

**T.C.  
BALIKESİR ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI**



**ÖĞRENEBİLEN, WEB TABANLI, DÜŞÜK ENERJİ  
TÜKETİMLİ, MODÜLER EV OTOMASYON SİSTEMİ  
GELİŞTİRİLMESİ**

**DOKTORA TEZİ**

**HÜSEYİN GÜNEŞ**

**BALIKESİR, ARALIK - 2016**

**T.C.**  
**BALIKESİR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI**



**ÖĞRENEBİLEN, WEB TABANLI, DÜŞÜK ENERJİ**  
**TÜKETİMLİ, MODÜLER EV OTOMASYON SİSTEMİ**  
**GELİŞTİRİLMESİ**

**DOKTORA TEZİ**

**HÜSEYİN GÜNEŞ**

**Jüri Üyeleri : Yrd. Doç. Dr. Davut AKDAŞ (Tez Danışmanı)**

**Prof. Dr. Fehmi ERZİNCANLI**

**Prof. Dr. Mehmet Hakan HOCAOĞLU**

**Doç. Dr. Murat Erhan BALCI**

**Yrd. Doç. Dr. Mehmet İREN**

**BALIKESİR, ARALIK - 2016**

## KABUL VE ONAY SAYFASI

**Hüseyin GÜNEŞ** tarafından hazırlanan “**ÖĞRENEBİLEN, WEB TABANLI, DÜŞÜK ENERJİ TÜKETİMLİ, MODÜLER EV OTOMASYON SİSTEMİ GELİŞTİRİLMESİ**” adlı tez çalışmasının savunma sınavı 23.12.2016 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Makina Mühendisliği Anabilim Dalı olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman  
Yrd. Doç. Dr. Davut AKDAŞ



Üye  
Prof. Dr. Fehmi ERZİNCANLI



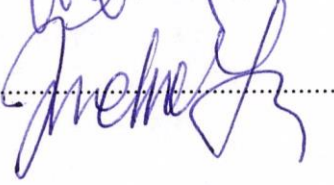
Üye  
Prof. Dr. Mehmet Hakan HOCAOĞLU



Üye  
Doç. Dr. Murat Erhan BALCI



Üye  
Yrd. Doç. Dr. Mehmet İREN



Jüri üyeleri tarafından kabul edilmiş olan bu tez Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Doç. Dr. Necati ÖZDEMİR



**Bu tez çalışması Balıkesir Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından 2015-28 (Öğrenebilen, Modüler, Web Arayüzü İle Kontrol Edilen, Açık Kaynak Kodlu Akıllı Bina Otomasyon Sistemi Geliştirilmesi) nolu proje ile desteklenmiştir.**

## ÖZET

**ÖĞRENEBİLEN, WEB TABANLI, DÜŞÜK ENERJİ TÜKETİMLİ,  
MODÜLER EV OTOMASYON SİSTEMİ GELİŞTİRİLMESİ**  
**DOKTORA TEZİ**  
**HÜSEYİN GÜNEŞ**  
**BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**  
**MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI**  
**(TEZ DANIŞMANI: YRD. DOÇ. DR. DAVUT AKDAŞ)**  
**BALIKESİR, ARALIK - 2016**

Akıllı ev otomasyon sistemleri evin aydınlatma, güvenlik, havalandırma, sıcaklık vb. sistemlerinin kullanıcı tarafından daha kolay kontrol ve takip edilebilmesini sağlamaktadır. Bu sistemler, ilk dönemlerinde yalnızca kullanıcı tarafından evin kolay bir şekilde kontrol edilebilmesini sağlarken, zamanla birlikte ev sakinlerinin yaşam şekillerini öğrenerek otonom olarak evin kontrolünü sağlamaya başlamışlardır. Bu çalışmada, öğrenebilen, modüler ve düşük enerji tüketimli çoklu ajan sistemleri temelli özgün bir akıllı ev otomasyon sistemi tasarlanmıştır. Tasarlanan sistem, merkezde yer alan bir bulut sunucu ve ona bağlı olarak her odada bulunan oda kontrolcüler ile ısıtma sistemini kontrol eden termostat bileşeni, lambaları kontrol eden aydınlatma bileşeni, prizleri kontrol eden priz bileşeni ve sensör bileşenlerinden oluşmaktadır. Sistemin kontrolü, geliştirilen web arayüzü ve her odada bulunan dokunmatik ekranlar ile sağlanmaktadır.

Akıllı ev sisteminin kullanıcı davranışlarını öğrenerek otonom hareket edebilmesi için çeşitli yapay zekâ algoritmaları geliştirilmiştir. Bu yapay zekâ algoritmalarının geliştirilmesi, iyileştirilmesi ve test edilebilmesi için sistemin belirli bir süre farklı kullanıcılar tarafından kullanılması gerekmektedir. Gerçek test ortamında yaşanabilecek sorunlardan dolayı sanal bir aile ve bir ev oluşturularak geliştirilen akıllı evi bu aile kullanıyormuş gibi kayıtlar üreten bir yazılım geliştirilmiş ve bir yıllık kayıtlar oluşturulmuştur. Daha sonra geliştirilen yapay zekâ algoritmalarının gerçek ortamda ve gerçek zamanda testi için bu sanal aile ile gerçek bir bireyi, gerçek bir odaya sahip sanal bir evde beraber yaşıyormuş gibi sunabilen hibrit bir akıllı ev simülasyonu geliştirilmiştir. Geliştirilen akıllı ev otomasyon sisteminde yapay zekâ; etki tepki, yetkilendirme, kişi konum tespit, aygıt durum tespit, kişi tercihleri tespit ve oda ısınma soğuma sürelerinin tespit işlemlerini gerçekleştirmekte ve bu tespit işlemlerinden elde edilen verileri gerçek zamanlı olarak çalışan sistem üzerinde kullanarak otonom olarak hareket edebilmektedir. Sonuç olarak kullanıcı davranışlarını öğrenerek her kullanıcıya özel olarak otonom hareket edebilen, kullanıcıların evin aydınlatma, ısıtma ve elektrik hatlarını kontrol edebildiği ve oda sıcaklığı, nemi ve parlaklık düzeyi gibi bilgileri takip edebildiği, web tabanlı, düşük enerji tüketimli, özgün tasarıma sahip prototip bir akıllı ev otomasyon sistemi gerçekleştirilmiştir.

**ANAHTAR KELİMELER:** Akıllı ev otomasyonu, yapay zekâ, yapay veri, akıllı ev simülasyonu, hibrit simülasyon.

## **ABSTRACT**

### **LEARNING, WEB BASED, LOW ENERGY CONSUMPTION, MODULAR HOME AUTOMATION SYSTEM DEVELOPMENT**

**PH.D THESIS**

**HUSEYIN GUNES**

**BALIKESIR UNIVERSITY INSTITUTE OF SCIENCE**

**MECHANICAL ENGINEERING**

**(SUPERVISOR: ASST. PROF. DAVUT AKDAS)**

**BALIKESİR, DECEMBER 2016**

Smart automation systems enable the systems of house's lighting, security, air-conditioning, heating, etc. to be controlled and monitored easily by users. While these systems were enabling the house to be controlled easily by only the user in the beginnings, they have started to autonomously assume control of the house by learning the life styles of house dwellers in time. In this study, multiple operative system based a unique smart automation system, with modular and less energy consumption, which is able to learn has been designed. Designed system consists of thermostat constituents controlling heating system, lighting constituents controlling lights, socket constituents controlling sockets and sensor constituents with cloud server in the center and room controllers existing in each room depending on cloud server. The control of the system is assumed with developed web interface and touch screens in each room.

The various artificial intelligence algorithm should be developed in order that smart house system is able to act autonomously by learning user behavior. The system should be used by users for a length of time in order to develop, enhance and test this artificial intelligence algorithm. A software, which produces registrations as if the family uses the smart house developed by creating imaginary family and a house, has been developed and annual registrations have been created due to problems that might occur in the real test environment. Afterwards, a hybrid smart house simulation that presents as if this imaginary family and a real person live together in an imaginary house having a real room has been developed for the test of developed artificial intelligence algorithms in the real environment and time. In the developed system of smart house automation, artificial intelligence carries out the processes of action and reaction, authorization, the detection of person and place, the detection of device status, the detection of personal preferences and the detection of heating and cooling time of the room and acts autonomously by using the data acquired from detection processes on a real and simultaneously working system. As a result, a prototype smart house automation system; web-based, with low energy consumption and having a unique design, which is able to dedicatedly act for each user by learning their behaviors; control lighting, heating and the powerline of the users' houses; and monitor information such as room temperature, humid and brightness level, has been created.

**KEYWORDS:** Intelligent home automation, artificial intelligence, artificial data, smart home simulation, hybrid simulation.

# İÇİNDEKİLER

Sayfa

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>İÇİNDEKİLER</b> .....	<b>iii</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>v</b>
<b>TABLO LİSTESİ</b> .....	<b>vii</b>
<b>ÖNSÖZ</b> .....	<b>viii</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. LİTERATÜR</b> .....	<b>4</b>
2.1 Akıllı Evlerin Yapabildikleri .....	4
2.2 Akıllı Evlerde Kullanılan Teknolojiler.....	6
2.2.1 Uzaktan Erişim İçin Kullanılan Teknolojiler.....	7
2.2.2 Aygıtlar Arası Erişim İçin Kullanılan Teknolojiler .....	8
2.3 Günümüzdeki Ev Sistemleri ve Özellikleri .....	11
2.4 Akıllı Evlerde Yapay Zekâ.....	12
2.4.1 Yapay Zekâ .....	13
2.4.2 Yapay Zekâ Teknikleri .....	15
2.4.2.1 Yapay Sinir Ağları .....	15
2.4.2.2 Bulanık Mantık .....	20
2.4.2.3 Destek Vektör Makinaları (Çekirdek Makinaları) .....	22
2.4.2.4 Markov Modelleri .....	24
2.4.2.5 Naive Bayes Teoremi .....	26
2.4.2.6 Karar Ağaçları .....	27
2.4.2.7 Çok Ajanlı Sistemler (MAS: Multi Agent Systems) .....	27
2.5 Yapay Veri Üretme .....	28
<b>3. YÖNTEM</b> .....	<b>31</b>
3.1 Sistemin Geliştirme Süreci .....	31
3.2 Akıllı Ev Sisteminin Gerçekleştirilmesi Gereken İşlemler .....	33
3.3 Sistemin Genel Yapısı, İletişim Şekilleri ve İşleyişi .....	34
3.4 Web Arayüzü.....	36
3.5 Merkez Sunucu .....	38
3.6 Oda Kontrolcü .....	39
3.7 Tümüleşik Sensör Bileşeni .....	43
3.7.1 Işık Sensörü.....	44
3.7.2 Sıcaklık ve Nem Sensörü .....	45
3.7.3 Hareket Sensörü .....	45
3.7.4 Gaz Sensörü .....	46
3.7.5 Alev Algılayıcı Sensör .....	47
3.8 Aydınlatma Bileşeni .....	48
3.9 Priz Bileşeni .....	49
3.10 Termostat Bileşeni.....	50
3.11 Dış Sensör Bileşeni .....	52
3.12 Kullanıcı Takip Bileşeni.....	53

3.13	Birimler Arası Bağlantılar ve İletişim .....	55
3.14	Sistemin Veri Yapısı ve İşlemlerin Kayıt Altına Alınması .....	61
3.14.1	Oda.....	62
3.14.2	Aygıt .....	62
3.14.3	Kullanıcı.....	63
3.14.1	İşlem.....	64
3.14.1	Sensör Verileri .....	64
3.15	Yapay Veri Üretici .....	65
3.15.1	Senaryo .....	66
3.15.2	Yapay Kayıt Üretici Yazılım .....	69
3.15.3	Senaryonun Yazılıma Girişi ve Sanal Kayıt Üretimi.....	74
3.16	Akıllı Ev Simülasyonu (Hibrit Simülasyon) .....	75
3.16.1	Aygıtlar .....	77
3.16.2	Kullanıcılar .....	78
3.16.3	Odalar.....	79
3.16.4	Olay Ekle .....	79
3.16.5	Yaklaşan Olaylar.....	80
3.17	Yapay Zekâ.....	82
3.17.1	Etki Tepki İşlemleri .....	83
3.17.2	Yetkilendirme .....	84
3.17.3	Kişilerin Konumunu Tespit Etme .....	85
3.17.4	Aygıtların Durumlarını Tespit Etme.....	91
3.17.5	Kullanıcı Tercihleri Belirlenmesi .....	93
3.17.6	Odaların Isınma Soğuma Süreleri Hesaplanması .....	94
3.17.7	Yapay Zekâ Algoritmalarının Sistem Üzerinde İşleyişi .....	96
<b>4.</b>	<b>BULGULAR .....</b>	<b>99</b>
4.1	Web Arayüzü.....	99
4.2	Yapay Veri Üretici .....	101
4.3	Yapay Zekâ.....	104
4.4	Sistemin Çalışması ve Donanımlar .....	110
<b>5.</b>	<b>SONUÇ VE ÖNERİLER .....</b>	<b>112</b>
<b>6.</b>	<b>KAYNAKLAR.....</b>	<b>116</b>
<b>7.</b>	<b>EKLER.....</b>	<b>128</b>
EK A.	Sanal ailenin senaryosu .....	129
EK B.	Sistemin genel yapısı.....	128
EK C.	Veritabanı tabloları.....	129
EK D.	Akıllı Ev Simülasyonu (Hibrit Simülasyon) .....	139
EK E.	Odada Daha Yetkili Biri Var mı? Program Kodu .....	140
EK F.	Kullanıcı Konum Tespit İşlemi Kodları .....	141
EK G.	Karar Ağacı .....	143



## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1: Akıllı evlerin yapabildikleri .....	5
Şekil 2.2: Yapay öğrenmenin hiyerarşik yapısı .....	15
Şekil 2.3: Basit nöron modeli .....	16
Şekil 2.4: Yapay sinir ağının genel yapısı .....	17
Şekil 2.5: Bulanık sistemin genel yapısı.....	21
Şekil 2.6: DVM ile doğrusal sınıflandırma.....	23
Şekil 2.7: Çekirdek fonksiyonu ile boyut dönüşümü.....	23
Şekil 3.1: Sistemin genel yapısı.....	36
Şekil 3.2: Web arayüzü.....	37
Şekil 3.3: Merkez Sunucu.....	38
Şekil 3.4 : Raspberry Pi 3 .....	40
Şekil 3.5 : LCD dokunmatik ekran .....	41
Şekil 3.6: Oda Kontrolcü .....	42
Şekil 3.7: Arduino Mega .....	43
Şekil 3.8: 5mm LDR ışık sensörü kartı .....	44
Şekil 3.9: Sıcaklık ve nem sensörü .....	45
Şekil 3.10: PIR hareket algılama sensörü .....	46
Şekil 3.11: Yanıcı gaz ve karbonmonoksit sensörü.....	46
Şekil 3.12: Alev algılayıcı sensör .....	47
Şekil 3.13: AC dimmer modül.....	48
Şekil 3.14: Aydınlatma Bileşeni .....	49
Şekil 3.15: Röle shield.....	49
Şekil 3.16: Priz Bileşeni .....	50
Şekil 3.17: Google Nest termostat .....	51
Şekil 3.18: Arduino Uno ve BLE shield.....	52
Şekil 3.19: Dış Sensör Bileşeni .....	53
Şekil 3.20: RFID okuyucu .....	54
Şekil 3.21: Odadaki sistemin genel görünümü.....	55
Şekil 3.22: Sistemin genel yapısı, bağlantı şekilleri ve iletişim yolları.....	56
Şekil 3.23: Oda bilgileri tablosu .....	62
Şekil 3.24: Aygıt bilgileri tablosu.....	63
Şekil 3.25: Kullanıcı bilgileri tablosu.....	63
Şekil 3.26: İşlem bilgileri tablosu.....	64
Şekil 3.27: Sensör verileri tablosu .....	65
Şekil 3.28: Sanal evin görünümü.....	68
Şekil 3.29: Kişi Ekle menüsü .....	70
Şekil 3.30: Oda Ekle menüsü .....	71
Şekil 3.31: Aygıt Ekle menüsü .....	71
Şekil 3.32: Kişisel bilgileri giriş ve senaryo oluşturma.....	72
Şekil 3.33: Kayıt üretim seçenekleri.....	73
Şekil 3.34: Üretilmiş yapay kayıtlar .....	74
Şekil 3.35: Yapay veri üretim akış diyagramı .....	75

Şekil 3.36: Akıllı ev simülasyonu arayüzü .....	77
Şekil 3.37: Odalarda bulunan termostat bilgileri .....	78
Şekil 3.38: Kullanıcıların konum gösterimi .....	79
Şekil 3.39: Oda bilgileri.....	79
Şekil 3.40: Olay Ekle menüsü .....	80
Şekil 3.41: Yaklaşan Olaylar menüsü.....	81
Şekil 3.42: Etki tepki işlemi ile kart okuyucunun etrafının aydınlatılması .....	84
Şekil 3.43: Tarihe göre kullanıcı konum kayıtları .....	85
Şekil 3.44: Saate göre kullanıcı konum kayıtları.....	86
Şekil 3.45: Kişinin son 10 pazartesi günü belirtilen saatteki konumları .....	89
Şekil 3.46: Termostat durum kayıtları .....	92
Şekil 3.47: Oda sıcaklık ve termostat verileri dakikalık kayıtları .....	95
Şekil 3.48: Yapay zekâ karar ağacı.....	97
Şekil 4.1: Android tablet, Android tarayıcı ile arayüzün test görüntüsü .....	100
Şekil 4.2: Windows 10 bilgisayar, Microsoft Edge ile test görüntüsü .....	100
Şekil 4.3: Kullanıcı konum tespit ekranı .....	106
Şekil 4.4: Konum hareketinin olduğu dakikadaki konum ihtimalleri.....	107
Şekil 4.5: Aygıt durum tespit.....	108
Şekil 4.6: Kişi tercihlerinin tespiti .....	109
Şekil 4.7: Salon oda ısınma soğuma tahmin verileri .....	110
Şekil A.1: Sistemin genel yapısı .....	128
Şekil B.1: Oda bilgileri tablosu .....	129
Şekil B.2: Aygıt bilgileri tablosu .....	130
Şekil B.3: Aygıt tipi tablosu .....	131
Şekil B.4: Kullanıcı bilgileri tablosu .....	132
Şekil B.5: İşlem bilgileri tablosu .....	133
Şekil B.6: Sensör verileri tablosu .....	134
Şekil B.7: Yapay kayıtlar tablosu .....	135
Şekil D.1: Hibrit simülasyon arayüzü.....	139
Şekil G.1: Yapay zekâ karar ağacı .....	143

## TABLO LİSTESİ

### Sayfa

<b>Tablo 2.1:</b> Dünya çapında yıllara göre internet kullanımı. ....	7
<b>Tablo 2.2:</b> Kablosuz iletişim teknolojilerinin teknik özellikleri. ....	11
<b>Tablo 2.3:</b> Kablosuz iletişim teknolojilerinin enerji tüketimleri. ....	11
<b>Tablo 3.1:</b> Sanal ailenin özellikleri. ....	66
<b>Tablo 3.2:</b> Sanal ailenin sıcaklık ve aydınlık tercihleri. ....	68
<b>Tablo 3.3:</b> Sanal ailenin oda yetki seviyeleri. ....	69
<b>Tablo 4.1:</b> Aydınlık değerleri dağılımı. ....	101
<b>Tablo 4.2:</b> Sıcaklık değerleri dağılımı. ....	103
<b>Tablo 4.3:</b> Ev giriş çıkış zaman dağılımı. ....	103
<b>Tablo 4.4:</b> Oda ısınma soğuma tahminleri. ....	110
<b>Tablo A.1:</b> Sanal ailenin senaryosu. ....	136

## ÖNSÖZ

Bizi yokluktan var eden Allah'a şükürler olsun. Allah tüm bu evrenin yaratılmasına sebep olan, bize doğru yolu gösteren peygamberimiz Hz. Muhammed(sav)' e rahmet etsin.

Farklı bir bölümden olmama karşın beni doktora öğrencisi olarak kabul eden, 6 yıldır birlikte çalışarak güzel bir ürün ortaya çıkardığımız danışman hocam Yrd. Doç. Dr. Davut AKDAŞ'a teşekkür ederim.

Bütün doktora süresince o değerli vakitlerini benim için harcayan hocalarım Prof. Dr. Fehmi ERZİNCANLI ve Yrd. Doç. Dr. Mehmet İREN'e teşekkür ederim.

Çalışmalarım süresince bana her zaman destek olan Yrd. Doç. Dr. Sabri BİCAKCI ve İngilizce çevirilerde bana her zaman yardımcı olan Arş. Gör. Erman KÖYBAŞI'na teşekkür ederim.

Yüksek lisans çalışmamın son döneminde beni şimdiki danışmanım ile tanıştıran doktora öğrenimime başlamama sebep olan ve bu dönemde desteklerini benden esirgemeyen hocalarım Yrd. Doç. Dr. Ayşen KARAMETE ve Yrd. Doç. Dr. Mehmet Emin KORKUSUZ'a ayrıca bir başka bölümden olmama rağmen bana güvenerek beni bölümlerine kabul eden Prof. Dr. İrfan AY'a teşekkür ederim.

Bugüne kadar benim için hiçbir fedakârlıktan kaçınmayan annem, babam ve ağabeyime teşekkür ederim.

# 1. GİRİŞ

İnsanlar için tarihin başlangıcıyla birlikte doğa şartlarından ve vahşi hayvanlardan korunmak amacıyla başlayan ev serüveni, zaman içerisinde farklı ihtiyaçlar ve amaçlar doğrultusunda çeşitli şekillerde karşımıza çıkmış ve günümüze kadar uzanmıştır. İlk zamanlarda mağaralar vb. doğal yapılar ev amaçlı kullanılmış, daha sonra hayvan derileri ve kemiklerinden çadırlar yapılmıştır. Zamanla ağaçlar, kayalar ve toprak kullanılarak evler geliştirilmiştir. Evler geliştikçe odaları artmış, süsleme sanatları ile daha güzel bir görünüme sahip olmuşlardır. Ayrıca evlerle birlikte ev içerisinde kullanılacak ev aletleri ve eşyalar da geliştirilmiştir (Albert, 2015). Tüm bu gelişimin altındaki temel amaç, insan ihtiyacını karşılayarak konforu artırmak ve yaşam standardını yukarı çekmektir.

Son yüzyılda elektriğin evlere girmesi ve elektronik teknolojisinin gelişimi evlerde büyük değişimlere sebep olmuştur. İnsanların ihtiyaçları doğrultusunda çamaşır makinası, bulaşık makinası, televizyon, kahve makinası, mutfak robotu vb. birçok elektronik ev aleti ve beyaz eşya evlerde bulunan standart araçlar haline gelmiştir. Günümüzde ise ev aletlerinin sayısının artmasıyla birlikte tüm bu aletleri tek bir noktadan kontrol ihtiyacı ortaya çıkmış ve dolayısıyla bu ihtiyaç ile birlikte akıllı ev kavramı ortaya çıkmıştır.

Akıllı Ev (Smart Home) kavramı ilk defa resmi olarak 1984 yılında Amerikan Ev İnşacıları Derneği (American Association of House Builders) tarafından kullanılmıştır. Ancak akıllı evlerin tarihi daha uzun bir geçmişe dayanmaktadır. 1960'ların başından itibaren meraklılar tarafından akıllı evlerin ilk temelleri olan "kablolu evler (wired homes)" geliştirilmeye başlanmıştır (Harper, 2003). O günden günümüze değin yapılan farklı çalışmalarda akıllı ev yerine, ev otomasyonu, zeki yapı, yapı otomasyon sistemi, akıllı bina ve entegre ev sistemleri gibi terimler kullanılmıştır (Pande & Sen, 2014; Fernandez, Losada, & Domonte, 2014).

İlk zamanlarda akıllı ev Lutolf (1992) tarafından: "Ev içindeki farklı hizmetlerin, ortak bir iletişim ağı üzerinden ekonomik, güvenli ve konforlu bir biçimde birbirine entegre edilmesidir" şeklinde tanımlanmıştır. Yakın zamanda ise

akıllı ev terimini farklı arařtırmacılar benzer řekillerde tanımlamıřtır. Akıllı ev Mennicken, Vermeulen ve Huang (2004) tarafından bilgisayar teknolojileri kullanarak ev sakinlerinin konforunu artırmak amacıyla evde yapılamayan iřlerin yapılması ya da yapılan iřlerin kolaylařtırılmasını saęlayan “Ev” olarak tanımlanmıřtır. Fernandez, Losada ve Domonte (2014) ise akıllı evi, bilgisayar ve iletiřim teknolojileriyle donatılmıř, ev sakinlerinin konfor, rahatlık ve g¼venlięi iin kolaylıklar sunan bir ev olarak tanımlamıřtır. Bir bařka arařtırmacı Yılmaz (2014) akıllı evleri, bir merkezden kontrol edilebilen, birbirleriyle haberleřebilen, iliřki kurabilen ve b¼t¼n bu teknolojiler sayesinde kiřilere daha tasarruflu, daha g¼venli, daha konforlu, ihtiyalara cevap verebilen ve hayatı kolaylařtıran imk¼nlar sunan evler olarak tanımlamıřtır. Akıllı ev bu tanımlamalar g¼z ¼n¼ne alındıęında “Bilgisayar ve iletiřim teknolojileri kullanarak ev sakinlerinin yařam standardını y¼kselten ev” olarak genel bir ifadeyle tanımlanabilir.

G¼n¼m¼zde birok kiři tarafından uzaktan eriřilerek eřitli aygıtların aılıp kapatılabildięi evler “akıllı (smart)” olarak nitelendirilmektedir. Ancak bu t¼r evlerde herhangi bir akıllı otomasyon iřlemi yapılmamaktadır. Bu t¼r evler akıllı olarak nitelendirilse de bunun sebebi “Akıllı” kelimesinin bir pazarlama terimi olarak kullanılmasından ileri gelmektedir. ¼rneęin Riverside Concept Houses bir evin akıllı olarak tanımlanabilmesi iin birbiri ile haberleřebilen ve uyum iinde alıřabilen aydınlatmalar, perdeler - panjurlar, televizyonlar, klimalar, m¼zik sistemleri, vb. gibi t¼m elektrikli aletlerin tek bir kumanda ve-veya internet ¼zerinden kontrol edebilen sisteme sahip olması gerektięini belirtmiřtir (Riverside Concept Houses, 2014). Arařtırmacılar ise kiřilere duyarlı ve onlara karmařık yollarla oz¼mler sunan, ¼rneęin ¼ęrenme algoritmaları kullanarak kiři davranıřlarını ¼ęrenen ve otomatik olarak ısıtma ve aydınlatma sistemini kontrol eden evleri “Akıllı” olarak nitelendirmiřtir (Mennicken, Vermeulen, & Huang, 2014).

Akıllı evlerin ilk ıkıř amacı binalarda yer alan elektrikle alıřan cihazların ve g¼venlik aygıtlarının kullanıcı dostu bir sistemle kontrol edilebilmesidir (Stefanov & Bien, 2004). G¼n¼m¼zde ise evin iinde ve dıřında yer alan birok donanımın kontrol¼ ve takibi akıllı ev otomasyon sistemleri aracılıęıyla yapılabilmektedir. Bu donanımlar aydınlatma, ısıtma, g¼venlik, su ve elektrik sistemleridir.

Modern akıllı evlerde, evde yer alan donanımların kontrolleri yanı sıra ev konforunu artıracak farklı uygulamalarda yer almaktadır. Bunlara; bebekler ve çocuklar ile yaşlı ve özürlü bireylerin takibini sağlayan sağlık sistemlerini, evin temizliğini otomatik yapabilen temizlik sistemlerini ve kişisel güvenlik sistemleri örnek verilebilir (Burkhard & Bouée, 2013).

Tüm bu özelliklerin yanında, son yıllarda akademik çalışmalarda geliştirilen akıllı evlerde yapay zekâ algoritmaları da kullanılmaya başlanmıştır. Araştırmacılar yapay zekâ algoritmaları sayesinde ev içerisinde kullanıcı davranışlarını takip ederek kayıt altına almakta, daha sonra bu kayıtlardan kullanıcının günlük davranışlarını öğrenmekte ve zaman içerisinde bu davranışlar doğrultusunda otonom hareket edebilen ev otomasyon sistemleri geliştirmektedirler. Bu sayede evde konforu artırarak enerji tasarrufu da sağlayabilen sistemler ortaya çıkabilmektedir(Alam, Reaz, & Mohd Ali, 2012).

Alam, Reaz ve Mohd Ali (2012) gelecekte akıllı evler; insan yaşamında daha önemli bir noktaya gelecektir. Görsel, işitsel ve dokunsal kullanıcı arayüzleri ile yönetilebileceklerdir.

Chan, Campo, Esteve ve Fourniols (2009)'a göre sağlık hizmetlerinin mevcut yüksek maliyetleri göz önüne alındığında akıllı evler uzaktan yönetilebilen sağlık hizmetlerinin bir parçası haline gelecektir. Ayrıca yaşlı ve engelliler için yardımcı sağlık hizmetleri sunulmasını sağlayacaktır.

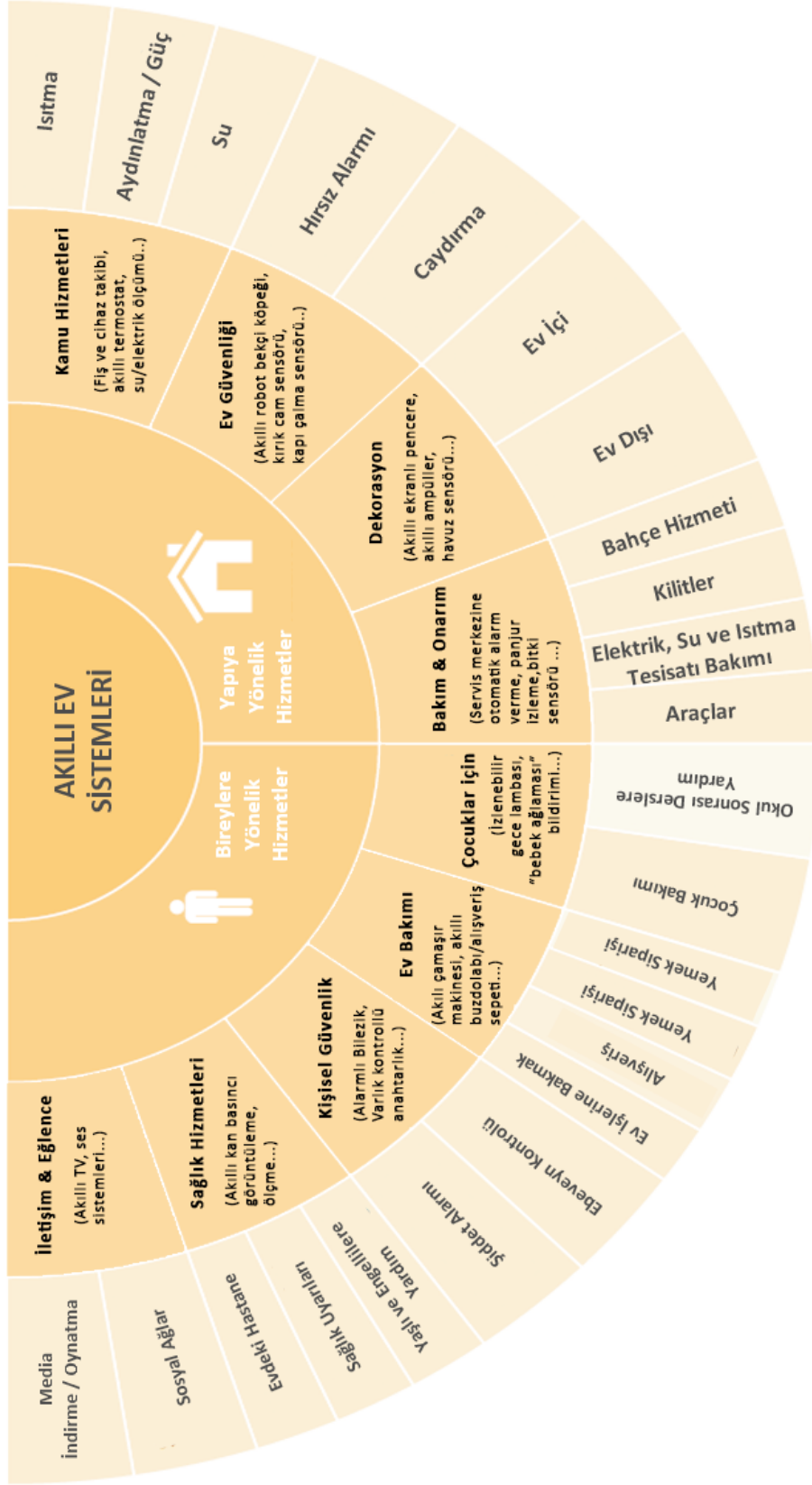
## **2. LİTERATÜR**

Bu bölümde, akıllı evlerin yapabildikleri, akıllı evlerde kullanılan teknolojiler akıllı evlerin çeşitleri ve akıllı evlerde kullanılan yapay zekâ teknikleri hakkında yapılan çalışmalar sunulmuştur.

