

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

## **AKILLI TELEFON YARDIMIYLA DEPREM TESPİTİ**

**YÜKSEK LİSANS TEZİ**

**Ahmet Furkan SÖNMEZ**

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ**

**Tez Danışmanı : Yrd. Doç. Dr. Seçkin ARI**

**Haziran 2016**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

AKILLI TELEFON YARDIMIYLA DEPREM TESPİTİ

YÜKSEK LİSANS TEZİ

Ahmet Furkan SÖNMEZ

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ

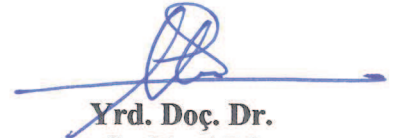
Bu tez 15 / 06 / 2016 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.



Doç. Dr.  
Hüseyin Serdar KÜYÜK  
Jüri Başkanı



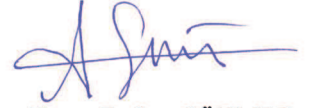
Yrd. Doç. Dr.  
Atınç YILMAZ  
Üye



Yrd. Doç. Dr.  
Seçkin ARI  
Üye

## BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.



Ahmet Furkan SÖNMEZ

27.07.2016

## TEŐEKKÜR

Bu alıőmanın gerekleőtirilmesinde, deęerli bilgilerini bizlerle payőalaőan, saygıdeęer danıőman hocam Yrd. Do. Dr. Sekin ARI'ya, bize ayırdıęı deęerli zamanı ve saęladıęı destek iin Yrd. Do. Dr. Veysel Harun ŐAHİN'e ve Do. Dr. Hüseyin Serdar KÜYÜK'e, bugünlere gelmemde büyük pay sahibi olan canım aileme ve yardımlarını esirgemeyen tüm mesai arkadaşlarıma teőekkürü bir bor bilirim.

# İÇİNDEKİLER

TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
SİMGELER VE KISALTMALAR LİSTESİ .....	v
ŞEKİLLER LİSTESİ .....	vi
TABLolar LİSTESİ .....	viii
ÖZET .....	ix
SUMMARY .....	x
BÖLÜM 1.	
GİRİŞ .....	1
1.1. Literatür Taraması .....	1
1.2. Tezin Amacı .....	4
1.3. Tez Planı .....	4
BÖLÜM 2.	
ANDROİD İŞLETİM SİSTEMİ .....	6
2.1. Android'in Tarihçesi .....	6
2.2. Android İşletim Sistemi'nin Yapısı ve Özellikleri .....	7
2.3. Android Mimarisi ve Sistem Özellikleri .....	7
2.4. Android Versiyonları .....	9
2.5. Android İşletim Sisteminde Uygulama Geliştirilmesi İçin Gerekli Metaryeller .....	10
2.5.1. Android yazılım geliştirme kiti .....	11
2.5.2. Android studio ve özellikleri .....	11

BÖLÜM 3.	
ANDROİD UYGULAMASININ GELİŞTİRİLMESİ .....	13
3.1. İvmeölçer Sensörü .....	13
3.2. Aktiviteler ve Uygulama Yaşam Döngüsü .....	14
3.3. AndroidManifest Dosyası .....	15
3.4. Uygulama Başlangıç Ekranı .....	16
3.5. Verilerin Kaydedildiği Ekran .....	17
BÖLÜM 4.	
BULANIK MANTIK .....	19
4.1. Bulanık Mantığa Giriş .....	19
4.2. Bulanık Denetleyici .....	21
4.3. Bulanık Mantık Üyelik Fonksiyonu ve Üyelik Derecesi .....	23
4.4. Sugeno Bulanık Modeli .....	25
BÖLÜM 5.	
AKILLI DEPREM TESPİT SİSTEMİ .....	27
5.1. Akıllı Telefon İle Verilerin Toplanması .....	27
5.2. Özellik Çıkartılması .....	31
5.2.1. AR model .....	31
5.2.2. Verilerin işlenmesi .....	31
5.3. Bulanık Deprem Tespit Sistemi Gerçeklenmesi .....	39
5.3.1. Bulanık deprem tespit sistemi .....	39
5.3.2. Sistemin üyelik fonksiyonları .....	40
5.3.3. Sistemin kural tabanı .....	42
5.3.4. Sistem çıkışı .....	42
5.4. Hassasiyet Analizi .....	44
BÖLÜM 6.	
SONUÇ .....	45
KAYNAKLAR .....	46

EKLER .....	49
ÖZGEÇMİŞ .....	61

## SİMGELER VE KISALTMALAR LİSTESİ

APK	: Android Package / Android Paketi
AR	: Autoregressive / Otoresif
ART	: Android Run Time / Android Çalışma Zamanı
AWGN	: Add White Gaussian Noise / Beyaz Gauss Gürültü Ekleme
FIS	: Fuzzy Inference System / Bulanık Çıkarım Sistemi
IDE	: Integrated Development Environmt / Entegre Geliş. Ort.
IO	: Input Output / Giriş Çıkış
JDK	: Java Development Kit / Java Geliştirme Kiti
MLP	: Multi-Layer Perceptron / Çok Katmanlı Algılayıcı
NFC	: Near Field Communication / Yakın Alan İletişimi
SDK	: Software Development Kit / Yazılım Geliştirme Kiti
SNR	: Signal-to-Noise Ratio / Sinyal Gürültü Oranı
SVM	: Support Vector Machine / Destek Vektör Makinesi
USB	: Universal Serial Bus / Evrensel Seri Veriyolu



## ŞEKİLLER LİSTESİ

Şekil 2.1. Android işletim sistemi mimarisi katmanı.....	9
Şekil 2.2. Örnek android studio ekran görüntüsü.....	12
Şekil 3.1. Aktivite yaşam döngüsü.....	14
Şekil 3.2. AndroidManifest.xml dosyası.....	16
Şekil 3.3. Uygulama ekranı.....	17
Şekil 3.4. Kaydetme ekranı.....	18
Şekil 4.1. Bulanık mantık örnek sıcaklık gösterimi.....	20
Şekil 4.2. Bulanık denetleyici.....	22
Şekil 4.3. Bulanık küme teorisi.....	23
Şekil 4.4. Triangular fonksiyon gösterimi.....	24
Şekil 4.5. Trapezoidal fonksiyon gösterimi.....	24
Şekil 4.6. Gaussian fonksiyon gösterimi.....	25
Şekil 4.7. Sugeno bulanık modelinin diyagramı.....	26
Şekil 5.1. Veritabanına kaydedilen veri örnekleri.....	28
Şekil 5.2. Merdiven inme sinyali.....	28
Şekil 5.3. Sabit zemin sinyali.....	29
Şekil 5.4. Yürüme sinyali.....	29
Şekil 5.5. Referans cihazı ile akıllı telefon karşılaştırması (sakarya depremi).....	30
Şekil 5.6. Referans cihazı ile akıllı telefon karşılaştırması (elcentro depremi).....	30
Şekil 5.7. Veri setlerinin AR1 katsayıları.....	35
Şekil 5.8. Veri setlerinin AR2 katsayıları.....	36
Şekil 5.9. Veri setlerinin AR3 katsayıları.....	37
Şekil 5.10. Veri setlerinin AR4 katsayıları.....	39
Şekil 5.11. Bulanık deprem tespit sistemi.....	39
Şekil 5.12. Sistemin üyelik fonksiyonları.....	41
Şekil 5.13. Eğitim setinin çıkış değerleri.....	43

Şekil 5.14. Test setinin çıkış değerleri .....	43
Şekil 5.15. Snr değerine göre sistemin doğruluk performansı .....	44

## TABLolar LİSTESİ

Tablo 2.1. Android versiyonları .....	9
Tablo 5.1. Sistemin kural tabanı .....	42
Tablo 5.2. Sistemin çıkış fonksiyonları.....	42

## ÖZET

Anahtar kelimeler: İvmeölçer sensörü, autoregressive model, akıllı telefon, deprem tespiti

Günümüzde akıllı telefonlar ile bir çok işlem gerçekleştirilmektedir. İçinde bulundurduğu ivmeölçer sensörleri sayesinde de çeşitli çalışmalarda kullanılmaktadır. İvmeölçer sensörü; bir kütleye uygulanan kuvvetin büyüklüğünü ölçen sensördür. İlk olarak akıllı telefonlarda bulunan ivmeölçer sensörü ile veriler toplanmıştır. Daha doğal sonuçlar alabilmek için veri toplama esnasında akıllı telefon pantolonun yan cebinde tutulmuştur. Çeşitli aktiviteler (koşma,yürüme,merdiven inip-çıkma, sabit tutarak vb ) esnasında ivmeölçer sensörü ile veri toplanmıştır.

Ham ivmelenme verilerinde yönelim problemini ortadan kaldırmak için toplanan bu ham verilerin genlikleri kullanılmıştır. Kararlaştırılan kaydırma aralığına göre genlikleri hesaplanan veriler belirli uzunluğa sahip olan pencerelere ayrılmıştır. Verilerin deprem mi yoksa insan kaynaklı hareketler mi olduğuna karar vermek için bulanık sistem Autoregressive (AR) model kullanılarak Matlab de geliştirilmiştir.

Çıkış verileri deprem verisi için pozitif insan aktivitesi için negatif olacak şekilde seçilmiştir. Veriler eğitim ve test verileri olmak üzere iki gruba ayrılmıştır. Böylece farklı koşullarda toplanan ivmelenme verilerinin eğitim ve test veri grupları oluşturulmuştur.

Test verileri ile yapılan deneylerde sistem gerçek deprem kaynaklı olan verilerin tümünü deprem olarak; insan hareketlerinden kaynaklanan verilerin tümünü de insan aktivitesi olarak algılamıştır.

# EARTHQUAKE DETECTION USING A SMARTPHONE

## SUMMARY

Keywords: Accelerometer sensor; Autoregressive model; Smartphone; Earthquake detection

Nowadays, many transactions are achieved with smartphones. Through its holdings in accelerometer sensors it is also used in the various studies. Accelerometer sensor is a sensor that measures the size of a power applied onto a mass. First of all data was gathered with the help of accelerometer sensor on mobile phones. In order to produce more natural results, the smartphones in the study were carried in side pockets of trousers during data gathering. Data was gathered during a variety of activities like running, walking, going upstairs and downstairs, holding still and so on via accelerometer sensor.

In order to avoid orientation problem in raw acceleration data, amplitudes of these raw data were used. Pieces of data whose amplitudes were calculated according to the pre-determined slider range were divided into windows with certain length. A fuzzy system has been developed to decide whether the the shake is due to earthquake movements or human movements using AR (Autoregressive) model with Matlab.

Output data were chosen in such a way that they would be positive for earthquake data and negative for human activity. Data were divided as training and test data. In this way, training and set data groups of acceleration data gathered from various conditions were formed.

During experiments with test data, the system was able to differentiate between all earthquake-based data and human-caused data successfully.

# BÖLÜM 1. GİRİŞ

## 1.1. Literatür Taraması

Bu alanda ve buna benzer alanlarda yapılan birçok çalışma mevcuttur. Alham Fikri Aji ve arkadaşları telefonlarda bulunan ivmeölçer sensörü ile deprem tespiti çalışmasını gerçekleştirmiştir. Yapılan deneyler akıllı telefonlar ile kullanılarak kaydedilen deprem sinyallerin örüntüsünü öğrenmek için tasarlanmıştır. Makine öğrenmesi ile n-gram modelleme kullanılarak sinyaller işlenmiştir. Sınıflama içinse Naïve Bayes, Multi-Layer Perceptron (MLP), and Random Forest yöntemleri kullanılmıştır. Çalışmalarında en iyi sınıflandırma doğruluğu random forest metodu ile sağlanmıştır. En yüksek doğruluk oranı %93,15'tir. Akıllı telefonlar ile deprem tespit edilebileceğinin sonucuna varılmıştır [1].

Tieta Antaresti ve arkadaşları insanlara deprem ile ilgili bilgi vermeyi amaçlayan internete bağlı olmayan telefonların kullanıldığı bireysel bir çalışma yapmışlardır. Veriler sınıflandırma işleminden önce doğruluğu arttırmak için statik pencereleme yapıldı. Kutu sayma boyutu özelliği ve hurt katsayısı gibi özellikler verilere uygulanmıştır. Statik pencereleme uygulanmıştır. Çünkü mümkün olduğunca çok sayıda yararlı özellik elde edilmeye çalışılmıştır. Yine burada da makina öğrenme yolu ile deprem sinyalleri belirlenmeye çalışılmıştır. Sınıflandırma olarak destek vektör makinesi ve çok katmanlı algılayıcı seçilmiştir. %81 ve %82,5 hassasiyeti ile ölçüm yapılmıştır [2].

Bir çok sismik algılama yaklaşımı bulunmaktadır. Jack Reilly ve arkadaşları bilimsel topluluk veya acil müdahale ekiplerinin depremden sonra hızlı müdahale etmesi için yapılan bir sistem olan iShake projesini geliştirmiştir. iShake projesi mobil server mimarisi şeklinde tasarlanmıştır. Bu mimaride deprem hareketlerini ölçmek için

cihazların içinde bulunan sensörler kullanılmaktadır. Sarsma tablası testleri ölçüm kalitesini arttırmak için yapıldı. iPhone veya iPod ile 150 çeşit yer hareketi ölçülmüştür. Bir sismik olay yüksek doğruluk istasyonları tarafından doğrulandıktan sonra filtreleme algoritmaları düşen telefonu tespit etmek veya aygıtı özgü yanılgıları engellemek için kullanılır. Cihazın mutlak yönüne karar vermek için bir metod geliştirilmiştir. Bir "sanal deprem" pilot testi iShake sisteminin çalışmasını doğrulamak için UC Berkeley kampüsünde gerçekleştirilmiştir. iShake projesi mobil cihazlar ile sismik algılama alanında bir çok temel hizmet vermektedir [3].

Qingkai Kong deprem kaynaklı yer hareketlerini ölçmek için akıllı telefonlar içinde yer alan ivmeölçer sensörü kullanılarak yapılabileceği savunmuştur. Bunu anlamak için ise noise floor ve shake table testlerini uygulamıştır. Bu testlerin sonucunda ivmeölçer sensörünün uygun olduğu görülmüştür [4]. Qingkai Kong Nodal Seismic Company tarafından dağıtılan Nodal seismic Array dizisi kullanmıştır. Bu dizi petrol kaynaklarının aktif kaynak keşifleri için geliştirilmiştir [5]. Shideh Dashti iShake çalışmasında iphone ve ipodların hassaslığı ölçmüştür. Çünkü bu cihazlar kusursuz ölçüm yapamamaktadırlar. Bu cihazların hassasiyeti bir yere sabitlenerek veya sabitlemeden test edilmiştir [6]. Niksa Orlic ve arkadaşı en büyük problemin doğal ve yapay sismik hareketleri birbirlerinden ayırt etmek olduğunu savunmuştur. Genetik algoritmanın kullanıldığı yeni bir yaklaşım kullanılmıştır [7].

Zhen-Yu He bu çalışmada Zaman serisinin Autoregressive (AR) modelini; üç eksenli ivmelenme verileri ile insan aktivitelerini ayırt etmek için kullanmıştır. Koşma, zıplama, yürüme, durağan gibi insan hareketlerini tespit etmeye çalışmıştır. 4. derece ar model kullanılmıştır. Özellik çıkarma olarak AR katsayı yöntemi ve sınıflandırma olarak ise SVM (Support Vector Machine) kullanılmıştır. Ar model kullanılarak dört aktivite için ortalama ayırt etme sonucu %92,25 olduğu görülmüştür. Ortalama, standart sapma korelasyon ve fft gibi özelliklere oranla en yüksek doğruluk sonucuna ulaşılmıştır. Çıkan sonuçlar; ar katsayılarının farklı insan hareketlerini ayırt etmede kullanılabileceğini göstermektedir [8].

Dinesh Singh özellik çıkarma tekniğini astımlı hastalarda ve normal hastalarda uygulamıştır. Mel - Frekans Cepstral Katsayısı tekniği özellik çıkarma işlemi için kullanılır. Çerçeveleme ve hamming pencereleme işlemlerini uygulamıştır. Mel Frekans Cepstral Katsayılarını kullanarak Matlab'de özellik çıkartmıştır. Astım hastaları ile sağlıklı insanların tespiti yapılmıştır [9]. Jani Mantyjarvi bu çalışmada insan aktivitelerini tanıma amaçlı ivmeölçer sensörlerine sahip giyilebilir cihazları kullanmıştır. Temel bileşen analizi ve bağımsız bileşen analizinin dalgacık dönüşümü ile kullanımı özellik üretimi için test edilmiştir. İnsan aktivitelerini tanınması işlemi çok katmanlı algılayıcı sınıflandırma ile yapılmıştır. Farklı insan hareketlerinin tanınması için en iyi sınıflandırma sonuçları %83-90 olarak ölçülmüş. Bu sonuç bağımsız bileşen analizi ve temel bileşenler analizi kullanılarak elde edilmiştir. Bu iki yöntem arasındaki farkın önemsiz olduğu ortaya çıkmıştır [10].

Ashish Shrestha ve arkadaşları düşük maliyetli akıllı cihazların yerleşik mems ivmeölçer sinyali ile sismik ve yapısal tepki ölçümleri için kullanılabileceğini savunmuştur. Ölçüm uygulama programı; ivme ölçme, kayıt, veri depolama ve bulut sunucu senkronizasyonu için geliştirilmiştir. Ölçümlerin güvenilirliği sinüs dalgası yüklenen farklı cihazlar yardımıyla sarsma tablası testleri ile değerlendirilmiştir. Önerilen metodun avantajları ve sınırlamaları özetlenmiştir [11].

Muhammad Shoaib ve arkadaşları gömülü sensörler kullanılarak fiziksel aktivite tanıma işlemi sağlık alanı gibi bir çok alanda kullanıldığını belirtmiştir. Başlangıçta bir veya daha fazla sensör bu tür uygulamalar için kullanılmıştır. Fakat son zamanlarda bir çok araştırmacı bu amaç için akıllı telefonları kullanmaya başlamıştır. Çünkü bu cihazlar manyetik alan ve ivmeölçer gibi çeşitli algılayıcılar ile donatılmıştır. Geçerli çalışmaların çoğunda, aktivite tanınması için toplanan sensör verileri makine öğrenmesi araçlarını kullanarak çevrimdışı analiz edilmektedir. Ancak şimdilerde modern cep telefonları bu işlemci, bellek ve pil gibi mevcut donanımları ile her açıdan daha güçlü hale gelmiştir. Çevrimdışı aktivite tanıma konusunda bir çok araştırma gözden geçirilmiştir. Yapılan çalışmalar üzerinde karşılaştırmalar yapılmış kullanılmış işletim sistemleri ile sınıflama algoritmaları hakkında detaylar verilmiştir [12].



