgele Orman(RO)

Karar ağaçlarını temel alan rastgele orman, bir ağaç oluşturmak yerine bir orman oluşturup bu ormandaki ağaçları rastgele düzenleyen bir topluluk öğrenmesi yöntemidir. Ağaç oluşturmak için CART algoritmasını kullanmaktadır.

Tüm değişkenler arasından en iyi dalı kullanarak her düğümü dallara ayırmak yerine, rastgele orman her düğümde rastgele seçilen değişkenlerin en iyisini kullanarak her düğümü dallara ayırır. Her yeni eğitim verisi, orijinal eğitim verisinden önyüklemeye yöntemiyle üretilir. Ağaçlar daha sonra rastgele özellik seçimi kullanılarak büyütülür. Gelişmiş ağaçlar budanmaz [18].

Rastgele orman oluşturulacak ağaç sayısı(n) ve her düğümde seçilecek değişken sayısı(m) olmak üzere iki parametreye sahiptir. Az parametreye sahip olması ve yöntemin parametre ayarına çok duyarlı olmaması nedeniyle iyi sonuçlar vermektedir [19].

Her ağaçtaki bölme işleminde, dallanma gerçekleşmeden önce Gini indeksi hesaplanır. Bölmeden önceki Gini değeri ile bölmeden sonraki Gini değeri arasındaki fark bulunur ve tüm ağaçlar oluşturulduktan sonra farklar toplanır. Bulunan değer, değişkenin önem derecesini verir. Bu işlem tüm değişkenler için hesaplanır [20].

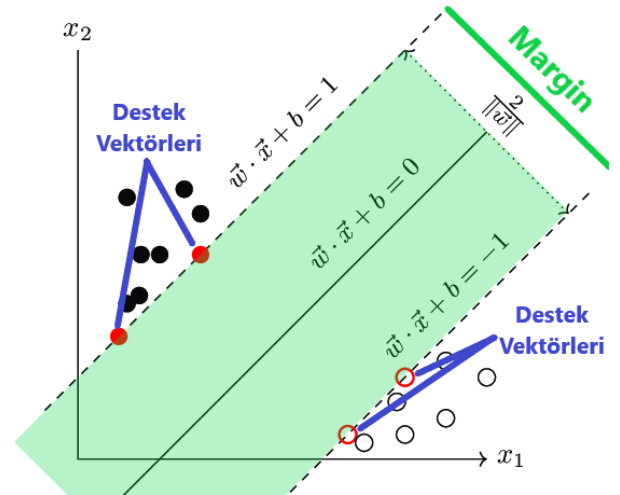
Hem sınıflandırma hem de regresyon için kullanılmaktadır. Sınıflandırma için kullanıldığında, her ağaçtan bir sınıf oyu alır ve ardından çoğunluk oyu kullanarak karar verir. Regresyon için kullanıldığında ise, her ağaçtan gelen tahminlerin ortalamasını almaktadır [19].

Rastgele orman çok hızlı, aşırı uyuma dayanıklı ve istediği kadar çok ağaçla çalışabilen bir yöntemdir. Karar ağaçlarında görülen ağaç yapısı rastgele orman da görülememektedir. Kara kutu olarak adlandırılan bu durum rastgele ormanın bir dezavantajıdır [18].

## 2.5. Destek Vektör Makinesi(DVM)

Her bir sınıfı diğer sınıflardan ayırmak için doğru ya da düzlem kullanarak sınıflandırma işlemini gerçekleştiren denetimli bir öğrenme algoritmasıdır.

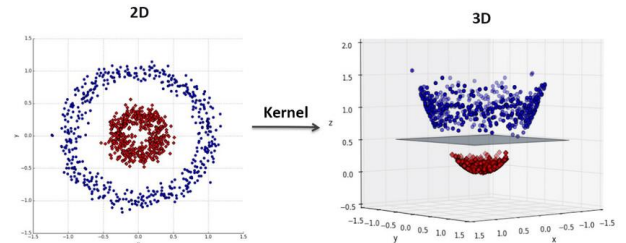
DVM, veri setinin doğrusal olarak ayrılabilir ve ayrılabilir durumuna göre ikiye ayrılır.