### **2.1 Akıllı Evlerin Yapabildikleri**

Yıllar içerisinde gelişen teknoloji, yeni fikirlerin ve yeni ihtiyaçların ortaya çıkması ile hem akademik olarak hem de ticari olarak birbirinden farklı fonksiyonlara sahip, farklı özellikleri olan farklı kişilere yönelik akıllı evler geliştirilmiştir. Günümüzde akıllı ev sistemlerinin yapabildikleri Şekil 2.1’de genel hatlarıyla gösterilmiştir (Schwenker, Bouée, & Oltmanns, 2013).





Şekil 2.1: Akıllı evlerin yapabildikleri

Akıllı evlerin yapabildikleri eve yönelik ve kişilere yönelik iki bölüme ayrılabilir. Eve yönelik olarak şunlardır:

- Isıtma, aydınlatma, güç (elektrik) ve su sistemlerinin kontrol edilebilmesi ve takip edilebilmesi,
- Dışarıdan eve müdahalelerin tespit edilmesi ve uyarı verilmesi, caydırmak için müdahale edilebilmesi,
- Bahçede bitkilerin sulama ve havuzun su seviyesini kontrol edebilmesi,
- Ev içerisinde yaşanan sorunları takip ederek tesisatçı vb. servis hizmetlerini doğrudan haberdar edebilmesi

Kişilere yönelik ise şunlardır:

- Bebeklerin gece ve gündüz takip edilmesi, çocukların okul sonrası çalışmalarına destek olması,
- Eksikleri takip ederek alışveriş yapabilmesi, pişirme işlemlerine ve temizliğe yardım etmesi,
- Kişisel güvenliğe yardım etmesi ve sağlık durumunu takip ederek acil durumlarda acil servise ulaşması,
- Kişisel hesaplarınıza erişerek, eposta ve sosyal ağlara ulaşabilmenizi sağlayabilmesi,
- Ev içerisinde tv ve ses sistemlerini kontrol edebilmenizi sağlaması (Schwenker, Bouée, & Oltmanns, 2013).

## **2.2 Akıllı Evlerde Kullanılan Teknolojiler**

Günümüzdeki akademik ve ticari akıllı ev sistemleri incelendiğinde; tüm bu belirtilen işlevleri (Şekil 2.1) gerçekleştirebilmek için benzer bir yapıda geliştirildikleri görülmektedir. Genel olarak sistemler uzaktan kontrol edilebilen bir merkez sunucu ve ev içerisinde işlemleri gerçekleştiren sunucuya bağlı uç bileşenlerden meydana gelmektedir. Ancak tüm bu işlevleri gerçekleştirirken farklı teknolojiler, yöntemler ve donanımlar kullanılmaktadır (Alam, Reaz, & Mohd Ali, 2012).

Shwehdi ve Khan (1996) odalarda yer alan termostatların kontrollerini merkezi bir bilgisayardan gerçekleştirebilen bir ev otomasyon sistemi geliştirmişlerdir. Yine 90'lı yıllarda benzer bir çalışmada röle tabanlı akıllı ev sistemleri geliştirilmiştir. Akıllı evlerin bilgisayar ve iletişim teknolojilerinin gelişip yaygınlaştığı 90'lı yıllardaki ilk örnekleri incelendiğinde sadece ev içerisinde kontrol edilebildikleri görülmektedir. Günümüzde ise akıllı ev sistemleri için uzaktan kontrol bir standart haline gelmiştir; havalandırma, ısıtma, soğutma, kapı ve pencere kontrolleri vb. uzaktan mobil aygıtlar ya da bilgisayarlar ile internet üzerinden kontrol edilebilmektedir (Chan, Est`eve, Escriba, & Campo, 2008).

### 2.2.1 Uzaktan Erişim İçin Kullanılan Teknolojiler

Geçmişten günümüze akıllı evlerin ev dışarısından da kontrol edilebilmesi için internet (Deokar & Nagmode, 2014) ve SMS teknolojileri kullanılmıştır (Thinagaran, Sulaiman, Yatim Sharif, Ramli, & Leong, 2013). SMS özellikle bileşenleri fazla olan sistemler için çok uygun olmasa da internet bağlantısının çökmesi durumunda alternatif olarak sisteme ani müdahale amacıyla kullanılabilir.

Son zamanlarda yapılan çalışmalarda ise yaygın olarak internet kullanılmaktadır. Buna sebep olarak internetin gün geçtikçe yaygınlaşması (Tablo 2.1) ve uzaktan erişim için uygun bir altyapı sunması gösterilebilir (Internet Live Stats : Internet Users, 2015).

**Tablo 2.1:** Dünya çapında yıllara göre internet kullanımı.

Yıl (1 Ocak)	İnternet Kullanıcıları	Kullanıcı Artışı	Dünya Nüfusu	Nüfustaki Artış	İnternet Kullanıcıların Dünya Nüfusuna Oranı
2014*	2,925,249,355	7.9%	7,243,784,121	1.14%	40.4%
2008	1,562,067,594	13.8%	6,753,649,230	1.21%	23.1%
2004	910,060,180	16.9%	6,435,705,600	1.22%	14.1%
2000	413,425,190	47.2%	6,127,700,430	1.26%	6.7%
1996	77,433,860	72.7%	5,821,016,750	1.38%	1.3%
1993	14,161,570		5,578,865,110		0.3%

## 2.2.2 Aygıtlar Arası Erişim İçin Kullanılan Teknolojiler

Akıllı ev otomasyon sistemlerinde bileşenlerin birbirleri ile iletişimi için de tıpkı uzaktan erişimde olduğu gibi farklı iletişim teknolojileri farklı şekillerde kullanılmıştır. Bunlar:

- Kablolulu teknolojiler
  - KNX
  - Ethernet
  - X10
  - UPB
- Kablosuz teknolojiler
  - Wi-fi
  - Bluetooth low energy (BLE)
  - ZigBee
  - Z-Wave
  - Ant

KNX ev ve bina otomasyonlarında kullanılan açık sistem bir standarttır. 2002 yılında Avrupa'da üç farklı protokol kullanan üreticilerin bunu standartlaştırma isteğiyle ortaya çıkmıştır. Twisted Pair, Radyo Frekansı, Power Line ve IP/Ethernet gibi iletişim ortamlarını desteklemektedir (KNX Association, 2015). KNX bir yazılım ile bir akıllı ev otomasyon sistemi tasarlanmasına, görselleştirilmesine ve çalıştırmasına olanak verir. De Luca, Lillo, Mainetti, Mighali, Patrono ve Sergi (2013) evlerin Android tabanlı çalışan aygıtlardan kontrol edilebilmesi ile ilgili ev otomasyonu çalışmalarında, KNX teknolojisi kullanmışlardır. Lee ve Hong (2009) ise KNX tabanlı bir akıllı ev sistemlerini ZigBee protokolü kullanarak kablosuz hale getirmiştir. Akademik çalışmalar yanında birçok firma KNX teknolojisiyle uyumlu akıllı bina sistemlerinde kullanılacak ürünler üretmiştir (Manufacturers, 2015).

Ethernet bilgisayar ağları (LAN) için kullanılan bir teknolojidir. Ethernet IEEE 802.3 olarak standartlaştırılmıştır. Özellikle bilgisayar kontrollü akıllı ev otomasyonlarında, yüksek bant genişliği gerektiren noktalarda kullanılmaktadır. 10

Gbit bant genişliğini hâlihazırda desteklemektedir (IEEE 802.3™-2012 – IEEE Standard for Ethernet, 2015).

X10 akıllı evlerde kullanılan en popüler teknolojilerden biridir (Das, Cook, Bhattacharya, Heierman III, & Lin, 2003). Bu teknolojinin en büyük avantajı binada yer alan mevcut elektrik altyapısını kullanmasıdır. Ancak veri iletim hızı 20 bit/s olup bu yüzden ancak çok düşük bant genişliği gerektiren işlemler için kullanılabilir. Ayrıca bazı gürültü filtreleme araçları elektrik hattı üzerinden gönderilen X10 komutlarını filtreleyebilir (X10 Basics, 2015). UPB teknolojisi de tıpkı X10 gibi elektrik hatları üzerinden veri iletimini sağlayan bir teknolojidir. X10'dan farkı daha güçlü sinyal üreterek çalışmasıdır (Prindle, 2016).

Akıllı bina sistemlerinde bileşenlerin birbirleriyle kablosuz olarak iletişimi için mevcut duruma göre farklı teknolojiler kullanılmaktadır. Wi-fi yüksek bant genişliğine sahip ancak yüksek enerji tüketen, genellikle binalarda ve açık alanlarda bireylerin internete bağlanmak için kullandığı bir iletişim teknolojisidir. Wi-fi IEEE 802.11 protokolüne bağlı olarak standartlaştırılmıştır. ElShafee ve Ahmed (2012) çalışmalarında tamamen wi-fi tabanlı bir akıllı ev geliştirirken, farklı çalışmalarda wi-fi bağlantısı sadece belirli görevler için kullanılmıştır (Örn. cep telefonu ile wi-fi üzerinden binanın kontrolünü sağlamak vb.) (Lian, Hsiao, & Sung, 2013).

Bluetooth, Ericsson firması tarafından, 1994 yılında mobil cihazları birbirine kablosuz olarak bağlamak için RS-232'ye alternatif olarak geliştirilmiştir. Bu teknoloji 2.4 GHz ISM frekans bandında çalışmakta 54 Mbps'ye kadar veri aktarımı yapabilmektedir (Bluetooth, 2015). Ancak Bluetooth geliştiricileri en son sürümünde bant genişliğini artırmak yerine enerji verimliliğine yönelmiş ve bağlantı hızını düşürerek (0,3 Mbps) güç tüketimini olabildiğince aşağıya çekmiştir. Bu Bluetooth sürümünün ismi Bluetooth Low Energy (BLE) olarak adlandırılmıştır (Bluetooth Low Energy, 2015). Sriskartha, Tan ve Karande çalışmalarında bir evin ısıtma sisteminin kontrolünü Bluetooth teknolojisiyle gerçekleştirmişler ve bir evin tüm sistemlerinin kontrolünün nasıl gerçekleştirilebileceğini teorik olarak ispatlamıştır (Sriskanthan, Tan, & Karande, 2002). Yine benzer bir çalışma da Piyare ve Tazil cep telefonu ile kontrol edilen bir akıllı ev geliştirmişler ve bluetooth teknolojisi kullanmışlardır. Ev içerisinde yer alan ve sıcaklık, ışıklandırma vs. bileşenleri kontrol eden merkez bir kontrol aygıtı ile diğer birimler arasında kablolu iletişim

sağlamışlar, merkez kontrol aygıtının dışarıdan (cep telefonu) kontrolünü ise kablosuz Bluetooth teknolojisiyle gerçekleştirmişlerdir (Piyare & Tazil, 2011).

ZigBee küçük boyutlarda veri alışverişini gerçekleştiren, iletim hızı yavaş, maliyeti düşük, uygulanabilirliği ve genişletilmesi kolay, az güç tüketen bir kablosuz haberleşme teknolojisidir. ZigBee 802.15.4 protokolüne bağlı olarak standartlaştırılmıştır. Bu teknoloji günümüzde akıllı bina otomasyon sistemlerinde kullanılan en popüler kablosuz iletişim teknolojisidir (ZigBee Alliance, 2015). Jadhav vd. ZigBee teknolojisi kullanılarak bilgisayar ile bir kontrol kartı arasında kablosuz iletişim kuran ve kontrol kartı ile ev içerisinde yer alan bileşenleri yöneten (röle, motor, alarm) bir akıllı ev prototip çalışma gerçekleştirmişlerdir (Jadhav, Chaudhari, & Vavale, 2014). Li ve Ziang (2010) çalışmalarında bir önceki çalışmadan farklı olarak ev içerisinde yer alan merkez kontrolcü ile uç bileşenler (lamba kontrolcüsü vs.) arasındaki bağlantıyı kablosuz olarak ZigBee ile gerçekleştirmiştir.

Z-Wave binalardaki elektrik/elektronik cihazların birbirleri ve kullanıcılarla kablosuz ve kolayca haberleşmesine olanak sağlayan bir otomasyon standardıdır (ITU-T G.9959) (Z-Wave, 2015). Zigbee ve Z-Wave birbirine çok benzer özelliklere sahip olsa da Z-Wave kullanan tüm bileşenler aynı firma tarafından üretilen çipi kullandıkları için ZigBee de karşılaşılan uyumluluk sorunları Z-Wave de yaşanmamaktadır (Z-Wave vs Zigbee, 2015).

Ant, ZigBee ve Bluetooth gibi 2.4 GHz ISM (Industrial Scientific Medical) frekans bandında çalışmakta (birçok ülkede telsiz iletişimi için sertifika veya lisansa gerek olmadan belirli bir çıkış gücü sınırlamasına uyarak, üzerinden yayın yapılabilen banttır) olan bir sensör ağı teknolojisidir. Ticari akıllı ev sistemlerinde kullanılan teknolojilerden biridir. Son olarak Ant+ sürümü çıkmıştır (Ant Applying Technologies, 2015). Ant+ sürümü tıpkı Bluetooth teknolojisinde olduğu gibi günümüzde birçok akıllı cep telefonunda standart olarak kullanılmaktadır.

Insteon diğer teknolojilerden farklı olarak kablolu ve kablosuz olarak kullanılabilir. Özellikle akıllı ev sistemlerinde kullanılmak için geliştirilmiştir. Elektrik hatları üzerinden kurulan bağlantı ile kablosuz bağlantı arasında köprü görevini yerine getirmektedir. Bu teknolojiyi kullanan hâlihazırda birçok akıllı ev aygıtı bulunmaktadır (Prindle, 2016).

Kablosuz teknolojilerin teknik özellikleri Tablo 2.2’de enerji tüketimleri ise Tablo 2.3’te karşılaştırılmıştır (Dementyev, Hodges, Taylor, & Smith; Samosuyev, 2010; Pothuganti & Chitneni, 2014; Choperena, 2015; Frenzel, 2015). Tablolar incelendiğinde yüksek bant genişliği gerektiren (ses ve video iletimi) uygulamalarda wifi, düşük enerji tüketimi ve düşük bant genişliğinin yeterli olduğu uygulamalarda BLE (Bluetooth Low Energy), yine düşük bant genişliğinin yeterli olduğu ancak karmaşık bir ağ yapısına sahip olması gereken uygulamalarda ise ZigBee’nin uygun olduğu görülmektedir.

**Tablo 2.2:** Kablosuz iletişim teknolojilerinin teknik özellikleri.

Kablosuz Teknoloji	Bluetooth Low Energy	ZigBee	Ant+	Wi-Fi
Frekans Aralığı	2.4 GHz	868/915 MHz; 2.4 GHz	2.4 GHz	2.4 GHz; 5 GHz
En Yüksek Bant Genişliği	1 Mb/s	250 Kb/s	20 Kbps	54 Mb/s
Mesafe	10-100 m	10-100 m	10-30 m	100 m
Topoloji	Point-to-point, star	Point-to-point, star, tree, mesh	Point-to-point, star, tree, mesh	Point-to-point, star

**Tablo 2.3:** Kablosuz iletişim teknolojilerinin enerji tüketimleri.

Enerji Tüketimi				
Kablosuz Teknoloji		Bluetooth Low Energy	ZigBee	Ant+
Uyku Halinde		0,78 $\mu$ A	4,18 $\mu$ A	3,1 $\mu$ A
Uyanık Halde		4,5 mA	9,3 mA	2,9 mA
En Düşük (120 saniye aralıkta)		10,1 $\mu$ A	15,7 $\mu$ A	28,2 $\mu$ A
Ortalama Aralığı	Uyku	10.0 s	14.3 s	15.3 s

### 2.3 Günümüzdeki Ev Sistemleri ve Özellikleri

Günümüzde dünyanın en büyük teknoloji şirketleri olarak kabul gören Google, Apple, Samsung, Philips, Amazon vb. şirketler akıllı ev çalışmalarını yürütmektedir. Bunun yanında akademik olarak da birçok çalışma yürütülmektedir.

Google internete bağlı olarak çalışan ve öğrenebilen termostat ile gaz detektörü üreten Nest isimli firmayı 3,2 milyar dolarlık bir fiyata satın alarak bu alanda çalışmalarına başlamıştır. Öğrenebilen termostat çalışmaya başladıktan bir hafta sonra kullanıcı davranışlarını öğrenerek otonom olarak evin sıcaklığını ayarlamaya başlamaktadır. Google şu anda bu firmanın ürünlerini daha da ileri

götürerek yeni sürümlerini çıkarmış ve bu ürünlere bir ev güvenlik kamerası ile birlikte Google Home isimli evi ses ile kontrol etmeye yarayan bir ürün geliştirmiştir. Amazon'da benzer bir ürün olarak Echo'yu geliştirmiştir (Amazon Echo, 2016). Ayrıca birçok firmanın ürünlerinde olduğu gibi Google da farklı firmaların ürünlerini kendi akıllı ev sistemi bünyesinde kontrol edilebilmesini sağlamıştır (Nest, 2016).

Apple, diğer firmaların aksine Homekit isminde Apple ürünlerinde çalışan bir uygulama geliştirmiştir. Bu uygulama firmaların ürettiği akıllı ev donanımlarını sesli ve normal bir arayüz ile kontrol edebilmektedir (Homekit, 2016).

Samsung, Google benzeri bir hareketle Smartthings isimli bir şirketi satın alarak akıllı ev alanında çalışmalarına başlamıştır. Sistemleri merkezde yer alan bir kontrolcü ile hareket, nem, kapı pencere sensörlerini takip edebilmekte, akıllı priz elektrikli cihazları kontrol edebilmekte, aydınlatmayı kontrol eden cihazı ile bahçe ışıklarını kontrol edebilmektedir (Smartthings, 2016).

Philips, Hue isimli ürünü ile akıllı ev alanında en popüler ürünlerden birini geliştirmiştir. Bu aygıt uzaktan kontrol edilebilen farklı istenilen renkte yanabilen bir lambadır. Google ve Apple'ın başta olmak üzere birçok firmanın sistemlerinden bu aygıt kontrol edilebilmektedir (Meet Hue, 2016). Bu ürün benzeri birçok firmanın farklı ürünleri olduğu gibi birçok farklı akıllı ev sistemleri olmasına karşın geneli birbirine çok benzemektedir.

## **2.4 Akıllı Evlerde Yapay Zekâ**

Akıllı evlerde kullanıcı konforunu artırmak ve enerji tasarrufu sağlamak vb. işlemler için uzaktan erişim, tek bir aygıtle kontrol gibi hane halkının işini kolaylaştıran çeşitli çözümler üretilmiştir. Ancak bu çalışmaların tümünde en basit işlemler dahi insan kontrolü ile gerçekleştirilmek zorundadır. Örneğin bir evdeki bireyler eve gelindiğinde lambayı ve hava durumuna göre termostatı açmak, aynı şekilde evden çıktığımızda da bunları kapatmak zorundadır. Son yıllarda yapılan bazı çalışmalarda ise kullanıcıların yaptıkları işlemleri takip ederek, zaman içerisinde bu işlemleri yapay zekâ teknikleri ile öğrenen ve otonom hareket eden, enerji tasarrufu sağlayabilen, akıllı sistemler geliştirmeye başlanmıştır.



### 2.4.1 Yapay Zekâ

Kavramlar ve algılar yardımıyla soyut ya da somut nesnelere arasındaki ilişkiyi kavrayabilme, soyut düşünme, muhakeme etme ve bu zihinsel işlevleri uyumlu şekilde bir amaca yönelik olarak kullanabilme yetenekleri zekâ olarak adlandırılmaktadır. Zekânın farklı tanımlarının olmasına karşılık zekâyâ ilişkin kuramların tümü zekânın geliştirilebilecek bir kapasite ya da potansiyel olduğu ve biyolojik temellerinin bulunduğu noktalarında birleşir. Buna göre zekâ, bireyin doğuştan sahip olduğu, kalıtımla kuşaktan kuşağa geçen ve merkez sinir sisteminin işlevlerini kapsayan; deneyim, öğrenme ve çevreden kaynaklanan etkenlerle biçimlenen bir bileşimdir (Türkiye Zeka Vakfı, 2015).

Zekâ bir başka tanımda; yeni bir duruma, bu durumun unsurları arasında var olan ilişkileri kavrayarak uyum gösterme yetisi olarak tanımlanmaktadır. İnsan yeni bir duruma uyum sağlayabilmek için geçmiş bilgilerini ve deneyimlerini kullanmaktadır. Zekâ, geçmiş bilgiler ve deneyimler bir araya geldiğinde ise çevrede oluşan yeni duruma göre yeni bir tepki üretilmekte ve öğrenme gerçekleşmektedir. Öğrenme genel anlamda, çevresi ile etkileşimi sonucu kişide oluşan düşünce, duyu ve davranış değişikliği şeklinde tanımlanmaktadır (Özden, 2003). Örneğin doğduğunda konuşamayan ve konuşulanları anlamayan bir bebek, insanlar içerisinde yaşayarak çeşitli sesler duymakta ve bu sesler ile ilişkili olarak çevreden farklı tepkiler almaktadır. Zekâsı yardımıyla bebek bu gelen sesler ve tepkileri işleyerek zamanla seslerin anlamlarını kavrayabilmekte ve kendi ses tellerini kullanarak sesleri taklit edebilmektedir.

Günümüzde insan zekâsının bir benzerini bilgisayar ortamında oluşturarak bilgisayarların öğrenmesini amaçlayan “Yapay Zekâ” isminde bir araştırma alanı ortaya çıkmıştır (Nabiyev, 2012). Bu alan, insan zekâsı ve öğrenmesi örnek alınarak bilgisayar ortamında geliştirilen algoritmalar ile yapay öğrenmenin gerçekleşmesini hedeflemektedir. Böylece bilgisayarların da insanlar gibi yeni durumlara (dışarıdan yeni bir komut almadan) uyum sağlaması sağlanabilecektir. Yapay zekânın bir başka

hedefi de insanın kolayca yapabildiği konuşma ve görüntü tanıma gibi işlemleri bilgisayar ortamında gerçekleştirebilmektir.

Normal şekilde programlanmış bir bilgisayar, programında ne komut verilmiş ise sırasıyla verilen komutları gerçekleştirmektedir. Yeni bir durumla karşılaştığında ise, o yeni durum için programlama yapılmadığından dolayı herhangi bir işlem yapamamaktadır. Yeni bir işlem yapılması için bilgisayarın yeniden programlanması gerekmektedir. Yapay zekânın amacı da, böyle yeni bir durumla karşılaşan bilgisayarların tekrar programlamaya gerek duyulmadan otomatik olarak ne yapması gerektiğini tahmin ederek o işlemi yapmasıdır. Bilgisayarların bu tahmin işlemini gerçekleştirebilmesi için sahip oldukları yapay zekâ ile öğrenmesi gerekmektedir.

Bilgisayarların öğrenmesine yapay öğrenme (makine öğrenmesi) denilmektedir. Yapay öğrenme, bilgisayarların örnek veri ya da geçmiş deneyimi kullanarak bir ölçüte göre başarımlarını artıracak biçimde programlanması olarak tanımlanmıştır (Alpaydın, 2013).

Bilgisayarların öğrenmesinin farklı türleri ve hiyerarşik bir yapısı vardır. Birinci seviyede öğrenme programlı öğrenmedir. Verilen komutu uygulayan tipik bilgisayar yazılımları bu türdendir. İkinci seviyede öğrenme ezber öğrenmedir. Klasik koşullanma benzeri bu öğrenme de sistem değişen şartlara göre tepkisini de değiştirebilmektedir. Sistemin davranış değişikliği doğru ve yanlış sonuç verdiğiğinde aldığı tepkiler yardımıyla gerçekleşmektedir. Üçüncü seviye öğrenme istatistiksel öğrenmedir. Sistem birçok kez çalıştırılması sonucu elde ettiği istatistiksel verilere göre öğrenme gerçekleşmektedir. Dördüncü seviye örneklere göre öğrenmedir. Sistem mevcut örneklere göre genelleştirme yapılabilmekte ve böyle öğretmenli-danışmanlı öğrenme gerçekleşmektedir. Beşinci seviye öğrenme ise yeni bilgiler kazanarak öğrenmedir. Bu öğrenmede sistem yeni varsayımlar ve durumlar türeterek kendi kendine (danışmansız) öğrenebilmektedir. Şekil 2.2’de yapay öğrenmenin basamakları sunulmuştur.



Şekil 2.2: Yapay öğrenmenin hiyerarşik yapısı

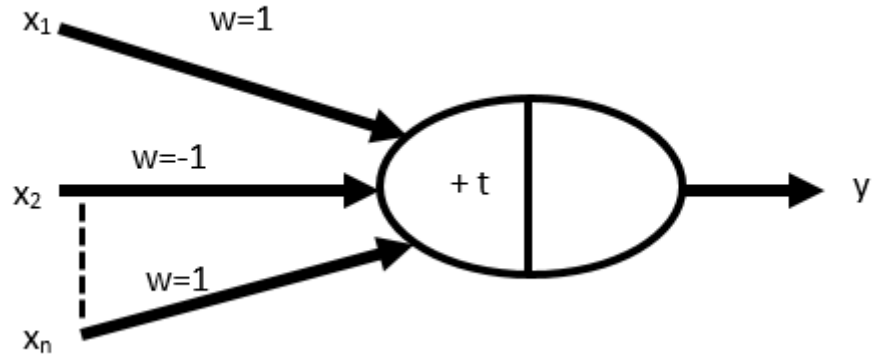
## 2.4.2 Yapay Zekâ Teknikleri

Uzun yıllardır yapılan yapay zekâ çalışmaları sonucunda farklı alanlarda kullanılmak üzere birçok yapay zekâ tekniği geliştirilmiştir.

### 2.4.2.1 Yapay Sinir Ağları

Yapay sinir ağları, insan beyninde yer alan sinir hücrelerinin çalışma mantığını temel alarak geliştirilmiş bilgisayar sistemleridir. Yapay sinir ağları sadece yazılım olabileceği gibi, sadece donanım ya da yazılım donanım bir arada hibrit sistemlerde olabilirler (Cinsdikici, 2016).

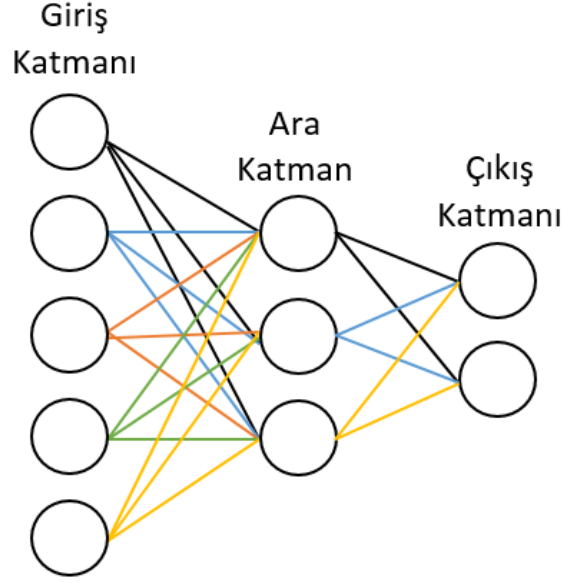
Beynin tam olarak nasıl çalıştığı günümüzde halen gizemini korumaktadır. Ancak beyni oluşturan milyarlarca sinir hücresinin (nöron) beynin çalışmasında önemli rol oynadığı tespit edilmiştir. 1943 yılında psikiyatrist Warren McCulloch ve matematikçi Walter Pitts ilk olarak bir nöronun matematik modellemesini yapmışlardır (Warren & Pitts, 1990). Bu model Şekil 2.3'te verilmiştir. Sonraki yıllarda teknolojinin gelişimi ile birlikte bu konuda yapılan çalışmalar da hız kazanmıştır.



Şekil 2.3: Basit nöron modeli

Günümüzde yapay sinir ağları sınıflandırma, modelleme ve tahmin olmak üzere birçok farklı alandaki sorunları çözmek için kullanılmaktadır. Ayrıca sıklıkla, sistem girdi verilerinin hatalı olmasının yüksek ihtimalli olduğu veya sorunu çözecek net bir algoritmanın geliştirilemediği durumlarda yüksek başarımlı sağladığı için kullanılmaktadır (Çayıroğlu, 2016).

Yapay sinir ağları yapay sinir hücrelerinin birbirine bağlanması ile oluşturulan yapılardır. Üç ana katmandan meydana gelmektedirler. Bunlar giriş, çıkış ve ara katmandır. Bu katmanlar Şekil 2.4'te sunulmaktadır.



Şekil 2.4: Yapay sinir ağının genel yapısı

Yapay sinir ağları giriş kanalından gelen verileri ağırlıklandırma katsayıları ile çarparak ara katmana iletirler. Ara katman kendisine bağlanan sinir hücrelerinden gelen verileri toplar ve kendi sahip olduğu eşik değere göre bir sonraki çıkış katmanına değerini ağırlıklandırma katsayısı ile çarparak gönderir. Çıkış katmanı da ara katmanın benzeri bir mantıkla kendisine gelen değerleri toplar ve net ismi verilen çıkış değerini üretir. Yapay sinir ağları şu şekillerde öğrenebilmektedir (Kriesel, 2016):

- Yeni bağlantılar kurarak,
- Mevcut bağlantıları silerek,
- Bağlantı ağırlıklarını değiştirerek,
- Eşik değerlerini değiştirerek,
- Temel fonksiyonları üzerinde değişiklikler yaparak,
- Yeni sinir hücreleri ekleyerek,
- Mevcut sinir hücrelerini silerek.

Literatürde yapay sinir ağlarının akıllı evlerde çok çeşitli şekillerde kullanıldığı belirlenmiştir. Chan, Hariton, Ringear ve Campo (1995) tarafından, yaşlı ve engelli bireylerin yaşam kalitesini artırmak amacıyla sıcaklık, ışık vb. sensörlerden veri toplama işlemini, bu bireyleri rahatsız etmeden gerçekleştiren bir

sistem kurulmuştur. Sistem yapay sinir ağı ile bireylerin hangi saatlerde neler yaptığını, hangi sıcaklık değerlerini tercih ettiğini öğrenmiş ve bu verilere göre alarm verme, sıcaklık ayarlama görevlerini akıllı evin otonom olarak gerçekleştirmesini sağlamıştır.

Bir başka çalışmada yine kullanıcı eylemlerini tanımak için yapay sinir ağlarından faydalanılmıştır. Çalışmada Washington Üniversitesi'nde yürütülmekte olan CASAS akıllı ev projesinden elde edilen sensör verileri kullanılmıştır. Sensör verileri zaman, aygıt ve aygıtın değerinden oluşmaktadır. Yapay sinir ağı, Geri besleme (Back-propagation – BP) algoritması kullanılarak on adet eylem için eğitilmiştir. Bu eylemler kahvaltı, öğle yemeği, evden ayrılma vb. günlük ev içerisinde rutin yapılan eylemlerdir. Altı farklı veri setinde bu eylemler için yapay sinir ağı çalıştırılarak eylemler tespit edilmeye çalışılmış ve sonuç olarak bir veri setinde %91,8 oranında başarılı sağlanmıştır. Ancak yapay sinir ağlarında kullanılan farklı nöron sayıları ve veri setinde tahmin edilmeye çalışılan olayla ilgili yeterli veri olup-olmaması durumunun başarımı etkilediği tespit edilmiştir. Ayrıca son olarak veri setleri Naive Bayes ve Saklı Markov Modelleri ile de test edilmiş ve alınan sonuçlar yapay sinir ağlarından elde edilen sonuçlar ile karşılaştırılmıştır. Yapay sinir ağlarının %91,8 doğru tespit oranıyla en başarılı olduğu belirlenmiş ancak farklı veri setlerinde saklı Markov modellerinin ve Naive Bayes'in de yapay sinir ağlarından daha başarılı olabileceği tespit edilmiştir (Fang, He, Si, Liu, & Xie, 2014).