Qingkai Kong ve arkadaşları geçerli deprem erken uyarı sistemlerini geliştirmek için sosyal ve mobil hesaplamayı da içeren yeni teknolojilerin ilgi odağı olduğunu düşünmüşlerdir. Akıllı telefonlar son bir kaç yıldır önemli gelişmeler kaydetmiştir. İçerisinde bulundurduğu çeşitli sensörler ile çeşitli insan hareketlerini ve deprem hareketlerini kaydedebilmektedir. Bu amaçla deprem ve insan hareketlerinin ayrımını sağlayacak olan bir akıllı telefon yazılımı geliştirilmiştir. Bunu yaparken amaçları akıllı telefon yardımıyla yeni bir sismik ağ türü kurmaktır. Bu çalışmada akıllı telefonların tespit cihazı olarak kullanılması değerlendirildi. Deprem ve insan aktivitelerini ayıracak olan algoritma geliştirildi. Sonuç olarak kullandıkları algoritma ile akıllı telefonların sadece kayıt cihazları olmadığı aynı zamanda son derece doğru deprem algılama araçları olduğu görülmüştür [13].

## **1.2. Tezin Amacı**

Bu çalışmanın amacı akıllı telefonlardaki ivmeölçer algılayıcıları kullanarak telefon üzerinde meydana gelen sarsıntının insan aktivitesinden mi yoksa deprem kaynaklı mı olduğunu tespit etmektir.

## **1.3. Tez Planı**

Tezin 2. bölümünde android işletim sisteminden anlatılmıştır. Android işletim sisteminin tarihçesi, yapısı, özellikleri, mimarisi ve versiyonlarından bahsedilmiştir. Uygulama geliştirmek için gerekli metaryeller hakkında bilgiler verilmiştir.

Tezin 3. bölümünde gerçekleştirilen android uygulaması anlatılmıştır. Aktivitelerden ve Uygulama Yaşam Döngüsünden bahsedilmiştir. Android manifest dosyası hakkında açıklamalar yapılmıştır. Uygulamanın başlangıç ekranı ve verilerin kaydedildiği ekran resimleri eklenmiştir.

Tezin 4. bölümünde bulanık mantık hakkında genel bilgiler verilmiştir. Bulanık mantığın temelinden ve yapısından bahsedilmiştir. Bulanık mantık ile klasik mantık

karşılaştırılmıştır. Bulanık mantığın avantaj ve dezavantajları belirtilmiştir. Bulanık denetleyiciler anlatılmıştır. Üyelik fonksiyonu ve üyelik derecesi kavramları açıklanmıştır.

Tezin 5. bölümünde gerçekleştirilen akıllı deprem tespit işleminden detaylı bir şekilde bahsedilmiştir. Kullanılan ar model anlatılmıştır. Özellik çıkartma yöntemleri açıklanmıştır. Bulanık deprem tespit sistemin giriş ve çıkış değerleri belirtilmiştir. Sistemin kural tabanının ve çıkış fonksiyonlarının tablosu verilmiştir.

Tezin 6. bölümünde uygulamanın sonucundan ve daha sonra yapılabilecek çalışmalardan bahsedilmiştir.

## **BÖLÜM 2. ANDROID İŞLETİM SİSTEMİ**

Android, Google ve Open Handset Alliance tarafından kodlanmış, taşınabilir bilişim cihazları için üretilmiş linux tabanlı açık kaynak kodlu bir mobil işletim sistemidir. Düşük maliyetli ve özelleştirilebilir bir işletim sistemidir. Bu sayede hem kullanıcılar hem de geliştiriciler için büyük kolaylık sağlamaktadır.

Açık kaynaklı olmasının etkisiyle birlikte çok hızlı bir şekilde ilerleme kaydetmiştir.

Android işletim sistemi, günümüzde büyük çoğunluğu akıllı telefonlar olmak üzere, dizüstü bilgisayarlarda, tabletlerde, saatlerde televizyonlarda ve gibi birçok elektronik cihazlarda kullanılmaktadır.

Android'in kullanıcı arayüzü genellikle dokunma kaydırma gibi gerçek dünya hareketlerine dayanır. Aynı zamanda çoğunlukla sanal klavye ile metin girişleri yapılmaktadır.

Android'in uygulama uzantısı .apk'dır [14], [15].

### **2.1. Android'in Tarihçesi**

2003 yılında Andy Rubin, Nick Sears, Rich Miner ve Chris Whie Android Inc. şirketini kumuştur. Bu şirket, yaptığı çalışmalar sonucunda Linux çekirdeğine sahip işletim sistemini cep telefonunda çalıştırmayı denemiştir ve başarılı olmuştur. Bu şekilde ilk Android çalışmalarına adım atılmıştır. Daha sonra Google Android Inc.'yi 2005 yılında satın almıştır. Google Android firmasını satın almasıyla bu firmada çalışanlar Google ekibine katılmıştır.

Open Handset Alliance, mobil teknoloji alanında büyük önem taşıyan uluslararası birliktir. Farklı donanım, yazılım ve telekom şirketlerini bünyesinde barındıran mobil cihazlar için bir işletim sistemi geliştirmeyi amaç edinmiştir. Google'ın önderliğinde 5 Kasım 2007 tarihinde 34 farklı geliştirici firmayı bir araya getirerek Android'in kurulduğunu duyurmuştur [16].

Google'ın android kodlarını yayınlaması ile bir çok telefon üreticileri android işletim sistemini cep telefonlarında kullanmaya başlamıştır. Android işletim sistemli telefonu ilk üreten firma HTC 'dir.

## **2.2. Android İşletim Sistemi'nin Yapısı ve Özellikleri**

Android çok güçlü bir işletim sistemidir. Bir çok özelliğe sahiptir. Bunlardan bazıları aşağıda listelenmektedir.

1. Açık kaynak kodludur.
2. Android linux çekirdeğini kullanmaktadır.
3. Androidde java sanal makinası kullanılmaktadır [16].

Android işletim sistemli cihazların sahip olduğu bazı donanımlar aşağıda belirtilmiştir.

Bunlar şu şekilde sıralanabilir : Dokunmatik ve çoklu dokunmatik ekranlar, sanal veya gerçek klavyeli ekranlar, gps, 1 veya daha fazla kamera, dijital pusula, ışık sensörü, ısı sensörü, ivmeölçer sensörü, bluetooth, yakınlık sensörü, wifi, barometre, USB, NFC, 2G-3G-4.5 G vb...

## **2.3. Android Mimarisi ve Sistem Özellikleri**

Android işletim sistemi 5 bölüme ve 4 ana katmana ayrılır. Bu bölümler aşağıda listelenmektedir.

Çekirdek: Bu bölüm süreç yönetimi, hafıza yönetimi, cihaz yönetimi gibi temel sistem fonksiyonlarını gerçekleştirir. Ayrıca ağ ve aygıt sürücülerini gibi bir çok kısım bu bölüm tarafından yönetilir.

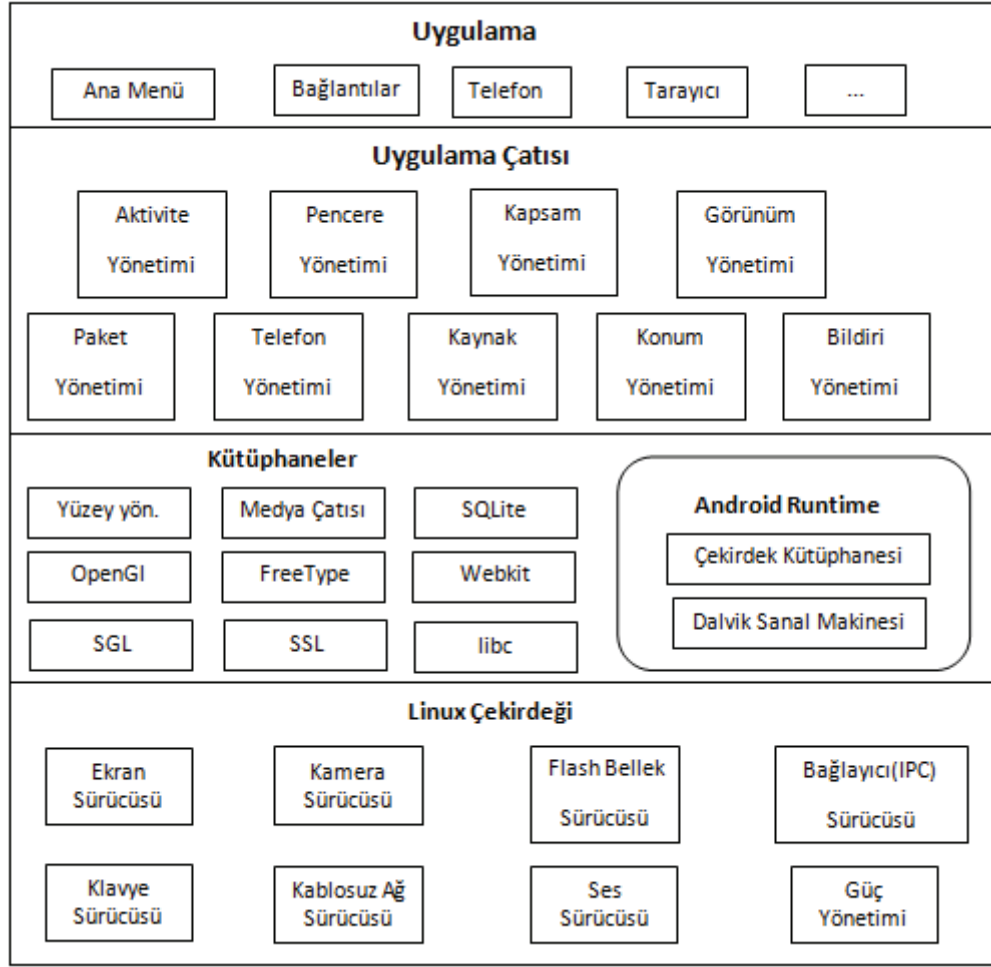
Android runtime: Dalvik Sanal Makinasının bulunduğu katmandır. Android'de yazılan kodlar Dalvik sanal makinasında kullanılmak üzere .dex uzantılı dosyalara dönüştürülmekte idi. Bu durum performans düşüşüne sebep olduğu için dalvik sanal makinası android 5.0 ile kaldırılmış ve ART'ye geçilmiştir.

Kütüphaneler: Linux çekirdeğine erişimin gerçekleşmesini sağlayan bileşenlerin bulunduğu bölümdür. Veritabanı kütüphaneleri, web tarayıcı kütüphaneleri, grafik ve arayüz kütüphanelerini kapsamaktadır.

Uygulama çatısı: Uygulama geliştiricilere zengin donanım ve yazılım desteği sunan bölümdür.

Uygulamalar: İşletim sisteminde kullanılan doğrudan java diliyle geliştirilmiş olan uygulamaların oluşturduğu bölümdür [17].

Şekil 2.1.'de android mimarisi görülmektedir.



Şekil 2.1. Android işletim sistemi mimarisi katmanı [18]

## 2.4. Android Versiyonları

Günümüze kadar çıkmış bir çok android işletim sistemi sürümü bulunmaktadır. Tablo 2.1.'de android versiyonları görülmektedir.

Tablo 2.1. Android versiyonları [14]

1.1	Astro	9 Şubat 2009
1.2	Bender	16 Mart 2009
1.5	Cupcake	30 Nisan 2009

Tablo 2.1. (Devamı)

1.6	Donut	15 Eylül 2009
2.0	Eclair	26 Ekim 2009
2.1		Ocak 2010
2.2	Froyo	20 Mayıs 2010
2.3	Gingerbread	6 Aralık 2010
3.0		Şubat 2011
3.1	Honeycomb	Temmuz 2011
3.2		Kasım 2011
4.0	Ice Cream Sandwich	19 Ekim 2011
4.1		9 Temmuz 2012
4.2	Jelly Bean	29 Ekim 2012
4.3		24 Temmuz 2013
4.4	KitKat®	31 Ekim 2013
5.0		12 Kasım 2014
5.0.1		2 Aralık 2014
5.0.2	Lollipop	19 Aralık 2014
5.1.0		10 Mart 2015
5.1.1		
6.0	Marshmallow	5 Ekim 2015

## 2.5. Android İşletim Sisteminde Uygulama Geliştirilmesi İçin Gerekli Metaryeller

Android işletim sisteminde uygulama geliştirmeye başlamadan önce yapılması gereken bazı işlemler bulunmaktadır. Bunlar aşağıda belirtilmiştir.

1. Öncelikle android işletim sisteminde uygulama geliştirmeden önce kullanılacak bilgisayarda java kurulu olması gerekmektedir. Bunu için <http://www.oracle.com/technetwork/java/javase/downloads/index.html> adresinden en güncel JDK indirilmelidir. Daha sonra kurulumu tamamlanıp path tanımı yapılmalıdır.
2. Android SDK kurulumu gerçekleştirilmelidir. (İstenildiği takdirde android studio ile birlikte de kurulabilmektedir.)
3. Android Studio veya Eclips gibi tümleşik geliştirme ortamı yüklenmelidir.

### **2.5.1. Android yazılım geliştirme kiti**

Android software development kit olarak bilinir. Belli bir yazılım paketi, yazılım çatısı, donanım platformu, bilgisayar sistemi, oyun konsolu, işletim sistemi veya bunlara benzer bir platform için uygulama geliştirilmesini sağlayan yazılımdır.

Genellikle yazılım geliştirme kiti hata ayıklama ve diğer yardımcı programları içerirler ve tamamı belli bir tümleşik geliştirme ortamı (IDE) içerisinde bulunur. Ayrıca SDK lar analiz veri sağlamak amacı ile tanınmış Google ve Facebook gibi uygulamalarda yüklü olarak gelebilmektedir. SDK ayrıca bazı noktaları açıklığa kavuşturmak için sık sık örnek kod ve destekleyici teknik notları veya dokümanları içerebilmektedir.

### **2.5.2. Android studio ve özellikleri**

Android Studio android için resmi tümleşik geliştirme ortamıdır. IntelliJ IDEA üzerine kurulmuştur. Android Studio 16 Mayıs 2013 tarihinde Google I/O etkinliğinde duyurulmuştur. Aşağıda bazı özellikleri listelenmektedir.

1. Gradle tabanlı esnek inşa desteği
2. Çoklu apk çıktısı
3. Kod şablonları ile yaygın uygulama üretimi
4. Sürükle ve bırak özellikli zengin editör



5. Uygulamanın performansını, kullanılabilirliği, farklı sürümlerde çalışmasını sağlayan test araçlarını içermeye
6. Google hizmetlerini uygulamaya ekleyebilme
7. SDK Android studio içerisinde gömülü olarak gelmektedir. Bu sebeple dışarıdan herhangi bir SDK kurulumuna gerek kalmamaktadır.
8. Yeniden yapılandırma ve kod tamamlama işlemleri ile kolay kod yazımı [19]

Şekil 2.2.'de android studio programının arayüzü görülmektedir.

```

package com.example.furkan.sensorapp;

import ...

public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {

    String akt;
    Spinner spinner;
    Spinner spinner2;
    Veritabanı db;
    private Veritabanı kayıtlar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final EditText ed = (EditText) findViewById(R.id.editText);
        final String girilen=ed.getText().toString();

        final AlertDialog.Builder alertbox2 = new AlertDialog.Builder(this);
        alertbox2.setTitle("UYARI");
        alertbox2.setMessage("Silmeğe istediğimize emin misiniz?");
        alertbox2.setNegativeButton("NO",
            (dialog, which) -> {
                // TODO Auto-generated method stub
                Toast.makeText(getApplicationContext(),
                    "hayır butonuna tıklandı", Toast.LENGTH_SHORT)
                    .show();
            });
        alertbox2.setPositiveButton("YES",

```

Şekil 2.2. Örnek android studio ekran görüntüsü

## **BÖLÜM 3. ANDROID UYGULAMASININ GELİŞTİRİLMESİ**

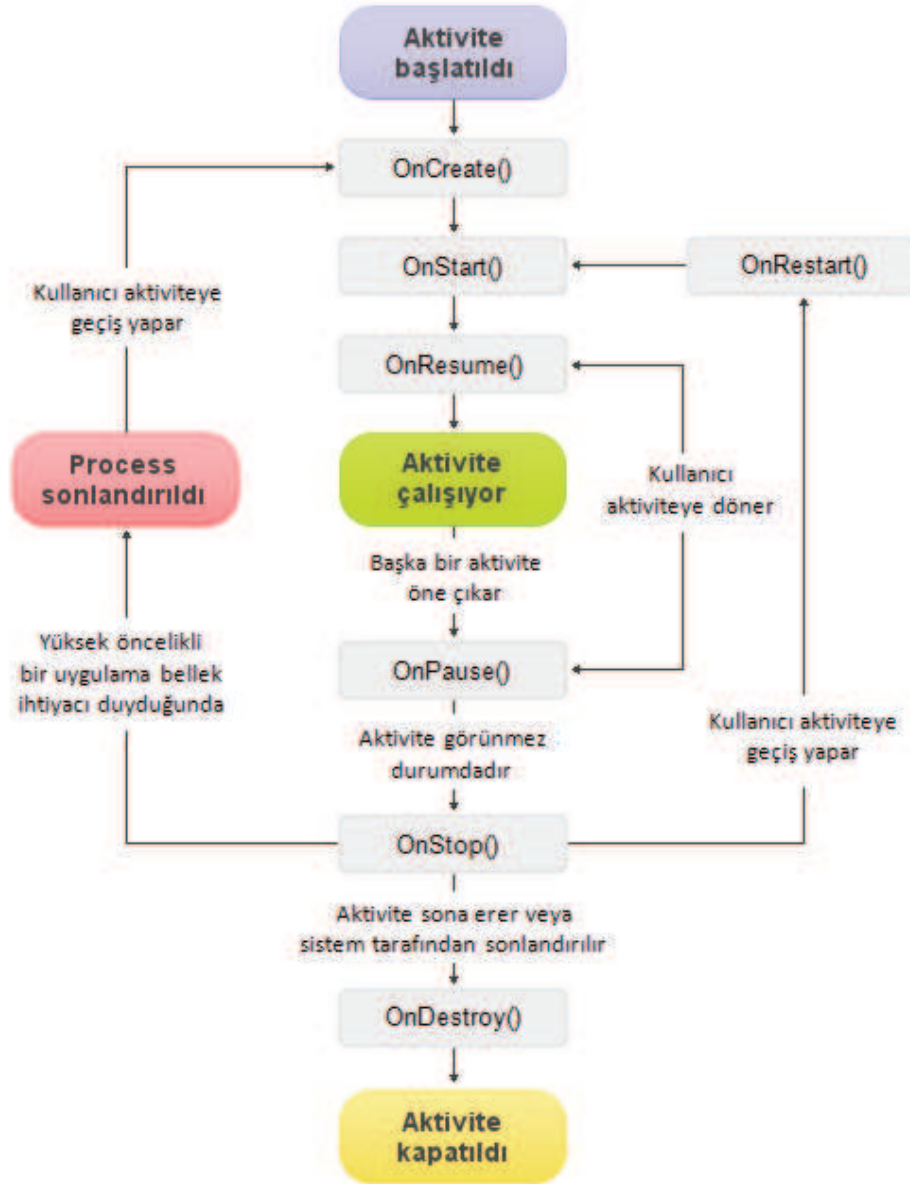
Bu bölümde geliştirilen android uygulaması anlatılacaktır. Android studio ile bu uygulama geliştirilmiştir. Bu program telefonda bulunan ivmeölçer sensörlerini kullanarak cihazın ivmelenmesini ölçüp kayıt etmektedir. Android programlamada sensörlerin örnekleme (örnek/ms) duyarlılığını gösteren Normal, UI, Game ve Fastest olmak üzere 4 çeşit sensör duyarlılığı modu bulunmaktadır. Bu programda sensör duyarlılığı game modu olarak seçilmiştir. Game modunda ivme ölçer sensörü yaklaşık 35-40 ms'de bir örnek almaktadır. Bu android programı ile insan hareketleri (koşma, yürüme, merdiven inip-çıkma, diğer,..) sonucu oluşan çeşitli veriler toplanmıştır. Bu verilere ek olarak deprem sarsma tablasından da gerçek deprem verileri toplanmıştır. Veriler Sony Xperia E4 adlı E2105 model numaralı akıllı telefon ile toplanmıştır. Her bir veri X eksenindeki ivmelenmeden, Y eksenindeki ivmelenmeden, Z eksenindeki ivmelenmeden, aktivite türünden ve başlama zamanından oluşmaktadır. Verilerin toplanması işlemi değişik sürelerde gerçekleştirilmiştir. Bu veriler sadece bir kişi tarafından toplanmıştır.