Şekil 2. Doğrusal durum örneği [21]

Şekil 2'de görülen DVM örneğinde siyah ve beyaz olarak ifade edilen iki sınıfı ayırmak için bir doğru çizilmiştir. Çizilen bu doğru her iki sınıfın noktaları için de maksimum uzaklıkta olacak şekilde belirlenmiştir. Sınıflara en yakın olarak çizilen doğrulara sınır çizgisi denilmektedir. Bu sınır çizgisine en yakın olan veri noktaları ise destek vektör olarak adlandırılmaktadır.

Doğrusal olmayan durumlarda doğrusal duruma dönüştürmek için çekirdek numarası(Kernel trick) olarak bilinen yöntem kullanılır [22].



Şekil 3. Doğrusal olmayan durumda çekirdek numarası örneği [21]

Şekil 3'te doğrusal olarak ayrılamayan bir veri kümesinin çekirdek numarası kullanılarak bir düzlem yardımı ile ayrılması gösterilmektedir.

En çok kullanılan çekirdek yöntemleri polinom çekirdeği(polynomial kernel) ve normal radyal tabanlı fonksiyon çekirdeği(Gaussian Radial Basis Function Kernel)'dir.

## 2.6. Lojistik Regresyon(LR)

Doğrusal regresyon modeli bağımlı değişken olarak sürekli değişkenleri alır, lojistik regresyon ise bağımlı değişken olarak kategorik değişkenlerin olduğu durumda kullanılan bir yöntemdir. Bu yöntem bir regresyon yöntemi değil, sınıflandırma yöntemidir. Bağımlı değişkenin iki veya daha fazla kategori içerdiği, bağımsız değişkenlerin ise sürekli veya kategorik bir yapıya sahip olduğu durumlarda

bağımsız değişkenler ile bağımlı değişkenler arasındaki ilişkiyi araştırmaktadır.

Lojistik regresyon analizi bağımlı değişken iki kategoriye sahipse ikili lojistik regresyon, ikiden fazla kategoriye sahipse çoklu lojistik regresyon ve son olarak ikiden fazla kategoriye sahip ve sıralı bir yapı söz konusu ise sıralı(ordinal) lojistik regresyon adını almaktadır.

Lojistik fonksiyonun amacı bağımsız değişkenin bir kategoriye dahil olma olasılığını ortaya koymaktır. Dolayısıyla 0 ile 1 arasında değer üretir [23].

$$p(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}} \quad (4)$$

Lojistik regresyon modelinde katsayıların yorumlanmasında odds oranı kullanılır [24]. Bir olayın olma oranının olmama oranına bölümü odds değeri ile ifade edilmektedir. İki farklı odds değerinin birbirine bölünmesi ise odds oranı olarak bilinmektedir.

$$Odds = \frac{p}{1-p} \quad (5)$$

Doğrusal regresyonda bağımsız ve bağımlı değişken arasında doğrusal bir ilişki olduğu varsayımı yerine lojistik regresyonda bağımsız ve bağımlı değişkenin logaritmik değeri arasında doğrusal bir ilişki olduğu varsayılmaktadır. Hatalar bağımsız olmalıdır. Çok bağlantı sorunu olmamalıdır. Kayıp veri olmamalıdır.

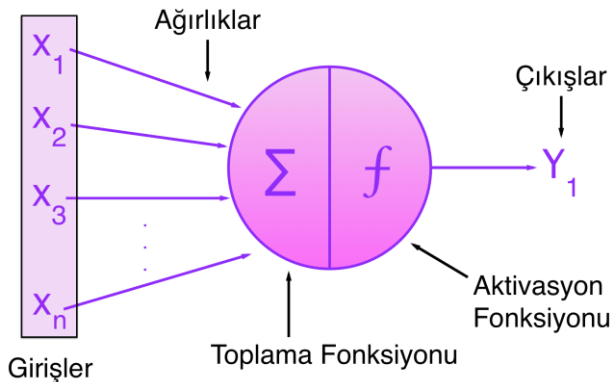
## 2.7. Yapay Sinir Ağları(YSA)

Canlı organizmalardaki biyolojik sinir ağlarını taklit eden veriye dayalı bir yöntemdir.