Ev sakinlerinin bir sonraki yapacakları hareketi belirlemek için yapılan başka bir çalışmada da geri beslemeli yapay sinir ağlarından faydalanılmıştır. Bu çalışmada bireylerin bir sonraki konumlarını belirlemek için 1 gizli katmanlı geri beslemeli yapay sinir ağı kullanılmıştır. Çalışmada bir sonraki adımı tahmin etmek için yerel ve evrensel olmak üzere iki yöntemden yararlanılmıştır. Birinci yöntemde, yalnızca kullanıcı hareketlerini takip eden ve tahmin eden bir yapay sinir ağı kullanılmış; ikinci yöntemde ise bir işyerinde çalışan beş elemanın tümünün verilerini kullanarak tahmin yapan bir yapay sinir ağı kullanılmıştır. Sonuçlarda sadece tek bir kişi için çalışan yapay sinir ağının daha başarılı olduğu tespit edilmiştir. Bunun sonucu olarak da çok kişi için çalışan yapay sinir ağının bireylerin arasındaki hareket bağıntılarını ilişkilendirememesi gösterilmiştir. Çalışmada kullanılan yöntemin %92 oranlarında başarılı tahminler ortaya koyduğu tespit edilmiştir (Vintan, Gellert, Petzold, & Ungerer, 2004).

Ev güvenliğinin ele alındığı farklı bir çalışmada, farklı kültürlerde ev güvenliği için farklı önlemlerin alınması gerektiği çünkü farklı kullanıcı davranışlarının olabileceği ve bu ihtiyaca karşılık olarak kendi kendine öğrenen bir algoritma geliştirilmesi gerekliliğine vurgu yapılmıştır. Bunun için çalışmada yapay sinir ağlarından yararlanılmıştır. Çalışma, diğer yapay sinir ağları çalışmalarına benzer şekilde üç aşamada gerçekleştirilmiştir. Bunlar eğitim aşaması, çapraz doğrulama aşaması ve test aşamasıdır. Eğitim aşamasında hareket, kızıl ötesi vb. sensörlerden gelen veriler nümerik verilere dönüştürülmüştür. Bilindiği üzere yapay sinir ağları nümerik verilerle çalışabilmektedir. Yapay sinir ağı gizli katmanlar aracılığıyla eğitilmiştir. Girilen veriler sonucunda yapay sinir ağı güvenlik açığı tespit edildi-edilmedi şeklinde tek tip çıkış üretmektedir. Yapay sinir ağı binlerce veri seti kullanılarak iteratif olarak eğitilmiştir. Çapraz karşılaştırma aşamasında eğitimden elde edilen sonuçlar ve beklenen sonuçlar karşılaştırılarak sistemin performansı ve doğruluğu ölçülmüştür. İstenen sonuçlar elde edildiğinde eğitim işlemi bitirilmiştir. Son olarak test aşamasında eğitilmiş olan yapay sinir ağı gerçek hayat verileri kullanılarak test edilmiştir. Sistemin test sonuçlarında %80 oranında başarılı olduğu tespit edilmiştir (Teoh & Tan, 2010).

Mozer (1998), yapay sinir ağları kullanarak kullanıcı ihtiyaçlarını takip edebilen ve bunlara cevap verebilen ve ayrıca enerji tasarrufu sağlayan ACHE (Adaptive Control of Home Environments – Ev Ortamlarının Uyarlamalı Kontrolü) isminde gerçek bir binada kurulmuş olan farklı bir akıllı ev sistemi geliştirmiştir. Bu sistem ile kullanıcının istediği sıcak su, aydınlatma, ısıtma ihtiyaçları otonom şekilde giderilirken, yapılan ayarlamalarla minimum enerji kullanılması sağlanmıştır.

Akıllı evlerde eylem tanıma, bir sonraki eylemi tahmin etme vb. işlemler yanında yan ya da yardımcı görev olarak nitelendirilebilecek yüz tanıma gibi işlemlerde de yapay sinir ağlarından yararlanılmaktadır. Zuo ve With (2005) yılında gerçekleştirdiği çalışmada yapay sinir ağlarından faydalanan akıllı evlerde kullanılacak gerçek zamanlı çalışan bir yüz tanıma sistemi geliştirmiştir. Sistem çalışma sonucunda %92 oranında başarımla elde etmiştir.

#### 2.4.2.2 Bulanık Mantık

Bulanık kelimesi “net olmayan” anlamına gelmektedir (Türk Dil Kurumu, 2016). Bulanık mantık (Fuzzy Logic) ise 1965 yılında Lütü Aliasker Zade tarafından ortaya atılmış bir durumun sadece evet-hayır, siyah-beyaz, 0-1 gibi kesin sonuçları olması yerine gri tonlara da sahip olabileceğini savunan bir yaklaşımdır (Zadeh, 1965).

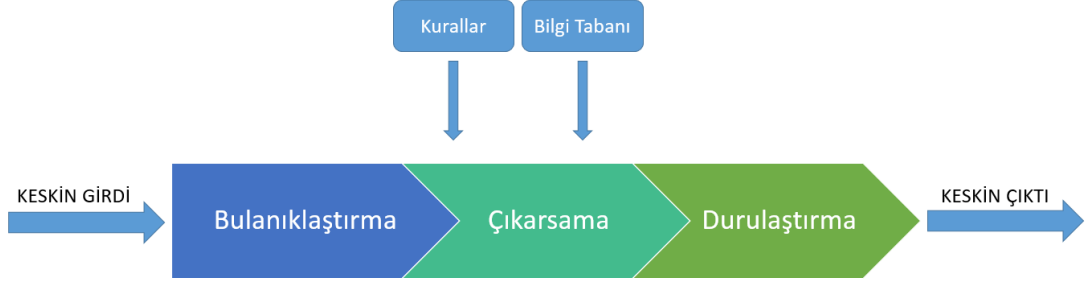
Bulanık mantık, küme teorisine dayanan matematiksel bir disiplindir. Klasik kümelerde bir eleman bir kümenin ya üyesidir ya da değildir. Bulanık kümelerde ise bir eleman birden fazla kümenin üyesi olabilir. Örneğin hava sıcaktır ya da soğuktur yerine bulanık mantık ile çok sıcak, sıcak, ılık, soğuk, çok soğuk gibi ara değerlerle durum daha açıklayıcı bir şekilde ifade edilebilmektedir.

Bulanık mantığın genel özellikleri ilk ortaya çıkaran Zade bu durumu şu şekilde açıklamıştır (Elmas, 2010);

- Bulanık mantıkta kesin değerler yerine, yaklaşık değerlere göre düşünme kullanılır.
- Bulanık mantıkta her şey  $[0,1]$  aralığında belirli bir derece ile gösterilir.
- Bulanık mantıkta bilgi büyük, küçük, çok az gibi dilsel terimlerle ifade edilir.
- Bulanık mantıkta çıkarım işlemi tanımlanmış kurallara göre yapılır.
- Matematiksel modeli çok zor elde edilen sistemler için uygundur.
- Bulanık mantık tam olarak bilinmeyen ya da eksik bilgi girişinin olabileceği durumlarda işlem yapmaya uygundur.

Bulanık mantık sınıflandırma, kümeleme ve karar verme süreçlerinde kullanılmaktadır. Bulanık mantık ile kurulan sistemin genel yapısı Şekil 2.5'teki gibidir.





Şekil 2.5: Bulanık sistemin genel yapısı

Bulanık sistemlere öncelikle sisteme kesin girdiler (37cm, 10kg vb.) veri olarak girilir. Bu girdiler bulanıklaştırma biriminde dilsel niteleyiciler olan sembolik değerlere dönüştürülür. Bulanıklaştırma işlemi sistemi tasarlayanlar tarafından belirlenen kurallara göre, girilen sayısal değere en küçük, çok büyük gibi dilsel değişken değerler verir. Bu aşamadan sonra veri çıkarılma birimine iletilir. Çıkarılma biriminde belirlenmiş kurallara göre çıkarım yapılmaktadır. Çıkarılma işlemi kurallar tablosuna ve bilgi tabanında yer alan verilere göre gerçekleştirilmektedir. A, B, C giriş, D çıkış olan bir sistem için;

$$\text{EĞER } A=x, B=y, C=z \text{ ise } D=i$$

şeklinde kuralları olan bir tabloya göre çıkış bulanık değeri hesaplanmaktadır. Son olarak durulaştırma birimi ise, tıpkı sisteme girildiği gibi kesin bir çıktı verebilmek için kendisine gelen bulanık veriyi kesinleştirmektedir. Bunun için ağırlık merkezi yöntemi, maksimum üyelik yöntemi, ağırlık ortalaması yöntemi, mean-max üyelik yöntemleri vb. yöntemler kullanılmaktadır.

Literatür incelendiğinde bulanık mantığın akıllı ev sistemlerinde genellikle ısıtma sistemlerinin kontrolü için kullanıldığı görülse de aydınlatma ve havalandırma sistemlerinin kontrolü, aktivite tanıma gibi amaçlarla kullanıldığı da belirlenmiştir.

Sıcaklık sistemleri aç-kapat, doğrusal kestirimci kontrol ve PID ile kontrol edilmektedir. Ancak bulanık mantık ile bu yöntemlere göre daha uygun sıcaklık geçişleri ve daha düşük sıcaklık sapmaları sağlanabilmektedir. Ayrıca bulanık mantık ile daha kararlı ve karmaşık matematiksel modellemelere ihtiyaç duyulmayan, enerji tasarrufu sağlarken, bireylerin konforunu artıran sistemler oluşturulabilmektedir. Kobersi, Finaev, Almasani ve Abdo (2013) geliştirdikleri MATLAB simülasyonu ile

bulanık mantığın avantajlarını ispat eden bir çalışma ortaya koymuşlardır. Yine benzer bir çalışmada bir yıllık ev dışı sıcaklık verileri takip edilmiş, daha sonra kurulan bulanık mantık sistemiyle evin kapı ve penceresinin açıklık durumu ile dış sıcaklık oranlarına göre ev içerisindeki ısıtıcının çalışma düzeyi belirlenmiştir (Mol, Chinthamani, Kamini, & Sugashini, 2012). Bir başka çalışmada ise yine benzer bir şekilde klimaların soğutma işlemleri için bulanık mantıktan faydalanılmıştır (Dash, Mohanty, & Mohanty, 2012). Bu çalışmalara ek olarak sıcaklık ile birlikte nem oranını da kontrol eden bir başka çalışmada da yine bulanık mantıktan yararlanılmıştır (Yas & Das, 2013).

Bulanık mantık ısıtma ve soğutma sistemleri dışında aydınlatma kontrolü için de kullanılmıştır. Kiyanfar, Lotfibonab ve Lotfibonab (2012) bir ortamda bulunan kişi sayısına göre kaç tane aydınlatma cihazının çalıştırılması gerektiğini belirleyen bir çalışma yapmışlar ve bu çalışmalarında kullanıcı memnuniyetini sağlayarak başarılı olmuşlardır (Kiyanfar, Lotfibonab, & Lotfibonab, 2012).

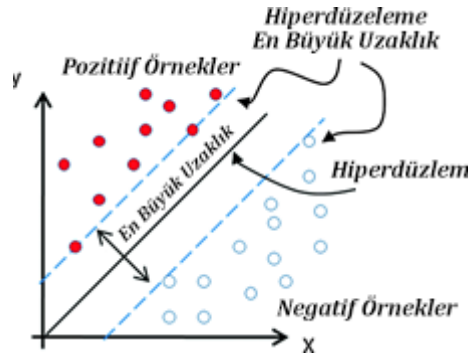
Yukarıda yapılan çalışmalardan farklı olarak Vainio, Valtonen ve Vanhala (2008) yılında yapmış oldukları çalışmalarında bir akıllı ev ortamını, orada yaşayan kişilerin tercihlerine göre değiştirebilecek bir kontrol sistemi geliştirmişlerdir. Bu kontrol sistemi ile akıllı ev ortamındaki ekipmanların ileriye yönelik kontrolü sağlanmakta ve bu sayede kişilerin tercihlerindeki değişimler çok daha önceden tahmin edilebilmektedir. Başka bir çalışmada ise Medjahed, Istrate, Boudy ve Dorizzi (2009) tarafından, bir ev ortamındaki aktivitelerin tanınması için birçok sensörü bir arada kullanan bulanık mantık ile çalışan bir sistem geliştirilmiştir. Bulanık mantığın kullanılması ile birlikte bu sistem, sağlık hizmeti konusunda önemli olan çeşitli günlük yaşamsal aktivitelerin tanınmasını sağlamıştır.

#### **2.4.2.3 Destek Vektör Makinaları (Çekirdek Makinaları)**

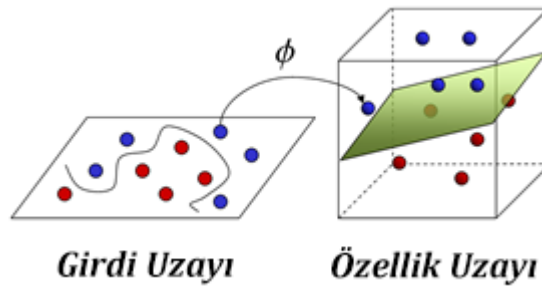
Destek vektör makinaları Vapnik ve Cortes (1995) tarafından iki sınıfın gruplandırması için kullanılabilecek yeni bir öğrenme yöntemi olarak ortaya atılmıştır. Bu öğrenme yönteminde doğrusal girdi vektörleriyle doğrudan işlem yapılabilirken, doğrusal olmayan girdi vektörleri daha üst boyutlarda doğrusal girdi

vektörlerine dönüştürülür. Son olarak elde edilen doğrusal vektördeki veriler düzlem üzerine çizilen bir hiperdüzlem ile sınıflandırılabilir.

Destek vektör makinaları sınıflandırma ve regresyon (bağlanım) problemlerini çözmekte kullanılır. Tıpkı yapay sinir ağlarında olduğu gibi destek vektör makinalarında da sistemin eğitimine ihtiyaç duyulmaktadır. Eğitimdeki amaç girdi vektörünü doğrusal olarak istenilen şekilde ikiye ayırabilen fonksiyonun belirlenmesidir (Şekil 2.6). Doğrusal olarak ayıramayan durumlarda ise çekirdek fonksiyonları kullanılarak üst boyutta verilerin doğrusal olarak ayrılabilirdiği bir boyuta veriler Şekil 2.7'deki gibi dönüştürülür (Çomak, 2008). Destek vektör makinalarının bir önemli özelliği de kendisine en yakın noktalar arasındaki uzaklığı en büyüğe çıkartan hiperdüzlemi bulabilmektir.



Şekil 2.6: DVM ile doğrusal sınıflandırma



Şekil 2.7: Çekirdek fonksiyonu ile boyut dönüşümü

Destek vektör makinaları kullanılan hasta ve yaşlılara yönelik bir akıllı ev çalışmasında, destek vektör makinalarından eylem tanıma ve yaşlı ve hastaların takip edilmesinde faydalanılmıştır. Sunulan bu aktivite tanıma yaklaşımında yaklaşan aktiviteler kategorilere ayrılmış ve bu kategoriye daha önce yapılan işlemler ve tahminler sonucunda atanmış skora göre değerlendirme yapılmıştır. Bu değerlendirme ile birlikte hatalı işlem oranı azaltılmıştır. Geliştirilen yöntem

temizlik, kahvaltı, telefonla görüşme, el yıkama vb. birçok günlük yaşam aktivitesi içeren farklı veri setlerinde sınanmış ve başarılı sonuçlar elde edilmiştir. Çalışmada son olarak geliştirilen yöntem Naive Bayes, J48 Karar Ağacı vb. yöntemlerle karşılaştırılmış ve diğer yöntemlere göre çok daha başarılı olduğu belirlenmiştir (Fahad, Khan, & Rajarajan, 2015).

Bir başka çalışmada yaşlı bakım sistemlerinin takip edilmesinde önceki çalışmadan farklı olarak ses tanıma için destek vektör makinalarından faydalanılmıştır. Sweet-Home isiminde yaşlı ve bakıma muhtaç insanlar için sıkıntılı durumları tespit etmek amacıyla ses tabanlı olarak hazırlanmış bir projenin parçası olan bu çalışmada, destek vektör makinaları ile Gaussian Karma Modelleri (Gaussian Mixture Models – GMM) kullanılarak ses tanıma işlemleri gerçekleştirilmiştir (esneme, gülme, su sesi, el çırpma vb. sesler). Çalışma sonucunda destek vektör makinaları daha başarılı olmuş ve %75 oranında ortalama başarı sağlamıştır (Sehili, Istrate, Dorizzi, & Boudy, 2012).

Yaşlı bireylere yönelik bir başka çalışmada gerçek bir ev kızıl ötesi hareket sensörleri, mikrofonlar, sıcaklık ve nem sensörleri ile donatılmıştır. Ayrıca kişilere giyilebilir sensörler bağlanılarak buldukları konumlar, hareket yolları ve yürüyüş sıklıkları takip edilmiştir. Tüm bu sensörlerden elde edilen veriler bir araya getirilerek bireylerin günlük yaşam aktiviteleri (temizlik, tuvalet kullanımı, yemek, dinlenme, uyuma) destek vektör makinaları aracılığıyla sınıflandırılmıştır. Sınıflandırmalar sonucunda en düşük temizlik %64,3, en yüksek yemek %97,8 oranında doğru olarak tahmin edilmiştir (Fleury, Vacher, & Noury, 2010).

#### **2.4.2.4 Markov Modelleri**

Markov modeli istatistiksel bir modeldir. Bu modelin daha iyi anlaşılabilmesi için modelle ilgili, rastgele değişken, stokastik süreç, Markovien özellik kavramlarının tanımlanması gerekmektedir. Rastgele değişken tesadüfi değişken anlamına gelmektedir. Stokastik süreç ise bu tesadüfi değişkenlere bağlı olan süreç olarak tanımlanmaktadır. Bir akıllı ev otomasyonu için düşünüldüğünde ev sakininin eve her geldiği saat tesadüfi değişken, on günlük eve geliş verisi ise stokastik bir süreci ifade etmektedir (Ulutaş & Özer, 2000). Markov özelliği, mevcut durum

verildiğinde, gelecek durumların geçmiş durumlardan bağımsız olduğu anlamına gelmektedir. Son olarak Markov modeli (Markov zinciri) ise Markov özelliğe sahip skotastik bir süreci ifade eder.

Markov modellerine göre bir durum belirli bir istatistiksel değere göre değişir veya değişmeden aynı kalır. Ayrıca geçmiş durumların mevcut durum üzerinde bir etkisi söz konusu değildir. Ancak şimdiki durum gelecek durumları etkileyebilir.

Markov modellerinden akıllı ev sistemlerinde genel olarak kullanılan model Saklı Markov modelidir (Hidden Markov Model). Normal Markov modellerinde durumlar gözlemci için görünebilir ancak saklı Markov modellerinde durum doğrudan görülemez ancak mevcut duruma bağlı çıkışlar görülebilir. Saklı Markov modelleri, sinyal işleme, ses, el yazısı ve vücut hareketleri tanımada, kripto analizlerinde, genetikte ve daha bir çok alanda kullanılmaktadır. Akıllı ev sistemlerinde ise ev sakinlerinin mevcut durumlarından sonra bir sonraki adımlarını tahmin etmek amacıyla kullanılmışlardır.

Akıllı evler için bir karar verme modülü oluşturmaya çalışan Babakura, Sulaiman, Mustapha ve Perumal (2014) Saklı Markov modelinden yararlanmışlardır. Çalışmada kamera, ev alarm sistemi, sesli çağrı sistemi ve enerji yönetim sisteminden oluşan bir akıllı ev sisteminde üretilen 6000 kayda sahip bir veri seti kullanılmıştır. Saklı Markov modellerinde önemli bir adım olan eğitim Baum Welch algoritması ile gerçekleştirilmiştir. Daha sonra saklı Markov modeliyle her tahmin edilmesi istenen olay ya da eylem için mevcut durumun daha önce belirlenmiş durumlarla en çok benzerlik taşıyanı Viterbi algoritmasıyla hesaplanmıştır. Çalışma sonucunda %95,7 oranında bir doğrulukla sistem tahminde bulunmuştur.

Günlük yaşamda bir odada hareket, sıcaklık, su akımı, ısıtıcı, telefon kullanımı ve telefon defteri kullanımı gibi verilerin sensörler aracılığıyla takip edildiği bir çalışmada, telefon kullanımı, el yıkama, yemek hazırlama, yemek ve ilaç kullanımı ve temizleme gibi işlemlerin elde edilen verilerle çıkarımı yapılmaya çalışılmıştır. Bunun için Saklı Markov modelin ilk aşamasına nazaran daha başarılı olan ikinci aşama Saklı Markov modelden faydalanılmıştır. Birinci aşama saklı Markov modelde sonraki durumların olasılığı ancak şu anki duruma bağlıyken ikinci aşama Saklı Markov modellerde şu anki ve ondan önceki iki durum da değerlendirmeye alınmaktadır. Böylece özellikle yanlış verilerin bulunduğu

durumlarda daha başarılı bir tanıma işlemi gerçekleştirilebilmektedir. Bu çalışmada da üç farklı veri seti ile çalışılmış ve her 3 veri setinde de ikinci aşama Saklı Markov model daha başarılı sonuçlar vermiştir (Zhang & Zhang, 2013). Saklı Markov modeli ile yapılan bu çalışmada, kullanıcı eylemleri kümelenerek görevler tanımlanmış ve bu kümeler geliştirilen Görev Tabanlı Saklı Markov Modeli (Task Based Markov Model - TMM) ile işlenerek kullanıcının bir sonraki adımı tahmin edilmeye çalışılmıştır. Öncelikle çalışma için sanal veriler üretilmiştir. Üretilen üç veri setinde toplamda 2825 günü kapsayan 30000 veri bulunmaktadır. Kullanılan sistemdeki eylem verileri etiketlenmemiş olduğu için danışmansız öğrenme yöntemi ile bu eylemlerin neler olduğu keşfedilmiştir. Bunun için tüm eylemler saat ve tarih verilerine göre sıralanmıştır. Daha sonra belirlenen kurallara uyan eylem grupları kümelenmiş ve bu eylemler bir görev olarak tanımlanmıştır. Son olarak belirlenen kümeler geliştirilen Markov Modeli'ne girdi olarak verilmiş ve tahmin işlemleri gerçekleştirilmiştir.

#### 2.4.2.5 Naive Bayes Teoremi

Naive Bayes, Bayes kuramını temel alan bir yaklaşımdır. Bayes kuramı, bir olayın olması ihtimalinin, o olayla ilgili başka koşullara bağlı olabileceğini savunmaktadır. Bayes teoremi aşağıdaki (2.1) denklemle ifade edilir;

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.1)$$

$P(A|B)$  ; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır.

$P(B|A)$  ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır.

$P(A)$  ve  $P(B)$  ; A ve B olaylarının önsel olasılıklarıdır.

Bayes kuramı az sayıda koşulun bulunduğu durumlarda kullanışlıdır. Ancak koşul sayısı arttıkça matematiksel hesaplamalar oldukça karmaşık bir hale gelmektedir. Naive Bayes ise koşulların birbirinden bağımsız olduklarını varsayarak

bu soruna bir çözüm getirmektedir. Naive Bayes sınıflandırma sorunlarını çözmek için kullanılmaktadır.

Akıllı evlerde engelli ya da çeşitli sağlık sorunları olan kişilerin takibi için yapılan bir çalışmada ev içerisinde yer alan hareket sensörlerinden gelen veriler ve aydınlatma için kullanılan anahtarlar takip edilmiştir. Daha sonra Naive Bayes kullanılan bir olasılıkçı öğrenme çerçevesi geliştirilerek sensör verilerine dayalı aydınlatma anahtarlarının kullanım tahminleri yapılmıştır (Wilson, Wolfe, & Atkeson, 2008).

#### **2.4.2.6 Karar Ağaçları**

Karar ağaçları, bir kararın tüm olası sonuçlarını ağaç dallarına benzeyen bir yapıda gösteren grafik modellerdir. Bu model karar alıcıya karar alırken hangi aşamaları takip etmesi, bu aşamalarda hangi faktörlerin göz önüne alması gerektiği ve geçmiş kararlar ve faktörler arasında nasıl bir ilişki olduğunun belirlenmesi konularında yardımcı olur. Bunun yanında karar ağaçları, algoritmaların ifade edilmesinde kullanılan alternatif bir yöntemdir (Cornel CIS Computer Science, 2016).

#### **2.4.2.7 Çok Ajanlı Sistemler (MAS: Multi Agent Systems)**

Çok ajanlı sistemler (Multi Agent Systems - MAS) birden çok ajanın (etmen) birbirleriyle etkileşim halinde çalıştığı sistemlere denmektedir. Buradaki ajan sahip olduğu sensörler ile ortamda bulunan bilgileri toplayan ve gerektiğinde duruma göre tepki verebilen nesnelere sahiptir. Çoklu ajan sistemleri dağıtık bilgisayar sistemleridir. Ancak klasik dağıtık bilgisayar sistemlerinin aksine zekidirler (Singh & Huhns).

Çok ajanlı sistemler genellikle akıllı evlerde donanım mimarisi kurgulanırken kullanılmaktadır. Çoklu ajan sistemlerinin akıllı ev sistemlerinde kullanıldığı bir çalışmada yedi ajan geliştirilmiştir. Bunlar; bulaşık makinası, kahve makinası, su ısıtıcı, robot, ısıtıcı, klima ve geri kalan işlemler için bir ajandadır. Bu ajanların çalıştığı bir simülasyon tasarlanmıştır. Simülasyonda ajanlar kullanılarak kaynak

harcamanın koordine edilmesi sağlanmıştır. Örneğim tüm odalarda 25°C sıcaklık istendiğinde, farklı odalarda yer alan ısıtıcı ve klimalar sıcaklık geçişlerini kontrol ederek bu sıcaklığı sağlayabilmektedir. Bunun yanında, su ısıtıcı sürekli olarak istenilen en büyük ve en küçük aralıklarda belirli miktarda suyun sürekli olacak sıcak kalmasını sağlamaktadır (Lesser, ve diğerleri, 1999).

Genel olarak akıllı evler kullanıcılar tarafından kontrol edilen ve kullanıcıların verdiği emirleri uygulayan evlerdir. Ancak akıllı evlerin kendi kontrollerini yaparak kendi başlarına karar alarak hareket edebilmelerini sağlayabilmek mümkündür. Bu tür evler yapay zekâ teknikleri kullanılarak geliştirilebilmektedir.

Akıllı evler yapay zekâ teknikleri kullanarak;

- kullanıcı takibi,
- yapılacak işleme karar verme,
- bir sonraki durumu tahmin etme,
- sistemdeki değişiklikleri takip etme,
- günlük olayları belirleme

işlemlerini yapabilmektedir (Reaz, 2013).

## 2.5 Yapay Veri Üretme

Akıllı ev sistemleri, yapay zekâ teknikleri ile işlemleri otomatik olarak gerçekleştirebilmek için kullanıcının daha önceki gerçekleştirdiği işlemleri takip ederek kayıt altına alırlar. Daha sonra bu verileri kullanarak kullanıcı davranışlarını belirleyerek bir sonraki adımı tahmin ederler. Ancak bu noktada bazı zorluklarla karşılaşmaktadır. Öncelikle yapay zekâ algoritmalarının tam olarak doğru çalışması için sürekli olarak sınanması ve iteratif olarak geliştirilmesi gerekmektedir. Bunu yapabilmek için kullanıcının ya da kullanıcıların ev içerisinde yeterli bir süre takip edilmesi ve verilerinin kaydedilerek veri setlerinin oluşturulması gerekmektedir. Ancak kayıt esnasında oluşabilecek her türlü aksaklık (kişinin hasta olması, acil durumların ortaya çıkması vb. rutini bozabilecek her türlü olay), veri setlerinin içerisinde gürültülere neden olacaktır ve yapay zekâ algoritmalarının düzgün



çalışmasını engelleyecektir (Heierman & Cook, 2003). Ayrıca bu kayıt işlemleri için hâlihazırda tüm işlevleri ile çalışan bir ev otomasyon sistemine ve hedef kitleye uyan örnek bireylere ihtiyaç vardır. Ek olarak, gerçek ortamda elde edilecek veriler bir ailenin özel yaşamı hakkında bilgiler içereceğinden çeşitli güvenlik sorunlarının doğmasına neden olabilir (Jarmin, Louis, & Miranda, 2014). Böyle durumlar için araştırmacılar, gerçek veriler ile çalışmak yerine yapay veriler ile çalışmayı tercih etmektedirler.

Yapay veriler, doğrudan ölçüm ile değil de sanal olarak üretilen ve istenilen duruma uygulanabilen verilerdir (Hill & Parker, 2002). Bu veriler, yapay veri üreten yazılımlar ile üretilmektedir (Korel, 1990).

Literatürde yapay veri üreten yazılımlar üzerine çeşitli çalışmalar bulunmaktadır. Arasu, Kaushik ve Li (2011) istenilen verileri esnek bir şekilde özel algoritmalar kullanarak üreten bir yazılım geliştirmişlerdir. Bu yazılım ile belirlenen bir senaryo üzerinden örneğin bir muhasebe firmasının girdi, çıktı, sipariş, müşteri, tedarikçi vb. bilgileri otomatik olarak üretilir. Ayrıca elde gerçek bir firmanın verileri varsa bu veriler maskelenerek benzer yapay veriler üretilmektedir.

Bir başka çalışmada ise endüstri düzeyinde (terabaytlar boyutunda) yapay veri üreten bir yazılım geliştirilmiş ve bu işlemi hızlı gerçekleştirebilmek için paralel programlama tekniklerinden faydalanılmıştır. Yazılım, XML tabanlı olarak çalışmaktadır. Belirli aralıklarda yüzdesel olarak rastgele değerler ve belirtilen oranlarda tekrarlanan değerler üretilmektedir. Yapılan üretim işlemlerinde işlemcilerin farklı çekirdeklerinden aynı anda faydalanarak (paralel programlama tekniği) çok sayıda veriyi daha hızlı bir şekilde üretilmektedir (Hoag & Thompson, 2007).

Farklı bir çalışmada araştırmacılar, esnek olarak farklı alanlarda kullanılacak geniş kapsamlı yapay veriler üretmeye yarayan DGL (Data Generation Language) isimli bir dil geliştirmişlerdir. Bu dilin yapısı, genel bir programlama dili yapısına benzemektedir. Fonksiyonlar aracılığıyla tablolar, alanlar ve bu alanlar için istenilen sayıda ve farklı türlerde veriler üretilmektedir (Bruno & Chaudhuri, 2005).

Akademik alıřmaların yanında yapay veri reten ticari yazılımlar da bulunmaktadır (CA Test Data Manager, 2016; Generatedata, 2016; SQL Data Generator, 2016). Ancak literatürde yapılan arařtırmalarda, akıllı ev otomasyonu sistemlerine yönelik yapay veri retmek iin zelleřtirilmiř herhangi bir akademik ya da ticari alıřmaya rastlanamamıřtır.

### 3. YÖNTEM

Bu çalışmanın amacı öğrenebilen, modüler, düşük enerji tüketimli ve web tabanlı bir akıllı ev sistemi geliştirilmesidir. Bundan önceki bölümlerde günümüze kadar geliştirilmiş olan akademik ve ticari akıllı ev sistemleri, bu sistemlerin yapabildikleri ve bu sistemlerde kullanılan yapay zekâ teknikleri incelenmiştir. İncelemeler sonucunda nihai hedef olarak geliştirilmesi planlanan akıllı ev sisteminin genel yapısı, işlevleri ve sahip olacağı yapay zekâ belirlenmiştir.