### **3.1. İvmeölçer Sensörü**

Bir kütleye uygulanan ivmeyi ölçen cihazlardır. Ölçüm yaparken, içerisinde bulundurduğu konumlandırılmış test kütlelerinin, değişken konumundan faydalanır. Sensörden alınan değerler  $m/s^2$  veya yer çekimi (g-Force) türünden sonuç üretir. Tek ve çok eksenli modelleri bulunmaktadır. Askeri, otomobil, endüstriyel ve bir çok teknolojik alanda kullanılmaktadır.

### 3.2. Aktiviteler ve Uygulama Yaşam Döngüsü

Android programlamada uygulamalar onCreate() metoduyla başlar ve başka metodlar sırasıyla devam ederek aktivite yaşam döngüsünü tamamlar. Şekil 3.1.'de aktivite yaşam döngüsü gösterilmektedir.



Şekil 3.1. Aktivite yaşam döngüsü [20]

Yukarıda gösterilen metodların tamamının kullanılması zorunlu değildir. Bu metodlar aşağıda açıklanmıştır.

onCreate(): Aktivite ilk oluşturulduğunda bu metod çağrılır.

onStart(): Bu metod aktiviteyi artık kullanıcı tarafında görünür (visible) duruma getirir.

onResume(): Bu metod ile birlikte kullanıcı daha önceden duraklattığı aktivite ile tekrar iletişim haline geçebilir. onPause() işlemini takip eder.

onPause(): Daha öncesinde devam eden uygulamayı durdurmak için kullanılır. Bununla beraber artık kullanıcıdan herhangi bir girdi alamaz ve hiçbir kod çalıştırılmaz.

onStop(): Artık aktivite görünür (visible) durumda değildir. Bu metoddan sonra kullanıcı ya prosesi sonlandırır yada onRestart() metodu ile aktiviteye geçiş yapar.

onDestroy(): Bu metod ile aktivite sistem tarafından yok edilir. Aktivite bu metod ile yaşam döngüsünü tamamlar.

onRestart(): Aktivite durdurulduktan sonra tekrar başlatılmak için bu metod kullanılır. onStop işlemini takip eder.

### **3.3. AndroidManifest Dosyası**

Her android uygulaması AndroidManifest.xml dosyasını bulundurmak zorundadır. Manifest dosyası hangi izinlerin istendiği, hangi servislerin kullanılacağı gibi gerekli android sistem bilgilerini içermektedir. Şekil 3.2.'de AndroidManifest.xml dosyası gösterilmiştir.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.furkan.sensorapp" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SensorApp"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="SensorApp" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".sayfaiki"
            android:label="SensorApp" >
            <intent-filter>
                <action android:name="android.intent.action.SAYFAIKI" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Şekil 3.2. AndroidManifest.xml dosyası

### 3.4. Uygulama Başlangıç Ekranı

Uygulama ekranında başlat, veri temizle, ve kapat butonları bulunmaktadır. Ayrıca aktivite ve süre isimli spinner yer almaktadır. Kayıt işlemine başlamadan önce aktivite türü ve süre uzunluğu seçilmelidir.

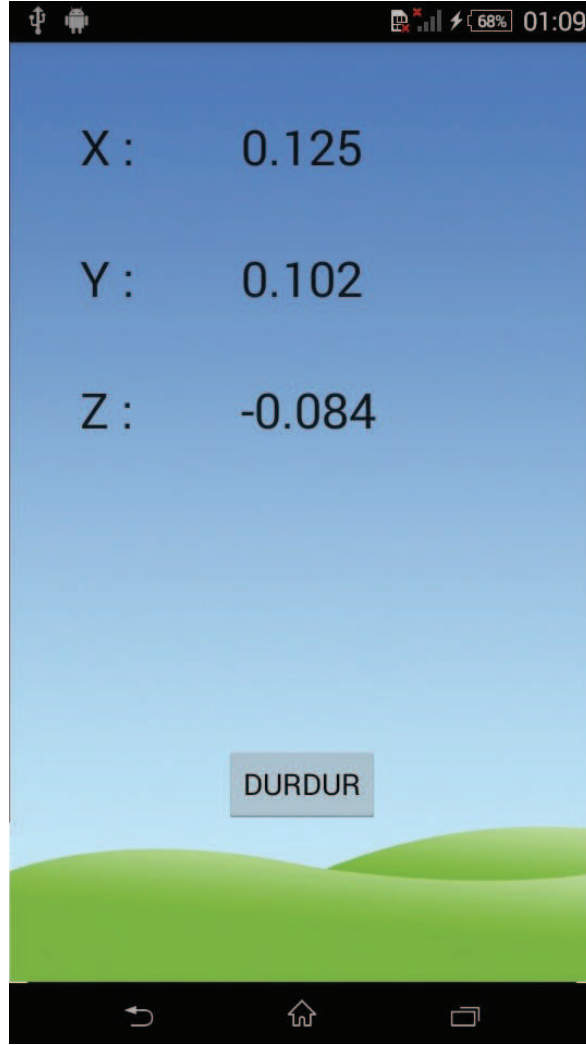
Şekil 3.3.'te geliştirilen android uygulamasının başlangıç ekranı görülmektedir.



Şekil 3.3. Uygulama ekranı

### 3.5. Verilerin Kaydedildiği Ekran

Verilerin kaydedildiği ekranda bir önceki sayfada belirtilen süreye kadar veriler toplanmaktadır. Eğer istenildiği takdirde durdur butonu ile o anda veri toplama işlemi bitirilmektedir. Şekil 3.4. verilerin kaydedildiği ekranı göstermektedir.



Şekil 3.4. Kaydetme ekranı

## **BÖLÜM 4. BULANIK MANTIK**

### **4.1. Bulanık Mantığa Giriş**

İkili mantık sistemine karşı geliştirilen ve günlük hayatta kullandığımız değişkenlere üyelik dereceleri atayarak, olayların hangi oranlarda gerçekleştiğini belirleyen çoklu mantık sisteme bulanık mantık denir.

Lotfi A. Zadeh tarafından geliştirilen bulanık mantığın ve bu mantık kurallarını kullanan bulanık küme teorisi 1965 tarihli orijinal makalesinde yayınlanmasından sonra belirsizlik içeren sistemlerin incelenmesi yeni bir boyut kazanmıştır. 1965 yılında ortaya atılmasına rağmen, bulanık küme kavramı ancak 1970 yılından sonra kullanılmaya başlanmıştır. Bu şekilde olmasında özellikle Zadeh'in 1965 deki ilk makalesinden [21] daha fazla etkili olan ve bulanık mantığın belirsizlik içeren sistemlere uygulanabilirliğini açıklayan makaleleri [22], [23] etkili olmuştur. 1980'li yılların sonrasında Japonların çeşitli sektörlerde bulanık mantığı kullanmasıyla, bu mantığın popüleritesi daha da artmıştır. Böylece yeni bir uygulama ve araştırma alanı doğmuştur.

Günlük hayatta kullanılan ifadelerin bir çoğu genellikle bulanık bir yapıya sahiptir. Herhangi bir şeyi tanımlarken, açıklarken kullanılan ifadeler bulanıklık içerir. Bu ifadeler örnek olarak; hızlı, yavaş, yaşlı, genç, sıcak, soğuk, pahalı, ucuz, biraz, fazla gibi bir çok ifade gösterilebilir. Bizler bir olayı anlatırken bu tür kesinlik belirtmeyen ifadeler kullanırız. Havanın sıcaklığına göre soğuk, çok soğuk, sıcak, çok sıcak veya ılık diyebiliriz. Herhangi bir arabanın fiyatına göre pahalı veya ucuz diyebiliriz. Bu durumlar insan beyninin belirsiz ve kesinlik içermeyen durumlarda



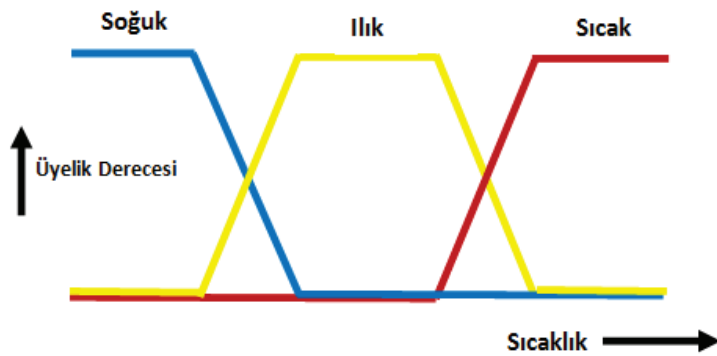
nasıl davrandığına ve olayları nasıl değerlendirip, tanımlayıp, komut verdiğiine dair birer örnektir [24].

Bulanık mantığın temelinde bulanık küme ve alt kümeler bulunmaktadır. Klasik mantık yaklaşımında bir varlık ya kümenin elemanıdır ya da değildir. Matematiksel olarak ifade edildiğinde varlık küme ile olan üyelik ilişkisi bakımından kümenin elemanı olduğunda "1", kümenin elemanı olmadığı zaman "0" olarak değerlendirilmektedir. Bulanık mantık klasik küme gösteriminin genişletilmiş hali olarak gösterilebilir. Bulanık varlık kümesinde her bir varlığın üyelik derecesi vardır ve bu üyelik derecesi (0,1) aralığında herhangi bir değer olabilir. Üyelik fonksiyonu  $M(x)$  ile gösterilir.

Örnek olarak normal oda sıcaklığını 23 derece olduğunu kabul edersek klasik küme kuramına göre 23 derecenin üzerindeki tüm sıcaklık derecelerini sıcak olarak kabul ederiz ve bu derecelerin sıcak kümesindeki üyelik dereceleri "1" olur. 23 derecenin altındaki tüm sıcaklık dereceleri ise soğuktur ve sıcak kümesindeki üyelik dereceleri "0" olur. Soğuk kümesini temel aldığımızda bu değerler tam tersine döner [25].

Bulanık mantıkta üyelik dereceleri 0 ve 1 arasında sonsuz sayıda değerle tanımlanabildiği için üyelik derecesi, elemanın o kümeye ne kadar ait olduğunu gösterir. Örneğin; üyelik derecesi 0,3 olan bir eleman o kümeye %30 aitken, üyelik derecesi 0,85 olan bir eleman o kümeye %85 ait olur.

Şekil 4.1. bulanık mantıkta örnek bir sıcaklık grafiğini göstermektedir.



Şekil 4.1. Bulanık mantık örnek sıcaklık gösterimi

Bilinen geleneksel yöntemlerine alternatif olarak kabul edilen bulanık mantık, doğadaki işleyişi taklit ederek sonuca ulaşır. Bulanık mantık kavramı iki temel öğeden oluşur;

1. Bulanık kümeler
2. Karar verme süreci [26]

#### Bulanık Mantığın Avantajları

1. İnsanın düşünme tarzına daha yakındır.
2. Matematiksel modele ihtiyaç duyulmaz.
3. Yazılımı basittir.
4. Bulanık mantıkta sonuç klasik gibi 0 ve 1 lerden ibaret değildir. Bunların ara değerlerini de içerir.
5. Kullanım ve uygulama alanları çok geniştir.
6. İnsana özgü tecrübe ile öğrenme olayının kolay modellenebilmesi sağlar ve belirsiz kavramların bile matematiksel olarak ifade edilmesine imkan tanır.
7. Problemin karmaşıklığı hızı çok fazla etkilememektedir.
8. Diğer tekniklere göre daha esnektir.

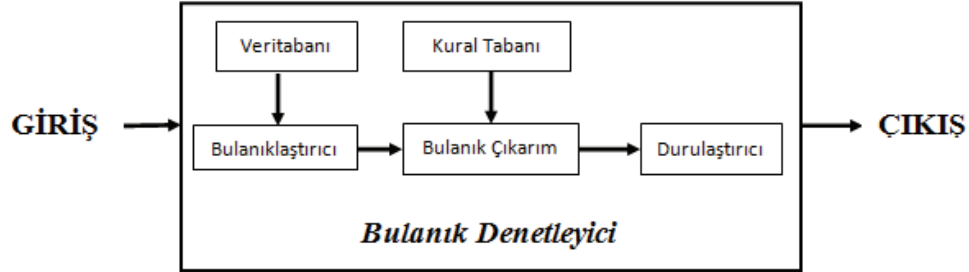
#### Bulanık Mantığın Dezavantajları

1. Kararlılık analiz işlemlerinin zorluğu
2. Üyelik fonksiyonlarının bulunmasının uzun zaman alabilmesi
3. Uygulamada ki kuralların oluşturulmasının bulanıklığa bağlı olması [27]

### **4.2. Bulanık Denetleyici**

Bulanık denetim sistemleri bulanık mantık metotları ile tanımlanan kural tabanlı sistemleridir.

Şekil 4.2.'de bulanık denetleyicisinin örnek şablonu görülmektedir. Bulanık denetleyici; giriş, veritabanı, kural tabanı, bulanıklaştırıcı, bulanık çıkarım, durulaştırıcı ve çıkış işlemlerinden oluşmaktadır.



Şekil 4.2. Bulanık denetleyici

**Giriş/Veri Tabanı:** Giriş değerlerini ve bu değerler hakkında tüm bilgileri içerir.

**Bulanıklaştırıcı:** Sisteme verilen keskin girişleri bulanıklaştırır. Ortaya çıkan veriler bulanık çıkarımda kullanılır.

**Bulanık kural tabanı:** Sistemin giriş ve çıkışlarını if ... then ... else ... şeklinde olan koşul cümleleri ile birbirine bağlar. Bu işlemin sonucunda durulaştırıcıya gönderilmek üzere bulanık çıkış elde edilir.

**Bulanık Çıkarım:** Giriş değerlerinin bulanıklaştırılması ve kuralların uygulanmasından sonra yapılan işlemdir. Her bir kuralın çıkarımlarını bir araya getirerek sistemin tüm girdilerine göre nasıl bir çıktı vereceğinin belirlenmesini sağlar.

**Durulaştırma:** Sistemin son elemanı olarak devreye girer. Bulanıklaştırılan giriş değerlerine kuralların hepsi uygulandıktan her giriş için bulanık bir çıkarım değeri oluşur. Bu oluşan bulanık değerlerin tekrar giriş değerleri gibi keskin değerler haline çevrilmesi işlemine durulaştırma denir.

**Çıkış:** Bulanık kural tabanlarının bulanık çıkarım vasıtasıyla işleme girmesi sonucuyla elde edilen çıktı grubuna denir [28], [29], [30].

### 4.3. Bulanık Mantık Üyelik Fonksiyonu ve Üyelik Derecesi

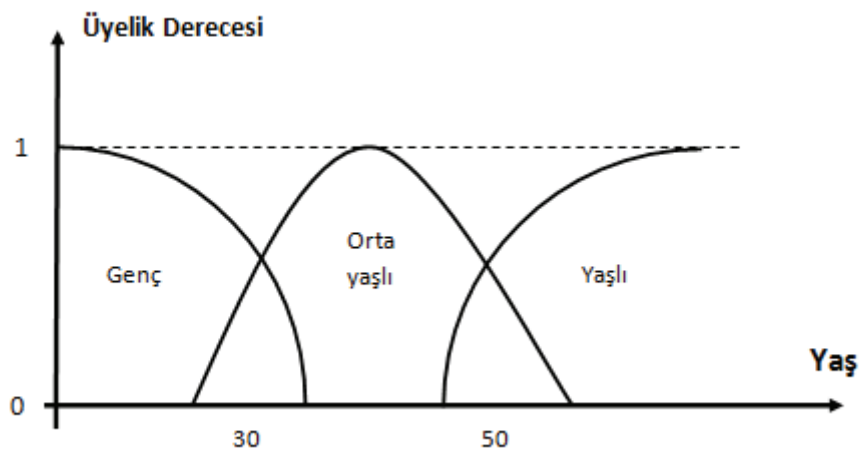
Bulanık kümelerin üyelik fonksiyonu, klasik kümelerde gösterge fonksiyonun bir genellemesidir. Bulanık mantıkta üyelik fonksiyonu bir değerlendirme uzantısı olarak gerçeğin derecesini temsil eder.