**Tablo 1.** İstatistik ve YSA

İstatistik	YSA
Parametre	Ağırlık
Bağımsız Değişken	Girdi
Artık	Hata
Tahmin Değeri	Çıktı

Tablo 1’de de görüldüğü üzere herhangi bir istatistiksel model ile YSA aynıdır. Sadece terminolojileri farklıdır.



**Şekil 4.** Yapay sinir ağı hücresi [25]

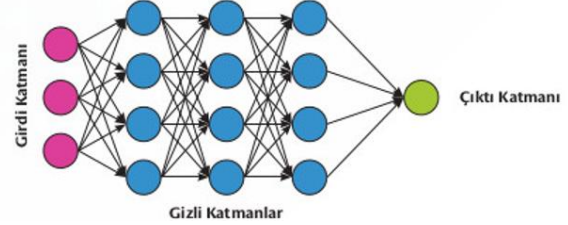
Girişler, ağı öğrenmesi istenen bilgilerdir. Dış ortamdan alınabildiği gibi başka hücrelerden de alınabilir. Verideki değişkenler olarak düşünebiliriz. Ağırlıklar, hücreye gelen girdilerin etkisini belirler. Toplama fonksiyonu net giriş bilgisinin hesaplanmasını sağlamaktadır [25]. En yaygın olan toplama fonksiyonu eşitlik 6 ile gösterilen hücreye gelen girdiler ile bu girdilerin ağırlıklarının çarpımının toplanmasıdır.

$$Net = \sum_{i=1}^n W_i X_i \quad (6)$$

Aktivasyon fonksiyonu hücreye gelen net bilgiyi analiz ederek hücrenin bu giriş bilgisine yönelik bir çıkış bilgisinin üretilmesini sağlamaktadır.

$$f(net_i) = Y_i \quad (7)$$

Yapay sinir ağlarının aktivasyon fonksiyonu, mimari yapı ve öğrenme algoritması olmak üzere üç temel bileşeni vardır. Aktivasyon fonksiyonu verideki eğrisel olup olmama durumunu algılayan bir fonksiyondur. Aktivasyon fonksiyonu tek kutuplu ya da çift kutuplu olabilir. Mimari yapının ileri beslemeli, doğrudan bağlantı içeren mimari, geri beslemeli gibi birçok çeşidi bulunmaktadır. Öğrenme algoritması ise ağırlıkların en iyi şekilde bulunma işlemidir. Bu işleme yapay sinir ağı eğitimi denir.



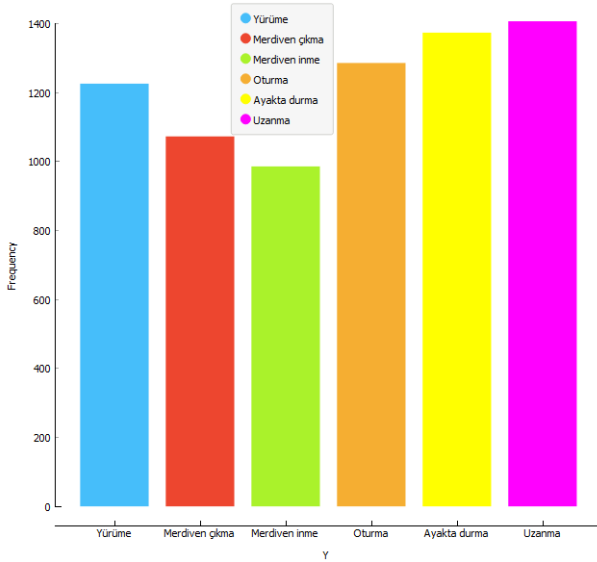
**Şekil 5.** Yapay sinir ağları mimarisi [26]

Gizli katman sayısı birden fazla olduğu durumda çok katmanlı sinir ağı olarak adlandırılmaktadır.

## 3. Bulgular

Uygulama veri bilimi platformu olan Anaconda aracılığı ile Orange 3 kullanılarak gerçekleştirilmiştir.