Bu bölümde geliştirilen akıllı ev otomasyon sisteminin geliştirme süreci, genel yapısı, işleyişi, kullanılan donanımlar ve teknolojiler, web arayüzü, merkez sunucu, işletim sistemi, aygıtlar arası iletişim ve yapay zekâ algoritmaları sırasıyla sunulacaktır.

#### 3.1 Sistemin Geliştirme Süreci

Akıllı ev sistemi geliştirilmesi için öncelikle mevcut akıllı ev sistemleri incelenmiştir. Böylece bir akıllı ev sisteminin gerçekleştirilmesi gereken işlemleri ve sahip olması gereken sistem yapısı belirlenmiştir. Daha sonra mevcut olarak akıllı ev sistemlerinde kullanılan ve kullanılabilecek teknolojiler belirlenmiştir. Donanımlar, teknolojiler ve sistemin gerçekleştirilmesi gereken işlemler belirlendikten sonra farklı sistemlerde hangi işlemlerin otonom olarak ne şekilde gerçekleştirildiği araştırılmıştır. Ayrıca genel olarak yapay zekâ teknikleri incelenmiştir. Tüm bu araştırmalar sonucunda akıllı ev sisteminin yapay zekâ yardımı ile gerçekleştirilmesi gereken işlemler ve hangi yapay zekâ teknikleri ile gerçekleştireceği belirlenmiştir.

Geliştirilecek olan akıllı ev sisteminin teorik altyapısı oluşturulduktan sonra sistemin başlangıç noktası olan web arayüzü geliştirilmiştir. Daha sonra web arayüzünün sürekli olarak çalışacağı ve sistemin merkez noktasını oluşturan merkezi sunucusu gerekli yazılımlarla birlikte kurulmuş ve gerekli ayarlamaları yapılmıştır. Bu işlemten sonra kişilerin web arayüzünden yaptığı işlemleri sonuçlandıracak ve

temel akıllı ev işlemleri gerçekleştirecek olan oda kontrolcüsü ve diğer akıllı ev donanımları geliştirilmiştir.

Donanımsal altyapının tamamen hazırlanmasının ardından sistem belirli bir süre çalıştırılarak test edilmiştir. Testlerin başarılı şekilde geçilmesinden sonra yapay zekâ işlemlerine geçilmiştir. Yapay zekâ işlemleri için sistemde gerçekleşen işlemlerin geçmiş kayıtlarına ihtiyaç vardır. Bu amaçla sistemdeki tüm birimlerin ve bu birimlerde gerçekleşen işlemlerin ne şekilde kayıt altına alınacağı ve nasıl saklanacağı belirlenmiştir.

Verilerin şablonları belirlendikten sonra yapay zekâ çalışmaları için zemin hazırlanmış olmuştur. Çünkü yapay zekâ algoritmaları öğrenme işlemlerini kaydedilmiş veriler ile gerçekleştirmektedir. Ancak yapay zekâ algoritmalarının geliştirilebilmesi için veri şablonları tek başına yeterli değildir. Kaydedilmiş geçmiş verilere de ihtiyaç vardır. Bu durum için sanki geliştirilecek olan akıllı ev sistemi geliştirilmiş, bir aile tarafından belirli bir süre kullanılmış ve verileri kayıt altına alınmış gibi kaydedilmiş sanal kayıtlar üretilecek bir yapay kayıt üretici sistem geliştirilmiştir. Daha sonra yapay kayıt üretici üzerinde sanal bir aile oluşturulmuş, bu aile için çeşitli senaryolar belirlenmiş ve bir yıl boyunca mevcut akıllı ev sistemi bu sanal aile tarafından kullanmış gibi bir yıllık sanal kayıtlar oluşturulmuştur. Daha sonra bu veriler kullanılarak bireylerin eve geliş zamanları, ısıtma tercihleri vb. işlemleri gerçekleştiren yapay zekâ algoritmaları geliştirilmiştir.

Yapay zekâ algoritmalarının geliştirilmesinin ardından, bu algoritmaların sonuçlarının gerçek zamanlı olarak çalıştığına nasıl çalıştığı, performans değerleri, doğru tahmin oranlarını ayrıca rutin olaylar yanında rastgele olaylara karşı nasıl tepki verdiğini test edebilmek için yapay kayıt üretici sistem için oluşturulan ailenin davranışlarını anlık olarak simüle eden bir akıllı ev simülasyonu geliştirilmiştir. Bu simülasyon ile sanal ailenin hareketleri anlık olarak izlenebilmekte, istenilen vakitlerde rastgele olaylar gerçekleştirilebilmektedir. Ayrıca yapay zekâ algoritmaları simülasyonda anlık olarak çalışmakta ve böylece anlık olarak yapay zekâ algoritmalarının nasıl çalıştığı gözlemlenebilmektedir.

Son olarak tüm sistemin ve yapay zekâ algoritmalarının gerçek ortamda nasıl çalıştıklarının test edilebilmesi için akıllı ev simülasyonu gerçek akıllı ev otomasyon sistemi ile entegre ederek akıllı ev hibrit simülasyon geliştirilmiştir. Bu hibrit

simülasyon ile sanal olarak üretilen ailenin yanına gerçek bir birey ve sanal olarak üretilen eve gerçek bir oda eklenmiştir. Böylece gerçek bir odada yaşayan gerçek bir kişinin tüm işlemleri ve bu kişi için çalışan yapay zekâ işlemleri simülasyon penceresinde takip edilebilir hale gelmiştir. Bu hibrit simülasyon ile tüm sistemler ve algoritmalar zaman içerisinde takip edilerek iteratif olarak geliştirilmeye devam edilerek nihai hedefe ulaşılmıştır.

### **3.2 Akıllı Ev Sisteminin Gerçekleştirilmesi Gereken İşlemler**

Mevcut akıllı ev otomasyon sistemleri incelendiğinde çok farklı işlemleri gerçekleştirebildikleri belirlenmiştir. Bu işlemlerden daha önce literatür bölümünde ayrıntılarıyla bahsedilmiştir. Ayrıca bir tablo ile bu işlemler sunulmuştur (Bkz. Şekil 2.1).

Bu çalışmada geliştirilecek olan akıllı ev sisteminin gerçekleştirilmesi gereken işlemler belirlenirken yapay zekâ algoritmalarının birbirinden farklı olarak ne tür işlemleri gerçekleştirilmesi gerektiği göz önüne alınmıştır. Ayrıca mevcut sistemlerin gerçekleştirdiği temel işlemlere örnek teşkil edecek işlemleri de gerçekleştirilmesi gerektirdiğine karar verilmiştir. Bunların sonucunda akıllı evin tür olarak gerçekleştirilmesi gereken işlemler şunlar olarak belirlenmiştir:

- Analog işlemler
- Dijital işlemler
- Etki tepki işlemleri

Analog işlemler ev içerisinde belirli aralıkta değerlerin kullanıcı tarafından belirlenerek gerçekleştirildiği işlemlerdir. Geliştirilen akıllı ev otomasyon sisteminde odanın aydınlık düzeyinin kontrol edilebilmesini sağlayan “Aydınlık Kontrol Bileşeni” ve oda sıcaklığının kontrol edilebilmesini sağlayan “Termostat Bileşeni” analog işlemlere örnektir. Kullanıcı aydınlık için 0-100 aralığında, sıcaklık içinse 16-32 aralığında bir değer belirleyebilmektedir. Yapay zekâ ise, bu noktada kullanıcının genel tercihlerini geçmişte yapmış olduğu analog işlem kayıtlarına bakarak tahmin etmektedir. Sonuç olarak analog tahminleri gerçekleştiren yapay zekâ algoritması

daha sonra akıllı eve eklenebilecek farklı analog işlemler için de kullanılabilir (Örn: perde kontrolü).

Dijital işlemler kullanıcı tarafından sadece aç-kapat emirlerinin verilebildiği işlemlerdir. Geliştirilen akıllı ev otomasyon sisteminde odada bulunan prizin kontrol edilebilmesini sağlayan “Priz Kontrol Bileşeni” dijital işlemlere örnektir. Kullanıcı bu işlemde sadece aç ya da kapat emirleri vererek prizi açıp kapatabilmektedir. Yapay zekâ analog işlemlerde olduğu şekilde kullanıcı davranışlarının kayıtlarına bakarak kullanıcının açma ya da kapatma işlemlerinin zamanlarını öğrenmekte ve otonom olarak işlemi gerçekleştirmektedir. Sonuç olarak bu geliştirilen yapay zekâ algoritması sisteme eklenebilecek farklı dijital işlemler içinde kullanılabilir (Örn: cam kontrolü).

Etki tepki işlemleri ise pasif yapay zekâ gerektiren işlemlerdir. Kullanıcı bir işlemi gerçekleştirdiğinde ya da sensörlerden bir veri alındığında otomatik olarak aynı tepki verilmektedir. Örnek olarak bu sistemde oda içerisinde hareket algılandığında kart okuyucu bileşen etrafındaki ışıklar otomatik olarak yanmaktadır. Bu noktada sistem bir yapay zekâ işlemi gerçekleştirmekte ancak bir öğrenme işlemi gerçekleştirmemektedir. Sonuç olarak akıllı ev sistemi bu işlemi gerçekleştirerek buna benzer farklı işlemleri de kolaylıkla gerçekleştirebileceğini ispat etmiş olacaktır (Örn: koridorda otomatik yanan lambalar).

Analog, dijital ve etki tepki işlemleri yanında son yıllarda geliştirilen akıllı ev sistemlerinin en temel özelliklerinden biri de farklı üreticiler ya da araştırmacılar tarafından geliştirilmiş olan akıllı ev bileşenlerinin de mevcut sistemler tarafından kullanılabilmesidir (Örneğin Google’ın NEST isimli termostatu, Philips’in HUE isimli ayarlanabilir lambaları vb.). Geliştirilecek olan akıllı ev sisteminin de hem bu tip farklı üreticilerin geliştirdikleri aygıtları kullanabilmesi hem de bu aygıtları öğrenebilen hale getirebilmesi için Google firmasının geliştirmiş olduğu NEST isimli termostat ısıtma aygıtı olarak mevcut sistemde kullanılmıştır.

### **3.3 Sistemin Genel Yapısı, İletişim Şekilleri ve İşleyişi**

Akıllı ev sisteminin genel yapısı tasarlanırken çoklu ajan sistemleri (multi agent systems) örnek alınmıştır. Çoklu ajan sistemleri birbirleriyle etkileşimde olan

birçok ajandan oluşan sistemlerdir. Ajanlar yazılımlar ya da robotlar gibi özerk birimlerdir. Bencil ya da işbirlikçi olarak sistemin ortak hedefleri ve kendi bireysel hedefleri doğrultusunda amaçlarına ulaşmaya kadar sürekli olarak çalışırlar. Bu çalışma için çoklu ajan sistemlerinden uyarlanarak merkeze bağlı ancak bağımsız olarak da çalışabilen ajanlardan oluşan bir akıllı ev sistemi yapısı tasarlanmıştır.

Geliştirilen akıllı ev otomasyon sistemi çeşitli donanımlar ve bu donanımlar üzerinde çalışan çeşitli yazılımlar bir araya getirilerek oluşturulmuştur. Genel yapı şu şekilde ifade edilebilir: kullanıcı web arayüzü ile sistemi kontrol etmektedir. Yani kullanıcı lamba açmak istediğinde, priz kapatmak istediğinde ya da termostatı ayarlamak istediğinde bu işlemi sistemin kontrol noktası olan web arayüzü ile gerçekleştirmektedir. Web arayüzü bir internet sayfasıdır. Bu internet sayfası merkezi sunucu üzerinde kurulu olan bir web sunucu yazılım ile çalıştırılmaktadır.

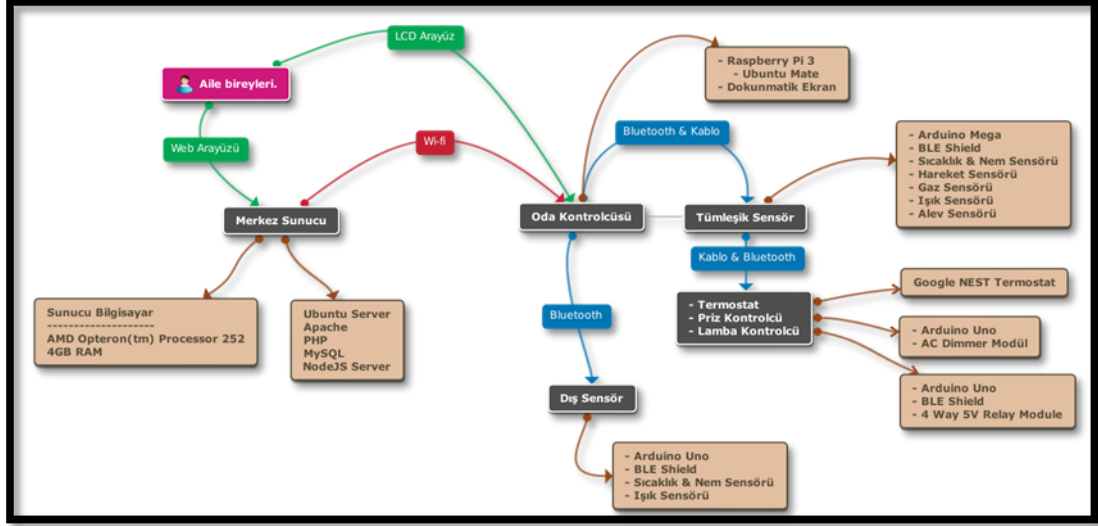
Merkez sunucu sürekli ve internete bağlı olarak çalışan fiziksel olarak herhangi bir yerde bulunabilecek bir sunucu bilgisayardır (Mevcut sistemde üniversite sunucu odasında bulunmaktadır). Bu sunucu web arayüzünde kullanıcı tarafından yapılan işlemleri kendisine internet üzerinden bağlı olan ilgili oda kontrolcüsüne iletmektedir.

Oda kontrolcüleri sistemin çoklu ajan sistemlerinde bahsedilen ajanlardır. Bu ajanlar web arayüzünden sunucu aracılığıyla kendilerine gelen istekleri tümleşik sensör aracılığıyla termostat bileşeni, aydınlatma bileşeni ve priz bileşenine iletmektedir. Bu bileşenlerde de kendilerine gelen talepleri yerine getirmektedir. Ayrıca tümleşik sensör bileşeni topladığı sensör verilerini oda kontrolcüsüne, oda kontrolcüsü de aynı verileri internet bağlantısı ile merkez sunucuya iletmektedir. Merkez sunucu da bu verileri web arayüzü ile kullanıcıya sunmaktadır.

İnternet bağlantısının bulunmadığı durumlarda oda kontrolcüsü kendi üzerinde yer alan arayüz ile kişilerin akıllı ev sistemini kontrol edip kullanabilmesini sağlamaktadır. Ayrıca daha önce almış olduğu emirlere göre otonom olarak çeşitli temel yapay zekâ işlemlerini de gerçekleştirebilmektedir.

Tüm bunların dışında kullanıcı tarafından gerçekleştirilen işlemler, sensörlerden elde edilen veriler, kullanıcı ile aygıt bilgileri ve her türlü diğer veriler merkez sunucuda kayıt altına alınmaktadır. Daha sonra bu veriler yapay zekâ

işlemlerinde kullanılmaktadır. Yapay zekâ işlemleri ise merkez sunucu üzerinde gerçekleştirilmektedir. Sistemin genel görünümü Şekil 3.1’deki gibidir (Daha geniş görünüm için: EK B, Şekil B.1).



Şekil 3.1: Sistemin genel yapısı

### 3.4 Web Arayüzü

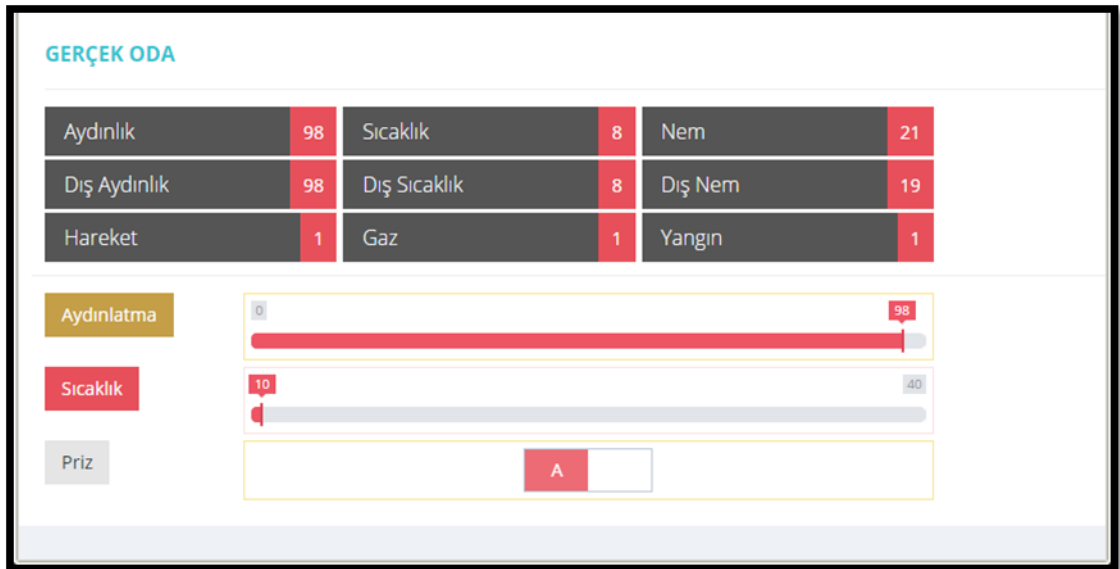
Sistemin başlangıç noktası kullanıcılarıdır. Kullanıcılar ev içerisinde yer alan aygıtları her yerden kolaylıkla kontrol edebilmelidir. Bunun için kullanıcının tüm sistemi kontrol edebileceği bir web arayüzü geliştirilmiştir. Sistemin kontrolü için web arayüzü tercih edilmesinin sebebi günümüzde hemen her aygıtın web arayüzünü sorunsuz gösterebilmesidir (cep telefonu, tablet, bilgisayar vb.). Farklı çalışmalarda sistemin kontrolü için, masaüstü uygulamalar, mobil uygulamalar ve özelleştirilmiş kontrol aygıtları kullanılmıştır. Ancak tüm bunların farklı dezavantajları vardır. Masaüstü uygulamalar sadece geliştirilen işletim sisteminde ya da masaüstü bilgisayarlarda, mobil uygulamalar ise sadece geliştirildiği platformlarda çalışabilmektedir. Web arayüzü ise hem masaüstü bilgisayarlarda, hem de mobil aygıtlarda çalışabilmektedir. Ayrı bir kontrol aygıtı kullanılması durumunda ise evin her bireyi için ayrı bir kontrol aygıtına ihtiyaç duyulacak ve ev dışından sistemin kontrolü mümkün olmayacaktır.

Web arayüzü html, css ve javascript programlama dilleri ile modern görünüme ve kolay bir kullanıma sahip olacak şekilde geliştirilmiştir. Arayüzde



gösterilen veriler anlık olarak güncellenmektedir. Ayrıca kullanıcı bir işlem yaptığında genel olarak web formlarında olan gönder düğmesi gibi bir düğmeye tıklamaya gerek kalmadan anında yapılan işlem merkez sunucuya iletilmektedir.

Tasarlanan web arayüzü her zaman erişilebilir olabilmesi için sürekli olarak internete bağlı olan merkez sunucu üzerinde çalıştırılmaktadır. Böylece ev içerisinde web arayüzünü yayınlayacak, sürekli olarak çalışan bir bilgisayara ihtiyaç duyulmamaktadır. Bunun yanında eğer akıllı ev sistemi birden fazla ortam için kullanılırsa ek bir web sunucusu kullanımına gerek duyulmayacaktır. Mevcut merkez sunucu yüzlerce web arayüzünü kapasitesinde barındırabilecektir. Web arayüzünün görünümü Şekil 3.2’teki gibidir.



Şekil 3.2: Web arayüzü

Web arayüzü yanısıra kullanıcıların internete ulaşamadıkları durumlarda akıllı ev otomasyon sistemini kontrol edebilmesi için oda içerisinde de bir dokunmatik ekran bulunmaktadır. Bu dokunmatik ekran oda kontrolcüsüne bağlı olarak çalışmaktadır. İnternet bağlantısının sağlanamadığı durumlarda ya da kolayca sistem kontrol edilmek istendiğinde bu ekran kullanılabilir. Dokunmatik arayüz kullanıcıların kolayca bu arayüzü de kullanabilmesi için web arayüzü ile tamamen aynı tasarıma sahip olacak şekilde geliştirilmiştir.

### 3.5 Merkez Sunucu

Merkez sunucu aktif internet bağlantısı olan ve sürekli olarak çalışan bir sunucu bilgisayardır. Tüm akıllı ev sisteminin merkezidir. Merkez sunucu, “Server” olarak tabir edilen sunucu olarak özelleştirilmiş bir bilgisayardır. Üzerinde AMD Opteron(tm) Processor 252 işlemci, 4 GB RAM ve RAID 1 teknolojisi ile yansılama yapılmış 250 GB’lık 2 adet SAS Disk yer almaktadır (Şekil 3.3).



Şekil 3.3: Merkez Sunucu

RAID (Redundant Array of Independent Disks) teknolojisi “Bağımsız Disklerin Artıklıklı Dizisi” olarak tanımlanmaktadır. RAID ile aynı veri birden fazla diske aynı anda ya da bir veri dizisi farklı disklere aynı anda yazılabilmektedir (RAID Modlar, 2015). Sistemde kullanılan RAID 1 teknolojisi ile sunucuya kaydedilen veriler otomatik olarak 2 diske aynı anda kaydedilmektedir. Böylece disklerde oluşabilecek herhangi bir arıza sonucunda sistem çalışmaya devam edecek, ayrıca herhangi bir veri kaybı yaşanmayacaktır. Özellikle yapay zekâ algoritmalarının geçmişteki kullanıcı verilerini işleyerek çalıştığı göz önüne alındığında verilerin kaybolmaması çok önemlidir. Sistemin sürekli olarak çalışabilmesi için elektrik kesintilerine karşın sistemin UPS’ye bağlı olması sağlanmıştır.

Sunucu bilgisayarın işletim sistemi olarak Ubuntu Server seçilmiştir. Ubuntu Server sunucular için özelleştirilmiş, Linux tabanlı bir işletim sistemidir. Ubuntu en yaygın olarak kullanılan Linux dağıtımıdır (Ubuntu, 2015).

Merkezi sunucu üzerinde sistemin ara yüzünü oluşturan web sayfalarının yayınlanabilmesi için bir web sunucu yazılımı kullanılmıştır. Web sunucu yazılımı olarak sistemin sürekli olarak sorunsuz çalışabilmesine olanak veren, dünya genelinde en çok kullanılan yazılım olan Apache kullanılmıştır. Apache tüm web sunucuların %56'sında kullanılmaktadır (Usage of web servers for websites, 2015). Ayrıca Apache web sunucusuna PHP programlama dili ile hazırlanmış web sayfalarını çalıştırabilmesi için PHP kurulmuştur.

Sistemdeki kullanıcı verilerinin, hareket kayıtlarının vb. tüm verilerin düzenli bir şekilde kaydedilmesi ve okunabilmesi için MySQL veritabanı kullanılmıştır. MySQL çoklu iş parçacıklı (multi-threaded), çok kullanıcılı (multi-user), hızlı ve sağlam (robust) bir veritabanı yönetim sistemidir (MySQL Nedir?, 2015).

Kullanıcıların web ara yüzünü kullanarak yaptığı ayarların (lamba parlaklığı, sıcaklık, priz) ve merkez sunucunun diğer bileşenlerle iletişiminin sağlanabilmesi için NodeJs Server kullanılmıştır. Node.js, Chrome web tarayıcısının da üzerinde çalıştığı V8 javascript motoru üzerinde, açık kaynaklı, sunucu tarafında çalışan, ağ ilintili uygulamalar için geliştirilmiş bir çalışma ortamıdır (Taş, 2015). Ayrıca sunucunun diğer birimlerle iletişiminin olabilecek en kısa sürede sağlanması ve düşük bant genişliğinin yeterli olması için Node.js kütüphanelerinden Socket IO kullanılmıştır. Socket IO Node.js Server üzerinde farklı tarayıcılarla sorunsuz olarak soket iletişim kurabilen bir kütüphanedir (socket.io, 2015). Bu kütüphane oda kontrolcü ile merkez sunucu arasındaki bağlantı ve iletişim için de kullanılmaktadır.

Kullanıcılar web arayüzünde bir işlem yaptıktan sonra işlem önce merkez sunucuya iletilir ve kayıt altına alınır. Bu işlem daha sonra merkez sunucuya hâlihazırda bağlı olan ilgili oda kontrolcüsüne soket bağlantı ile gönderilir.

### **3.6 Oda Kontrolcü**

Oda kontrolcüsü ev içerisindeki bir odada tüm sensörlerden gelen verilerin toplanıp merkezi sunucuya gönderildiği, çevrimdışı durumlarda yapay zekâ işlemlerinin uygulandığı, oda içi bilgilerin oda içerisinde dokunmatik LCD ekran aracılığıyla kullanıcılara gösterilmesinin sağlandığı, kullanıcıların da oda ile ilgili işlemleri yapabildiği ve oda giriş çıkış işlemlerinin gerçekleştirildiği birimdir. Oda

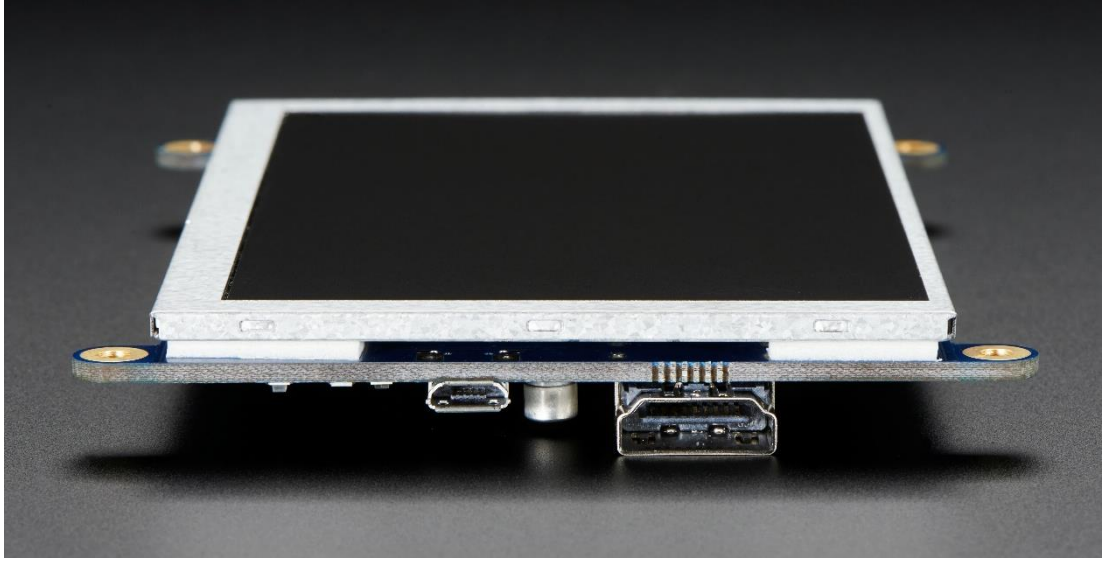
kontrolcleri tm sistemin bařarılı bir řekilde alıřması iin duruma gre zerk duruma gre merkez sunucuya baėımlı olarak alıřan ajanlardır.

Oda kontrolcs donanım olarak Raspberry Pi (řekil 3.4) ve ona baėlı bir dokunmatik LCD ekrandan oluřmaktadır. Rapsberry Pi 3 1.2 GHz 4 ekirdekli 64-bit ARM Cortex-A53 iřlemci, 1 GB Ram ile birlikte dhili wi-fi ve bluetooth modllerine sahiptir (Raspberry Pi 3 Model B, 2016). Bu iřlemci sayesinde zerinde eřitli Linux trevi iřletim sistemleri alıřabilmektedir. Ayrıca HDMI ıkıřına sahiptir. Bu ıkıř ile HDMI giriřine sahip btn grnt aygıtlarıyla alıřabilmektedir.



**řekil 3.4 :** Raspberry Pi 3

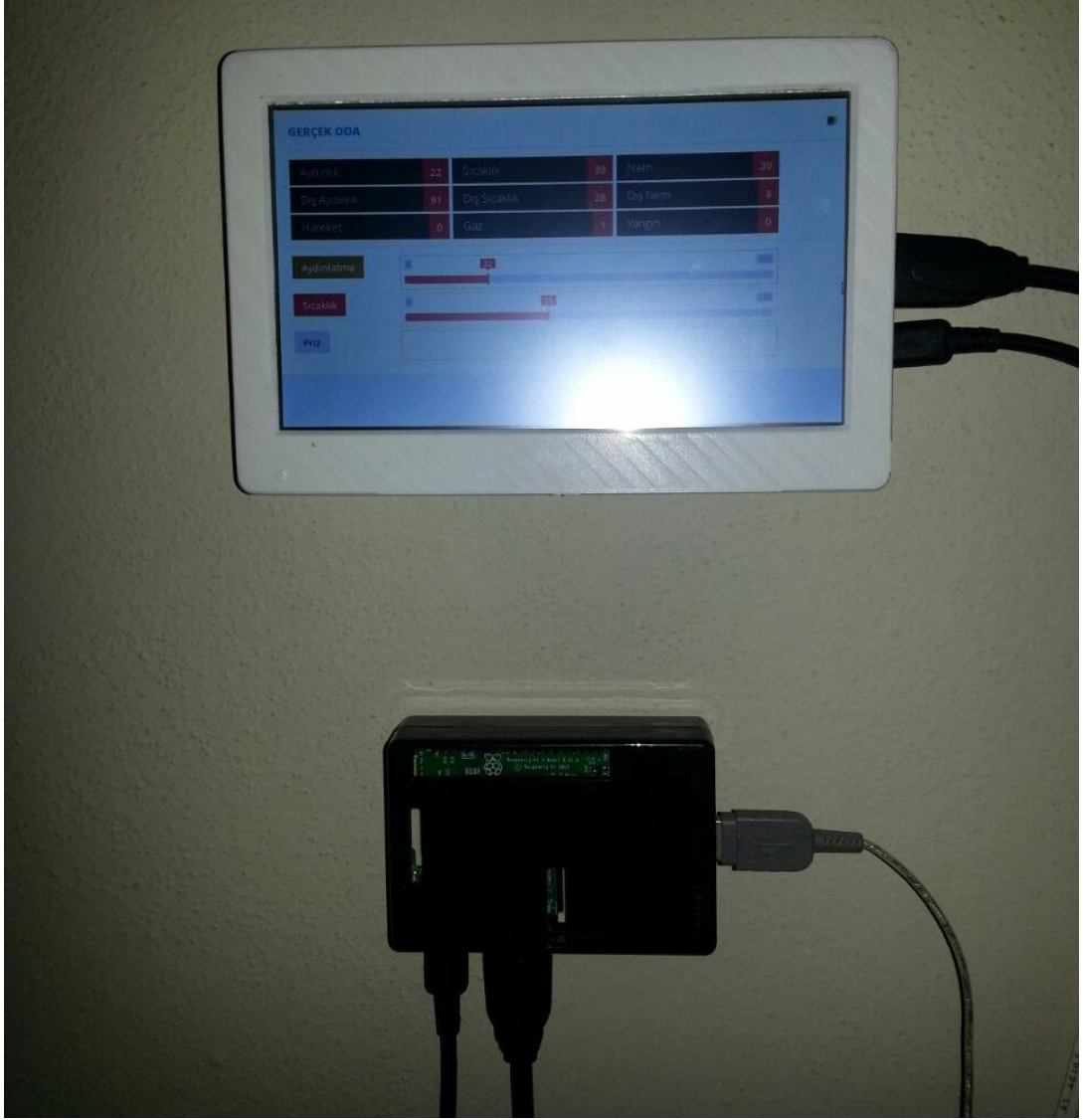
Oda ierisindeki bilgileri bir akıllı telefonun ya da bilgisayarın web arayzne bakmadan ev sakinlerinin takip edebilmelerini ve eřitli iřlemleri yapabilmelerini saėlamak amacıyla alıřmada odadaki tm iřlemleri takip eden Rapsberry Pi'ye HDMI yoluyla baėlanabilen dokunmatik bir ekran kullanılmıřtır (řekil 3.5).