Üyelik fonksiyonları ilk kez Zadeh tarafından 1965 yılında tanıtıldı. Zadeh bu teorisinde 0 ile 1 aralığında değer alan üyelik fonksiyonlarının kullanılmasını önerdi.

Üyelik fonksiyonları her elemanına 0 ile 1 aralığında bir değer atar [31].

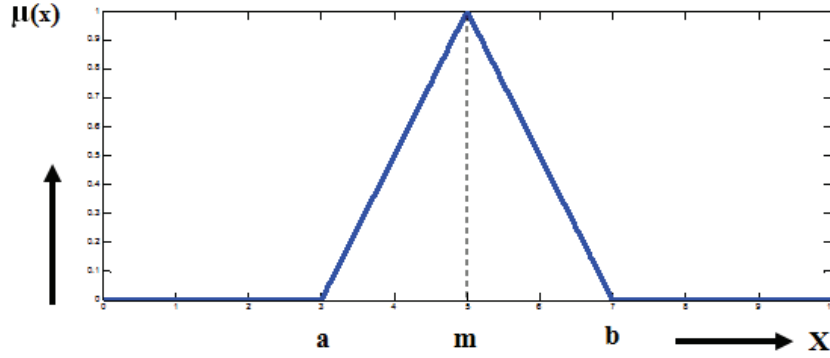
Üyelik fonksiyonu bulanık mantıkta ile temsil edilir.  $\mu_A(x)$  üyelik fonksiyonundaki bir  $x$  noktasının  $A$  bulanık kümesindeki üyelik derecesidir.  $\mu_A(x)=1$  ise  $x$ 'in  $A$  bulanık kümesinin kesin bir elemanı olduğu anlaşılır.  $\mu_A(x)=0$  olduğunda ise  $x$ 'in  $A$  bulanık kümesinin tamamen dışında olduğu anlaşılır.  $0<\mu_A(x)<1$  olduğu durumlarda ise her değer  $x$ 'in  $A$  bulanık kümesindeki belirsiz değeridir.

Şekil 4.3.'te bulanık mantık teorisine göre insan yaşının derecelendirilmesi görülmektedir [32].



Şekil 4.3. Bulanık küme teorisi

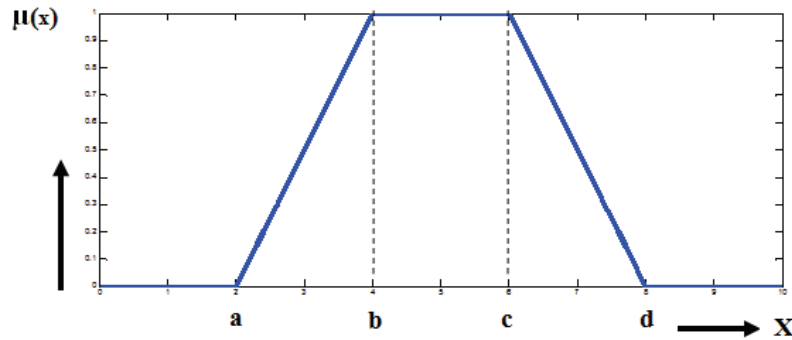
Triangular fonksiyon: Alt limit olan  $a$ , üst limit olan  $b$  ve  $m$  değerlerinde oluşur.  $a < m < b$  şartını sağlamaktadır. Şekil 4.4. triangular fonksiyonu göstermektedir. Denklem 4.1 ise triangular fonksiyonuna göre  $x$  noktasının üyelik derecesini belirtmektedir.



Şekil 4.4. Triangular fonksiyon gösterimi [33]

$$\begin{aligned} \mu(x) &= 0, & x \leq a \\ \mu(x) &= \frac{x-a}{m-a}, & a < x \leq m \\ \mu(x) &= \frac{b-x}{b-m}, & m < x \leq b \\ \mu(x) &= 0, & x \geq b \end{aligned} \quad (4.1)$$

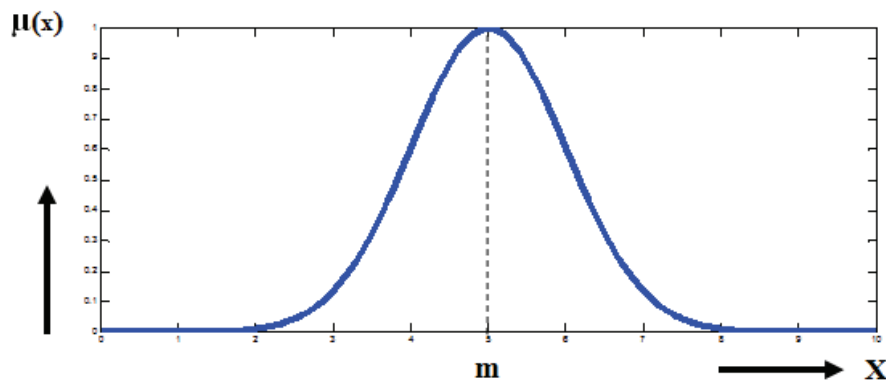
Trapezoidal fonksiyon: Alt limit olan  $a$ , üst limit olan  $d$ ,  $b$  ve  $c$  değerlerinde oluşur.  $a < b < c < d$  şartını sağlamaktadır. Şekil 4.5. trapezoidal fonksiyonu göstermektedir. Denklem 4.2 ise trapezoidal fonksiyonuna göre  $x$  noktasının üyelik derecesini belirtmektedir.



Şekil 4.5. Trapezoidal fonksiyon gösterimi [33]

$$\begin{aligned}
\mu(x) &= 0, & x < a \text{ or } x > d \\
\mu(x) &= \frac{x-a}{b-a}, & a \leq x \leq b \\
\mu(x) &= 1, & b \leq x \leq c \\
\mu(x) &= \frac{d-x}{d-c}, & c \leq x \leq d
\end{aligned} \tag{4.2}$$

Gaussian fonksiyon: Merkez değeri olan  $m$  ve standart sapma değeri olan  $k$  ile tanımlanır. Standart sapma değeri olan  $k > 0$  olmak zorundadır. Şekil 4.6. gaussian fonksiyonu göstermektedir. Denklem 4.3 ise gaussian fonksiyonuna göre  $x$  noktasının üyelik derecesini belirtmektedir.



Şekil 4.6. Gaussian fonksiyon gösterimi [33]

$$\mu(x) = e^{-\frac{(x-m)^2}{2k^2}} \tag{4.3}$$

### 4.3. Sugeno Bulanık Modeli

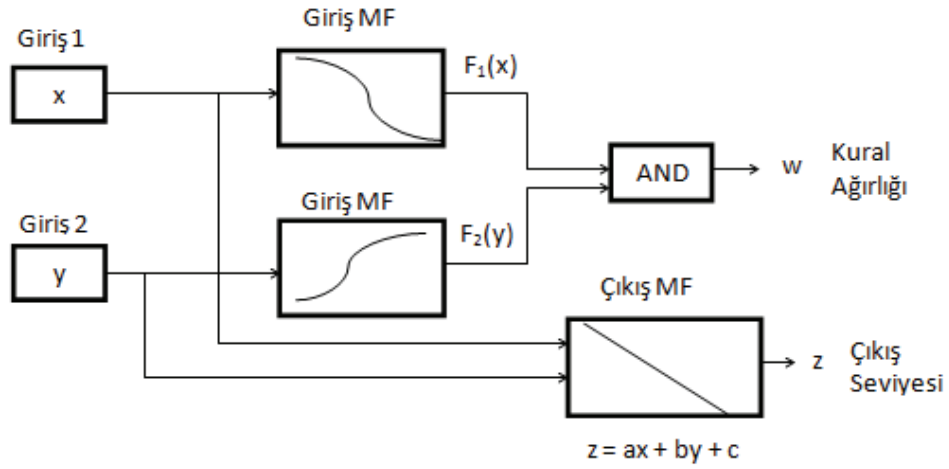
Sugeno bulanık mantık 1985 yılında tanıtılmıştır. Mamdani bulanık mantık yöntemine bir çok açıdan oldukça benzemektedir. Bulanık çıkarım işlemlerinin ilk ikisi olan girdi değişkenlerinin bulanıklaştırılması ve bulanık mantık işlemleri Mamdani bulanık modellemeden farksızdır. İki yöntem arasındaki en temel fark çıktı

üyelik fonksiyonlarıdır. Sugeno bulanık modellemenin çıkış üyelik fonksiyonları ya doğrusaldır ya da sabittir [34].

Eğer Giriş 1 = x ve Giriş 2 = y ise Çıkış = ax + by + c şeklinde ifade edilir. Her bir kuralın kendi çıkış seviye ağırlığı vardır. Sistemin çıkışı tüm kuralların ağırlıklı ortalamasıdır. Denklem 4.4 ile sistemin çıkışı hesaplanır.

$$\text{Çıkış} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (4.4)$$

Denklemdeki N değeri kural sayısını, w değeri kural ağırlığını ve z değeri ise çıkış seviyesini göstermektedir. Şekil 4.7. sugeno bulanık modelinin çalışma şeklini göstermektedir.



Şekil 4.7. Sugeno bulanık modelinin diyagramı [34]

## **BÖLÜM 5. AKILLI DEPREM TESPİT SİSTEMİ**

### **5.1. Akıllı Telefon ile Verilerin Toplanması**

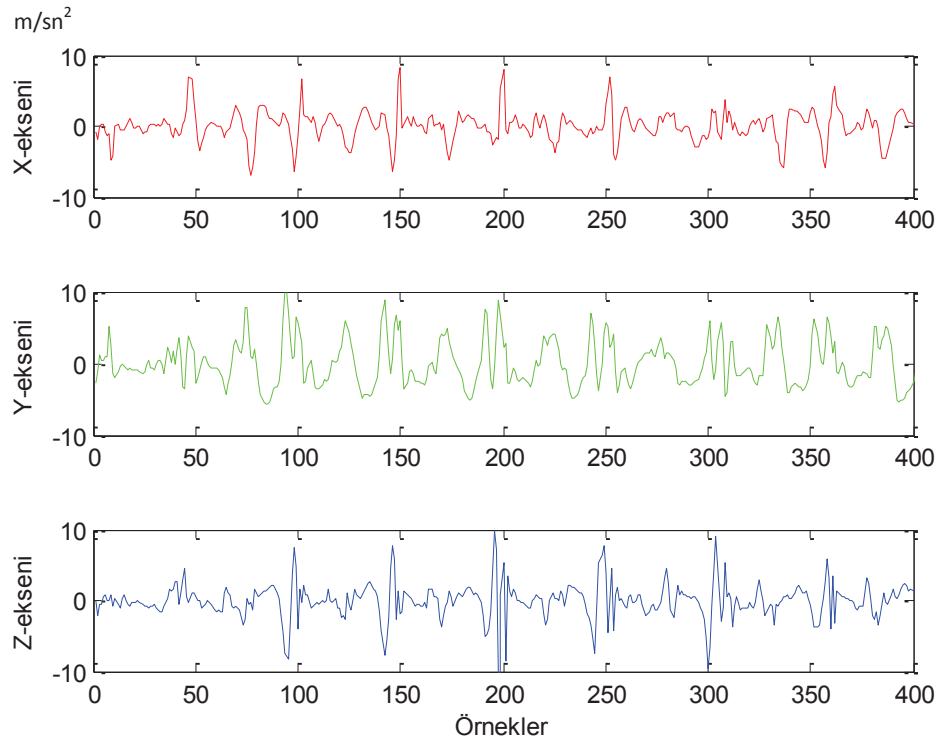
Bu uygulamada ilk olarak akıllı telefonlardaki ivme ölçer sensörünün kullanıldığı bir android programı geliştirilmiştir. Bu program telefonda bulunan ivmeölçer sensörlerini kullanarak cihazın ivmelenmesini ölçüp kayıt etmektedir. Android programlamada sensörlerin örnekleme (örnek/ms) duyarlılığını gösteren Normal, Fastest, Game, UI olmak üzere 4 çeşit sensör duyarlılığı modu bulunmaktadır. Bu programda sensör duyarlılığı game modu olarak seçilmiştir. Game modunda ivme ölçer sensörü yaklaşık 35-40 ms'de bir örnek almaktadır. Bu android programı ile insan hareketleri (koşma, yürüme, merdiven inip-çıkma, diğer,..) sonucu oluşan çeşitli veriler toplanmıştır. Bu verilere ek olarak deprem sarsma tablasından da gerçek deprem verileri toplanmıştır. Veriler Sony Xperia E4 adlı E2105 model numaralı akıllı telefon ile toplanmıştır. Her bir veri X eksenindeki ivmelenmeden, Y eksenindeki ivmelenmeden, Z eksenindeki ivmelenmeden, aktivite türünden ve başlama zamanından oluşmaktadır. Verilerin toplanması işlemi değişik sürelerde gerçekleştirilmiştir. Bu veriler sadece bir kişi tarafından toplanmıştır. Elcentro ve sakarya depremi sinyalleri Sakarya Üniversitesi İnşaat Mühendisliğine ait deprem sarsma tablası ile ölçülmüştür. Elcentro ve sakarya olmak üzere 2 adet deprem kaydı, koşma, yürüme, merdiven inme ve merdiven çıkma olmak üzere 4 adet insan aktivitesi ve 1 adet sabit zemin kaydı bulunmaktadır.

Şekil 5.1. veritabanında kayıtlı veri örneklerini, Şekil 5.2. merdiven inme sinyalini, Şekil 5.3. sabit zemin sinyalini, Şekil 5.4. yürüme sinyalini, Şekil 5.5. ve Şekil 5.6. ise referans cihazı ile akıllı telefonun deprem sinyalleri üzerindeki karşılaştırmasını göstermektedir.

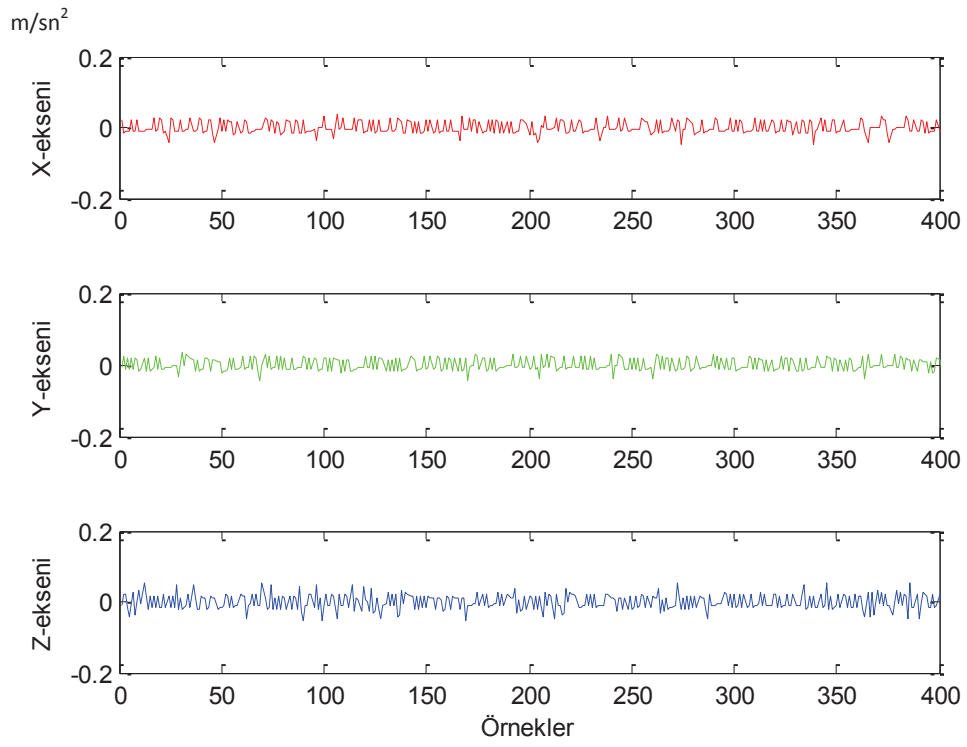


X eksen	Y eksen	Z eksen	Baş. Zamanı	Akt. Türü
2811	-1.45	-0.049	1.151	17.11.2015 21:47 yurume1
2812	-0.241	0.696	2.606	17.11.2015 21:47 yurume1
2813	0.359	0.557	1.962	17.11.2015 21:47 yurume1
2814	0.501	0.354	0.037	17.11.2015 21:47 yurume1
2815	-0.211	0.528	-0.92	17.11.2015 21:47 yurume1
2816	0.014	0.33	-0.92	17.11.2015 21:47 yurume1
2817	-0.019	-0.747	-0.398	17.11.2015 21:47 yurume1
2818	-0.23	-2.253	-0.105	17.11.2015 21:47 yurume1
2819	-0.276	-2.537	0.222	17.11.2015 21:47 yurume1
2820	-0.006	-2.643	-0.19	17.11.2015 21:47 yurume1
2821	0.118	-2.421	-0.182	17.11.2015 21:47 yurume1
2822	0.186	-1.692	0.068	17.11.2015 21:47 yurume1
2823	-0.066	-1.506	0.116	17.11.2015 21:47 yurume1
2824	-0.635	-1.695	-0.121	17.11.2015 21:47 yurume1
2825	-1.029	-2.275	-0.556	17.11.2015 21:47 yurume1
2826	-1.037	-2.679	-0.843	17.11.2015 21:47 yurume1
2827	-0.768	-2.664	-1.012	17.11.2015 21:47 yurume1
2828	-0.4	-1.212	-1.453	17.11.2015 21:47 yurume1
2829	-0.106	-0.203	-2.02	17.11.2015 21:47 yurume1
2830	0.344	0.665	-3.179	17.11.2015 21:47 yurume1
2831	0.551	1.727	-4.719	17.11.2015 21:47 yurume1
2832	0.686	3.466	-5.247	17.11.2015 21:47 yurume1
2833	1.835	5.224	-3.401	17.11.2015 21:47 yurume1
2834	2.418	4.088	-0.238	17.11.2015 21:47 yurume1
2835	0.862	-0.224	-0.957	17.11.2015 21:47 yurume1