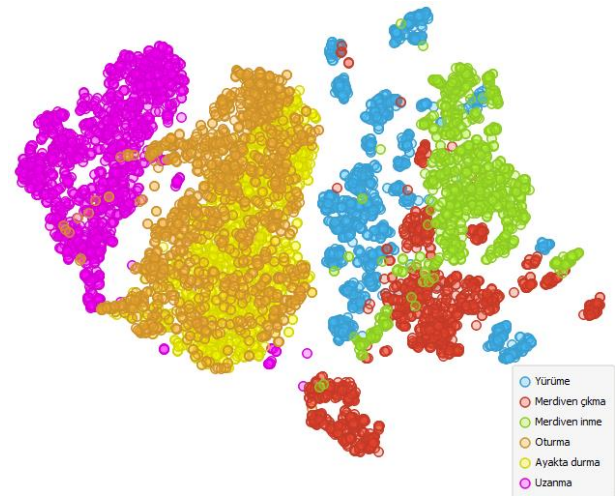
Keşifsel veri analizi aşamasında herhangi bir kayıp veriye rastlanmamaktadır. Aktivite sınıflarının dengesiz olmadığı Şekil 6’da görülmektedir.



**Şekil 6.** Aktivite sınıflarının sıklığı

Doğrusal olmayan, denetimsiz bir boyut azaltma yöntemi olan t-dağıtılmış stokastik komşu yerleştirme (t-SNE) yöntemi verileri görselleştirmek için kullanılır. Bu işlemi çok boyutlu verilerde noktalar arasındaki uzaklıkları olabildiğince koruyarak daha düşük boyutlu bir uzaya eşleyerek yapmaktadır [27].

Kullanılan yöntemin bir kümeleme yöntemi olmadığı unutulmamalıdır. Bu yöntem orijinal uzayda verilerin nasıl düzenlendiğine dair bir fikir vermeye çalışmaktadır [27]. Bu yöntemin verimize uygulanmış hali Şekil 7’de görülmektedir.



**Şekil 7.** Aktivite sınıflarının t-SNE ile görselleştirilmesi

Oturma ve ayakta durma verilerinin üst üste bindiği görülmektedir. Sınıflandırma aşamasında bu iki sınıfı ayırt etmekte model zorlanabilir.

Verideki değişkenler arasındaki ilişkiyi yok etmek ve boyutu azaltmak için veriye TBA uygulanmıştır. TBA analizi sonucunda 100 temel bileşen ile açıklanan varyans %95’tir. 561 öznelilik yerine birbirinden bağımsız 100 temel bileşen ile model kurulur.

Veriye karar ağaçları, naive bayes, rastgele orman(RO), destek vektör makinesi(DVM), lojistik regresyon(LR) ve yapay sinir ağları(YSA) uygulanmıştır. Uygulama aşamasında 10-katlı çapraz doğrulama kullanılmıştır. Sonuçlar Tablo 2’de verilmiştir.

**Tablo 2.** Modellerin performans sonuçları

Model	AUC	Sınıflandırma Doğruluğu	F1	Kesinlik (Precision)	Duyarlılık (Recall)
Karar Ağacı	%92,1	%85,3	%85,3	%85,3	%85,3
Naive Bayes	%98,1	%85,9	%86	%86,5	%85,9
RO	%98,8	%88,7	%88,7	%88,7	%88,7
DVM	%99,6	%94,6	%94,5	%94,7	%94,6
LR	%99,8	%97	%97	%97	%97
YSA	<b>%99,9</b>	<b>%97,5</b>	<b>%97,5</b>	<b>%97,5</b>	<b>%97,5</b>

Tablo 2’ye bakıldığında en yüksek sınıflandırma doğruluğu %97,5 ile YSA olduğu görülmektedir. Diğer performans değerlendirme yöntemleri(AUC, F1, kesinlik, duyarlılık) de destekler niteliktedir.

Her aktivite sınıfı için modellerin sınıflandırma doğruluğu Tablo 3’te verilmiştir. Her model için karışıklık matrisi ekler kısmında yer almaktadır.

**Tablo 3.** Modellerin aktivite sınıflarına göre doğru sınıflandırma oranları

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Karar Ağacı	%89,9	%86,2	%83,5	%76	%78,9	%96,4
Naive Bayes	%86,7	%86,6	%88,1	%76,6	%85	%92,3
RO	%94,6	%87,3	%88	%78,5	%84,8	%98,2
DVM	%99,3	%99,3	%99,5	%80,6	%92,1	%98,4
LR	%99,9	%98,4	%99,4	%92,2	%93	%99,9

YSA	%99,8	%99,1	%99,2	%93,9	%94,3	%99,7
-----	-------	-------	-------	-------	-------	-------

Tablo 3'e bakıldığında oturma ve ayakta durma aktivitelerinin doğru sınıflandırma oranlarının %76'ya kadar düştüğü görülmektedir. Keşifsel veri analizi yaparken fark ettiğimiz gibi bu iki sınıfı ayırmakta bazı modeller zorlanmaktadır.