**Şekil 3.5** : LCD dokunmatik ekran

Bu ekran 800\*480 çözünürlüğe ve dokunmatik özelliklere sahiptir. Raspberry Pi'ye hem HDMI hem de USB aracılığıyla bağlanarak, hem görüntüyü vermekte, hem de üzerinde ev sakinlerinin çeşitli ayarlar yapabildiğini sağlamaktadır (Adafruit 5" and 7" 800x480 TFT HDMI Backpack, 2016).

Oda kontrolcüsü yani Raspberry Pi 3 aslında mini bir bilgisayardır. İşletim sistemi olarak Ubuntu Mate kullanılmıştır. Bu işletim sistemi Mate masaüstü ortamının ARM işlemcilerde çalışabilen Ubuntu işletim sistemine uyarlanmış halidir. Ubuntu daha önce de bahsedildiği üzere açık kaynak kodlu bir Linux dağıtımdır. İşletim sistemi yanında oda kontrolcüsü üzerinde merkez sunucu ile iletişimde kullanılan NodeJs Server yazılımı, web arayüzünün yayını için NGinx web sunucusu, yapay zekâ işlemlerinin ve diğer işlemlerin kaydının tutulması için SQLite veritabanı yazılımı kullanılmıştır. Bu noktada benzer yazılımlar olan Apache yerine NginX, MySql yerine SQLite kullanılmasının sebebi oda kontrolcüsünde daha düşük işlem gücü bulunması ve alternatif yazılımların çok daha az sistem kaynağı dolayısıyla sistemin daha az enerji harcamasıdır. Sistemin çalışır hali Şekil 3.6'da gösterilmektedir.



**Şekil 3.6:** Oda Kontrolcü

Oda kontrolcüsü merkez sunucu ile diğer donanımsal birimler arasında köprü görevi görürken her iki tarafla çift yönlü iletişim kurmaktadır. Son olarak kullanıcıdan merkez sunucuya gelen işlem buradan, oda kontrolcüsüne iletilmiştir. Bu işlem oda kontrolcüsünden ilgili birçok noktaya gidebilmektedir. Çünkü oda içerisindeki bütün birimler doğrudan ya da dolaylı olarak oda kontrolcüsüne bağlıdır.

Oda kontrolcüsüne bağlı olarak ev içerisinde işlem yapan 2 birim vardır. Bunlar tümleşik sensör ve dış sensör bileşenleridir. Kullanıcı tarafından yapılan ve oda kontrolcüsüne kadar iletilen işlem bu noktadan sonra tümleşik sensör bileşenine iletilmektedir.

### 3.7 Tmleřik Sensr Bileřeni

Tmleřik sensr bileřeni oda ierisinde bulunan tm sensr ve iřlem yapan birimlerin (aydınlama, priz) baėlı olduėu bileřendir. Oda kontrolcsne usb baėlantı ile baėlıdır. Ancak baėlantı gerekli olması durumunda bluetooth ile kablosuz olarak da saėlanabilmektedir.

Tmleřik sensr bileřeni, donanım olarak bir Arduino Mega ve ona baėlı olan BLE Shield (Bluetooth Low Energy Shield) ile sıcaklık, nem, ıřık gaz, alev ve hareket sensrlerinden oluřmaktadır. Ayrıca tmleřik sensr bileřenine oda giriř ıkıř iřlemlerini takip etmeye yarayan bir NFC (Near Field Communication, Yakın Alan İletiřimi) kart okuyucu baėlanmıřtır.

Arduino Mega 2560 (řekil 3.7); Atmega2560 temelli bir mikro denetleyici kartıdır. Kartın zerinde 54 adet dijital giriř/ıkıř pini ve 16 analog giriř bulunmaktadır (Arduino MEGA 2560, 2016). Sensrler ve kart okuyucu baėlantıları ile diėer bileřenler bu giriř/ıkıř portlarına baėlanmaktadır.



řekil 3.7: Arduino Mega

Akıllı bir evde ev ierisinde olan durumları ve evreyi iyi bir řekilde takip edebilmek ok nemlidir. nk ne kadar ok ve saėlıklı bilgi elde edilebilirse bu gerekleřtirilecek yapay zekâ iřlemlerinin o denli bařarılı olmasını saėlayacaktır. Bunun iin bu alıřmada ev ierisine birok sensr yerleřtirilmiřtir. Bunlar ıřık, sıcaklık ve nem, hareket, gaz ve alev sensrleridir.



### 3.7.1 Işık Sensörü

Bu çalışmada 5mm boyutunda bir LDR kullanılmıştır. LDR (Ligh Dependent Resistor) ışığa bağlı olarak direnci değişen bir dirençtir. LDR Arduino uyumlu bir kart (Şekil 3.8) aracılığıyla Arduino Mega'ya bağlanmıştır.



Şekil 3.8: 5mm LDR ışık sensörü kartı

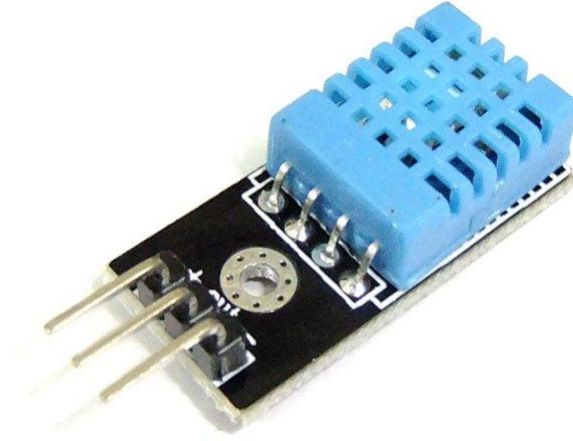
Sensörden analog çıkış alınarak ortamdaki ışık en karanlık konumdayken 1024, en aydınlık değerdayken 0 değeri elde edilmiştir. Bu değerler daha sonra bir fonksiyon aracılığıyla en parlak 100 en karanlık 0 olacak şekilde dönüştürülmüştür.

Aydınlanma için lüks (lux) birimi kullanılmaktadır. Lüks, 1 metre yarıçaplı bir kürenin merkezinde bulunan, 1 candela şiddetindeki ışık kaynağının 1 metrekarelik küre yüzeyinde oluşturduğu aydınlanma şiddetidir (Lüks (birim), 2016). Candela ise 1 atm basınç altında ve platinin erime sıcaklığındaki bir siyah cismin  $1/60000 \text{ m}^2$  büyüklüğündeki yüzeyinin kendisine dik olan bir doğruda verdiği ışık şiddetine denmektedir (Candela, 2016). Kullanılan sensör kart üzerinde yer alan ayar düğmesi aracılığıyla farklı aydınlık düzeylerinde en yüksek değeri verebilmektedir. Çalışmada en yüksek aydınlık değeri 100 lüks olarak belirlenmiştir.



### 3.7.2 Sıcaklık ve Nem Sensörü

Sıcaklık ve nemi ölçmek için tek bir sensör modülünden faydalanılmıştır (Şekil 3.9). Bu modül dijital çıkış vermektedir. Sıcaklık düzeyini 0-50°C aralığında 2°C hata ile nem oranını ise 5% RH (Relative humidity: belli bir sıcaklıkta bir hava kütleğinde bulunan nem miktarının yüzde ifadesi) hata payı ile ölçebilmektedir (DHT11 - Humidity and Temperature Sensor , 2016).



Şekil 3.9: Sıcaklık ve nem sensörü

### 3.7.3 Hareket Sensörü

Oda içerisindeki hareketleri algılamak için çalışmada PIR (Passive Infrared sensor) sensörü olarak bilinen insanların ve diğer sıcakkanlı canlıların ürettikleri kızılötesi ışıkları algılayarak ortamda herhangi bir hareketin olup olmadığını algılayabilen sensörler kullanılmıştır (Şekil 3.10). Çalışmada kullanılan sensör 5m'ye kadar hareketleri algılayabilmektedir (HC-SR501 PIR Motion Detector, 2016).



**Şekil 3.10:** PIR hareket algılama sensörü

### 3.7.4 Gaz Sensörü

Ev içerisinde oluşabilecek çeşitli gaz kaçaklarını tespit ederek güvenliği sağlamak için karbon monoksit ve yanıcı gaz algılayıcı sensörlerden faydalanılmıştır (Şekil 3.11). Kullanılan sensör ortamdaki 100 ila 10.000 ppm (Parts per million) yanıcı gaz ve 10 ila 10.000 ppm Karbon Monoksit (CO) yoğunluğunu algılayabilmektedir (MQ-9 Semiconductor Sensor for CO/Combustible Gas , 2016). Ppm, herhangi bir karışımda toplam madde miktarının milyonda 1 birimlik maddesine denilmektedir (What is ppm?, 2016).



**Şekil 3.11:** Yanıcı gaz ve karbonmonoksit sensörü

### 3.7.5 Alev Algılayıcı Sensör

Ev içerisinde çıkabilecek yangını tespit edebilmek için alev algılayıcı sensör kullanılmıştır (Şekil 3.12). Üzerinde bulunan kızılötesi algılayıcı ile PIR sensörlerine benzer mantıkta alevi algılayabilmektedir. Çalışmada kullanılan sensör 1m mesafeye kadar alev algılayabilmektedir (Flame Sensor Technical Data Sheet, 2016).



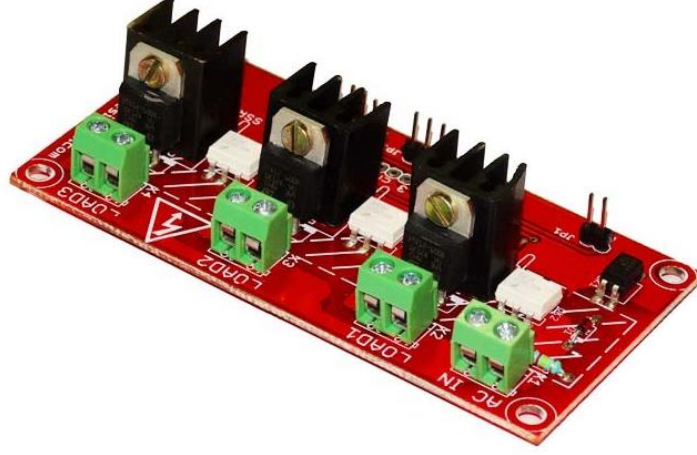
Şekil 3.12: Alev algılayıcı sensör

Tümleşik sensör bileşeni üzerine bağlı olan sensörlerden elde ettiği verileri sürekli ve düzenli olarak oda kontrolcüsüne aktarmaktadır. Bunun yanında oda içinde aydınlatma kontrolünü sağlayan aydınlatma kontrol bileşeni ile priz kontrolünü sağlayan priz kontrol bileşeni tümleşik sensöre bağlıdır. Tümleşik sensör bileşeni kendisine oda kontrolcüsünden iletilen işlem emrini gerekli işlemlerden geçirerek ilgili birime uygulamaktadır.

Bu bölümün ardından senarize edilen işlemin öncelikle aydınlatma kontrol, daha sonra priz kontrol son olarak ta termostat bileşenine iletildiği varsayılarak bu 3 bileşen anlatılacaktır.

### 3.8 Aydınlatma Bileşeni

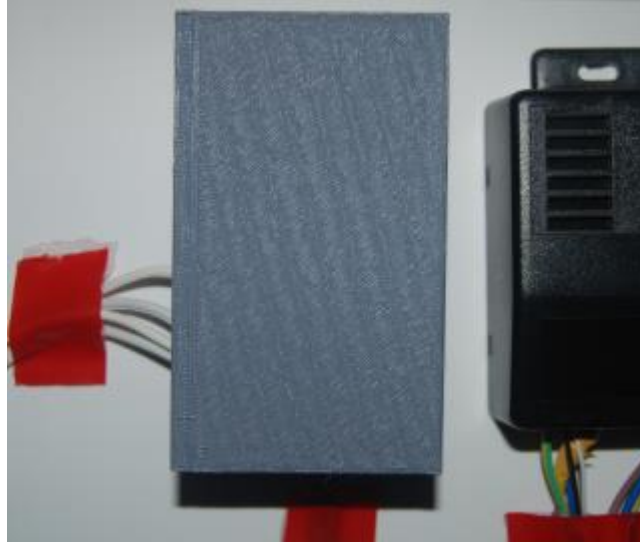
Aydınlatma bileşeni ev içerisinde odalarda yer alan lambaların parlaklık düzeyinin web arayüzünden ve oda içerisindeki panelden analog olarak 0-100 aralığında bir değer ile ayarlanabilmesini sağlamaktadır. Aydınlatma bileşeni bu işlemi bir AC Dimmer modül ile gerçekleştirmektedir (Şekil 3.13).



Şekil 3.13: AC dimmer modül

Web arayüzünde yapılan parlaklık ayarlama işlemi oda kontrolcüsüne geldikten ya da doğrudan oda kontrolcüsüne bağlı arayüzde lamba parlaklık ayarı yapıldıktan sonra gelen veri seri bağlantı ile tümleşik sensör bileşenine iletilmektedir. Bileşene gelen veri yine aynı şekilde seri bağlantı yolu ile AC Dimmer Modül'e iletilmekte ve lamba parlaklık ayarı gerçekleştirilmektedir.

AC Dimmer Modül aynı anda 3 adet lambayı desteklemektedir. Bunların kontrolü için seri bağlantı üzerinden önce hangi lambanın ayarı yapılmak isteniyorsa o lambanın bağlı olduğu bağlantı noktasının harf değerinin (1. Lamba için A, 2. Lamba için B, 3.lamba için C) gönderilmesi gerekmektedir. Daha sonra yazı tipinde 3 haneli olarak 0-100 aralığında parlaklık değeri iletilir. Örnek olarak 60 parlaklık düzeyi isteniyorsa "A060 " metni AC Dimmer Modül'e gönderilmelidir. AC Dimmer Modül kendisine iletilen değere göre şehir şebekesinden gelen 220V gerilimi 0-100 aralığına göre oranlayarak lambaya aktarmaktadır. Böylece lamba parlaklık düzeyi oransal olarak istenilen seviyede ayarlanabilmektedir. Aydınlatma bileşeninin görünümü Şekil 3.14'teki gibidir.



Şekil 3.14: Aydınlatma Bileşeni

### 3.9 Priz Bileşeni

Priz bileşeni ev içerisinde yer alan elektrik ihtiyacını karşılayan prizlerin web arayüzünden ve oda içerisindeki panelden açılıp kapatılabilmesini sağlamaktadır. Bu işlemin gerçekleşebilmesini sağlayan priz bileşeni bir röle shield (Şekil 3.15) ve prizlerden(çalışmada örnek olarak bir üçlü priz kullanılmıştır) oluşmaktadır.



Şekil 3.15: Röle shield

Web arayüzünde yapılan işlemler oda kontrolcüsüne geldikten ya da doğrudan oda kontrolcüsüne bağlı arayüzde priz işlemi gerçekleştirildikten sonra bluetooth sinyalleri ile priz bileşenine iletilmektedir. Priz bileşeni kendisine gelen

veri doğrultusunda röle shield a seri bağlantı üzerinden veri göndererek elektrik iletiminin röle shield ile açılıp-kesilmesini sağlamaktadır.

Röle shield 2'li röle kontrol kartı, 5V ile prizlerin kontrol edilebildiği, Arduino veya diğer başka mikrodenetleyeciler ile kullanılabilen bir röle kartıdır (Şekil 3.16). Mikrodenetleyeciden tetik sinyali sırasında 20mA'lık bir akım çekmektedir. 30VDC veya 220VAC gerilimde 10A'e kadar enerji anahtarlayabilmektedir.



Şekil 3.16: Priz Bileşeni

### 3.10 Termostat Bileşeni

Termostat, sıcaklığı istenilen ölçüde sabit tutmaya yarayan bir tür kontrol aracıdır. Çalışmada termostatın görevi bulunduğu odanın sıcaklığını kullanıcının belirlediği değerde sabit tutmaktır.

Termostat bileşeni diğer iki bileşen olan priz kontrol ve aydınlatma kontrol bileşenlerinden farklı olarak yeni bir donanım olarak geliştirilmemiştir. Bunun sebebi şudur: son zamanlarda geliştirilen birçok akıllı ev sisteminde farklı firmaların geliştirdikleri ürünler ya bir arada kullanılmaktadır ya da bu tür ürünlere destek verilmektedir. Ayrıca sadece farklı firmaların ya da araştırmacıların geliştirdikleri bileşenlerini kullanarak bunların kontrollerini sağlayan akıllı ev çözümleri de ortaya çıkmıştır (Eclipse Smart Home, Home Asistant vb). Bu çalışmada da farklı ürün ve

bileşenlere örnek olması açısından şu anda dünya da en iyi termostatlar arasında olan Google Nest termostat (Şekil 3.17) kullanılmıştır (Prospero, 2016).



Şekil 3.17: Google Nest termostat

Google Nest termostat kendi içeriğinde öğrenilebilir özelliklere sahip olmasına karşın bu çalışmada sadece sıcaklık sabitleyici olarak kullanılmıştır. Diğer bileşenlerin aksine Google Nest termostat ile tümleşik sensör bileşeni üzerinden kablo ile değil oda kontrolcüsü ile wi-fi üzerinden bağlantı kurulmaktadır. Kullanıcının web arayüzünden gerçekleştirdiği işlem oda kontrolcüsüne ulaştıktan sonra wi-fi bağlantı ile termostata iletilmekte ve termostat oda sıcaklığını istenilen seviyeye kendisi çekmektedir.

Bu kısma kadara bahsedilen bileşenler ve donanımlar kullanıcı bir işlem yaptıktan sonra bu işlemi hayata geçirmek için gerekli işlemleri gerçekleştirmektedirler. Ancak web arayüzünde sadece işlem yapılmamakta aynı zamanda odanın içerisinin ve dışarının sıcaklığı ve diğer bilgileri görülebilmektedir. Odanın içinde bulunan sensörler tümleşik sensör bileşenine bağlı olarak çalışmaktadır. Odanın dışındaki sıcaklık, nem ve ışık bilgilerini öğrenmek içinse dış sensör bileşeni geliştirilmiştir.



### 3.11 Dış Sensör Bileşeni

Akıllı ev sistemleri için evin dışından sensör verileri almak ev içerisinde yapılacak işlemler için önem teşkil etmektedir. Bu yüzden geliştirilen akıllı ev sisteminde çevre sıcaklık, nem ve parlaklık verilerini takip ederek sürekli olarak oda kontrolcüsüne aktaracak bir bileşen geliştirilmiştir.

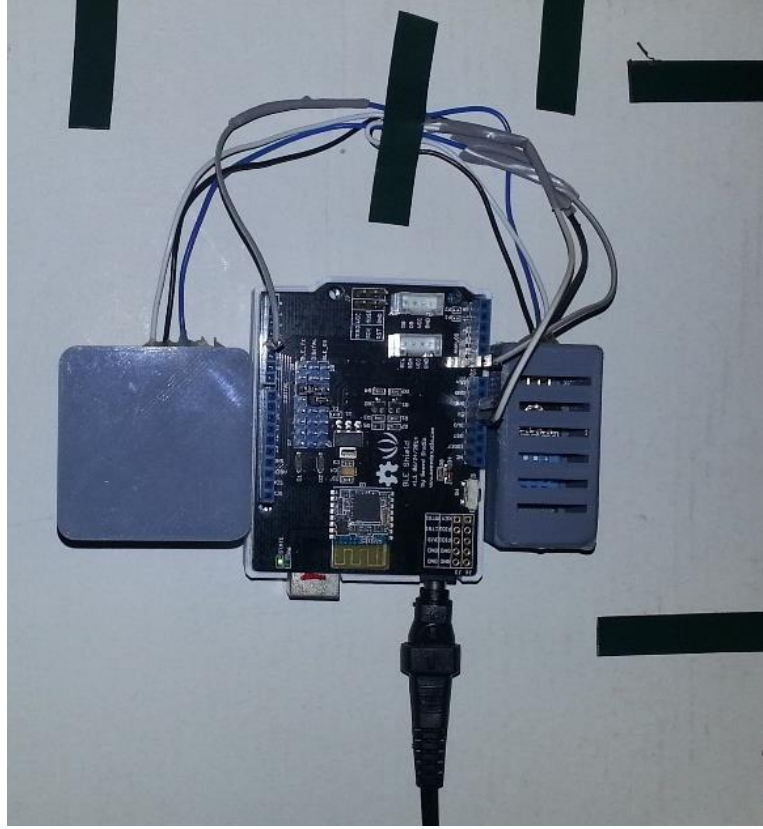
Dış sensör bileşeni donanım olarak bir BLE shield (Şekil 3.18), bir Arduino Uno ile sıcaklık, nem ve ışık sensörlerinden oluşmaktadır. Arduino Uno tıpkı Arduino Mega benzeri bir mikro denetleyici kartıdır. Aralarındaki fark Uno modelinin çok daha az bağlantı noktasına sahip olmasıdır. Ancak sahip olduğu 16 dijital ve 6 analog giriş çıkış pini dış sensör bileşeni için fazlasıyla yeterlidir.



Şekil 3.18: Arduino Uno ve BLE shield

Dış sensör bileşeni (Şekil 3.19) evin dışarısında bulunduğu için bağlantısı sadece kablosuz olarak bluetooth ile sağlanmaktadır. Bu sensör için kullanılan sıcaklık, nem ve ışık sensörleri tümleşik sensör bileşeninde kullanılan sensörler ile aynı özelliklere sahiptir.





Şekil 3.19: Dış Sensör Bileşeni

### 3.12 Kullanıcı Takip Bileşeni

Tüm bu birimler dışında son olarak tümleşik sensör bileşenine bağlı olarak çalışan kullanıcıların oda giriş çıkış işlemlerinde kim olduklarını tespit ederek akıllı ev sisteminin kullanıcıların buldukları konumları ve hareketlerini tespit edebilmesi için kullanıcı takip bileşeni geliştirilmiştir.

Kullanıcı takip bileşeni ile kişilerin ev içerisine girip ev dışına çıktıkları ve ev içerisinde hangi odalara giriş-çıkış yaptıkları takip edilebilmektedir. Literatür kısmında bahsedildiği üzere kullanıcı takibi için birçok yöntem bulunmaktadır. Bu çalışmada çalışmanın genel yapısına en uygun olan RFID (Radio-frequency identification) teknolojisi kullanılmaktadır.

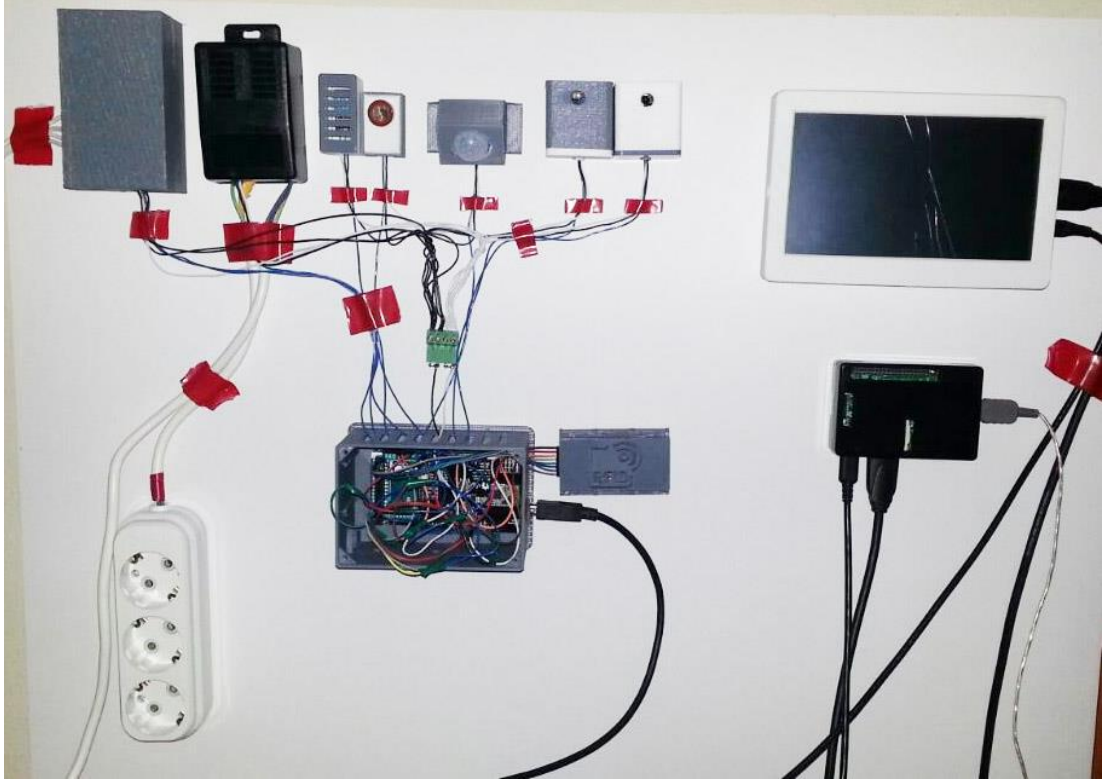
RFID teknolojisinin kullanıcılar tarafından kolayca kullanabilmesi için kapı girişlerine RFID okuyucu bileşenler (Şekil 3.20) yerleştirilmiştir. Bu bileşenler kullanıcılar üzerinde yer alan küçük bir madeni para boyutunda olan RFID etiketlerini okumaktadır.



**Şekil 3.20:** RFID okuyucu

Kullanıcılar bu etiketi bir yüzük şeklinde ya da kıyafetlerinin bir köşesine yerleştirerek oda giriş-çıkış ve ev giriş-çıkışlarında okuyucuya okuttuklarında sistem kullanıcıları takip edebilmektedir. Ayrıca sistem bu etiketler sayesinde kullanıcıları karıştırmamaktadır ve kullanıcıların tam olarak tespiti yapay zekâ algoritmaları için çok önemlidir. RFID okuyucu kullanıcı verisini etiketten okuduktan sonra bunu tümleşik sensör bileşenine aktarmakta, tümleşik sensör bileşeni oda kontrolcüsüne, oda kontrolcüsü de merkezi sunucuya veriyi iletmektedir. Kullanıcı daha önce başka bir konumdaysa o odaya girdi işlemi, daha önce o odadaysa odadan çıktı işlemi olarak bu işlem kayıt altına alınmaktadır.

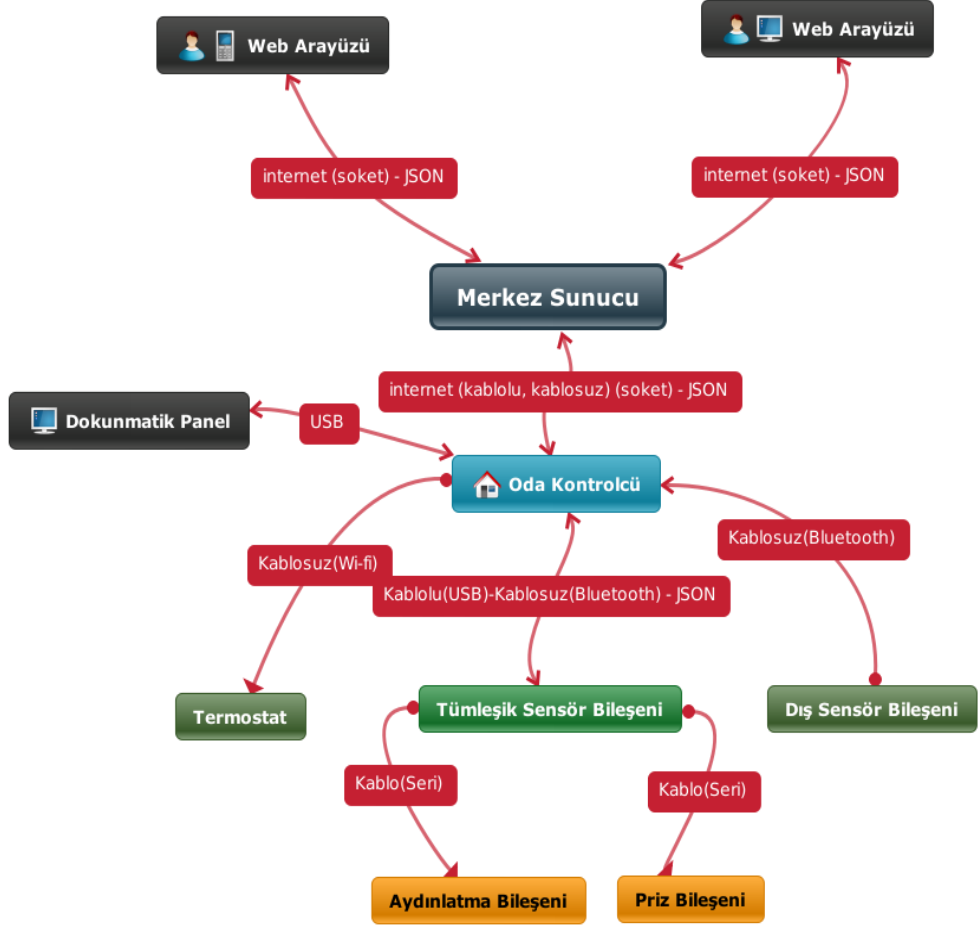
Bu noktaya kadar donanımlar ile ilgili bütün donanımsal işlemler bitmiştir. Sistemin son görünümü Şekil 3.21’de gösterilmektedir.



Şekil 3.21: Odadaki sistemin genel görünümü

### 3.13 Birimler Arası Bağlantılar ve İletişim

Şu ana kadar geliştirilen akıllı ev sisteminin gelen yapısı ve donanımsal altyapısı anlatılmıştır. Bu bölümde sistemin donanımlarının birbirleriyle ve kullanıcılarla ne şekilde etkileşime girdiği ve nasıl iletişim kurdukları anlatılacaktır. Sistemin genel yapısı, bağlantı şekilleri ve iletişim yolları Şekil 3.22'deki gibidir.



Şekil 3.22: Sistemin genel yapısı, bağlantı şekilleri ve iletişim yolları

Web arayüzü merkez sunucu üzerinde çalışan bir web sayfasıdır. Merkez sunucu ise sürekli olarak internete bağlı olarak çalışan bir sunucu bilgisayardır. Kullanıcıların sistemin web adresini ([http://sunucu.elektrogame.net/civa\\_smart\\_home/odaKontrolcu/index](http://sunucu.elektrogame.net/civa_smart_home/odaKontrolcu/index)) tarayıcılarına yazarak web arayüzüne erişmektedir. Bu işlem standart bir http işlemidir. Kullanıcı web arayüzünü açtığı anda web sayfası yüklenmesi işlemi bittikten sonra kullanıcının aygıtı (bilgisayar, cep telefonu vb.) ile merkez sunucu arasında soket bağlantı kurulmaktadır.