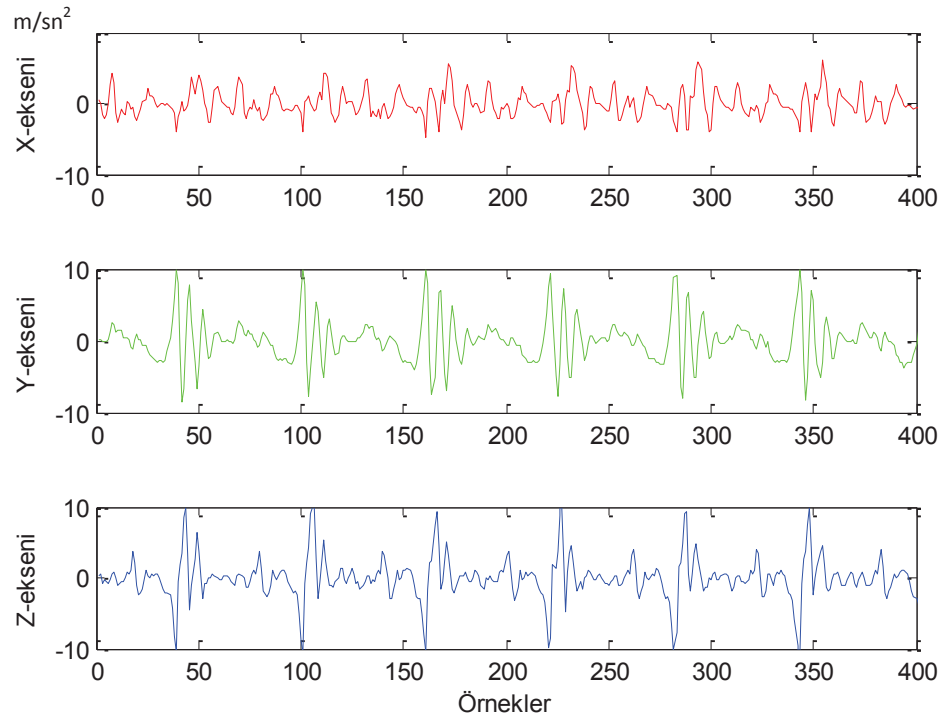
Şekil 5.1. Veritabanına kaydedilen veri örnekleri



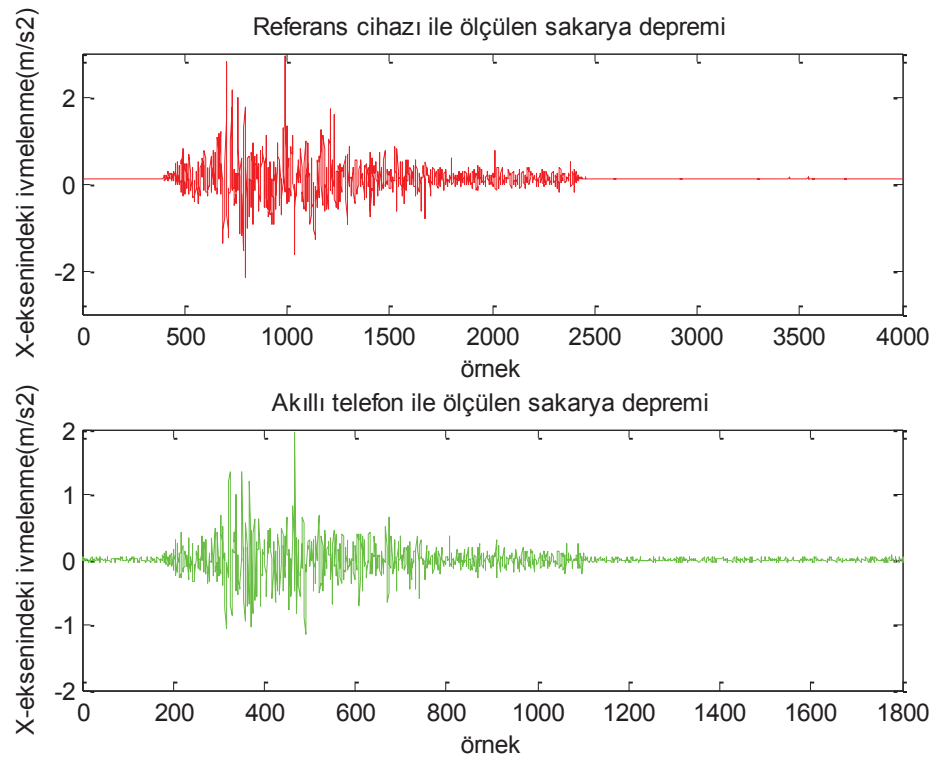
Şekil 5.2. Merdiven inme sinyali



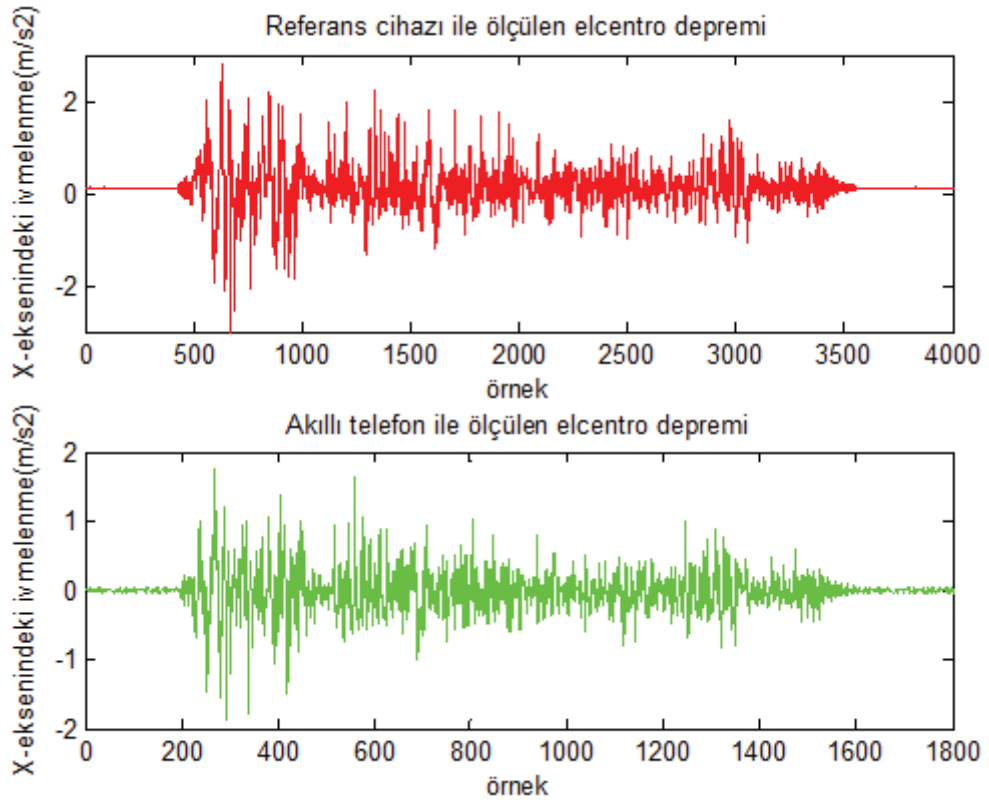
Şekil 5.3. Sabit zemin sinyali



Şekil 5.4. Yürüme sinyali



Şekil 5.5. Referans cihazı ile akıllı telefon karşılaştırması (sakarya depremi)



Şekil 5.6. Referans cihazı ile akıllı telefon karşılaştırması (elcentro depremi)

## 5.2. Özellik Çıkartılması

### 5.2.1. AR model

Bu çalışmada, akıllı cihazlar yardımıyla deprem ve insan hareketleri birbirinden ayrılmaya çalışılmıştır. Akıllı cihazlar üzerindeki üç eksenli ivmeölçer algılayıcılarla zaman serisi işaretler ölçülmüş ve ölçülen ivmelenme verisi 4. derece AR model kullanılarak modellenmiştir. Elde edilen parametreler eylemin insan ya da deprem kaynaklı olup olmadığını sınıflandıran bulanık sistemin giriş parametreleri olarak kullanılmıştır.

İkinci derece durağan işaretlerin modellenmesinde AR filtreler yaygın olarak kullanılmaktadır. Ar model derecesi işaretin o andaki değerini tahmin etmek için kendi geçmiş örnek değerlerinden ne kadar kullanılması gerektiğini belirler.

Normalize edilmiş AR katsayıları bulanık sistemin giriş parametreleri olarak kullanılmıştır. Kısa süreli hızlanma verileri durağan rastgele sinyallere iyi bir örnek olduğundan sinyallerin tanımlanmasında AR model kullanılması oldukça uygundur. Ar model Denklem 5.1 ile ifade edilir.  $x(n)$  random signal'ı,  $p$  ise dereceyi ifade eder.

$$x(n) = - \sum_{k=1}^p a(k) * x(n - k) + u(n) \quad (5.1)$$

### 5.2.2. Verilerin işlenmesi

İnsan aktivitesi ivmelenme verileri 3 eksenlidir.  $x$ ,  $y$  ve  $z$  eksenlerinde kayıt alınmıştır. Deprem ivmelenme verileri ise tek eksenlidir. Sadece  $x$  ekseninde kayıt alınmıştır. Yönelim problemini ortadan kaldırmak için insan aktivite verilerinin genlikleri hesaplanmıştır. Hesaplanan genliklere göre işleme alınmıştır. Deprem

verileri tek ekseninde kayıt edildiği için bu verilerinde mutlak değerleri işleme alınmıştır.

Kararlaştırılan kaydırma aralığına göre genlikleri hesaplanan veriler belirli uzunluğa sahip olan pencerelere ayrılmıştır. Pencere uzunluğunun sistem performansına etkisini ölçmek için çeşitli deneyler yapılmıştır. 5, 20, 50, 100, 200 değerlerinin seçildiği çalışmalarda 50 veya 50'den büyük bir değer seçilmesi öngörülmüştür. Çünkü sistem; 5 değeri için %71, 20 değeri için %96 doğruluk oranı ile çalışmıştır. 50, 100 ve 200 değerleri için ise sistem %100 doğruluk oranı ile çalışmaktadır. Bu değer azaltılırsa sistemin performansı da azalmaktadır. Bu çalışmada pencere uzunluğu 200 örnek olarak seçilmiştir. Kaydırma aralığı olarak ise 20 örnek seçilmiştir. Yani her bir adımda 200 veri işleme sokulmaktadır. İlk olarak ilk 200 örnek işleme alınıp ilk pencere oluşturulmuştur. Daha sonra 20 örnek atlanarak 21.-221. örnekler işleme alınıp ikinci pencere oluşturulmuştur. Daha sonra ise 41.-241. örnekler işleme alınmıştır. Bu şekilde tüm örnekler pencereleme işlemi uygulanarak Ar katsayıları çıkartılmıştır.

Elcentro depremi sinyali 1400 örnekten oluşmaktadır. 200 örnekten oluşan 60 adet pencereye ayrılmıştır. 50 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Sakarya depremi sinyali 1000 örnekten oluşmaktadır. 200 örnekten oluşan 40 adet pencereye ayrılmıştır. 30 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Koşma sinyali 2000 örnekten oluşmaktadır. 200 örnekten oluşan 90 adet pencereye ayrılmıştır. 80 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Sabit sinyali 2000 örnekten oluşmaktadır. 200 örnekten oluşan 90 adet pencereye ayrılmıştır. 80 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Yürüme sinyali 2000 örnekten oluşmaktadır. 200 örnekten oluşan 90 adet pencereye ayrılmıştır. 80 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Merdiven çıkma sinyali 2000 örnekten oluşmaktadır. 200 örnekten oluşan 90 adet pencereye ayrılmıştır. 80 tanesi eğitim için 10 tanesi test için ayrılmıştır.

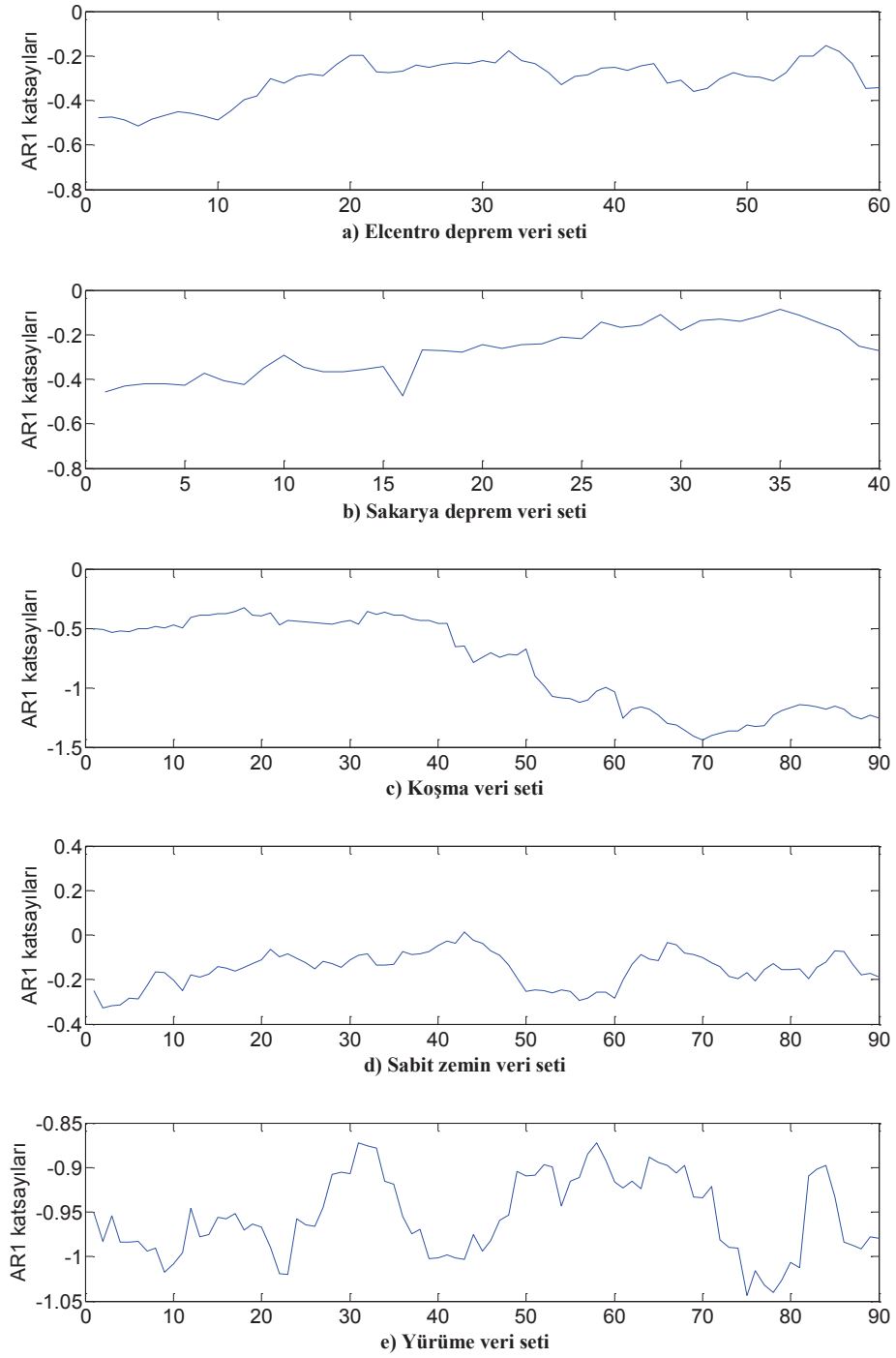
Merdiven inme sinyali 2000 örnekten oluşmaktadır. 200 örnekten oluşan 90 adet pencereye ayrılmıştır. 80 tanesi eğitim için 10 tanesi test için ayrılmıştır.

Özellik çıkarma işleminin gerçekleştirilmesi için 4.derece Autoregressive model kullanılmıştır. AR model bu pencerelerin her birine uygulanmıştır. Matlab'deki arburg fonksiyonu ile bu işlem gerçekleştirilmiştir. Bu fonksiyon  $a=arburg(x,p)$  olarak ifade edilebilir.  $x$  parametresi 200 örnekten oluşan bir pencereyi;  $p$  parametresi ise sıra numarasını göstermektedir. 2. derece, 3. derece, 4. derece ve 5. derece gibi farklı derecelerdeki AR model ile yapılan sonuçlar karşılaştırıldığında sistem doğruluk performansının değişmediği fakat 4. derece AR model ile sonuçların +1 ve -1 değerlerine daha yakın çıktığı görülmüştür. Bu sebepten dolayı AR model derecesi 4 olarak alınmıştır.

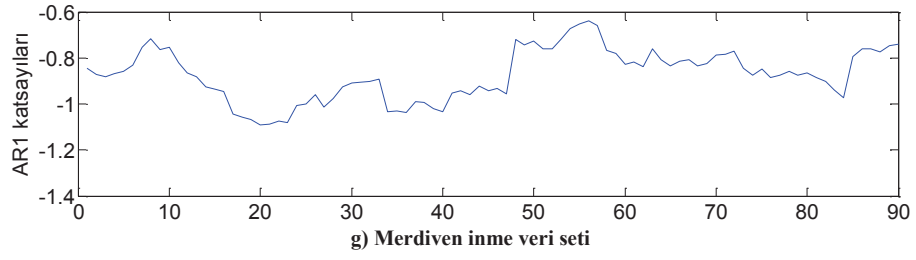
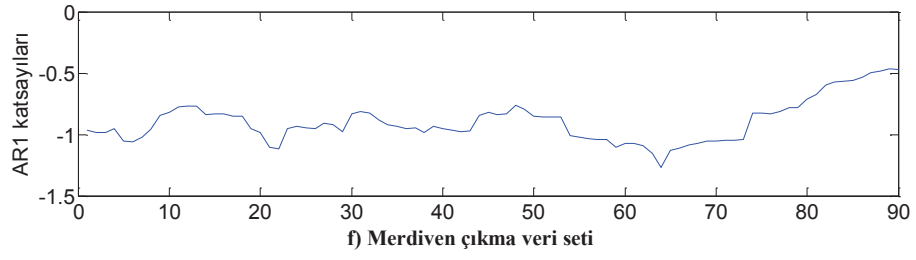
Bu pencerelere Ar model uygulanmış ve toplam 550 adet Ar katsayılarından oluşan verimiz oluşmuştur. Bu verilerin 480'i eğitim için 70'i ise test amaçlı kullanılmıştır.