Yürüme sınıfını en iyi LR algoritması, merdiven çıkma ve merdiven inme sınıflarını en iyi DVM algoritması, oturma ve ayakta durma sınıflarını en iyi YSA algoritması ve son sınıf olan uzanma sınıfını en iyi LR algoritması sınıflandırmıştır.

En yüksek sınıflandırma doğruluğuna sahip model olan YSA ile test verisi sonuçları Tablo 4 ile Tablo 5'te yer almaktadır.

**Tablo 4.** Yapay sinir ağları performans sonucu

Değerlendirme yöntemi	YSA
AUC	%99,9
Sınıflandırma Doğruluğu	%97,9
F1	%97,9
Kesinlik	%97,9
Duyarlılık	%97,9

**Tablo 5.** Yapay sinir ağları algoritmasının test verisi için karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%99,6	%0	%0,4	%0	%0	%0
Merdiven çıkma	%0,2	%99,6	%0,2	%0	%0	%0
Merdiven inme	%0,5	%0,5	%98,8	%0,2	%0	%0
Oturma	%0,4	%0	%0	%94,5	%5,1	%0
Ayakta durma	%0	%0	%0	%4,5	%95,5	%0
Uzanma	%0	%0	%0	%0,4	%0	%99,6

Tablo 4'e bakıldığında YSA algoritmasının test kümesi üzerinde de yüksek sınıflandırma doğruluğuna sahip olduğu görülmektedir.

Tablo 5'e bakıldığında YSA algoritmasının sınıfları ayırmada iyi performans gösterdiği görülmektedir.

#### 4. Tartışma ve Sonuç

İnsan aktivitelerini tanıma konusu nesnelerin interneti ile daha da popüler hale gelmiş ve giyilebilir cihazlar ile veri toplama işlemi kolaylaşmıştır. Bu konu ile ilgili Türkçe kaynakların az olduğu görülmektedir. İlerleyen dönemlerde teknolojinin de gelişmesiyle bu konu üzerinde daha da durulacağını öngörmek mümkündür.

Literatürde derin öğrenme algoritmalarının makine öğrenme algoritmalarına göre daha iyi sonuç verdiği kanısına ulaşılabilir. Çalışmamızda da makine öğrenme algoritmaları içerisinde derin öğrenmenin temelini oluşturan YSA algoritmasının en iyi sonucu vermesi literatürdeki sonuçlar ile örtüşmektedir.

Fiziksel aktiviteler içeren bir veri kümesi üzerinde çalışıldığında elde edilen sonuçların çok yüksek olması ele alınan aktivitelerin karışık veya örtüşen aktiviteleri içermemesinden kaynaklanıyor olabilir. Yürüme koşma gibi fiziksel aktiviteler yerine yemek pişirme, araba kullanma gibi karışık aktiviteler ya da yürürken müzik dinlemek, film izlerken yemek yemek gibi örtüşen aktivitelerin yer aldığı veri kümeleri

üzerinde çalışmalar yapılabilir. Karışık ya da örtüşen aktivite verileri ile çalışıldığında bu oranların düşebileceği göz önünde bulundurulmalıdır.

#### Kaynakça

- [1] T. Şahin, "Akıllı bileklik kullanarak aktivite tanıma uygulamaları," 2019. Accessed: Jun. 04, 2021. [Online]. Available: <https://tezarsivi.com/akilli-bileklik-kullanarak-aktivite-tanima-uygulamaları>.
- [2] M. C. Sorkun, A. E. Danişman, and Ö. D. Incel, "Human activity recognition with mobile phone sensors: Impact of sensors and window size," in *26th IEEE Signal Processing and Communications Applications Conference*, Jul. 2018, pp. 1–4, doi: 10.1109/SIU.2018.8404569.
- [3] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6669, pp. 289–296, doi: 10.1007/978-3-642-21257-4\_36.
- [4] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using AR model representation and SVM," in *Proceedings of the 7th International Conference on Machine Learning and Cybernetics*, 2008, vol. 4, pp. 2245–2250, doi:



- 10.1109/ICMLC.2008.4620779.
- [5] E. Bulbul, A. Cetin, and I. A. Dogru, "Human Activity Recognition Using Smartphones," in *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies*, Dec. 2018, pp. 1–6, doi: 10.1109/ISMSIT.2018.8567275.
- [6] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31314–31338, Dec. 2015, doi: 10.3390/s151229858.
- [7] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, p. 529, Mar. 2017, doi: 10.3390/s17030529.
- [8] İ. A. Metin and B. Karasulu, "İnsan Aktivitelerinin Sınıflandırılmasında Tekrarlayan Sinir Ağı Kullanan Derin Öğrenme Tabanlı Yaklaşım," *Veri Bilim. Derg.*, vol. 2, no. 2, pp. 1–10, 2019, Accessed: Jun. 04, 2021. [Online]. Available: [www.dergipark.gov.tr/veri](http://www.dergipark.gov.tr/veri).
- [9] N. Tufek and O. Ozkaya, "A comparative research on human activity recognition using deep learning," in *27th Signal Processing and Communications Applications Conference*, Apr. 2019, pp. 1–4, doi: 10.1109/SIU.2019.8806395.
- [10] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep Learning Models for Real-time Human Activity Recognition with Smartphones," *Mob. Networks Appl.*, vol. 25, no. 2, pp. 743–755, Apr. 2020, doi: 10.1007/s11036-019-01445-x.
- [11] Y. Chen and Y. Xue, "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Jan. 2015, pp. 1488–1492, doi: 10.1109/SMC.2015.263.
- [12] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, Nov. 2017, doi: 10.3390/s17112556.
- [13] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," 2013.
- [14] "UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set." <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones> (accessed Jun. 04, 2021).
- [15] "Temel bileşen analizi," *Vikipedi, Özgür Ansiklopedi*. [https://tr.wikipedia.org/wiki/Temel\\_bileşen\\_analizi](https://tr.wikipedia.org/wiki/Temel_bileşen_analizi) (accessed Jun. 05, 2021).
- [16] M. F. Akca, "Karar Ağaçları (Makine Öğrenmesi Serisi-3)," *Medium*, 2020. <https://medium.com/deep-learning-turkiye/karar-ağaçları-makine-öğrenmesi-serisi-3-a03f3ff00ba5> (accessed Jun. 06, 2021).
- [17] E. Hatipoglu, "Machine Learning — Classification — Naive Bayes — Part 11," *Medium*, 2018. <https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4> (accessed Jun. 06, 2021).
- [18] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. 2009.
- [20] M. Akman, "Veri madenciliğine genel bakış ve Random Forests yönteminin incelenmesi: Sağlık alanında bir uygulama," 2010. Accessed: Jan. 07, 2021. [Online]. Available: <https://www.ulusaltezmerkezi.net/veri-madenciligine-genel-bakis-ve-random-forests-yonteminin-incelenmesi-saglik-alaninda-bir-uygulama/>.
- [21] M. F. Akca, "Nedir Bu Destek Vektör Makineleri? (Makine Öğrenmesi Serisi-2)," *Medium*, 2020. <https://medium.com/deep-learning-turkiye/nedir-bu-destek-vektör-makineleri-makine-öğrenmesi-serisi-2-94e576e4223e> (accessed Jun. 06, 2021).
- [22] E. K. Ülgen, "Makine Öğrenimi Bölüm-4 (Destek Vektör Makineleri)," *Medium*, 2017. <https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-4-destek-vektör-makineleri-2f8010824054> (accessed Jun. 06, 2021).
- [23] C. Erden, "Lojistik Regresyon Analizi ve Uygulaması," *Youtube*, 2020. <https://www.youtube.com/watch?v=K4arTc6DYGs> (accessed Jun. 06, 2021).
- [24] H. Bircan, "Lojistik Regresyon Analizi: Tıp Verileri Üzerine Bir Uygulama," *Kocaeli Üniversitesi Sos. Bilim. Enstitüsü Derg.*, no. 2, pp. 185–208, 2004.
- [25] Y. Güzel, "Yapay Zeka Ders Notları 03 | Biyolojik Sinir Sistemi ve Yapay Sinir Ağı Hücresi," *Medium*. <https://medium.com/@yasinguzel/yapay-zeka-ders-notları-03-biyolojik-sinir-sistemi-ve-yapay-sinir-ağı-hücresi-6555add68d80> (accessed Jun. 06, 2021).
- [26] M. E. Ocak, "Yapay Sinir Ağları Nedir?," *TÜBİTAK Bilim Genç*, 2020. <https://bilimgenc.tubitak.gov.tr/yapay-sinir-aglari-nedir> (accessed Jun. 06, 2021).
- [27] A. Violante, "An Introduction to t-SNE with Python Example," *Towards Data Science*, 2018. <https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1> (accessed Jun. 05, 2021).