Bu web arayüzünün sürekli olarak yayınlanması için merkez sunucuda Apache web sunucu yazılımı sürekli olarak çalışmaktadır. Bunun yanında web

arayüzünde yapılan işlemlerin anında merkez sunucuya iletilmesi için web arayüzünü açan aygıt ile merkez sunucu arasında soket bağlantı kurulmasını sağlayan sunucu yazılımı da ayrıca geliştirilmiştir. Sunucu yazılımı aşağıdaki kodlarla gelen bağlantıları dinlemeye başlamaktadır:

```
var io_web_arayuzu = require('socket.io')();
io_web_arayuzu.listen(32032);
```

Web arayüzü ise ilk açıldığında şu kodlarla sunucuya bağlanmaktadır:

```
var socket;
$( document ).ready(function() {
    socket = io.connect(<?=SBT_MerkezSunucuPort?>);
    socket.on('connect',function() {
    });
});
```

Web arayüzü ile merkez sunucu arasında 32032 numaralı bağlantı noktasından (port) bağlantı kurulmasının ardından web arayüzünde iki tür işlem yürütülmektedir. Birinci işlem kullanıcı oda içerisinde yer alan priz, aydınlatma ve termostat bileşenlerini kontrol edebilmesidir. Kullanıcı bu birimlerle ilgili herhangi bir işlem yaptığında anında kurulu bağlantı üzerinden yapılan işlem ile ilgili veri merkez sunucuya gönderilmektedir (Örnek kodda priz açma/kapatma işlemi sunulmuştur).

```
$('#ayar_priz').on('switchChange.bootstrapSwitch',
function(event, state) {
    var okunan_deger = $(this).prop('checked');
    if (okunan_deger)
        prizAcKapatIslemYap(1, idoda);
    else
        prizAcKapatIslemYap(0, idoda);
});
function prizAcKapatIslemYap(ac_kapat, idoda) {
    if (ac_kapat == 1)
```

```

        {
            var islem_deger = 6; // Priz Aç
            var islemin_degeri=1;
        }
        else
        {
            var islem_deger = 7; // Priz Kapat
            var islemin_degeri=0;
        }

        socket.emit('priz_ac_kapat',
                    { kisi:<?=$person?>, dsi: d_aydinlik,
                    dss: d_sicaklik, dsn: d_nem, oda: idoda, isik: aydinlik,
                    sicaklik: sicaklik, islem: islem_deger, deger:
                    islemin_degeri, aygit_id:idaygit });
    }

```

Kodlardan da anlaşılacağı üzere veriler JSON (JavaScript Object Notation) biçiminde gönderilmektedir. Veriler sunucuya ulaştığında JSON biçiminde olduğu için öncelikle ayrıştırma işlemine tabi tutulmakta ve veriler anlamlı hale getirilmektedir. Daha sonra sunucu kendisine hâlihazırda internet aracılığıyla bağlı olan oda kontrolcüyeye veriyi yine JSON biçiminde(merkez sunucuya gelen veri hiç değiştirilmeden) gönderilmektedir (Örnek kodda önceki kodlarda gösterilmiş olan priz işlemi devam ettirilmektedir). Oda kontrolcü ile merkez sunucu arasında kurulan bağlantı web arayüzü ile merkez sunucu arasındaki bağlantı ile tamamen aynı şekilde kurulmaktadır. Fark olarak bu bağlantı için 42032 numaralı bağlantı noktası kullanılmaktadır. Bağlantı noktaları TCP-IP bağlantılarda kullanılmaktadır ve 0-65535 arası değer alabilmektedir.

```

socket.on('priz_ac_kapat', function (data) {
    arayuzde_yapilan_priz_ayari_veritabanina_logla(data)
;
io_rpi.sockets.emit('priz_ac_kapat', data);
});

```

Yapılan işlemin oda kontrolcüsüne ulaşmasının ardından oda kontrolcüsünde de tıpkı merkez sunucuda olduğu gibi gelen JSON biçimindeki veri ayrıştırılır. Ayrıştırılan verilerden elde edilen değere göre hangi aygıt ile işlem yapılacağı belirlenir ve tümleşik sensöre aygıt numarası, işlem tipi ve işlemin değeri iletilir. Bu iletim kablolu (usb) ve kablosuz (bluetooth) olmak üzere iki farklı şekilde olabilmektedir. Örnekte yer alan kod bluetooth ile kablosuz olarak yapılan iletişimi göstermektedir.

```
socket.on('priz_ac_kapat',function(data) {  
    priz_emri=data;  
    priz_emri.yapildi=0;  
    if (data.deger==1)  
        console.log('Priz açıldı...');  
    else  
        console.log("Priz kapatıldı...");  
    var priz = new Buffer([data.aygit, data.islem,  
data.deger]);  
    tumlesikSensorCharacteristic.write(temperature,  
false, function(err){  
        });  
});
```

Son olarak oda kontrolcüsünden tümleşik sensöre gelen işlem verisi tıpkı diğer birimlerde olduğu gibi öncelikle ayrıştırılır. Daha sonra priz bileşeninin bağlı bulunduğu bağlantı noktası açılır yada kapatılır.

```
if (mySerial.available())  
{  
    islem_aygit = mySerial.read();  
    islem_tipi = mySerial.read();  
    switch(islem_tipi.toInt())  
    {  
        case 6 : digitalWrite(prizPin(islem_aygit),  
LOW);
```

```
break;
case 7 : digitalWrite((prizPin(islem_aygit), HIGH);
break;
}
}
```

Bu noktaya kadar web arayüzünden yapılan işlemin ilgili aygıtı ulaşıncaya kadar nasıl bir yol izlenerek iletildiği sunulmuştur. Bu bölümden sonra tam tersi bir yol ile sensörlerden elde edilen verilerin web arayüzüne ulaşıncaya kadar nasıl bir yol izlendiği sunulacaktır.

Dış sensör bileşeninden ya da tümleşik sensör bileşeninden elde edilen veriler öncelikle oda kontrolcüsüne aktarılmaktadır. Dış sensörden oda kontrolcüsüne veri aktarılırken her zaman bluetooth ile kablosuz bağlantı kullanılmaktadır. Veriler diğer işlemlerin birçoğunda olduğu gibi JSON biçiminde ve saniyede bir kere olacak şekilde aktarılmaktadır.

```
StaticJsonBuffer<200> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
root["isik"] = degerSensorIsik;
root["sicaklik"] = degerSensorIsi;
root["nem"] = degerSensorNem;
if (BLE.available())
{
    sv = root.printTo(BLE);
    delay(1000);
}
```

Tümleşik sensör bileşeninden de benzer şekilde gönderilen veriler oda kontrolcüsüne geldikten sonra ulaşan her veri ters yönde de aynı olacak şekilde merkez sunucuya iletilir. Merkez sunucuya ulaşan sensör verileri veritabanında kayıt altına alınır (veri yapıları, kayıt işlemleri bir sonraki bölümde sunulacaktır). Daha sonra veriler bağlı olan web arayüzü tarafından 3 saniyede bir merkez suucudan çekilir ve kullanıcıya gösterilir.



```

$('#zamanlayici').timer({
    duration: '3s',
    callback: function() {
        aydinlik_verilerini_guncelle();
        sicaklik_verilerini_guncelle();
    },
    repeat: true});

function aydinlik_verilerini_guncelle(){
    $.getJSON("<?=base_url()?>civaSmartHome/JsonYayin/Ay
dinlikBilgileri", function(response) {
$('#label_aydinlik_os').html(response[0].os);
$('#label_aydinlik_oy').html(response[1].oy);
$('#label_aydinlik_om').html(response[2].om);
$('#label_aydinlik_ok').html(response[3].ok);
$('#label_aydinlik_oe').html(response[4].oe);
$('#label_aydinlik_og').html(response[5].og);
$('#label_aydinlik_go').html(response[6].go);
$('#label_aydinlik_ed').html(response[7].ed);

    });
}

});

```

### 3.14 Sistemin Veri Yapısı ve İşlemlerin Kayıt Altına Alınması

Akıllı ev otomasyon sistemi bu işlemlerin tamamlanmasıyla birlikte çalışır hale gelmiştir. Bu noktadan sonra sıra gerçekleştirilecek olan yapay zekâ için sistemde yapılan her işlemin kayıt altına alınmasındadır. Yapay zekâ, öğrenme işlemlerini kullanıcıların geçmiş davranışlarına bakarak gerçekleştireceği için

sistemin ürettiği verilerin anlaşılır ve kayıpsız bir şekilde kayıt altına alınması çok önemlidir.

Sistemde kayıt tutmak için merkez sunucu bölümünde de sunulduğu üzere MySQL veritabanı yönetim yazılımı kullanılmıştır. Kullanıcılar, sistemdeki aygıtlar, odalar, işlem tipleri ve yapılan bütün işlemler tablolar halinde kayıt altına alınmaktadır. Bu bölümde sırasıyla hangi verilerin neden ve ne şekilde kayıt altına alındığı sunulacaktır (Tüm tabloların ayrıntılı görünümü EK B’de sunulmuştur).

### 3.14.1 Oda

Geliştirilen akıllı ev otomasyon sistemi sahip olduğu çoklu ajan sistemleri yapısı itibariyle odalardan oluşmaktadır. Çünkü her oda için bir ajan(oda kontrolcü) bulunmaktadır. Yani oda aslında oda kontrolcüdür denilebilir. Bu yüzden oda kontrolcülerini bir donanım olmasına karşın sistemde bir aygıt olarak değil farklı bir yapı(oda) olarak kayıt altına alınmıştır. Çünkü oda ile ilgili aygıtların genel özelliklerinden bağımsız farklı bilgilerinde kaydı tutulmaktadır. Oda verileri veritabanında Şekil 3.23’teki gibi tutulmaktadır (Şekil B.1).

ad	tanım	kod	sicaklik	nem	aydinlik
Salon	Tüm aile bireylerinin tv seyretmek, sohbet etmek i...	os	34	0	10
Yatak Odası	Anne ve babanın yatak odası.	oy	34	0	11
Mutfak	Genellikle annenin vakit geçirdiği ve ailede herke...	om	35	0	12
Kız Çocuk Odası	Kız çocuğun evde genel olarak vakit geçirdiği oda.	ok	36	0	13
Erkek Çocuk Odası	Erkek çocuğun evde genel olarak vakit geçirdiği od...	oe	37	0	14

Şekil 3.23: Oda bilgileri tablosu

### 3.14.2 Aygıt

Sistemde elektriği, ısıtma sistemini ve aydınlatma sistemini kontrol eden birimler aygıtlardır. Bu aygıtların her an hangi konumda olduğunun bilinmesi ve hangi aygıtların sistemde bulunduğu anlaşılması için aygıt tanım bilgileri ve aygıtların durum bilgileri kayıt altına alınmaktadır. Aygıt bilgileri Şekil 3.24’teki gibi tutulmaktadır (Şekil B.2).

idaygit	tip	ad	acik_kapali	deger
1	1	Salon Termostat	0	0
2	1	Yatak Odası Termostat	1	20
3	1	Mutfak Termostat	1	27
4	1	Kız Çocuk Odası Termostat	1	30
5	1	Erkek Çocuk Odası Termostat	1	29
6	2	Salon Lambası	1	50

Şekil 3.24: Aygıt bilgileri tablosu

Aygıt bilgileri tablosunda yer alan tip alanı aygıtın tipini ifade etmektedir ve tip isimli bir başka tablo ile ilişkilidir (Şekil B.3). Termostat, priz ve lamba şeklinde 3 aygıt tipi bulunmaktadır.

### 3.14.3 Kullanıcı

Akıllı ev sisteminin yapay zekâ işlemlerini kullanıcılara özel yapabilmesi için sistemin kullanıcıları tanıması gerekmektedir. Ayrıca sistemin güvenliği açısından kullanıcılar sisteme kullanıcı adı ve şifre ile giriş yapmaktadır. Bunların yanında evin içerisinde her kullanıcı her şeyi yapma yetkisine sahip olmamalıdır. Kullanıcıların yapabilecekleri işlemlerin yetkilendirilmesi ve seviyelendirilmesi gerekmektedir. Tüm bu durumlar için kullanıcı bilgileri “kullanici” isimli tabloda Şekil 3.25’teki gibi kayıt altına alınmıştır (Şekil B.4).

idkullanici	isim	bulundugu_oda	yetki	kullanici_adi	sifre	goruntulem
1	Ahmet YILMAZ	3	10	aliyilmaz	1234	BA
4	Ayşe YILMAZ	4	10	ayseyilmaz	1234	AN
5	Zeynep YILMAZ	6	5	zeynepyilmaz	1234	KÇ
6	Mehmet YILMAZ	7	4	mehmetyilmaz	1234	EÇ1
7	Mustafa YILMAZ	5	1	mustafayilmaz	1234	EÇ2

Şekil 3.25: Kullanıcı bilgileri tablosu

### 3.14.1 İşlem

Akıllı ev sisteminde kullanıcıların yaptıkları tüm işlemler yapay zekâ çalışmaları için veri kaynağı olduğundan kayıt altına alınmaktadır. Böylece sistemde kimin, ne zaman, ne yaptığı bilinebilmektedir. Ayrıca işlemleri kayıt altına alınırken işlem yapılan odanın durum bilgileri de birlikte kaydedilmektedir. Bu bilgiler hangi şartlar altında kullanıcının nasıl davrandığını anlamak konusunda kaynak oluşturmaktadır. İşlem kayıtları tarih ve saat etiketiyle birlikte veritabanında “kayıt” isimli tabloda Şekil 3.26’daki tutulmaktadır (Şekil B.5).

idkayıt	tarih_saat	kullanici	oda	aygit	islem_tipi	islem_degeri	oda_sicaklik	oda_aydinlik
1	2015-01-03 17:05:38	1	0	19	7	0	2	98
2	2015-01-04 00:03:55	4	0	19	7	0	0	100
3	2015-01-04 12:56:19	1	0	9	7	0	2	98
4	2015-01-10 17:06:47	1	0	19	7	0	-4	98
5	2015-01-10 23:58:00	4	0	19	7	0	1	100
6	2015-01-11 12:59:27	1	0	9	7	0	0	98
7	2015-01-17 16:56:12	1	0	19	7	0	12	97
8	2015-01-18 00:08:27	4	0	19	7	0	-1	100
9	2015-01-18 13:05:58	1	0	9	7	0	11	98
10	2015-01-24 17:09:36	1	0	19	7	0	13	98

Şekil 3.26: İşlem bilgileri tablosu

### 3.14.1 Sensör Verileri

Akıllı ev sistemi tarafından sensörler aracılığıyla toplanan veriler oda kontrolcüsü tarafından merkez sunucuya iletilmektedir. Yapay zekâ işlemlerinde odaların ve dış şartların geçmişte ne durumda bulunduğu belirlenebilmesi için tüm sensör verileri “sensör\_kayıt” tablosunda Şekil 3.27’deki gibi kayıt altına alınmaktadır (Şekil B.6).

idsensor_kayit	tarih_saat	oda	sicaklik	nem	aydinlik	hareket	gaz	ates	dis_sicaklik	dis_nem	dis_aydinlik
1	2016-07-23 15:42:30	10	27	17	65	0	1	1	28	8	91
2	2016-07-23 15:42:31	10	27	17	65	0	1	1	28	8	91
3	2016-07-23 15:42:32	10	27	17	65	0	1	1	28	8	91
4	2016-07-23 15:42:33	10	27	17	63	0	1	1	28	8	91
5	2016-07-23 15:42:35	10	27	17	60	0	1	1	28	8	91
6	2016-07-23 15:42:36	10	27	17	60	0	1	1	28	8	91

**Şekil 3.27:** Sensör verileri tablosu

Sensörlerden toplanan veriler odalara göre kaydedilmektedir. Ayrıca oda içinde yer alan sensörlerden toplanan veriler ile birlikte dış sensör bileşeninden elde edilen sensör verileri de kaydedilmektedir.

Geliştirilen akıllı ev sisteminde şu ana kadar sistemin genel tasarımı yapılmış, donanımlar geliştirilmiş, donanımlarının birbiri ile iletişim ve entegrasyonu sağlanmış, sistem çalışır hale getirilmiş ve kullanıcıların yaptıkları işlemler kayıt altına alınmaya başlanmıştır. Bu aşamadan sonra sıra yapay zekâ algoritmalarının geliştirilmesine ve sistemin kullanıcı davranışlarını öğrenerek otonom hareket ettirilmesine gelmiştir.

### 3.15 Yapay Veri Üretici

Geliştirilen akıllı ev sisteminde yapay zekâ teknikleri ile işlemleri otonom gerçekleştirebilmek için kullanıcıların belirli bir süre akıllı ev sistemini kullanarak sistemde kayıtlar oluşturmaları gerekmektedir. Yapay zekâ algoritmaları da bu kayıtları inceleyerek kullanıcı davranışlarını belirlemektedir. Ancak bu noktada bazı zorluklarla karşılaşılmaktadır. Öncelikle yapay zekâ algoritmalarının farklı şartlar altında sınanması ve iteratif olarak geliştirilmesi gerekmektedir. Yani akıllı evi aynı anda farklı kişilerin farklı şekillerde kullanarak kayıtlar oluşturması gerekmektedir. Ayrıca mevcut sistemin farklı sayıda odalar içerecek şekilde kurulması gerekmektedir. Bunların yanında sistem işletilirken kayıt esnasında oluşabilecek her türlü aksaklık (kişinin hasta olması, acil durumların ortaya çıkması vb. rutini bozabilecek her türlü olay), veri setlerinin içerisinde tutarsızlıklara neden olacak ve yapay zekâ algoritmalarının test edilmesini ve geliştirilmesini olumsuz yönde etkileyecektir. Buna rağmen kayıtlar istenilen şekilde sorunsuz olarak elde edilse dahi bu tür kayıtlar gerçek bireyler hakkında özel yaşama dair bilgiler içereceğinden


farklı sorunlarında doğmasına neden olacaktır (güvenlik, özel hayatın gizliliği). Son olarak gerçek bir sistemin çok uzun vadede (Örn: bir yıl) nasıl çalışacağını belirlemek için sistemin çok uzun sürelerde açık kalması ve kayıt tutması gerekmektedir. Tüm bu durumları aşabilmek için çalışmada yapay zekâ algoritmalarının geliştirilmesi ve sınanması aşamasında sanal veriler ile çalışılmasına karar verilmiştir.

Çalışmanın bu bölümünde geliştirilecek olan yapay zekâ algoritmalarının iteratif olarak iyileştirilebilmesi ve test edilebilmesi için günlük insan davranışlarını taklit ederek geliştirilecek olan akıllı ev sisteminin üreteceği verilerin benzeri yapay kayıtlar üreten bir yazılım geliştirilmiştir.





### 3.15.1 Senaryo

Yapay veri üretici yazılım için öncelikle beş kişilik sanal bir aile oluşturulmuştur. Bu aile baba, anne, kız çocuk ve iki erkek çocuktan oluşmaktadır. Aile için çok sık kullanılan soyadlarından YILMAZ soyadı seçilmiştir. Aile bireylerinin diğer özellikleri sırayla Tablo 3.1’de sunulmuştur.

**Tablo 3.1:** Sanal ailenin özellikleri.

Kişi	Özellikleri
<b>Ahmet YILMAZ (BA)</b> 	<ul style="list-style-type: none"><li>• Ailenin babasıdır.</li><li>• 45 yaşındadır.</li><li>• Hafta içi çalışmaktadır.</li><li>• Hafta sonu ailesiyle vakit geçirmektedir.</li><li>• Spor yapmaya ve spor müsabakalarına ilgi duymaktadır.</li></ul>

**Tablo 3.1:** Sanal ailenin özellikleri (devam).

<p><b>Ayşe YILMAZ (AN)</b></p> 	<ul style="list-style-type: none"><li>• Ailenin annesidir.</li><li>• 40 yaşındadır.</li><li>• Çalışmamaktadır.</li><li>• Ev Hanımı, evdeki temizlik ve yemek işlerini kendisi yapmaktadır.</li><li>• Çocuklara ev ödevlerinde yardımcı olmaktadır.</li></ul>
<p><b>Mustafa YILMAZ (EÇ2)</b></p> 	<ul style="list-style-type: none"><li>• Ailenin en büyük çocuğudur.</li><li>• 19 yaşındadır.</li><li>• Üniversite öğrencisidir.</li><li>• Gece geç vakitlerde gelebilmektedir.</li><li>• Hafta sonlarını arkadaşlarıyla geçirebilmektedir.</li></ul>
<p><b>Zeynep YILMAZ (KÇ)</b></p> 	<ul style="list-style-type: none"><li>• Ailenin tek kız çocuğudur.</li><li>• 16 yaşındadır.</li><li>• Lise öğrencisidir.</li><li>• Hafta içi okula gidip gelmektedir.</li><li>• Hafta sonları ebru kursuna ve arkadaşlarıyla vakit geçirebilmektedir.</li></ul>
<p><b>Mehmet YILMAZ (EÇ1)</b></p> 	<ul style="list-style-type: none"><li>• Ailenin en ufak çocuğudur.</li><li>• 10 yaşındadır.</li><li>• Ortaokula gitmektedir.</li><li>• Hafta içi okula gidip gelmektedir.</li><li>• Hafta sonları evde ya da anne-babasıyla birlikte vakit geçirmektedir.</li></ul>

Aile ve özellikleri belirlendikten sonra bu ailenin yaşayacağı sanal bir ev oluşturulmuştur. Sanal ev salon, mutfak, erkek çocukların odası, kız çocuk odası ve yatak odasından oluşmaktadır. Sanal evin şeması Şekil 3.28’de gösterilmiştir.



Şekil 3.28: Sanal evin görünümü

Sanal bir aile ve ev oluşturulmasının ardından ailenin bir haftalık yaşamı senarize edilmiştir. Senarize işlemi yapılırken aile bireylerinin ve evin daha önce belirlenen özellikleri dikkate alınmıştır. Ailenin bir haftalık senaryoları EK C’de sunulmuştur.

Ailenin haftalık tüm faaliyetleri belirlenerek kişilerin hangi odalara ne zaman girip çıktığı belirlenmiş olmaktadır. Geliştirilmiş olan akıllı ev sisteminin ürettiği kullanıcı kaydı verileri benzeri veriler üretmesi gerektiği için sadece oda giriş-çıkış verilerinin üretimi yeterli olmamaktadır. Bunun dışında akıllı ev sisteminde yer alan termostat, aydınlatma bileşeni vb. bileşenlerinde kullanım verilerinin üretilmesi gerekmektedir. Bunun için kişilerin aydınlatma ve sıcaklık tercihleri de belirlenmiştir. Bu tercihler Tablo 3.2’deki gibidir.

Tablo 3.2: Sanal ailenin sıcaklık ve aydınlık tercihleri.

	Ahmet		Ayşe		Mustafa		Zeynep		Mehmet	
Sıcaklık	25-15	22-13	30-17	27-15	28-20	4-15	30-20	28-18	28-20	24-15
Aydınlatma	70-90		75-95		50-85		80-100		75-90	



Yukarıdaki tabloda yer alan sıcaklık değerlerinin ilki (Ahmet: 25-15) gündüz vakitlerinde kişinin ısıtma ya da soğutma açma sınır dereceleridir. Kişi bulunduğu odada sıcaklık 25 derece üstüne çıktığında soğutma açmakta, 15 derecenin altına indiğinde ise ısıtma sistemini açmaktadır. Diğer sıcaklık değerleri ise gece için geçerli sıcaklık aralığıdır. Geliştirilen akıllı ev sisteminde lamba parlaklığı 0-100 aralığında ayarlanmaktadır. Aydınlatma değerleri de kişilerin lamba parlaklığını yüzdesel olarak belirledikleri değerlerdir.

Son olarak kişilerin aynı odada buldukları durumlarda ya da bir odada işlem yapılırken o odada başka birilerinin bulunduğu durumlarda ne yapılacağını karar verilmesi gerekmektedir. Örneğin mutfakta anne varken baba geldiğinde odada bulunan aydınlık veya sıcaklık tercihleri kime göre ayarlanacağına sistem karar verebilmelidir. Bu sorunun çözümü için her oda için her kullanıcıya özel 1-10 aralığında yetki seviyeleri belirlenmiştir. 10 yetki seviyesi odadaki en yetkili kişiyi 1 ise yetkisi en düşük kişiyi ifade etmektedir. Kişilerin oda yetki seviyeleri Tablo 3.3'te gösterilmiştir.

**Tablo 3.3:** Sanal ailenin oda yetki seviyeleri.

	Ahmet	Ayşe	Mustafa	Zeynep	Mehmet
Salon	10	10	9	8	7
Mutfak	8	10	5	9	4
Yatak Odası	9	10	1	1	1
Erkek Çocuk Odası	5	7	10	1	9
Kız Çocuk Odası	5	8	1	10	1

### 3.15.2 Yapay Kayıt Üretici Yazılım

Ailenin, senaryoların ve yetkilerin belirlenmesinin ardından tüm bu verilerin girilebileceği ve bu verilere göre veri üretecek olan yapay veri üretici yazılım geliştirilmeye başlanmıştır. Geliştirilmeye başlanmış olan yazılımda temel hedef her ne kadar belirtilen senaryoya uygun yapay akıllı ev kayıtları üretmek olsa da oluşabilecek yeni durumlar ve farklı şekillerde yapay zekâ tekniklerinin test edilebilmesi için olabildiğince esnek bir yazılım geliştirilmesine karar verilmiştir. Bunun için sadece belirtilen senaryodaki kişiye ve eve göre değil farklı durumlara göre de yapay kayıt üretebilen bir yazılım tasarlanmıştır. Sonuç olarak tasarlanan

yapay kayıt üreten yazılım ile istenilen sayıda bireye sahip bir ailenin, istenilen sayıda odaya sahip bir evde, istenilen sayıda analog ve dijital giriş - çıkışa sahip aygıtı, günün istenilen saatlerinde kullanımlarının istenen tarih aralığında kayıtları üretilebilmektedir. Ayrıca yazılım farklı arařtırmacılar tarafından da esnek olarak kullanılabilmesi için web tabanlı olarak geliřtirilmiřtir. Arařtırmacılar birey, oda ve aygıt ekleme-silme-düzenleme vb. iřlemlerini bu web tabanlı yazılımdan yapabilmektedir. Ayrıca bireylerin, akıllı ev içerisinde gerçekleřtirdikleri günlük ve haftalık tüm iřlemler bu arayüzden girilerek yapay kayıtlar oluşturulabilmektedir.

Yapay kayıt üretici yazılım daha önce belirtilen iřlemleri gerçekleřtirebilmesi için Kiři, Oda, Aygıt ve Kayıt Üret menülerinden oluřmaktadır. Kiři menüsünde evde yařayan kiřiler ve özellikleri girilebilmekte ve düzenlenebilmektedir. Oda ve aygıt menülerinde de ilgili birimler eklenebilmekte, düzenlenebilmekte ve silinebilmektedir. Son olarak “Kayıt Üret” menüsü ile girilen kiři, oda ve aygıt verilerine göre kayıt üretilebilmektedir.

Yazılımda ilk olarak, aile bireylerinin verileri girilmektedir. Bunun için “Kiři Ekle” menüsüne tıklanarak aile bireyinin bilgileri girilerek “Kaydet” düğmesi ile kayıt edilir (řekil 3.29).

řekil 3.29: Kiři Ekle menüsü

Bu ařamadan sonra “Oda Ekle” menüsüne tıklanarak odanın bilgileri girilerek “Kaydet” düğmesi ile kayıt edilir (řekil 3.30).

**Şekil 3.30:** Oda Ekle menüsü

Odaların belirlenmesinin ardından odalarda yer alan aygıtlar, “Aygıt Ekle” menüsünden aygıtların bilgileri girilerek kayıt edilir (Şekil 3.31).

**Şekil 3.31:** Aygıt Ekle menüsü

Sisteme temel bilgiler girildikten sonra kayıt üretilebilmesi için gerekli olan zamanlama vb. bilgilerin girileceği Kullanıcı Adı (Örn. Ahmet YILMAZ) menüsüne girilir (Şekil 3.32). Bu menüden bir kişinin, hafta boyunca uyuma – uyanma, oda giriş – çıkış, aygıtları kullanım zamanlamaları ve sıcaklık, aydınlık düzeyi tercihleri girilir. Ayrıca kişilerin odalara göre farklı yetki seviyeleri olması istenirse de bu menü ile odalara göre kişi yetki seviyeleri belirlenebilmektedir. Örneğin mutfakta en yetkili kullanıcı anne olarak tanımlanırsa; anne mutfakta iken diğer kullanıcılar mutfakta herhangi bir işlem yapamamaktadır.

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi	Pazar
Uyanma	6:25	6:35	6:45	6:40	6:30	10:00	10:00
Uyuma	23:10	23:15	23:10	23:15	0:00	2:00	23:10
Ev Giriş	17:31	17:30-22:10	17:30	17:30-22:10	17:30	19:00-01:10	22:10
	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]
Ev Çıkış	07:30	07:30-19:50	07:30	07:30-19:50	07:30	17:00-23:50	19:50
	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]
Salon Giriş	21:00	21:00	21:00	21:00	21:00	21:00	20:00
	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]
Salon Çıkış	23:00	23:00	23:00	23:00	23:00	23:40	23:00
	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]	[09:00-17:30-21:50]

**Şekil 3.32:** Kişisel bilgileri giriş ve senaryo oluşturma

Kişiler, odalar ve aygıtlar eklendikten sonra, kişi bilgi girişi menüsü ile senaryolar oluşturulmaktadır. Bu işlemlerden sonra “Kayıt Üret” menüsüne girilerek kayıt üretilmesi istenen tarih aralığı girilir, kişiler ve işlemler seçilir. Ayrıca üretilen değerlerin gerçeğe uygun bir şekilde olması için belirli bir sapma değeri girilmesi gerekmektedir. Örnek olarak senaryoda bir oda girişi için saat 12.00 belirlenmiş ise gerçek hayatta bu kişinin her zaman tam 12.00 da gelmesi beklenemez. Burada girilen sapma değeri ile kişilerin gerçekleştirdikleri olayların önce ve sonra olarak kaç dakika sapma ile gerçekleşeceği belirlenebilmektedir. Önceki örnek ele alındığında 12.00’da gerçekleşen oda girişi olayı sapma 5 belirlenmiş ise 11.55-12.05 aralığında gerçekleşecektir. Son olarak “İşlem Kaydı Üret” düğmesine tıklanarak yapay kayıtlar üretilir (Şekil 3.33).

### İşlem Kaydı Üretme Seçenekleri

#### Tarih Aralığı

Başlangıç Tarihi

Bitiş Tarihi

#### Kişiler

- Ali YILMAZ
- Ayşe YILMAZ
- Mehmet YILMAZ
- Okan YILMAZ
- Zeynep YILMAZ

#### İşlemler

- Isıtma
- Aydınlatma
- Prizler
- Perde
- Cam
- Konum

Sapma

**Şekil 3.33:** Kayıt üretim seçenekleri

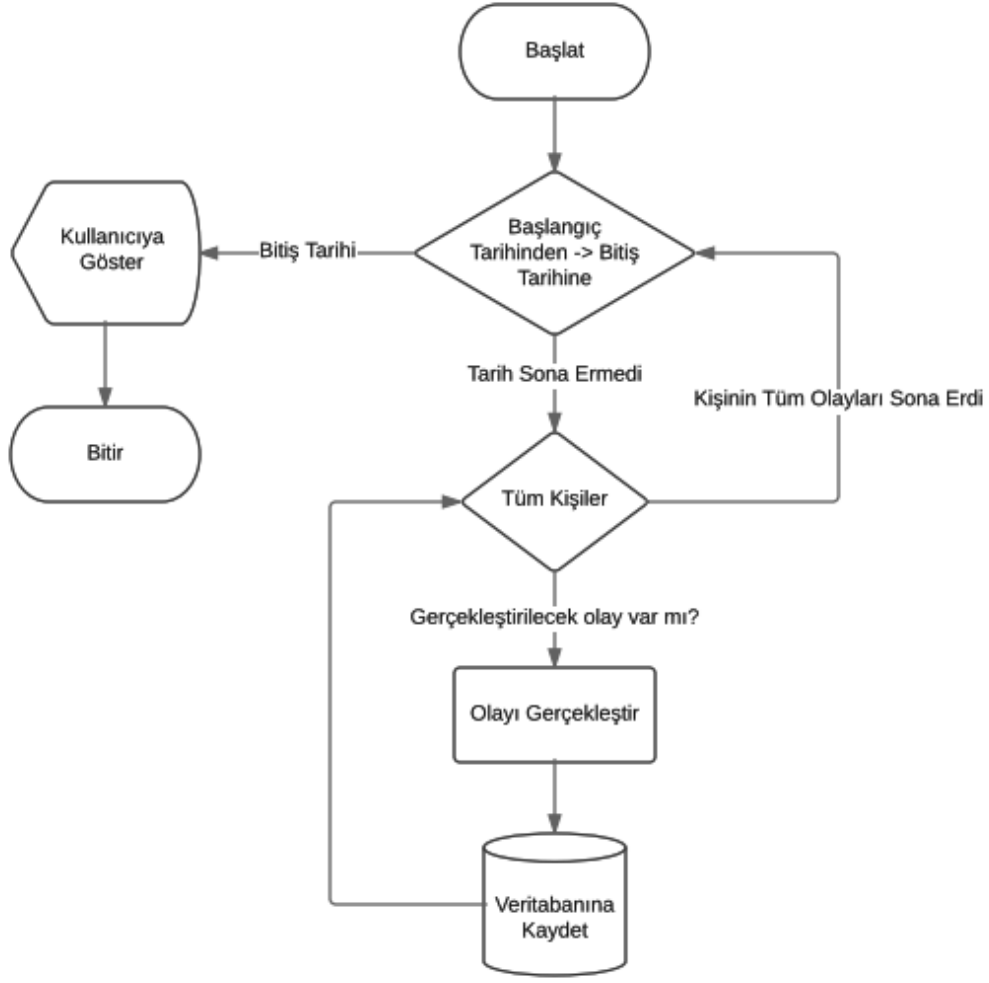
Yeni açılan sayfada üretilen kayıtlar tarih, saat, kişi, oda, işlem tipi, kullanılan aygıt vb. bilgiler tablo olarak gösterilmektedir (Şekil 3.34). Ayrıca üretilen kayıtlar, otomatik olarak veritabanına kaydedilmektedir.