Ar katsayılarından oluşan veriler karar kısmında bulunan bulanık sistemin giriş parametrelerini oluşturmaktadır. Bu matrisin her bir satırı 200 örneklilik ivmelenme verisinden oluşan bir pencereyi ifade etmektedir. Çıkış verileri deprem verisi için pozitif (+1) insan aktivitesi için negatif (-1) olacak şekilde seçilmiştir.

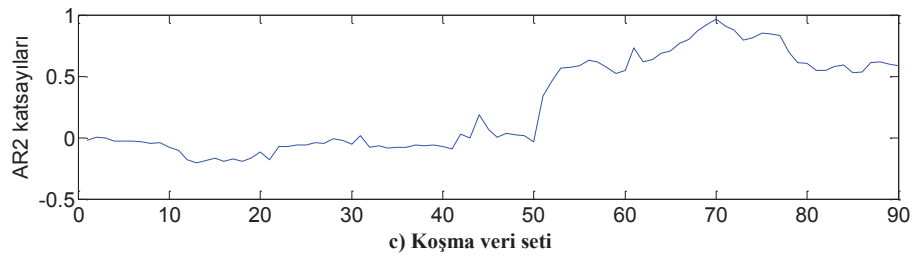
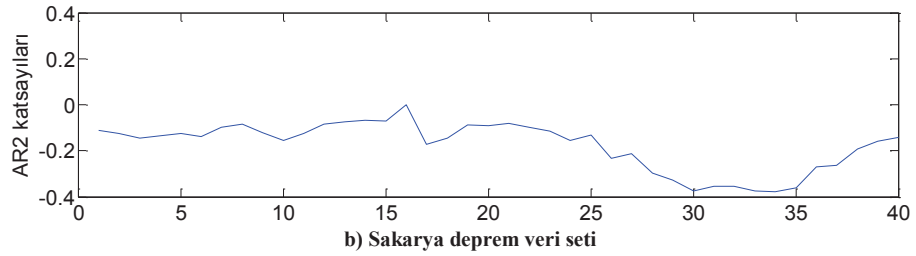
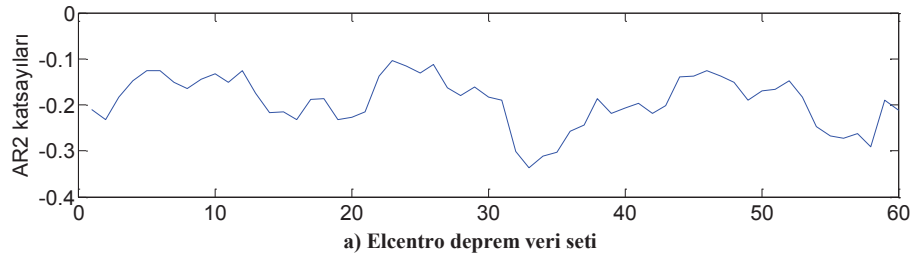
Şekil 5.7. AR1 katsayılarını, şekil 5.8. AR2 katsayılarını, Şekil 5.9. AR3 katsayılarını ve Şekil 5.10. ise AR4 katsayılarını göstermektedir.



Şekil 5.7. Veri setlerinin AR1 katsayıları a) Elcentro deprem veri seti b) Sakarya deprem veri seti c) Koşma veri seti d) Sabit zemin veri seti e) Yürüme veri seti f) Merdiven çıkma veri seti g) Merdiven inme veri seti

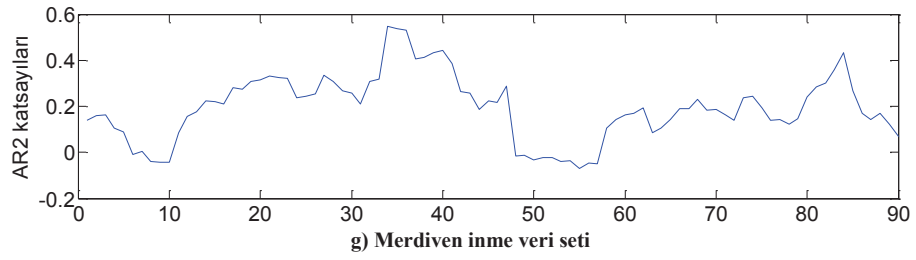
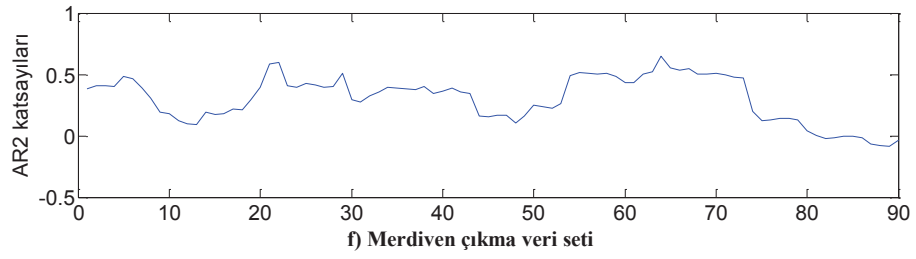
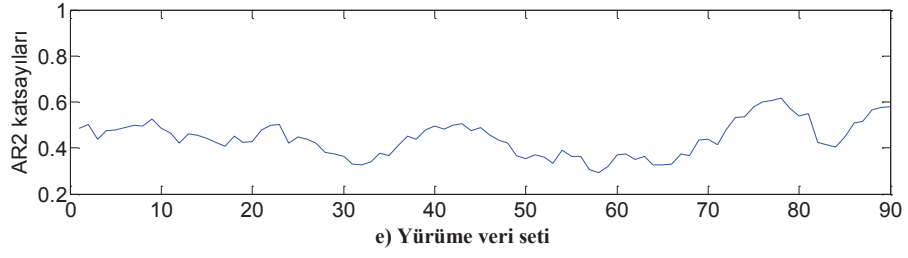
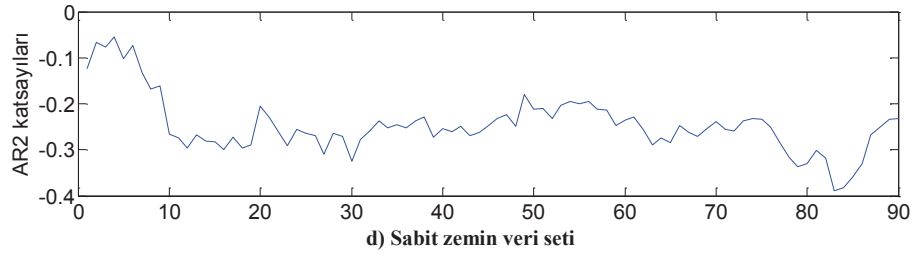


Şekil 5.7. (Devamı)

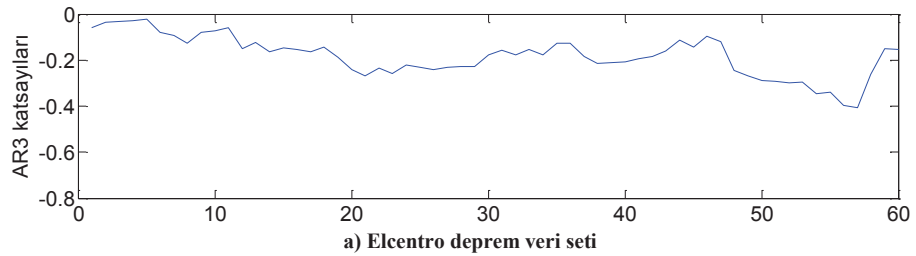


Şekil 5.8. Veri setlerinin AR2 katsayıları a) Elcentro deprem veri seti b) Sakarya deprem veri seti c) Koşma veri seti d) Sabit zemin veri seti e) Yürüme veri seti f) Merdiven çıkma veri seti g) Merdiven inme veri seti

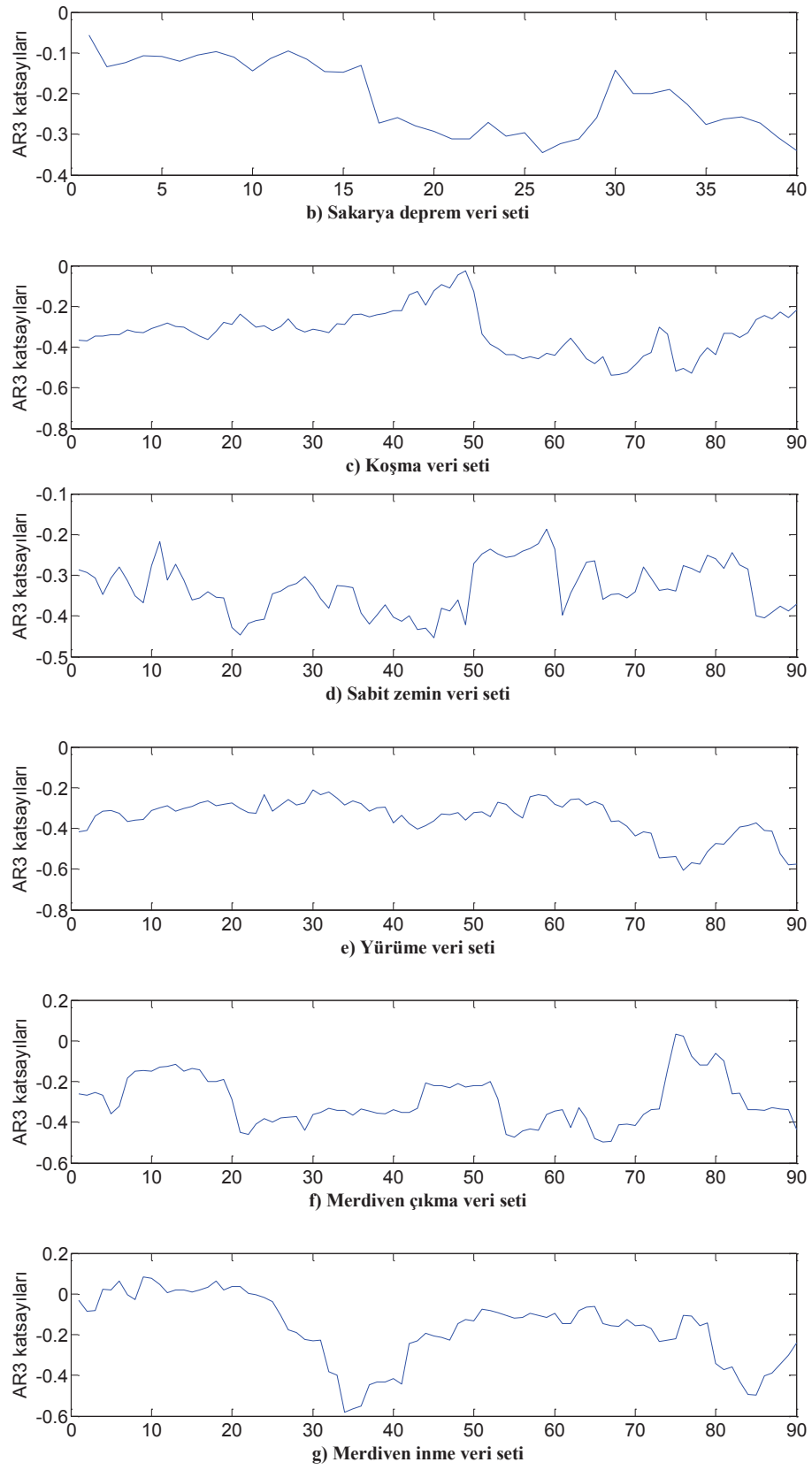




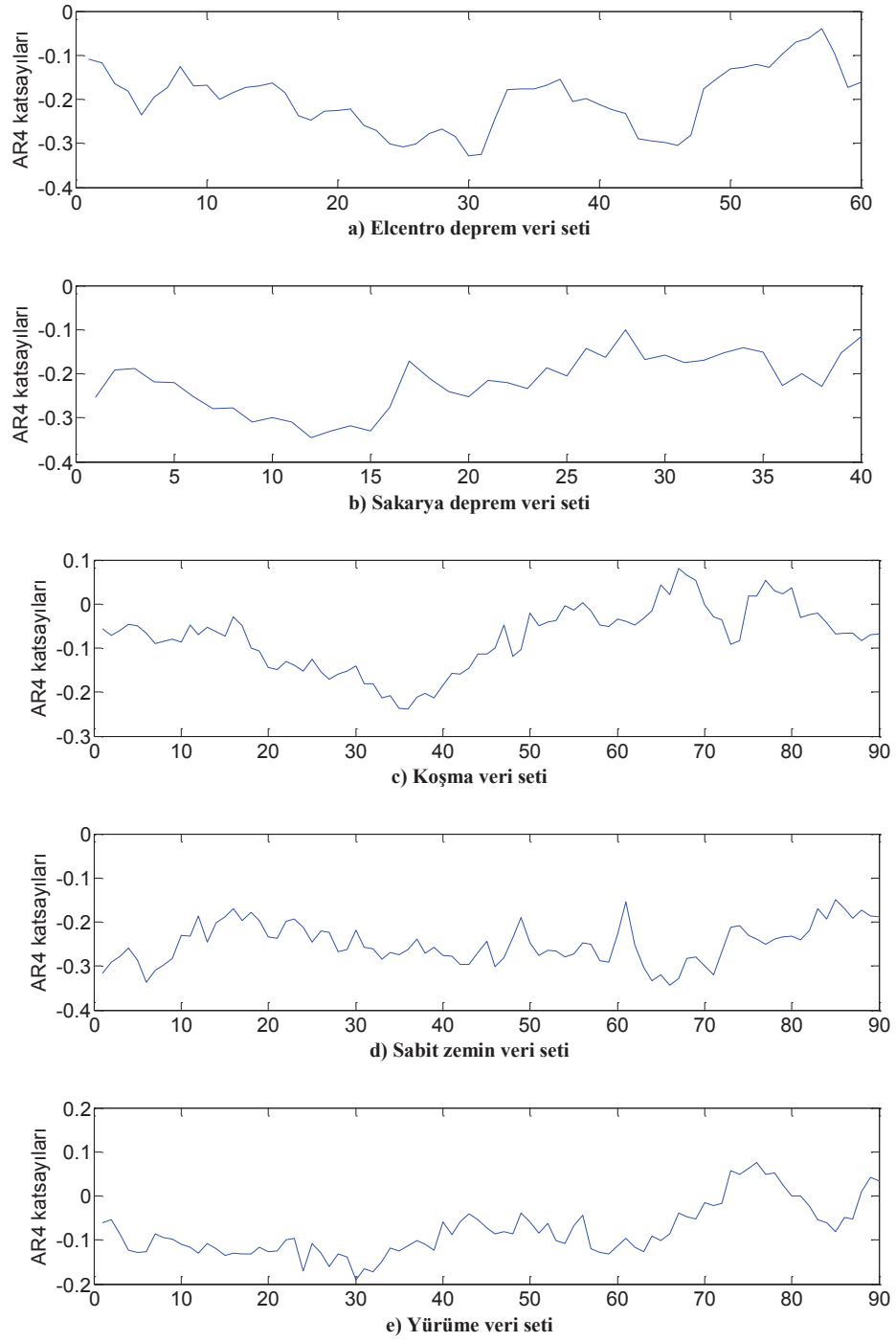
Şekil 5.8. (Devamı)



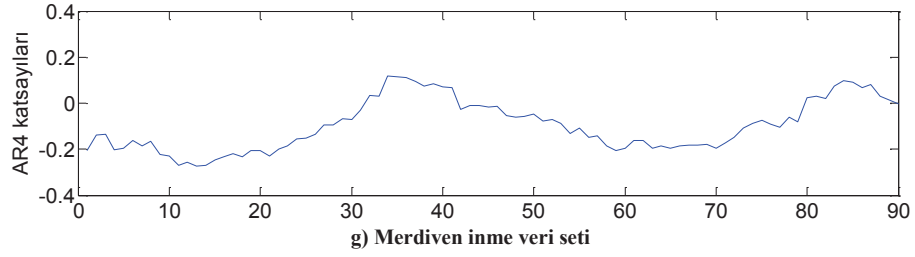
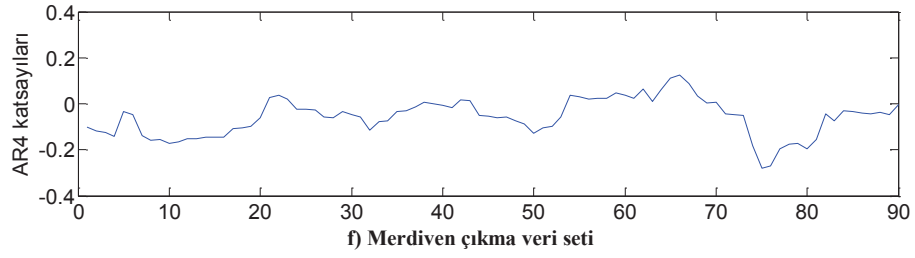
Şekil 5.9. Veri setlerinin AR3 katsayıları a) Elcentro deprem veri seti b) Sakarya deprem veri seti c) Koşma veri seti d) Sabit zemin veri seti e) Yürüme veri seti f) Merdiven çıkma veri seti g) Merdiven inme veri seti



Şekil 5.9. (Devamı)



Şekil 5.10. Veri setlerinin AR3 katsayıları a) Elcentro deprem veri seti b) Sakarya deprem veri seti c) Koşma veri seti d) Sabit zemin veri seti e) Yürüme veri seti f) Merdiven çıkma veri seti g) Merdiven inme veri seti



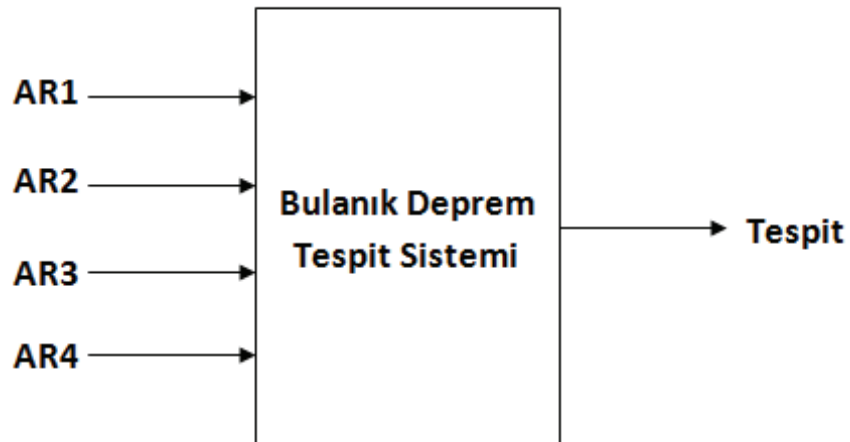
Şekil 5.10. (Devamı)

### 5.3. Bulanık Deprem Tespit Sistemi Gerçeklenmesi

İlk olarak verilerin deprem mi yoksa insan kaynaklı hareketler mi olduğuna karar verecek olan bulanık sistem geliştirilmeye çalışılmıştır.