## Ekler

Eğitim veri kümesi için her modelin karışıklık matrisi yer almaktadır.

**Tablo 6.** Karar ağacı algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%89,9	%4,5	%5,6	%0	%0	%0
Merdiven çıkma	%6,8	%86,2	%6,9	%0,1	%0	%0
Merdiven inme	%7,8	%8,7	%83,5	%0	%0	%0
Oturma	%0	%0,1	%0	%76	%20,8	%3,2
Ayakta durma	%0	%0,1	%0	%21	%78,9	%0
Uzanma	%0	%0,3	%0,1	%2,6	%0,6	%96,4

**Tablo 7.** Naïve bayes algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%86,7	%3,4	%9,6	%0,2	%0,1	%0
Merdiven çıkma	%5,5	%86,6	%7,8	%0	%0,1	%0
Merdiven inme	%2,8	%8,9	%88,1	%0,1	%0	%0
Oturma	%2,2	%0,7	%4,4	%76,6	%15,1	%1
Ayakta durma	%1,8	%1,7	%1,8	%9,3	%85	%0,4
Uzanma	%1,9	%0,7	%4,2	%0,1	%0,8	%92,3

**Tablo 8.** Rastgele orman algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%94,6	%2,4	%2,9	%0	%0	%0
Merdiven çıkma	%8,3	%87,3	%4,3	%0,1	%0	%0
Merdiven inme	%5,2	%6,8	%88	%0	%0	%0
Oturma	%0,2	%0,1	%0,1	%78,5	%18,1	%3,1
Ayakta durma	%0,1	%0,1	%0	%15	%84,8	%0,1
Uzanma	%0	%0	%0	%1,7	%0,1	%98,2

**Tablo 9.** Destek vektör makinesi algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%99,3	%0,2	%0,4	%0	%0	%0
Merdiven çıkma	%0,1	%99,3	%0,6	%0	%0	%0
Merdiven inme	%0,1	%0,4	%99,5	%0	%0	%0
Oturma	%0,1	%0	%0,4	%80,6	%18,9	%0,1
Ayakta durma	%0	%0	%0,2	%7,6	%92,1	%0
Uzanma	%0	%0	%1,5	%0,1	%0	%98,4



**Tablo 10.** Lojistik regresyon algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%99,9	%0,1	%0	%0	%0	%0
Merdiven çıkma	%0,6	%98,4	%0,7	%0,3	%0	%0
Merdiven inme	%0,1	%0,5	%99,4	%0	%0	%0
Oturma	%0	%0,2	%0	%92,2	%7,1	%0,5
Ayakta durma	%0	%0	%0	%7	%93	%0
Uzanma	%0	%0	%0	%0,1	%0	%99,9

**Tablo 11.** Yapay sinir ağı algoritmasının karışıklık matrisi

Model	Yürüme	Merdiven çıkma	Merdiven inme	Oturma	Ayakta durma	Uzanma
Yürüme	%99,8	%0,1	%0	%0,1	%0,1	%0
Merdiven çıkma	%0,5	%99,1	%0,3	%0,2	%0	%0
Merdiven inme	%0,1	%0,4	%99,2	%0,1	%0,2	%0
Oturma	%0,1	%0	%0	%93,9	%5,8	%0,2
Ayakta durma	%0,1	%0,1	%0	%5,5	%94,3	%0
Uzanma	%0	%0	%0	%0,3	%0	%99,7