No	Tarih - Saat	Kişi	Oda	Tip	Tercih Edilen Sıcaklık	Oda Sıcaklığı	Tercih Edilen Aydınlık	Aydınlık	Ayarlanmış Aydınlık	Aygıt
1	2015-01-24 18:54:14	1	og	Oda Giriş		13		89.5		
2	2015-01-24 01:09:35	1	og	Oda Giriş		9		2.2		
3	2015-01-24 16:50:03	1	og	Oda Çıkış		9				
4	2015-01-24 23:55:09	1	og	Oda Çıkış		9				
5	2015-01-24 20:50:38	1	os	Oda Giriş		9		2.3		
6	2015-01-24 20:50:38	1	os	Isıtma Aç	19	9				
7	2015-01-24 20:50:38	1	os	Aydınlatmayı Ayarla		9	70-90	2.3	76	

Şekil 3.34: Üretilmiş yapay kayıtlar

### 3.15.3 Senaryonun Yazılıma Girişi ve Sanal Kayıt Üretimi

Yapay kayıt üretici yazılımının geliştirilmesinin ve sanal ailenin tüm özelliklerinin belirlenmesini ardından ailenin tüm bilgileri senaryoya uygun şekilde sisteme girilmiştir. Bu işlemin ardından yapay zekâ algoritmalarında kullanılmak üzere bir yıllık sanal kayıt oluşturulmuştur. Sapma değeri olarak 10 girilmiştir. Sistemin kayıt üretme süreci Şekil 3.35'teki akış diyagramında gösterilmiştir:



Şekil 3.35: Yapay veri üretim akış diyagramı

### 3.16 Akıllı Ev Simülasyonu (Hibrit Simülasyon)

Çalışmada bu aşamada sistem donanımsal olarak geliştirilmiş ve tamamen çalışır hale getirilmiş, ayrıca yapay zekâ çalışmaları için sanal bir aile oluşturularak sistemi bir yıl boyunca bu aile tarafından kullanılmış gibi kayıtlar üretilmiştir. Son olarak yapay zekâ çalışmalarına geçilmeden önce bir eksik nokta kalmıştır. Sanal aile üretilmiş ve kayıtları tutulmuştur ve bu veriler ile yapay zekâ çalışmaları yapılabilir ancak bu ailenin olduğu bir evde çalışan sistemin geliştirilen algoritmalar ile gerçek zamanlı olarak nasıl test edileceği sorusu cevap bulmamıştır. Bu sorunun cevabı olarak üretilmiş olan sanal aileyi gerçekten üretilmiş olan akıllı evde yaşıyormuş gibi canlandırma yapacak olan bir akıllı ev simülasyonu geliştirilmesine, ayrıca simülasyonun sadece sanal bir simülasyon yanında geliştirilmiş olan gerçek

donanımlarla ve gerçek bir kişi ile de entegrasyonunun sağlanmasına karar verilmiştir.

Son durumda geliştirilecek olan akıllı ev sisteminin işleyişi şu şekilde olacaktır. Üretilen beş sanal kişi beş odalı sanal bir evde sanal aygıtları kullanarak yaşamaktadır. Bu aileye gerçek bir kişi(araştırmacı), geliştirilen donanımların içinde çalıştığı gerçek bir oda ve gerçek aygıtlar eklenecektir. Sanal kişi, oda ve aygıtların tüm işlemleri simülasyon yazılımı ile yürütülürken gerçek kişi, oda ve aygıtlar tamamen gerçek donanımlardan ve kişilerden elde edilen veriler ile takip edilecektir.

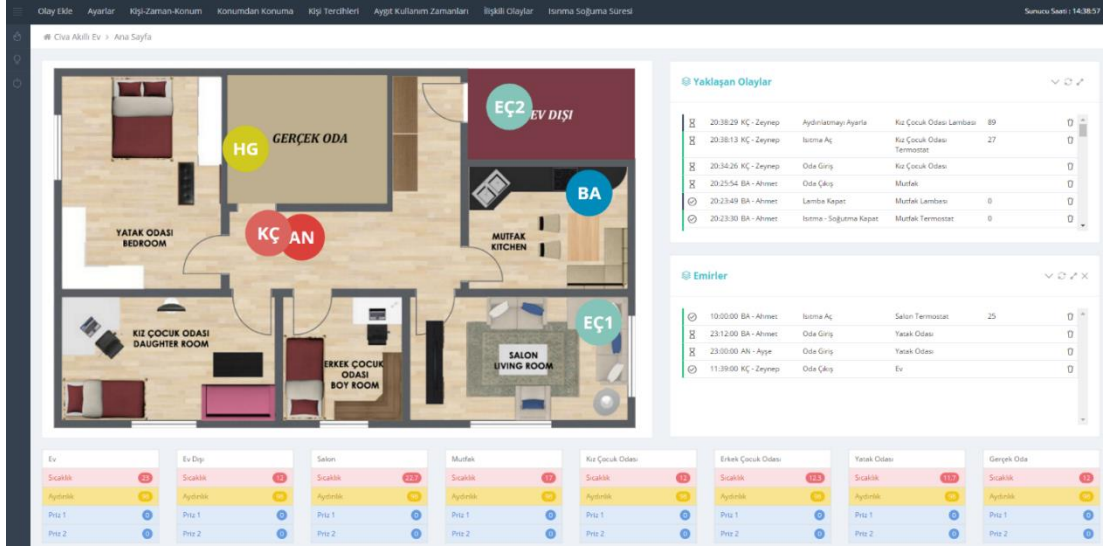
Akıllı ev hibrit simülasyonu için kişiler, odalar yapay kayıtlar üretilirken hâlihazırda üretilmiştir. Bunlara ek olarak gerçek kişi, gerçek oda ve gerçek aygıtları ile birlikte sanal evde sanal odaların her birinde bir lamba, bir termostat ve iki adet priz olduğu varsayılarak gerekli aygıt bilgileri ilgili veritabanı tablolarına eklenmiştir.

Hibrit simülasyon için kişiler, odalar, aygıt ve senaryoların hazır hale getirilmesinin ardından simülasyon geliştirilmeye başlanmıştır. İlk olarak hibrit simülasyonun sahip olması gereken özellikler belirlenmiştir. Her noktadan kolaylıkla takip edilebilmesi ve her zaman çalışması için web tabanlı olarak geliştirilmesine ve merkez sunucu üzerinde çalıştırılmasına karar verilmiştir. Gerçekleştirmesi gereken işlemler ise şu şekilde belirlenmiştir:

- Tüm kişiler, aygıtlar ve odaların durumu her an görülebilmeli,
- Kişilerin konumları değiştirilebilmeli,
- Aygıtların durumları değiştirilebilmeli,
- Sisteme olaylar eklenebilmeli,
- Sana olarak gerçekleşecek olaylar görülebilmeli.

İlk olarak tüm bu işlemlerin takip edilebileceği web arayüzü tasarlanmıştır. Web arayüzü Şekil 3.36'da gösterilmektedir (Şekil D.1).





Őekil 3.36: Akıllı ev simülasyonu arayüzü

### 3.16.1 Aygıtlar

Hibrit simülasyonda aygıtların anlık bilgileri sol üstte yer alan alev, lamba ve açma/kapama düğmesi simgelerinden takip edilmektedir. Ayrıca bu bölümden aygıtlar ile ilgili yapılabilecek her türlü işlem yapılabilmektedir. Őekil 3.37’de bütün odalardaki termostatların o anki durumları ve gösterdikleri deęerler sunulmaktadır.

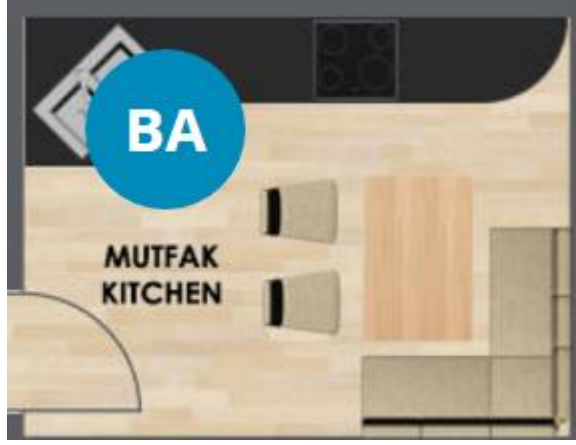


Şekil 3.37: Odalarda bulunan termostat bilgileri

Termostat ayarını değiştirmek için şeritler üzerinde yer alan çubukları sürüklemek yeterli olmaktadır. Burada yapılan işlem merkez sunucuya tıpkı akıllı ev sisteminin gerçek kontrolünün yapıldığı web arayüzünden olduğu gibi iletilmektedir. İşlem sanal bir aygıtla yapılmış ise merkez sunucuda veriler veritabanına kaydedilmekte ve bunun dışında herhangi bir işlem yapılmamaktadır. Eğer işlem gerçek termostat ile yapılmış bir işlemse daha önce aygıtlar arası işlemler bölümünde anlatıldığı gibi gerçek termostatın ayarı değiştirilinceye kadar yapılan işlem önce merkez sunucuya, daha sonra oda kontrolcüyeye ve son olarak termostata iletilmektedir.

### 3.16.2 Kullanıcılar

Kullanıcılar arayüzde daire şeklinde, renkli ve üzerlerinde onların kim olduğu gösteren simgelerle ifade edilmektedir. Simgelerin bulunduğu konum kişinin hangi odada bulunduğunu göstermektedir. Kişi konum değiştirdiğinde simgesi de birkaç saniye içerisinde eski konumundan alınarak yeni konumuna yerleştirilmektedir. Şekil 3.38’de BA kullanıcısının (Baba) mutfakta olduğu gösterilmektedir.



Şekil 3.38: Kullanıcıların konum gösterimi

### 3.16.3 Odalar

Oda bilgileri arayüzde alt bölgede gösterilmektedir. Burada odanın sıcaklığı, aydınlık düzeyi ve odada bulunan prizlerin durumları gösterilmektedir. Şekil 3.39’da mutfağın 17 °C sıcaklıkta olduğu, ortamın parlaklık düzeyinin 98 olduğu ve mutfakta bulunan her iki prizinde kapalı olduğu gösterilmektedir.

Mutfak	
Sıcaklık	17
Aydınlık	98
Priz 1	0
Priz 2	0

Şekil 3.39: Oda bilgileri

### 3.16.4 Olay Ekle

Normal şartlarda hibrit simülasyonda bütün olaylar ya senaryoya göre sanal olarak gerçekleşmektedir ya da gerçek kullanıcı tarafından gerçek aygıtlar kullanıldığında gerçekleşmektedir. Ancak yapay zekâ çalışmalarında algoritmaların farklı durumlara ve olaylara nasıl karşılık verdiğini tespit edebilmek için rutin işlemlerin dışına çıkılması gerekebilmektedir. Bunun için hibrit simülasyona

istenilen saatte istenilen kişinin istenilen işlemi yapabilmesini sağlayan bir olay ekleme menüsü eklenmiştir (Şekil 3.40).

Olay Ekle

Kişi AN - Ayşe

Oda Erkek Çocuk Odası

Tarih - Saat

Aygıt Salon Termostat

Ayar 0 50 100

İşlem Oda Giriş

İptal Olay Kaydet

Şekil 3.40: Olay Ekle menüsü

Bu menüde öncelikle kullanıcı seçilmektedir. Daha sonra eğer işlem bir oda giriş/çıkış olayı ise oda, değilse bir aygıt seçilmektedir. Son olarak tarih, saat ve işlemin tipi seçilerek olay kaydedilmektedir. Kaydedilen olay vakti gelince tıpkı web arayüzünden işlem yapılmış gibi gerçekleştirilmektedir.

### 3.16.5 Yaklaşan Olaylar

Hibrit simülasyon gerçek bir kullanıcının kendi hayatı ve sanal bir ailenin üretilmiş hayat senaryosunu yaşaması üzerine kurulmuştur. Örneğin sanal ailenin babası pazartesi akşamları saat 19.00'da eve gelmektedir. Bu bilgi kişilerin senaryosu üretildiğinde veritabanına kaydedilmiştir.

Hibrit simülasyon kişilerin haftalık hayat senaryolarını on dakika önceden olacak şekilde kontrol etmektedir. Eğer senaryoya göre on dakika sonra

gerçekleşecek herhangi bir olay varsa bu olay “yaklaşan olaylar” tablosuna eklenmektedir. Ayrıca yaklaşan olaylar arayüzde gösterilmektedir (Şekil 3.41).



Time	User	Action	Location	Value	Icon
20:38:29	KÇ - Zeynep	Aydınlatmayı Ayarla	Kız Çocuk Odası Lambası	89	🗑️
20:38:13	KÇ - Zeynep	Isıtma Aç	Kız Çocuk Odası Termostat	27	🗑️
20:34:26	KÇ - Zeynep	Oda Giriş	Kız Çocuk Odası		🗑️
20:25:54	BA - Ahmet	Oda Çıkış	Mutfak		🗑️
20:23:49	BA - Ahmet	Lamba Kapat	Mutfak Lambası	0	🗑️
20:23:30	BA - Ahmet	Isıtma - Soğutma Kapat	Mutfak Termostat	0	🗑️

**Şekil 3.41:** Yaklaşan Olaylar menüsü

Yaklaşan olaylar istenirse kullanıcı tarafından çöp kutusu simgesine tıklanarak gerçekleşmeden silinebilmektedir. Ekranda yaklaşan olayların yanında yer alan tik işareti olayın gerçekleştiğini, kum saati simgesi ise olayın henüz gerçekleşmediğini göstermektedir.

Listede yer alan olayların gerçekten o işlem yapılmış gibi anlık olarak takip edilmesi ve sisteme işlenmesi için merkez sunucuda her beş saniyede bir yaklaşan olay olup olmadığını kontrol ederek varsa işlemi gerçekleştiren bir yazılım geliştirilmiştir. Bu yazılım hem yaklaşan olayları hem de olay ekle menüsünden eklenen olayları sisteme işlemektedir (Yazılımda Türkiye zaman dilimi değişikliği iptal edildiği için buna uygun Bağdat zaman dilimi değeri kullanılmıştır).

```
var CronSaniyeJob = require('cron').CronJob;
require('log-timestamp');
new CronSaniyeJob('* /5 * * * * *', function() {
  async.parallel({
    saniyelik_olaylari_gerceklestir: function(callback)
    {
      setTimeout(function() {
        istemci.get('/saniyelik_olaylar/', function(err, res,
        data) {
          callback(null, 1);
        }
      }
    }
  }
});
```

```
        return;  
    });  
    }, 150);  
    }  
    }, function(err, results) {  
        console.log(results);  
    });  
    }, null, true, 'Asia/Baghdad');
```

Yaklaşan olayların işlenmesi sırasında bir noktaya da dikkat edilmektedir. Eğer gerçekleştirilecek işlemin bulunduğu odada işlemi gerçekleştirilecek kişiden o oda için daha yetkili biri varsa o işlem gerçekleştirilmemektedir. Sistemin böylece yapay zekâ işlemler için hatalı veriler üretmesinin de önüne geçilmektedir.

Hibrit simülasyonunda çalışı hale getirilmesi ile birlikte yapay zekâ algoritmalarının geliştirilmesi, test edilmesi ve iteratif olarak iyileştirilmesi çalışmalarına geçilmiştir.

### 3.17 Yapay Zekâ

Geliştirilen akıllı ev sisteminde kullanıcı konforunu artırmak ve birçok işlemin otomatik yapılabilmesini sağlamak için birçok farklı işlemi gerçekleştiren yapay zekâ algoritmaları geliştirilmiştir. Bu yapay zekâ algoritmalarının neler olduğu, ne tür görevleri yerine getirdiği, nasıl çalıştıkları ve nasıl geliştirildikleri bu bölümde anlatılacaktır.

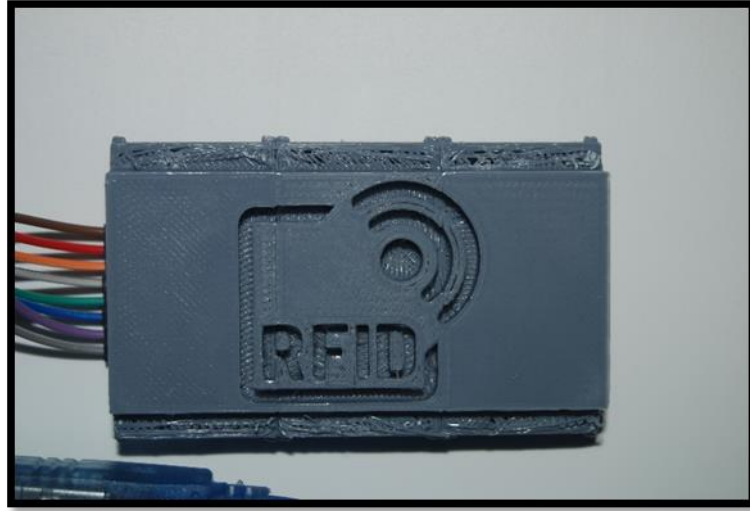
Genellikle mevcut akıllı ev sistemlerinde yapay zekâ tek tip işlem gerçekleştirmektedir ancak geliştirilen bu akıllı ev sisteminde yapay zekâ ile farklı türde birçok işlem gerçekleştirilmiştir. Bu işlemler etki tepki işlemleri, yetkilendirme, kişilerin konum tespiti, belirli zaman aralığında tekrarlanan olayların tespiti, birbirleriyle ilişkili olayların tespiti, kişilerin akıllı ev sistemini kullanım tercihlerinin tespiti, evin ısınma ve soğuma zaman aralıklarının belirlenmesi ve son olarak kişi hareketlerine ve değişen şartlara göre kullanım alışkanlıklarına uygun şekilde sistemin otonom yeni duruma uyum sağlamasıdır.

### 3.17.1 Etki Tepki İşlemleri

Etki tepki işlemleri akıllı ev sisteminde sensörlerden algılanan bir değer sonucu ya da evde herhangi bir olay gerçekleştiğinde otomatik olarak gerçekleştirilecek işlemlerdir. Binaların merdivenlerindeki lambaların hareket algılandığında otomatik açılması ve hareket bittiğinde belirli bir süre sonra otomatik olarak kapanması bu olaya örnek verilebilir. Bunun dışında son zamanlarda yeni yapılan evlerde standart hale gelen evin içerisinde koridorlara yerleştirilen led lambalarla koridorun aydınlatılmasının otomatik olarak gerçekleştirilmesi de örnek olarak gösterilebilir.

Bu çalışmada bu işleme benzer olarak akıllı ev sistemindeki oda kontrolcüsüne yaklaşıldığında otomatik olarak akıllı kart okuyucunun (Şekil 3.42) etrafı aydınlatılmaktadır. Kişiler böylece kolay bir şekilde oda giriş çıkış işlemlerini gerçekleştirebilmektedir. Ayrıca bu işlem tamamen tümleşik sensör kontrolünde gerçekleştirilmektedir. Böylece sistemde bağlantılarda olabilecek sorunlardan bu işlem etkilenmeyecektir. Bu vb. işlemlerin genel mantığı sensörlerden gelen verilerin sürekli olarak her an kontrol edilmesi ve beklenen değerler okunduğunda işlemin gerçekleştirilmesi şeklindedir. Geliştirilen sistemde de tümleşik sensör bileşeninin işlemci döngüsünde sürekli olarak hareket sensörünün durumu kontrol edilmektedir. Hareket algılandığı anda akıllı kart okuyucunun etrafındaki ışıkların (led lamba) bağlı olduğu bağlantı noktasına dijital çıkış verilmektedir. Hareket işlemi olmadığında da dijital çıkış sonlandırılmaktadır.

```
degerSensorHareket = digitalRead(bnSensorHareket);  
if (degerSensorHareket == 1)  
    digitalWrite(akilliKartAydinlatma, HIGH);  
else  
    digitalWrite(akilliKartAydinlatma, LOW);
```



Şekil 3.42: Etki tepki işlemi ile kart okuyucunun etrafının aydınlatılması

### 3.17.2 Yetkilendirme

Yetkilendirme, akıllı ev otomasyon sisteminin en önemli noktalarından birisidir. Çünkü bu tür sistemlerde her kişinin her ayarı yapması düşünülemez. Örneğin mutfakta anne bulunurken evin ufak çocuğu mutfağın sıcaklık ayarını değiştirememelidir. Bu vb. durumların önüne geçebilmek ve sistemin daha doğru bir şekilde çalışabilmesi için sistemde yetkilendirme işlemlerinin yapılabilmesini sağlayan bir algoritma geliştirilmiştir.

Yapay veri üretici bölümünde kullanıcılar için yetkilendirme değerleri belirlendiği sunulmuştu. Her kullanıcı için her odaya göre 0-10 aralığında bir değer belirlenmektedir. 10 en yüksek yetki seviyesini 0 ise en düşük yetki seviyesini belirtmektedir. Bir kullanıcının her oda için yetkileri veritabanında oda kodları ile birlikte şu şekilde tutulmaktadır:

```
[{"og":5}, {"os":"10"}, {"om":"8"}, {"ok":"5"}, {"oe":"5"}, {"oy":"9"}, {"go":"0"}]
```

Kullanıcı herhangi bir işlem gerçekleştğinde algoritma işlemin gerçekleştirildiği odada kullanıcının kendisinden daha yetkili bir kişi bulunup bulunmadığını kontrol etmektedir. Eğer kullanıcıdan daha yetkili bir yoksa işlem gerçekleştirilmekte varsa işlem gerçekleştirilmemektedir (Program Kodu: EK E).



### 3.17.3 Kişilerin Konumunu Tespit Etme

Kişilerin ne zaman nerede olacağını bilmesi akıllı ev sisteminin en önemli noktalarından birisidir. Çünkü kişiler genellikle gerçekleştirdikleri işlemleri odada buldukları zamanlarda ya da oda giriş çıkışları ile ilişkili zamanlarda gerçekleştirmektedir. Örneğin kişiler genellikle odaya girerken lambayı açarlar ve çıkarken de kaparlar veya odaya gelmeden ya da eve gelmeden belirli bir süre önce termostat açılır.

Bu çalışmada kişilerin konum tespiti için iki farklı yöntem bir arada kullanılarak daha sağlıklı ve daha hızlı sonuçlar elde edilmeye çalışılmıştır. Tüm yöntemler için öncelikle sistemi kullanan kullanıcıların sürekli olarak konumları ve oda giriş çıkış işlemleri zaman değerleriyle birlikte kayıt altına alınmıştır.

Kişilerin konum ve hareket kayıtları, iki farklı konum tespit işlemi için farklı şekillerde tutulmuştur. Bunlardan ilkinde tüm kullanıcıların hangi tarihte hangi odada bulunduğu bilgileri dakikalık olarak kayıt altına alınmıştır. Kayıtlar Şekil 3.43'teki gibi tutulmaktadır.

zaman	kisi_1_konum	kisi_4_konum	kisi_5_konum	kisi_6_konum	kisi_7_konum	kisi_8_konum
2015-06-25 07:35:00	8	4	6	7	7	10
2015-06-25 07:34:00	8	4	6	7	7	10
2015-06-25 07:33:00	8	4	6	7	7	10
2015-06-25 07:32:00	8	4	6	7	7	10
2015-06-25 07:31:00	8	4	6	7	7	10
2015-06-25 07:30:00	8	4	6	7	7	10
2015-06-25 07:29:00	8	4	6	7	7	10
2015-06-25 07:28:00	8	4	6	7	7	10

Şekil 3.43: Tarihe göre kullanıcı konum kayıtları

Bir diğer kayıt tutma şekli ise yine kullanıcıların dakikalık olarak kaydının tutulmasıdır ancak bu işlem daha farklıdır. Bu işlem için kullanıcıların haftalık periyotlarda alışkanlıklarının tekrarlandığı, ayrıca hafta içi ve hafta sonu davranışlarının ya da haftanın her günü için buldukları konumların (Örn: her pazartesi) birbirine benzeyebileceği veya her gün benzer şekillerde hareket ediyor olabilecekleri varsayılmıştır. Bu varsayımlardan hareketle kullanıcıların gerçekleştirdikleri oda giriş çıkış işlemlerinin tarih yerine sadece saat bilgileri

alınarak bulunulan haftanın gününe, hafta içi ve hafta sonuna ayrıca her güne göre kayıtları tutulmuştur. Kayıtlar Şekil 3.44'teki görünmektedir.

saat	kisi	bulundugu_oda	gunluk	gunluk_haftaici	gunluk_haftasonu	pazartesi	salı	carsamba	persembe	cuma	cumartesi	pazar
18:34:00	1	3	125	87	38	19	14	13	21	20	20	18
18:34:00	1	11	0	0	0	0	0	0	0	0	0	0
18:34:00	1	4	0	0	0	0	0	0	0	0	0	0
18:34:00	1	10	0	0	0	0	0	0	0	0	0	0
18:34:00	1	8	24	18	6	4	3	3	4	4	4	2
18:34:00	1	7	0	0	0	0	0	0	0	0	0	0
18:34:00	1	5	0	0	0	0	0	0	0	0	0	0
18:34:00	1	6	0	0	0	0	0	0	0	0	0	0
18:33:00	1	8	24	18	6	4	3	3	4	4	4	2

Şekil 3.44: Saate göre kullanıcı konum kayıtları

Tablodan da görüleceği üzere her kullanıcı için bulunduğu her konuma göre dakikalık olarak veriler tutulmaktadır. Bu veriler kullanıcının daha önce o saatte o odada günlük(tüm günler), hafta içi günlük (sadece hafta içi günlerde), hafta sonu günlük(sadece hafta sonu günlerde) ve haftanın o gününe göre kaç defa kayıt altına alındığını ifade etmektedir. Bu değerler her dakika merkez sunucuda çalışan bir algoritma ile sürekli olarak artırılarak güncellenmektedir.

Örneğin pazartesi günü 18.34'te 1 numaralı kişi 6 numaralı odada bulunduğu varsayılırsa, saatin 18.34, kişinin 1 odanın 6 olduğu kayıt değeri günlük 1 olarak, günlük hafta içi 1 olarak, pazartesi 1 olarak güncellenecektir. Bu güncellemeleri yapan algoritma aşağıdaki gibidir ve her dakika bir defa çalıştırılmaktadır.

*KişiKonumArttır*

*saat <- Sistem saati oku*

*hafta\_ici\_sonu <- Hafta içi/sonu oku*

*gun <- Haftanın gününü oku*

*Tüm kişiler (kişi)*

*konum <- Kişi tablosundan kişinin şu anki konumunu oku*

*Kayıdı Bul(kişi, saat, konum)*

*kayıt <-kişi, saat ve konumun eşit olduğu kaydı oku*

*kayıt günlük değeri 1 arttır.*

*kayıt haftanın günü değerini 1 arttır*

*Eğer hafta\_ici\_sonu = Hafta içi ise*

*günlük haftaiçi değeri 1 arttır*

*Değilse*

*Günlük haftasonu değeri 1 arttır.*

Bu işlem için geliştirilen algoritma Markov modeli ile karınca koloni algoritmasından esinlenilerek geliştirilmiştir. Karınca koloni algoritmasının temeli karıncaların geçtikleri konumlara bıraktıkları feromon maddesinin yoğunluğuna göre hangi yolun daha kısa olduğunu tespit etmeye dayanmaktadır. Markov modeli ise stokastik süreçler sonucunda sistemin bulunulan durumdan bir sonraki duruma eski durumlardan bağımsız olarak geçiş olasılıklarını belirlemektedir. Bu algoritmada da bu yöntemlere benzer şekilde kullanıcı bir konumda kaldıkça o karıncaların feromon seviyesinin artmasına benzer şekilde konumdaki değeri bir artırılmaktadır. Bu değerler ile Markov modelinde olduğu gibi eski değerlerden çıkarım yapılarak mevcut durumda kullanıcının hangi konumda bulunabileceği olasılıkları hesaplanabilmektedir.

Kullanıcı konum tespiti için verilerin kayıt altına alınmasından sonra kullanıcı konum tespit işlemine geçilmiştir. Yapılmak istenilen şey belirtilen kullanıcının, belirtilen saatte evde mi yoksa değil mi, evde ise hangi odada bulunduğu sorularının cevaplarını bulmaktır.

Kullanıcı tespit işlemi için belirtilen saatte kullanıcının son olarak bahsedilen tablodan kayıtları çekilmektedir. Örneğin Şekil 3.44'te de görülen 1 numaralı kullanıcının saat 18.34'te nerede olduğunun bulunmak istendiği varsayılırsa saat 18.34 ve kullanıcının 1 olduğu tüm kayıtlar çekilmektedir. Kayıtlar incelendiğinde kullanıcının daha önce belirtilen dakikada yalnızca 3 ve 8 numaralı odalarda bulunduğu anlaşılmaktadır. Kullanıcı belirtilen saatte 135 defa 3 numaralı odada 24 defa ise 8 numaralı odada kayıt altına alınmıştır. Günün hafta içi olduğu varsayılırsa 87-18, pazartesi olduğu varsayılırsa 19-4 şeklinde veriler elde edilecektir. Bu veriler Min-Max normalleştirilmesi ile normalleştirildiğinde kullanıcının belirtilen saatte hangi konumda bulunabileceği oransal olarak tespit edilmiş olacaktır.

Normalleştirme işlemi için öncelikle kişinin belirlenen konumlardan en az olarak bulunduğu konumun değeri (Min) ile tüm bulunduğu konumların değerlerinin

toplamını (Max) tespit etmek gerekmektedir. Örnekte verilen günlük verilere bakıldığında değerler şu şekilde hesaplanmaktadır:

$$\text{Min} = 0 \quad (3.1)$$

$$\text{Max} = 135 + 24 = 159$$

Kişinin hangi konumda bulunacağı olasılıklarını hesaplamak için değerler formülde yerlerine yerleştirildiğinde:

$$P(K_3|x) = \frac{K_3 - K_{min}}{K_{max} - K_{min}} = \frac{135 - 0}{159 - 0} = 0,84 \quad (3.2)$$

x (belirtilen saat) bilinenine göre kişinin konum 3'te bulunma olasılığı 0-1 aralığında 0,84 olarak hesaplanmıştır. Yani kullanıcının belirtilen saatte %84 oranında konum 3'te bulunacağı sonucuna varılmıştır.