#### 5.3.1. Bulanık deprem tespit sistemi

Bu sistemin AR katsayılarından oluşan 4 adet girişi ve 1 adet çıkışı bulunmaktadır. Şekil 5.11. tasarlanan bulanık deprem tespit sistemini göstermektedir.



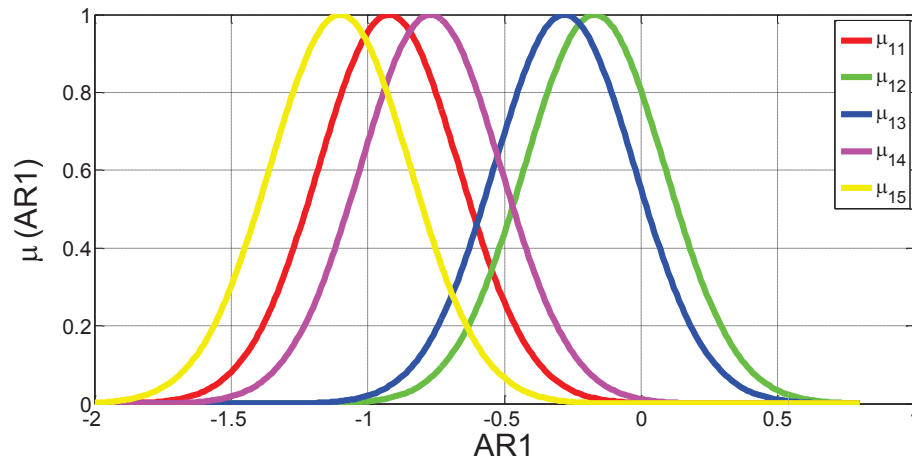
Şekil 5.11. Bulanık deprem tespit sistemi

Matlab fuzzy logic araç kutusunda bulunan genfis 2 fonksiyonu kullanılmıştır. Sınıflandırma algoritması için 0,5 değeri kullanılmıştır.

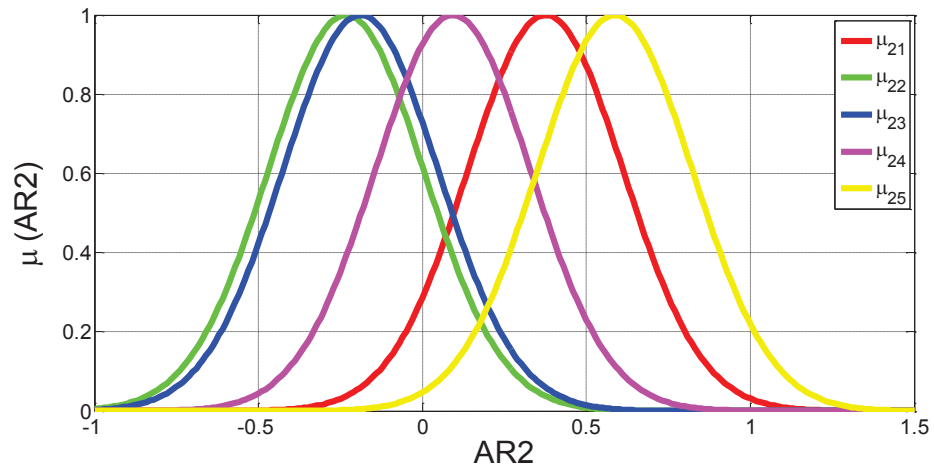
genfis 2 sugeno tipi FIS (bulanık çıkarım sistemi) yapısı oluşturur. Eksiltici kümelemeyi kullanır. Giriş argümanı olarak ayrı giriş ve çıkış verilerine sahiptir. Sadece bir çıkış olduğunda; başlangıç FIS yapısını oluşturmak için kullanılır. Bir dizi kural çıkartarak veri davranış modeli gerçekleştirir. Kural çıkarma metodu kural sayısını ve önceki üyelik fonksiyonlarını belirlemek için eksiltici fonksiyonu kullanır. Daha sonra her kuralın sonucunu belirlemek için linear least squares estimation yöntemini kullanır. Bu fonksiyon bir dizi bulanık kural içeren FIS yapısı geri döndürür. Giriş üyelik fonksiyonu tipi gaussmf ve çıkış üyelik fonksiyonu tipi doğrusaldır [35].

### 5.3.2. Sistemin üyelik fonksiyonları

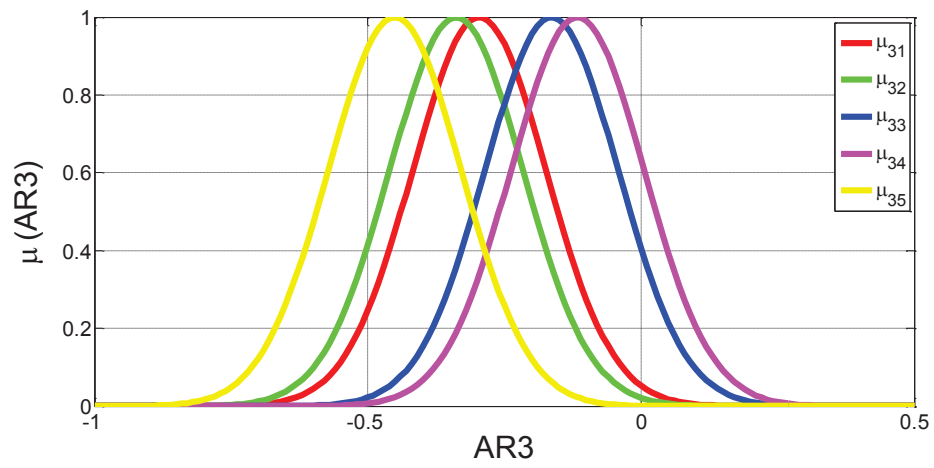
Şekil 5.12. tasarlanan sistemin üyelik fonksiyonları göstermektedir.



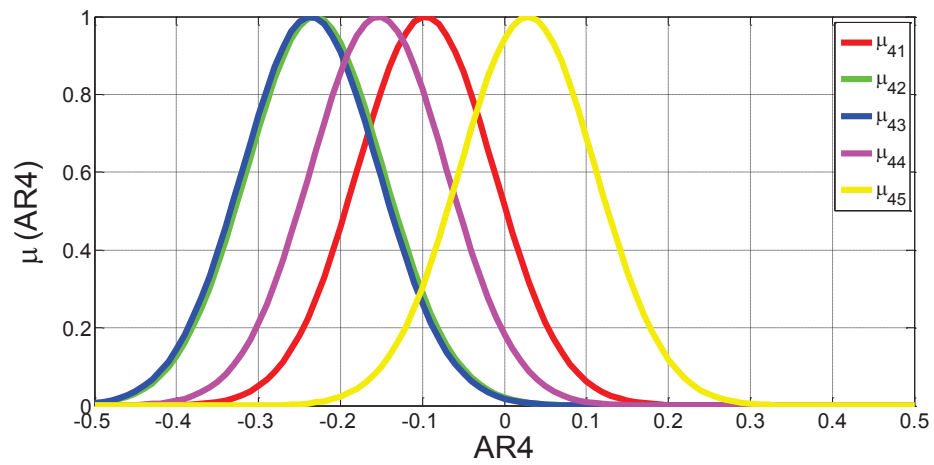
Şekil 5.12. Sistemin üyelik fonksiyonları a) Sistemin 1. girişinin üyelik fonksiyonları b) Sistemin 2. girişinin üyelik fonksiyonları c) Sistemin 3. girişinin üyelik fonksiyonları d) Sistemin 4. girişinin üyelik fonksiyonları



b) Sistemin 2. girişinin üyelik fonksiyonları



c) Sistemin 3. girişinin üyelik fonksiyonları



d) Sistemin 4. girişinin üyelik fonksiyonları

Şekil 5.12. (Devamı)

### 5.3.3. Sistemin kural tabanı

Tablo 5.1. ise sistemin kural tabanını göstermektedir. Her bir girişin üyelik fonksiyonu diğer girişlerin üyelik fonksiyonu ile sisteme girerek bir çıkış üretmektedir.

Tablo 5.1. Sistemin kural tabanı

i									
1	if	ar1 $\mu_{11}$	and	ar2 $\mu_{21}$	and	ar3 $\mu_{31}$	and	ar4 $\mu_{41}$	then out <sub>1</sub>
2	if	ar1 $\mu_{12}$	and	ar2 $\mu_{22}$	and	ar3 $\mu_{32}$	and	ar4 $\mu_{42}$	then out <sub>2</sub>
3	if	ar1 $\mu_{13}$	and	ar2 $\mu_{23}$	and	ar3 $\mu_{33}$	and	ar4 $\mu_{43}$	then out <sub>3</sub>
4	if	ar1 $\mu_{14}$	and	ar2 $\mu_{24}$	and	ar3 $\mu_{34}$	and	ar4 $\mu_{44}$	then out <sub>4</sub>
5	if	ar1 $\mu_{15}$	and	ar2 $\mu_{25}$	and	ar3 $\mu_{35}$	and	ar4 $\mu_{45}$	then out <sub>5</sub>

### 5.3.4. Sistem çıkışı

Geliştirilen sistem eğitildikten sonra Tablo 5.2.'deki kural çıkış fonksiyonlarına ait olan katsayılara ulaşılmıştır.

Tablo 5.2. Sistemin çıkış fonksiyonları

out <sub>1</sub> =	-4.842*ar <sub>1</sub> - 5.058*ar <sub>2</sub> - 4.676*ar <sub>3</sub> - 5.001*ar <sub>4</sub> - 5.433
out <sub>2</sub> =	21.25*ar <sub>1</sub> + 21.38*ar <sub>2</sub> + 23.34*ar <sub>3</sub> + 17.98*ar <sub>4</sub> + 20.09
out <sub>3</sub> =	22.23*ar <sub>1</sub> + 21*ar <sub>2</sub> + 28.18*ar <sub>3</sub> + 19.58*ar <sub>4</sub> + 20.63
out <sub>4</sub> =	3.409*ar <sub>1</sub> + 3.33*ar <sub>2</sub> + 3.454*ar <sub>3</sub> + 2.901*ar <sub>4</sub> + 2.1
out <sub>5</sub> =	1.93*ar <sub>1</sub> + 1.995*ar <sub>2</sub> + 1.925*ar <sub>3</sub> + 1.783*ar <sub>4</sub> + 0.801

i kural numarası olduğu durumda her bir kuralın ağırlığı Denklem 5.2 kullanılarak hesaplanabilmektedir.

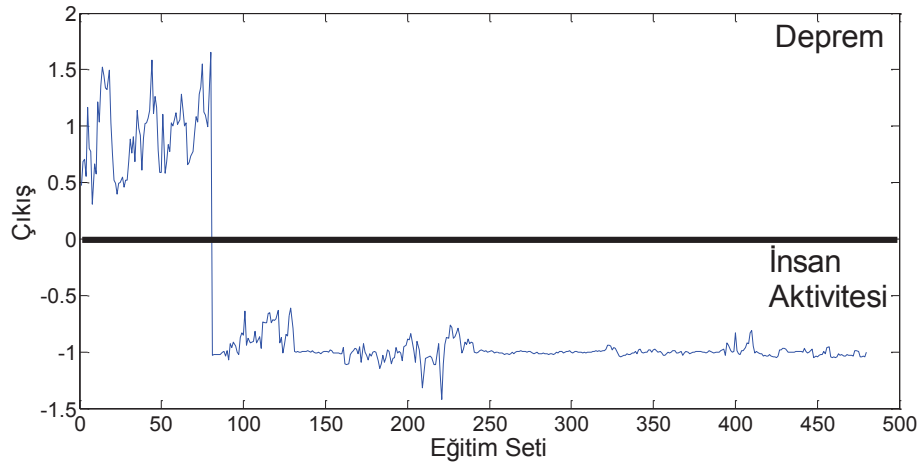
$$W_i = \mu_{1i}(AR1) \cdot \mu_{2i}(AR2) \cdot \mu_{3i}(AR3) \cdot \mu_{4i}(AR4) \quad (5.2)$$

Sistemde 5 adet kural bulunmaktadır. Denklem 5.3 ise sistemin çıkışını göstermektedir.

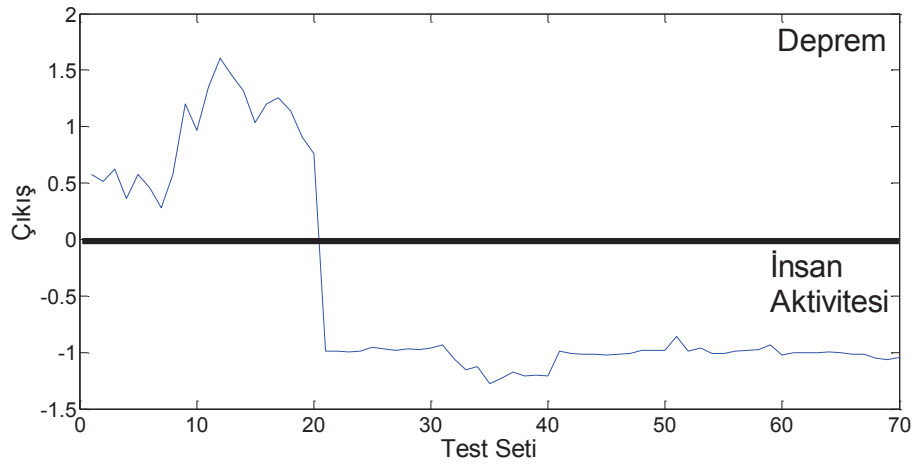
$$\text{Çıkış} = \frac{\sum_{i=1}^N W_i \text{out}_i}{\sum_{i=1}^N W_i} \quad (5.3)$$

Deprem kaynaklı olan verilerin çıkış değerleri pozitif (+1); insan hareketlerinden kaynaklanan verilerin çıkış değerleri negatif (-1) olacak şekilde değerlendirilmiştir.

Şekil 5.13. eğitim setinin çıkış değerlerini Şekil 5.14. test setinin çıkış değerlerini göstermektedir.



Şekil 5.13. Eğitim setinin çıkış değerleri



Şekil 5.14. Test setinin çıkış değerleri

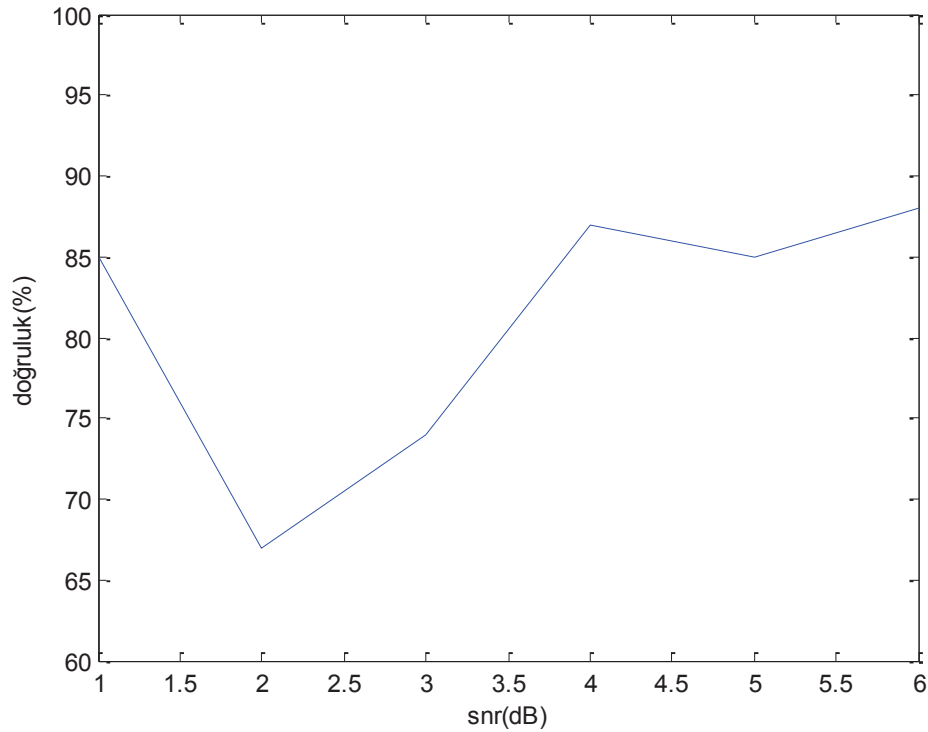


#### 5.4. Hassasiyet Analizi

Kaydedilen verilere awgn (beyaz gürültüsü) fonksiyonu ile gürültü eklenmiştir.  $y = \text{awgn}(x, \text{snr})$  fonksiyonunda bulunan  $x$  parametresi gürültü eklenecek sinyali;  $\text{snr}$  parametresi ise örnek başına sinyal-gürültü oranını belirtir.  $\text{snr}$  değeri Denklem 5.4'te gösterilen şekilde hesaplanmaktadır.

$$SNR_{dB} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (5.4)$$

Sinyal gücünün gürültü gücüne oranına  $\text{snr}$  denir.  $\text{snr}$  değerinin birimi desibeldir. Hassasiyet analizinde farklı dB'deki gürültü eklenmiş ivmelenme verileri kullanılarak sistemin doğruluğu ölçülmüştür. Tespit işleminin doğruluk oranı  $\text{snr}$  değeri ile birlikte değişmektedir. Farklı  $\text{snr}$  değeri ile sonuçlar gözlenmiştir. Şekil 5.15.'de farklı  $\text{snr}$  değerlerine göre sistemin doğruluk performansı görülmektedir.



Şekil 5.15.  $\text{snr}$  değerine göre sistemin doğruluk performansı

## BÖLÜM 6. SONUÇ

Test verileri ile yapılan deneylerde sistem gerçek deprem kaynaklı olan verilerin tümünü deprem olarak; insan hareketlerinden kaynaklanan verilerin tümünü de insan aktivitesi olarak algılamıştır. Böylece akıllı telefonlardaki ivmeölçer sensörleriyle deprem kaynaklı ve insan kaynaklı ivmelenmelerin tespit işleminin yapılabileceği görülmüştür.

Bu çalışma daha sonra yapılması planlanan bir projeye ışık tutmaktadır. Geliştirilecek uygulamanın amacı büyük sarsıntuların akıllı cihazlar ile tespit edilmesidir. Bir erken uyarı sistemi olmasından daha ziyade amaç büyük sarsıntı tespiti ile göçük altında kalma riski olan kişilerin daha sonra geliştirilecek devam uygulamaları ile konum ve derinlik/yükseklik bilgisinin arama kurtarma ekiplerine ulaştıracak bir büyük projeye zemin hazırlamaktır. Geliştireceğimiz uygulama internet omurgasından bağımsız olarak çalışacak ve akıllı cihazlar içerisindeki ivmeölçerleri kullanılacaktır. Sarsıntı tespiti tek bir cihaz ile değil bir grup cihazın aralarında haberleşerek ortaklaşa karar vermesi şeklinde olacaktır. Birbirlerinin yakınında olan akıllı cihazlar belirli bir sapma payı ile benzeri sarsıntı hissettiklerinde birbirlerine haber verecek ve bir karar destek sistemi ile büyük bir sarsıntının olup olmadığına karar vereceklerdir. Böylece kendilerine acil durum moduna alarak konum, derinlik, sağlık bilgisi gibi bilgileri toplamaya başlayacaklardır.

## KAYNAKLAR

- [1] Aji, Alham F., Petrus Mursanto, and Setiadi Yazid. "Can smartphones be used to detect an earthquake? Using a machine learning approach to identify an earthquake event." Systems Conference (SysCon), 2014 8th Annual IEEE. IEEE, 2014.
- [2] Antaresti, Tieta, Anggha Satya Nugraha, and Setiadi Yazid. "Wavelet denoising and fractal feature selection for classifying simulated earthquake signal from mobile phone accelerometer." Multi-Conference on Systems, Signals & Devices (SSD), 2014 11th International. IEEE, 2014.
- [3] Reilly, Jenny, et al. "Mobile phones as seismologic sensors: Automating data extraction for the iShake system." Automation Science and Engineering, IEEE Transactions on 10.2 (2013): 242-251.
- [4] Kong, Qingkai, and Richard Allen. "31 Evaluation of Smartphones as Seismometers."
- [5] Kong, Qingkai, and Richard Allen. "28 Potential Applications Using a Smartphone Network."
- [6] Dashti, S., et al. "iShake: The Reliability of Phones as Seismic Sensors."
- [7] Orlic, Niksa, and Sven Loncaric. "Earthquake—explosion discrimination using genetic algorithm-based boosting approach." Computers & geosciences 36.2 (2010): 179-185.
- [8] He, Zhen-Yu, and Lian-Wen Jin. "Activity recognition from acceleration data using AR model representation and SVM." Machine Learning and Cybernetics, 2008 International Conference on. Vol. 4. IEEE, 2008.
- [9] Student, M. Tech. "Feature Extraction From Asthma Patient's Voice Using Mel-Frequency Cepstral Coefficients."
- [10] Mäntyjärvi, Jani, Johan Himberg, and Tapio Seppänen. "Recognizing human motion with multiple acceleration sensors." Systems, Man, and Cybernetics, 2001 IEEE International Conference on. Vol. 2. IEEE, 2001.

- [11] Feasibility Of Smart Devices In Structural Vibration Measurement Ashish Shrestha 1), Ji DANG 2) and Xin WANG 3).
- [12] A Survey of Online Activity Recognition Using Mobile Phones Muhammad Shoaib 1;\*, Stephan Bosch 1, Ozlem Durmaz Incel 2, Hans Scholten 1 and Paul J.M. Havinga 1.
- [13] Kong, Qingkai, et al. "Smartphone-based networks for earthquake detection."Innovations for Community Services (I4CS), 2015 15th International Conference on. IEEE, 2015.
- [14] [https://tr.wikipedia.org/wiki/Android\\_\(i%C5%9Fletim\\_sistemi\)](https://tr.wikipedia.org/wiki/Android_(i%C5%9Fletim_sistemi)), Eriřim Tarihi: 29.04.2016.
- [15] [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)), Eriřim Tarihi: 29.04.2016.
- [16] Taç, M. 2013 Android programlama 978-605-61677-7-5.
- [17] ANDROID TUTORIAL Simply Easy Learning by tutorialspoint.com ([http://www.tutorialspoint.com/android/android\\_tutorial.pdf](http://www.tutorialspoint.com/android/android_tutorial.pdf)), Eriřim Tarihi: 10.04.2016.
- [18] <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-201/android-mimarisi-ve-sistem-ozellikleri>, Eriřim Tarihi: 19.04.2016.
- [19] <http://developer.android.com/tools/studio/index.html>, Eriřim Tarihi: 04.05.2016.
- [20] [http://www.bilgigunlugum.net/android/2android\\_aktivite.html](http://www.bilgigunlugum.net/android/2android_aktivite.html), Eriřim Tarihi: 20.04.2016.
- [21] L.A. Zadeh, "Fuzzy sets", Information and Control 8, 338-353, 1965.
- [22] L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", IEEE Transactions on Systems, man, and Cybernetics, Vol. SMC-3, No. 1,28-44, January 1973.
- [23] L.A. Zadeh, "The concept of linguistic variable and its application to approximate reasoning", Part 1, 2, and 3. Information Sciences, 8: 199-249, 8: 301-357-, 9: 43-80, 1975.
- [24] Altař, İ. Bulanık Mantık : Bulanıklılık Kavramı Enerji, Elektrik, Elektromekanik-3e, Temmuz 1999, Sayı 62, Sayfalar:80-85, Bileřim Yayıncılık.
- [25] [https://tr.wikipedia.org/wiki/Bulan%C4%B1k\\_mant%C4%B1k](https://tr.wikipedia.org/wiki/Bulan%C4%B1k_mant%C4%B1k), Eriřim Tarihi: 06.05.2016.

- [26] Şen, Z. 2003 Modern Mantık Bilge Kültür Sanat 9789758509652.
- [27] <http://www.herturbilgi.com/2013/01/bulanik-mantik/>, Erişim Tarihi: 07.05.2016.
- [28] Eleren, A. İmkb'ye Kayıtlı Çimento İşletmelerinin Finansal Tablolarının Bulanık Mantık Yaklaşımı İle Değerlendirilmesi Afyon Kocatepe Üniversitesi İ.İ.B.F. Dergisi 2007.
- [29] Altaş, İ. Bulanık Mantık : Bulanık Denetim Enerji, Elektrik, Elektromekanik-3e, Eylül 1999, Sayı 64, Sayfalar:76-81, Bileşim Yayıncılık.
- [30] <http://slideplayer.biz.tr/slide/3627204/>, Erişim Tarihi: 12.05.2016.
- [31] [https://en.wikipedia.org/wiki/Membership\\_function\\_\(mathematics\)](https://en.wikipedia.org/wiki/Membership_function_(mathematics)), Erişim Tarihi: 09.05.2016.
- [32] Klima Sistem Kontrolünün Bulanık Mantık İle Modellemesi Ahmet ÖZEK, Mahmut SİNECEN Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi 2004.
- [33] [http://www.dma.fi.upm.es/recursos/aplicaciones/logica\\_borrosa/web/fuzzy\\_inferencia/funpert\\_en.htm](http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/funpert_en.htm), Erişim Tarihi: 10.05.2016.
- [34] <http://www.mathworks.com/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html>, Erişim Tarihi: 19.05.2016.
- [35] <http://www.mathworks.com/help/fuzzy/genfis2.html>, Erişim Tarihi: 14.05.2016.

## EKLER

### EK 1: Veritabani.java kodları

```
package com.example.furkan.sensorapp;

import android.content.Context;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Message;
import android.widget.Toast;

public class Veritabani extends SQLiteOpenHelper {

    private static final String VERITABANI="kayitlar.db";
    private static final int SURUM = 3;

    public Veritabani(Context cont){

        super(cont, VERITABANI,null,SURUM);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        db.execSQL("CREATE TABLE degerler (id INTEGER PRIMARY KEY
AUTOINCREMENT, veriX TEXT, veriY TEXT, veriZ TEXT, time DATETIME,
akturu TEXT);");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        db.execSQL("DROP TABLE IF EXISTS degerler");
        onCreate(db);
    }

}
```

## EK 2: MainActivity.java kodları

```

package com.example.furkan.sensorapp;

import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity implements
AdapterView.OnItemClickListener {

    String akt;
    Spinner spinner;
    Spinner spinner2;
    Veritabani db;
    private Veritabani kayitlar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final EditText ed = (EditText) findViewById(R.id.editText);
        final String girilen=ed.getText().toString();

        final AlertDialog.Builder alertbox2 = new
AlertDialog.Builder(this);
        alertbox2.setTitle("UYARI");
        alertbox2.setMessage("Silmek istediğimize emin misiniz?");
        alertbox2.setNegativeButton("NO",
            new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog,
int which) {

                    // TODO Auto-generated method stub
                    Toast.makeText(getApplicationContext(),
                        "hayır butonuna tıklandı",
Toast.LENGTH_SHORT)
                        .show();

                }

            });
        alertbox2.setPositiveButton("YES",
            new DialogInterface.OnClickListener() {

```

```

        @Override
        public void onClick(DialogInterface dialog,
int which) {
            // TODO Auto-generated method stub
            SQLiteDatabase db =
kayitlar.getWritableDatabase();

            db.execSQL("DELETE FROM degerler");
            db.close();

            kayitlar.close();
            Toast.makeText(getApplicationContext(),
                "evet butonuna tiklandi",
Toast.LENGTH_SHORT)
                .show();
        }
    });

    kayitlar=new Veritabani(this);

    Button basla=(Button)findViewById(R.id.btnBasla);
    basla.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            akt = spinner.getSelectedItem().toString();
            String sur= spinner2.getSelectedItem().toString();
            Intent yeni=new Intent(MainActivity.this,
sayfaiki.class);
            if (spinner.getSelectedItemPosition()==5)
            {
                akt=ed.getText().toString();
            }
            yeni.putExtra("aktivite_adi", akt);
            yeni.putExtra("surr", sur);
            startActivity(yeni);

        }
    });

    Button kapat=(Button)findViewById(R.id.btnKapat);
    kapat.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();

        }
    });

    Button temiz=(Button)findViewById(R.id.btnTemizle);
    temiz.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            alertbox2.show();

        }
    });

```



```

        spinner=(Spinner) findViewById(R.id.spinner);
        ArrayAdapter adapter=ArrayAdapter.createFromResource(this,
R.array.aktivite, android.R.layout.simple_spinner_item);
        spinner.setAdapter(adapter);
        spinner.setOnItemClickListener(this);

        spinner2=(Spinner) findViewById(R.id.spinner2);
        ArrayAdapter<CharSequence> adapter2 =
ArrayAdapter.createFromResource(this,
        R.array.sure,
android.R.layout.simple_spinner_item);

adapter2.setDropDownViewResource(android.R.layout.simple_spinner_d
ropdown_item);
        spinner2.setAdapter(adapter2);

        spinner.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                ed.setText("");

                if(position == 5)
                {
                    Toast.makeText(getApplicationContext(),
"Test", Toast.LENGTH_LONG).show();
                    ed.setEnabled(true);
                    //akt=ed.getText().toString();

                } else {
                    ed.setEnabled(false);
                }
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {

            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so
long

```

```
        // as you specify a parent activity in
        AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {

    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

    @Override
    public void onPause() {
        super.onPause();
    }

}
}
```

### EK 3: Sayfaiki.java kodları

```

package com.example.furkan.sensorapp;

import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

public class sayfaiki extends Activity implements
SensorEventListener {

    private SensorManager sensorManager;
    private float last_x, last_y, last_z = 0;
    private Veritabani kayitlar;
    int indis=0;
    String akt;
    String sur;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sayfaiki);

        Intent intent = getIntent();
        akt = intent.getStringExtra("aktivite_adi");
        sur= intent.getStringExtra("surr");

        Toast.makeText(getApplicationContext(), akt,
Toast.LENGTH_LONG).show();

        kayitlar=new Veritabani(this);

        final TextView X=(TextView)findViewById(R.id.textView4);
        final TextView Y=(TextView)findViewById(R.id.textView5);
        final TextView Z=(TextView)findViewById(R.id.textView6);

        Button dur=(Button)findViewById(R.id.button2);
        dur.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                kayitlar.close();
                finish();
            }
        });
    }
}

```

```

    });
}

    sensorManager = (SensorManager)
getSystemService(SENSOR_SERVICE);
    sensorManager.registerListener(this,
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
SensorManager.SENSOR_DELAY_GAME);
    sensorManager.registerListener(this,
sensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY),
SensorManager.SENSOR_DELAY_GAME);
}

    private void KayitEkle(String bir, String iki, String uc,
String tur){
    SQLiteDatabase db=kayitlar.getWritableDatabase();
    ContentValues veriler=new ContentValues();
    veriler.put("veriX", bir);
    veriler.put("veriY", iki);
    veriler.put("veriZ", uc);
    veriler.put("time", getDateTIme());
    veriler.put("akturu", tur);

    db.insertOrThrow("degerler", null, veriler);
    db.close();
}

    private String getDateTIme() {
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS", Locale.getDefault());
    Date date = new Date();
    return dateFormat.format(date);
}

    float[] gravity = new float[3];

    long r=System.currentTimeMillis();

    public void onSensorChanged(SensorEvent event) {

        long w=System.currentTimeMillis();

        indis++;

        if (event.sensor.getType() == Sensor.TYPE_GRAVITY) {
            gravity[0] = event.values[0];
            gravity[1] = event.values[1];
            gravity[2] = event.values[2];
        }

        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {

            final TextView
X=(TextView) findViewById(R.id.textView4);
            final TextView
Y=(TextView) findViewById(R.id.textView5);
            final TextView

```

```

Z=(TextView) findViewById(R.id.textView6);

    float[] values = event.values;
    float x = values[0];
    float y = values[1];
    float z = values[2];

    final float alpha = 0.8f;

    float[] linear_acceleration = new float[3];
    linear_acceleration[0] = 0;
    linear_acceleration[1] = 0;
    linear_acceleration[2] = 0;

    gravity[0] = alpha * gravity[0] + (1 - alpha) *
event.values[0];
    gravity[1] = alpha * gravity[1] + (1 - alpha) *
event.values[1];
    gravity[2] = alpha * gravity[2] + (1 - alpha) *
event.values[2];

    linear_acceleration[0] = event.values[0] - gravity[0];
    linear_acceleration[1] = event.values[1] - gravity[1];
    linear_acceleration[2] = event.values[2] - gravity[2];

    last_x = linear_acceleration[0];
    last_y = linear_acceleration[1];
    last_z = linear_acceleration[2];

    last_x = Math.round(last_x*1000.f)/1000.f;
    last_y = Math.round(last_y*1000.f)/1000.f;
    last_z = Math.round(last_z*1000.f)/1000.f;

    X.setText(String.valueOf(last_x));
    Y.setText(String.valueOf(last_y));
    Z.setText(String.valueOf(last_z));

    KayitEkle(String.valueOf(last_x),
String.valueOf(last_y), String.valueOf(last_z), akt);

    if(Integer.valueOf(sur)==15 && (w-r)>15000)
    {
        sensorManager.unregisterListener(this);
        finish();
    }

    if(Integer.valueOf(sur)==30 && (w-r)>30000)
    {
        sensorManager.unregisterListener(this);
        finish();
    }
    if(Integer.valueOf(sur)==45 && (w-r)>45000)
    {
        sensorManager.unregisterListener(this);
        finish();
    }
    if(Integer.valueOf(sur)==60 && (w-r)>60000)

```

```
        {  
            sensorManager.unregisterListener(this);  
            finish();  
        }  
    }  
}  
  
@Override  
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
    }  
}
```

## EK 4: Strings.xml kodlari

```
<resources>
  <string name="app_name">SensorApp</string>

  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>

  <string-array name="aktivite">
    <item>Yurume</item>
    <item>Kosma</item>
    <item>Merdiven</item>
    <item>Sabit zemin</item>
    <item>Arabada</item>
    <item>Diger</item>
  </string-array>

  <string-array name="sure">
    <item>15</item>
    <item>30</item>
    <item>45</item>
    <item>60</item>
  </string-array>
</resources>
```

**EK 5: Sayfaiki.xml kodları:**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/androidres">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="X :"
            android:textSize="30dp"
            android:id="@+id/textView"
            android:layout_marginTop="44dp"
            android:layout_alignParentTop="true"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_marginLeft="44dp"
            android:layout_marginStart="44dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Y :"
            android:textSize="30dp"
            android:id="@+id/textView2"
            android:layout_marginTop="40dp"
            android:layout_below="@+id/textView"
            android:layout_alignLeft="@+id/textView"
            android:layout_alignStart="@+id/textView" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Z :"
            android:textSize="30dp"
            android:id="@+id/textView3"
            android:layout_below="@+id/textView2"
            android:layout_alignLeft="@+id/textView2"
            android:layout_alignStart="@+id/textView2"
            android:layout_marginTop="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:textAppearance="?android:attr/textAppearanceLarge"

```



```

        android:text="00"
        android:textSize="30dp"
        android:id="@+id/textView4"
        android:layout_alignTop="@+id/textView"
        android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="00"
    android:textSize="30dp"
    android:id="@+id/textView5"
    android:layout_alignTop="@+id/textView2"
    android:layout_alignLeft="@+id/textView4"
    android:layout_alignStart="@+id/textView4" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="00"
    android:textSize="30dp"
    android:id="@+id/textView6"
    android:layout_alignTop="@+id/textView3"
    android:layout_alignLeft="@+id/textView5"
    android:layout_alignStart="@+id/textView5" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="DURDUR"
    android:id="@+id/button2"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="98dp" />
</RelativeLayout>
</LinearLayout>

```

## ÖZGEÇMİŞ

Ahmet Furkan SÖNMEZ, 10.10.1991'de Sakarya'da doğdu. İlk, orta ve lise eğitimini Sakarya'da tamamladı. 2009 yılında Figen Sakallıođlu Anadolu Lisesi'nden mezun oldu. 2009 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü 2013 yılında bitirdi. 2013 yılında Bülent Ecevit Üniversitesi'nde ÖYP Araştırma Görevlisi olarak göreve başladı. Şubat 2015 tarihinden itibaren Sakarya Üniversitesi Bilgisayar ve Bilişim Bilimleri Fakültesi'nde, Bilişim Sistemleri Mühendisliği'nde Araştırma Görevlisi olarak çalışmaktadır.