İşlemlerin bir sonraki adımı kullanıcının tüm günlere yönelik olarak belirlenen konum olasılığının yanında hafta içi ve hafta sonu ile haftanın sadece o gününe göre konum olasılıklarının hesaplanmasıdır. Bu hesaplamalarda aynı şekilde yapıldığında öncelik olarak önce haftanın gününe göre, sonra hafta içi ve hafta sonuna göre son olarak ta her güne göre çıkan olasılıklardan en büyük olasılığa göre kullanıcının hangi konumda olabileceğine karar verilmektedir. Eğer herhangi bir olasılık değeri 0,5 üzerindeyse kullanıcının o konumda olacağı tahmin edilmektedir. Eğer böyle bir değer yoksa haftanın o gününe göre en yüksek olasılık kullanıcının bir sonraki konumu olarak tespit edilmiş olmaktadır.

x: Belirtilen saat

K: Günlük (Haftanın tüm günlerinde o saate göre)

I: Hafta içi ya da hafta sonu

C: Haftanın o gününe göre (Örn: pazartesi) (3.3)

$$\text{Seç Konum eğer} \begin{cases} P(C_x|x) \geq 0,5 \\ P(I_x|x) \geq 0,5 \\ P(K_x|x) \geq 0,5 \end{cases} \text{ değilse } P(C_x|x)_{max}$$

Literatür taramasında ve incelenen programlarda çok çeşitli konum tespit yöntemleri olmasına karşın bu çalışmada birçok çalışmada gözden kaçan bir nokta

tespit edilmiştir. Bu noktada kişilerin alışkanlıklarının değişebileceği noktasıdır. Örneğin senaryoda da belirtildiği üzere haftanın belirli bir günü belirli bir saatte halı saha maçlarına giden kişi maçların farklı bir güne alınması ile o gün ve maçın zamanının değiştiği diğer güne göre alışkanlıklarını tamamen değiştirebilir. Yukarıda geliştirilmiş olan yapay zekâ konum tespit algoritması ve birçok çalışmada yer alan algoritmalar böyle bir durumda yeni duruma adapte olabilmek için çok uzun zamana ihtiyaç duymaktadır. Örneğin bir kişinin bir yıllık verilerinin kayıt altına alındığı varsayıldığında, kullanıcının davranış değişikliğini sistemin fark edebilmesi ve ona göre doğru tahminler yürütebilmesi de bir yıl sürecektir. Bu sorunun çözümü için bu çalışmada geliştirilen algoritmanın üzerine yeni bir yapay zekâ algoritması daha geliştirilmiştir. Bu algoritma kullanıcının sadece son zamanlarda neler yaptığına odaklanmaktadır.

İkinci algoritma için kayıt olarak ilk bahsedilen kayıt yöntemi ile tutulan kayıtlar kullanılmaktadır. Bu kayıt şeklinde tüm kayıtlar tarih ve saat bilgisi ile birlikte saklanmaktadır. Böylece bu veriler kullanılarak kişinin genelde değil de son zamanlarda nasıl hareket ettiği tespit edilebilmektedir. Örneğin kişinin pazartesi günü saat 12.00’da nerede olduğunu tespit edebilmek için veri tabanından haftanın gününün pazartesi ve saatin 12.00 olduğu son on kayıt çekilir. Çekilen kayıtlar Şekil 3.45’teki gibi görünmektedir.

idyz_dakika_kayit	zaman	kisi_1_konum
81	2016-09-12 12:00:00	11
1520	2016-09-19 12:00:00	11
2294	2016-09-26 12:00:00	1
5009	2016-10-03 12:00:00	11
6447	2016-10-10 12:00:00	11
7887	2016-10-17 12:00:00	8
9316	2016-10-24 12:00:00	11
10749	2016-10-31 12:00:00	8
12111	2016-11-07 12:00:00	8
13545	2016-11-14 12:00:00	11

**Şekil 3.45:** Kişinin son 10 pazartesi günü belirtilen saatteki konumları

Kayıtlar 1 numaralı kişinin son on pazartesi gününde saat 12.00’da hangi konumlarda olduğunu göstermektedir. Bu verileri kullanarak kullanıcının son on gün

içinde hangi odada hangi olasılıkta bulunduğu veriler normalleştirilerek şu şekilde hesaplanır:

$$P(K_8|x) = \frac{\sum K_8}{\sum K} = \frac{3}{10} = 0,3$$

$$P(K_1|x) = \frac{\sum K_1}{\sum K} = \frac{1}{10} = 0,1 \quad (3.4)$$

$$P(K_{11}|x) = \frac{\sum K_{11}}{\sum K} = \frac{6}{10} = 0,6$$

Formüllerden elde edilen değerlere göre kişi %60 oranla 11 numaralı odada %30 oranla 8 numaralı odada %10 oranla 1 numaralı odada bulunacaktır.

Son olarak kullanıcı konum tespit işleminde elde edilen verilere göre kullanıcının hangi odada bulunabileceğine karar vermek gerekmektedir. Bu işlem için öncelikle son on haftanın o günü değerlerine bakılmaktadır. Eğer bu değerlerden herhangi biri 0,5 değerine eşit ya da üstündeyse kişinin o odada olacağına karar verilmektedir. Eğer herhangi bir değer 0,5'e eşit ya da üzerinde değilse bir önceki algoritmadaki seçeneklere göre işleme devam edilmektedir. O algoritmadan elde edilen değerlerden haftanın o gününe göre hesaplanan olasılık değeri 0,5'e eşit ya da üzerindeyse kişinin o odada olacağına karar verilmektedir. Eğer değilse aynı işlem sırasıyla hafta içi/sonu ve tüm günler için hesaplanan olasılık değerlerine göre devam ettirilmektedir. Eğer herhangi bir değer 0,5'e eşit ya da üzerinde değilse değerler yine sırasıyla 0,4'ün üzerinde olan en büyük değer konum olarak belirlenmektedir. Eğer herhangi bir değer 0,4'ün üzerinde değilse kişi için konum tespit edilmemektedir. Ayrıca karar algoritmasındaki 0,4 ve 0,5 parametreleri istenildiği zaman değiştirilebilmektedir. Sistemin karar algoritması şu şekildedir (Program kodları: EK F):

*KararVer*

*Son10 <-Seç ->En Büyük (Son 10 Haftanın O Günü Konum Olasılıkları)*

*Günlük <- Seç -> En Büyük Haftanın Tüm Günleri Konum Olasılığı*

*Hafta İçi Sonu <- Seç -> En Büyük Hafta içi/sonu Konum Olasılığı*

*OGün <- Seç ->En Büyük O Gün İçin Konum Olasılıkları*

*Eğer Son10  $\geq 0,5$*

*Kişi Konum <- Son10.Konum*

*Bitir*

*Değilse*

*Eğer OGün  $\geq 0,5$*

*Kişi Konum <- OGün.Konum*

*Bitir*

*Eğer Hafta İçi Sonu  $\geq 0,5$*

*Kişi Konum <- Hafta İçi Sonu.Konum*

*Bitir*

*Eğer Günlük  $\geq 0,5$*

*Kişi Konum <- Günlük.Konum*

*Bitir*

*Konum En Büyük <- Seç -> En Büyük (Tüm Olasılıklar)*

*Eğer Konum En Büyük  $\geq 0,4$*

*Kişi Konum <- Konum En Büyük.Konum*

*Değilse*

*Kişi Konum <- Tespit Edilemedi*

#### **3.17.4 Aygıtların Durumlarını Tespit Etme**

Kişilerin konumlarının tespit edilmesi gibi aygıtlarda da istenilen zamanda aygıtın durumunun tespit edilmesi için bir yapay zekâ algoritması geliştirilmiştir. Çünkü bütün aygıt işlemleri kişilerin oda giriş çıkış işlemleri ile ilgili olmayabilmektedir. Örneğin gece sürekli olarak belirli bir vakitte koridorda bulunan bir lamba belirli bir parlaklıkta olacak şekilde ayarlanabilir. Ayrıca bu işlem tek bir kullanıcı tarafından değil her gün farklı bir kullanıcı tarafından da gerçekleştiriliyor olabilir.

İstenilen aygıtın istenilen vakitte genel olarak hangi ayarla bulunduğunu tespit etmek için kullanıcının son zamanlarında hangi odada bulunduğunu tespit eden algoritma ile neredeyse aynı şekilde bir algoritma geliştirilmiştir.

Bu algoritma için dakikalık olarak sürekli şekilde tüm aygıtların durumları kayıt altına alınmaktadır. Örneğin saat 14.10 da 3 numara odada yer alan termostatın veriler Şekil 3.46’da gösterilmiştir.

zaman	oda_3_termostat_1_onoff	oda_3_termostat_1_value
2015-01-14 14:10:00	0	0
2015-01-21 14:10:00	1	26
2015-01-28 14:10:00	1	24
2015-02-04 14:10:00	1	25
2015-02-11 14:10:00	0	0
2015-02-18 14:10:00	0	0
2015-02-25 14:10:00	1	27
2015-03-04 14:10:00	1	25
2015-03-11 14:10:00	1	26
2015-03-18 14:10:00	1	25

Şekil 3.46: Termostat durum kayıtları

Kayıtlar salonda bulunan termostatın son 10 Çarşamba gününde saat 14.10’da açık kapalı olup olmadığını ve açıksa kaç dereceye ayarlandığını göstermektedir. Algoritma öncelikle bu şekilde haftanın o gününe göre son 10 günlük verileri çekmektedir. Daha sonra açık kapalı durumları saydırılarak 0-1 aralığında açıklık ve kapalılık durumları belirlenmektedir.

$$P(T_{açık} = \frac{\sum T_{açık}}{10} = \frac{7}{10} = 0,7 \quad (3.5)$$

Son olarak aygıtın bulunduğu durumun tespit edilmesinin ardından tespit işleminde elde edilen verilere göre aygıtın açılıp açılmayacağına karar verilmektedir. Eğer elde edilen açık kapalı değerlerden herhangi biri 0,7 veya üzerindeyse sistem o aygıtı belirlenen duruma otonom olarak getirecektir. Ayrıca eğer termostatın



açılmasına karar verilirse son 10 veride elde eden aygıt için belirlenen değerlerin aritmetik ortalaması alınarak bu değer aygıta uygulanacaktır.

### 3.17.5 Kullanıcı Tercihleri Belirlenmesi

Daha önce kullanıcı-konum-zaman algoritması ile kullanıcıların ne zaman nerede olacakları tespit edilmektedir. Ancak bu algoritma tek başına yeterli olmamaktadır. Çünkü kullanıcı eğer oda soğuksa daha önceden termostatu açabilir ya da odaya girdiği anda oda karanlıksa lambayı açabilir. Bunların yanında eğer termostat ve lamba istediği ayarlarda değilse ayarlarını değiştirebilir. Bu işlemleri otonom olarak gerçekleştirebilmek için kullanıcıların hangi sıcaklık değerlerinde termostatu açtıkları, hangi aydınlık değerlerinde ise lambayı açtıkları, ayrıca bu aygıtları açtıklarında hangi değerlere ayarladıklarının tespit edilmesi gerekmektedir.

Kullanıcıların sıcaklık ve aydınlık tercihlerini tespit edebilmek için daha önce yaptıkları işlemlerin kayıtlarını incelemek yeterli olmaktadır. İstatistiksel yöntemlerle bu kayıtlardan kullanıcı tercihleri belirlenebilmektedir. Bir kullanıcının tercihlerini belirlemek için veri tabanının kayıt tablosunda şu sorgu yapılmaktadır:

```
$CI = get_instance();
$sonuc = array();
$CI->db->select_max("deger", "deger_en_buyuk");
$CI->db->select_min("deger ", "deger_en_kucuk");
$CI->db->select("AVG(deger) AS ortalama");
$CI->db->select_max("sicaklik", "sicaklik");
$CI->db->where("kullanici", $kisi);
$CI->db->where("islem", 3);
$CI->db->where("HOUR(tarih_saat) >=", 7);
$CI->db->where("HOUR(tarih _ saat) <", 19);
$d_veri = $CI->db->get("kayit")->result();
```

Bu sorgulama sonucunda veritabanından kişinin gündüz vakitlerinde termostatu açtığı en yüksek sıcaklık, en düşük sıcaklık ve açtığı sıcaklık değerlerinin

ortalaması ile sıcaklık açtığı en yüksek oda sıcaklığı değeri dönmektedir. Bu sorgulamadan dönen değerlere göre eğer kişinin gireceği oda kullanıcının termostat açtığı en yüksek oda sıcaklığı değerinden düşük bir değere sahipse termostat belirli bir süre(süre tespiti başka bir algoritma ile tespit edilmektedir) önce açılmaktadır. Aydınlatma işlemi için de algoritma aynı şekilde çalışmaktadır, sadece aydınlatma işlemi kullanıcının oda giriş işlemi yapmasından sonra gerçekleştirilmektedir.

### **3.17.6 Odaların Isınma Soğuma Süreleri Hesaplanması**

Kullanıcıların sıcaklık tercihlerinin tespit edilmesinin ardından bu elde edilen verinin ne kadar süre önce kullanılacağı tespit edilmesi gerekmektedir. Örneğin kullanıcının 20 dakika sonra bir odada olacağı belirlendikten sonra termostat açma işleminin de yapılacağına karar verildiği varsayıldığında bu işlemin ne zaman gerçekleştirilmesi gerektiğinin belirlenmesi gerekmektedir. Çünkü eğer termostat erken açılırsa bu boşa enerji harcanmasına, geç açılırsa kullanıcı odaya geldiğinde odanın istenilen sıcaklığa ulaşamamasına neden olacaktır.

Odaların ne kadar sürede ısınıp soğuduğunu tespit etmek için tıpkı diğer yapay zekâ işlemlerinde olduğu gibi yine odanın sıcaklık değerleri ve odada bulunan termostata ait veriler sürekli olarak kayıt altına alınmaktadır (Şekil 3.47).

zaman	oda_3_sicaklik	oda_3_termostat_1_onoff	oda_3_termostat_1_value
2016-12-03 13:34:50	21.8	1	24
2016-12-03 12:09:54	20.9	1	24
2016-12-02 20:25:01	20	1	24
2016-12-02 20:24:02	19.1	1	24
2016-12-02 20:23:01	18.2	1	24
2016-12-02 20:22:02	17.3	1	24
2016-12-02 20:21:02	16.4	1	24
2016-12-02 20:20:02	15.5	1	24
2016-12-02 20:19:02	14.6	1	24
2016-12-02 20:18:01	13.7	1	24
2016-12-02 20:17:02	12.8	1	24
2016-12-02 20:16:02	11.9	1	24
2016-12-02 20:15:01	11	1	24
2016-12-02 20:14:01	11	0	0

**Şekil 3.47:** Oda sıcaklık ve termostat verileri dakikalık kayıtları

Şekildeki veriler bir odada 20.15'te açılan termostattan sonra odadaki sıcaklık değişimlerini göstermektedir. Geliştirilen yapay zekâ işlemi de oda sıcaklığındaki bu değişimlerin dakikalık olarak kaç derece gerçekleştiğini tespit etmeye çalışmaktadır. Bunu için istenilen odada son 1 haftalık kayıtlar veritabanından çekilmektedir (Son 10080 kayıt). Daha sonra bu çekilen kayıtlar sırayla incelenerek sıcaklık bir önceki değere göre düştü ise bu fark oda soğuma değerine eğer yükseldi ise oda ısınma değerine eklenmektedir. Ayrıca oda soğuma ve ısınma sayıları da saydırılmaktadır. Son olarak toplam oda ısınma ve soğuma değerleri soğuma ve ısınma sayılarına bölünerek odanın ısınma ve soğuma değerleri belirlenmektedir. Algoritma şu şekildedir:

*Oda Isınma Soğuma Hesapla*

*Son 10080 kayıt <-Seç-> Oda Aygıt Dakikalık Kayıt Tablosu*

*Tüm Kayıtlar İçin*

*Eğer Önceki Oda Sıcaklık <> Oda Sıcaklık*

*Eğer Önceki Oda Sıcaklık < Oda Sıcaklık*

*Artış Sayısı 1 arttır*

*Artış <- Artış + Oda Sıcaklık – Önceki Oda Sıcaklık*

*Eğer Önceki Oda Sıcaklık > Oda Sıcaklık*

*Azalış Sayısı 1 arttır*

$$\text{Azalış} <- \text{Azalış} + \text{Oda Sıcaklık} - \text{Önceki Oda Sıcaklık}$$

$$\text{Önceki Oda Sıcaklık} <- \text{Oda Sıcaklık}$$

$$\text{Artış Değeri} <- \text{Artış} / \text{Artış Sayısı}$$

$$\text{Azalış Değeri} <- \text{Azalış} / \text{Azalış Sayısı}$$

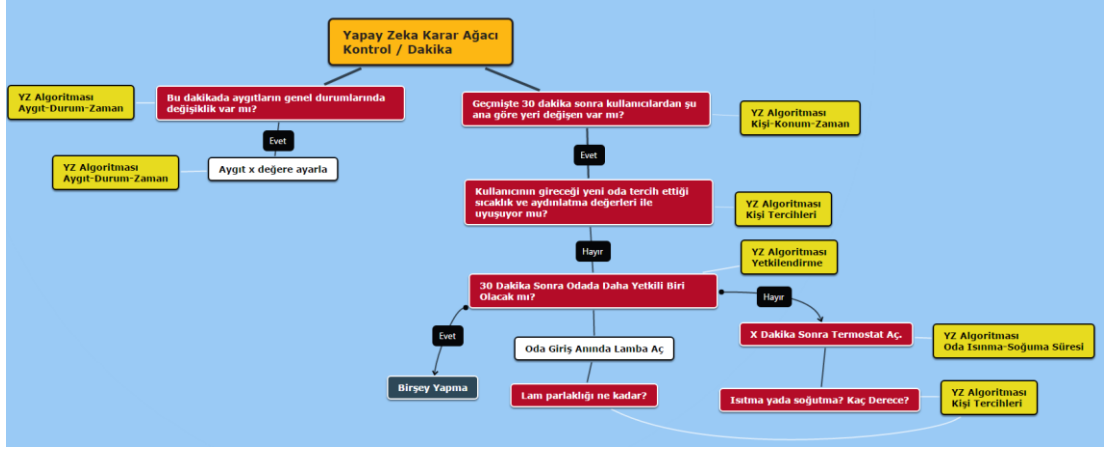
Odanın dakikalık olarak ısınma ve soğuma değerlerinin belirlenmesinin ardından kullanıcının odaya girmesinden kaç dakika önce termostatın açılması gerektiği ya da kullanıcının bulunduğu odadan çıkmasından kaç dakika önce termostatın kapanması gerektiği hesaplanabilmektedir. Bu hesaplama için kullanıcının tercih ettiği ortalama termostat değerinden mevcut oda sıcaklığı çıkarılır ve oda sıcaklık artış değerine bölünür. İşlemin sonucu kaç dakika önce termostatın açılması gerektiği bilgisini vermektedir.

$$x = \frac{\text{Kullanıcı Ortalama Sıcaklık Tercihi} - \text{Oda Sıcaklık}}{\text{Oda Sıcaklık Artış}} \quad (3.6)$$

### 3.17.7 Yapay Zekâ Algoritmalarının Sistem Üzerinde İşleyişi

Şu ana kadar akıllı ev sistemi geliştirilmiş, çalışır hale getirilmiş, sanal bir ev ve aile oluşturularak bu aile için haftalık yaşam senaryoları hazırlanmış, bu senaryolar ile yapay veriler üretilerek bu veriler üzerinde çalışan yapay zekâ algoritmaları yazılmış ayrıca bu sanal ailen ile gerçek bir evin odasında yaşayan bir kişiyi aynı evde yaşıyormuş gibi canlandıran bir hibrit simülasyon geliştirilmiştir. Çalışmanın nihai hedefine ulaşması için son nokta olarak geliştirilen yapay zekâ algoritmalarının hibrit simülasyona entegre olarak çalıştırılması kalmıştır.

Yapay zekâ algoritmalarının sistem ile birlikte gerçek zamanlı olarak çalışabilmesi için öncelikle bir karar ağacı geliştirilmiştir. Karar ağaçları çalışan sistemin bulunulan durumda değişen şartlara göre nasıl karar alması gerektiğinin ağaç dallanmaları şeklinde ifade edilmesidir. Sistem için geliştirilmiş karar ağacı Şekil 3.48'de gösterilmiştir (Büyük hali için: Şekil G.1).



Şekil 3.48: Yapay zekâ karar ağacı

Karar ağacı merkez sunucuda geliştirilen bir yazılım ile dakikada bir defa işletilmektedir. Öncelikle sistemdeki tüm kullanıcıların 30 dakika sonra (bir odanın en fazla 30 dakikada istenilen seviyede sıcaklığa ulaşacağı varsayılmıştır) bulunabilecekleri konumlar tespit edilerek şu anki konumları ile karşılaştırılmaktadır (Kişi-Konum-Zaman algoritması). Eğer bu karşılaştırma sonucunda kullanıcılardan konumunda değişiklik olanlar varsa kullanıcının gideceği odanın kullanıcının tercihleri ile uyuşup uyuşmadığı tespit edilmektedir (Kişi Tercihleri algoritması). Kullanıcı tercihleri odanın mevcut değerlerinden (sıcaklık) farklı ise 30 dakika sonra gidilecek odada daha yetkili birinin bulunup bulunmayacağına bakılmaktadır (Yetkilendirme algoritması). Daha yetkili biri odada bulunacaksa işlem bitirilmektedir, eğer daha yetkili biri o odada bulunmayacaksa termostat açıldığından kaç dakika sonra odanın istenilen seviyede sıcaklığa sahip olacağı belirlenmektedir (Oda Isınma-Soğuma Algoritması). Son olarak termostatın açıldığında kaç santigrat dereceye ayarlanacağı “Kişi Tercihleri” algoritması ile belirlenmektedir. Termostat açma işleminin tüm parametrelerinin belirlenmesinin ardından bu işlem “Yaklaşan Olaylar” tablosuna eklenmekte ve sistem tarafından zamanı geldiğinde işletilmektedir. Ayrıca eğer kullanıcı evin dışından gelmiyor da bir odasından başka bir odasına geçiyorsa çıktığı oda içinde benzer şekilde soğuma süreleri hesaplanmakta ve eğer odada 30 dakika sonra başka biri bulunmayacaksa “Yaklaşan Olaylar” tablosuna termostat kapama işlemi eklenmektedir. Bunlarla birlikte oda giriş ve oda çıkış anlarında benzer şekillerde odada bulunan lamba kapatılmakta ya da kişi tercihleri belirlenerek kullanıcının tercih ettiği düzeyde açılmaktadır.

Karar ağacının diğer dalında aygıtların zamana göre değişen durumları kontrol edilmektedir (Aygıt-Durum-Zaman algoritması). Çünkü kişilerin konum hareketlerinden bağımsız olarak gerçekleştirdikleri işlemler olabilir. Örneğin bir priz her gün saat 12.00'da açılıyorsa saat 12.00 olduğunda prizin otonom olarak açılması gerekmektedir. Bunu için aygıtların gelecek durumları "Aygıt-Durum-Zaman" algoritması ile tespit edilmekte ve eğer durumlarında bir değişiklik varsa değişen duruma göre aygıt için yeni işlem yapılmasına karar verilmektedir. Bu işlem de bir önceki işlemde olduğu gibi "Yaklaşan Olaylar" tablosuna eklenmekte ve 5 saniye içerisinde gerçekleştirilmektedir.

## **4. BULGULAR**

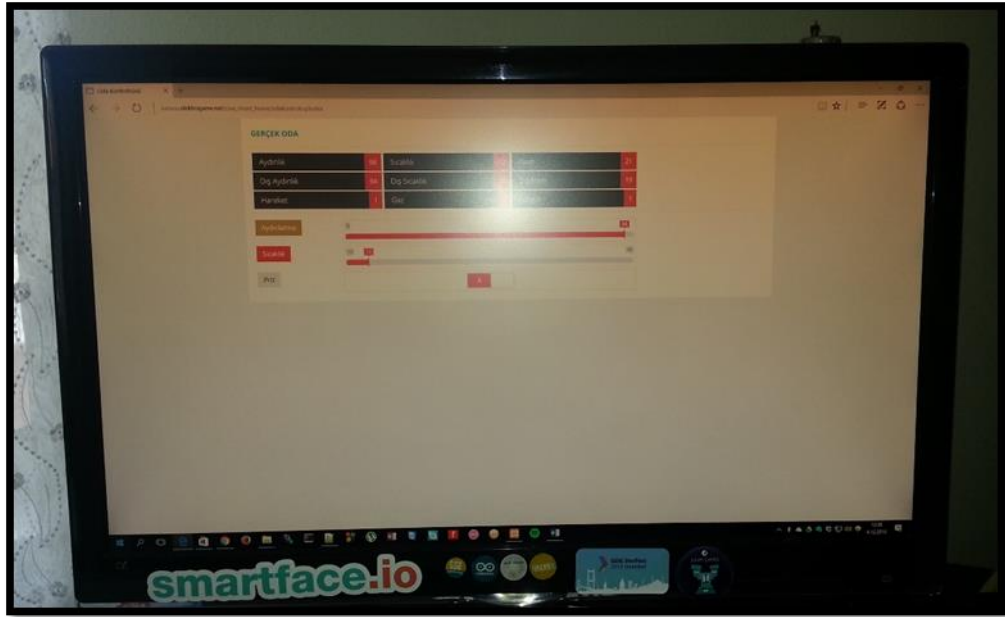
Bu bölümde bu çalışma sonucunda ortaya çıkarılan akıllı ev otomasyon sisteminin geliştirilmesi, çalıştırılması ve test edilmesi sırasında elde edilen veriler sunulacaktır.

### **4.1 Web Arayüzü**

Geliştirilen akıllı ev otomasyon sisteminin, hibrit simülasyonun ve yapay veri üreticinin yönetim paneli ile oda kontrolcünün LCD ekran ile ayar yapılmasını sağlayan arayüzünde web sayfaları kullanılmıştır. Web sayfaları kullanılmasının amacı tüm bu birimlerin farklı işletim sistemlerine ve yazılımlara sahip bilgisayar, dizüstü bilgisayar, tablet ve akıllı telefonlar ile sorun yaşamadan kontrol edilebilmesini sağlamaktır. Çalışma sonucunda geliştirilen web arayüzleri web desteği veren farklı işletim sistemlerine, farklı yazılımlara ve farklı çözünürlüklere sahip aygıtlarla test edilmiş ve tüm bu aygıtlarla sistemin sorunsuz olarak çalıştığı tespit edilmiştir (Şekil 4.1, Şekil 4.2).



Şekil 4.1: Android tablet, Android tarayıcı ile arayüzün test görüntüsü



Şekil 4.2: Windows 10 bilgisayar, Microsoft Edge ile test görüntüsü

Ayrıca işlemlerin sorunsuz ve hızlı gerçekleştirilmesi için http protokolü yavaş kalmaktadır. Bunun için web arayüzünden merkez sunucu ile Node.js kütüphanesi ile soket bağlantı kurulmuş ve JSON (JavaScript Object Notation) biçimi ile veriler olabildiğine ufaltılmış bir şekilde iletilmiştir. Böylece yapılan işlemlerin kullanıcı bağlantı hızına bağlı olarak saniyenin 1/10'undan daha düşük sürelerde gerçekleştirildiği tespit edilmiştir.



## 4.2 Yapay Veri Üretici

Yapay Veri Üretici ile üretilen veriler yapay zekâ teknikleri ile sınanmadan önce üretilen verilerin senaryo ile tutarlılığı test edilmiştir. Üretilen verilerin senaryo ile tutarlılığının test edilmesi için istatistiksel yöntemlerden faydalanılmıştır. Senaryoda belirlenen ısıtma ve aydınlatma tercihlerinin doğruluğu için üretilen verilerin frekans dağılımları incelenmiştir. Ayrıca eve giriş-çıkış, oda giriş-çıkış ile çeşitli aygıtları kullanım saatlerinin doğruluğu için üretilen kayıtlardaki zamanların frekans dağılımları incelenmiştir.

Yapay veri üretici yazılım, senaryosu girilen beş kişilik ailenin bir yıllık verisini yaklaşık olarak bir günde üretebilmiştir. Yazılım, bir gün içerisinde toplam 49302 kayıt oluşturularak bu verileri veritabanına kaydedebilmiştir. Kayıtlarda yapılan incelemede sisteme önceden girilen senaryoların dışında herhangi bir kaydın oluşturulmadığı belirlenmiştir. Ayrıca senaryoda olan her işlemin kaydı yazılım tarafından başarıyla üretilmiştir.

Yazılımın ürettiği verilerde en önemli noktalardan biri de işlemlerin kayıtlarının oda yetki seviyelerine göre üretilmesidir. Örneğin bir odada iki kişi yer almakta ve bu kişilerin yetki seviyeleri birinci kişinin 5 ve ikinci kişinin 10 olduğu varsayılırsa, bu durumda birinci kişi o odada herhangi bir işlem yapamamalıdır. Son olarak kayıtlar bu durum göz önüne alınarak odalara göre belirlenen yetki seviyeleri açısından incelenmiş ve bu konuda da herhangi bir hata olmadığı tespit edilmiştir.

Üretilen kayıtlardan ilk olarak kişinin tercih ettiği aydınlık düzeyi tercihlerinin frekans dağılımları incelenmiştir. Birinci kişinin (Baba) senaryoya göre tercih ettiği aydınlık düzeyi 70-90 arası belirlenmiştir. Buna göre frekans dağılımının 70-90 arası düzenli bir şekilde dağılması gerekmektedir. Yapılan analiz sonucunda sonuçlar senaryo ile tutarlı çıkmıştır (Tablo 4.1).

**Tablo 4.1:** Aydınlık değerleri dağılımı.

Değer	Frekans
70	65
71	63
72	76
73	79
74	67

**Tablo 4.1:** Aydınlık değerleri dağılımı (devam).

70	65
71	63
72	76
73	79
74	67
75	89
76	72
77	66
78	69
79	76
80	68
81	68
82	56
83	72
84	59
85	54
86	75
87	67
88	53
89	57
90	67
En Yüksek	90
En Düşük	70
Ortalama	79,65
Standart Sapma	5,99

İkinci olarak kişinin tercih ettiği sıcaklık düzeyi frekans dağılımı incelenmiştir. Senaryoda ısıtma-soğutma işlemi için kişinin, hangi aralıklarda ısıtma ve soğutma sistemini açacağı ile ısıtma-soğutma işlemi yaptığında kaç dereceyi tercih ettiği belirlenmiştir. Buna göre kullanıcının sıcaklık tercihleri, gündüz sıcak için 25 °C ve soğuk için 18 °C'dir. Gece ise sıcak için 20 °C, soğuk için 18 °C'dir. Bu tercihleri belirterek kişinin ısıtma-soğutma sistemini başlatma sıcaklık aralıkları ise gündüz 15-25 °C ve gece 13-22 °C'dir. Örneğin kullanıcının gündüz ısıtma işlemi başlatması için oda sıcaklığının 15 °C'nin altında olması gerekmektedir. Ayrıca veriler üretilirken kişinin sürekli olarak tam aynı sıcaklık-soğukluk derecelerini belirlemeyeceği varsayılarak tercih edilen sıcaklık derecelerine  $\pm 1$  derece rastgele bir sapma uygulanmıştır. Isıtma-soğutma işlemi başlatıldığında belirlenen derecelerin Tablo 4.2'de görüldüğü gibi senaryoya uygun olduğu belirlenmiştir.

**Tablo 4.2:** Sıcaklık değerleri dağılımı.

Değer	Frekans
17	78
18	54
19	254
20	167
21	166
24	48
25	39
26	34
En Yüksek	26
En Düşük	17
Ortalama	20,19
Standart Sapma	2,26

Son olarak üretilen verilerden ev ve oda giriş-çıkış saatlerinin dağılımlarının senaryoya uygunluğu araştırılmıştır. Örnek olarak kullanıcının pazartesi günleri eve giriş-çıkış saatleri için üretilen kayıtlar incelenmiştir. Senaryoya göre kullanıcının pazartesi günleri saat 7.30'da evden çıkmalı 17.31'de ise eve dönmelidir. Yazılım ile kayıt üretilirken tüm saat ile ilgili işlemlerde gerçek hayatla benzer olması için  $\pm 10$  dakika rastgele sapma uygulanmıştır. Üretilen kayıtların senaryo ile uygunluğunun sınanması için giriş-çıkış saatlerinin frekans dağılımları incelenmiştir. Tablo 4.2'de gösterilen sonuçlara göre üretilen kayıtlardaki değerlerin  $\pm 10$  dakika sapma aralığında olduğu ve üretilen kayıtların senaryoya uygun olduğu belirlenmiştir.

**Tablo 4.3:** Ev giriş çıkış zaman dağılımı.

Değer	Frekans
7.20	4
7.21	3
7.22	4
7.23	1
7.24	2
7.25	2
7.26	1
7.27	3
7.28	4
7.30	4
7.31	2
7.32	4
7.33	3
7.34	1
7.35	2

**Tablo 4.3:** Ev giriş çıkış zaman dağılımı (devam).

7.36	2
7.37	4
7.38	1
7.39	2
7.40	4
17.21	3
17.22	1
17.23	5
17.24	2
17.26	3
17.27	2
17.28	3
17.29	1
17.30	3
17.31	2
17.32	4
17.33	1
17.34	1
17.35	2
17.36	1
17.37	2
17.38	4
17.39	4
17.40	6
17.41	3

### 4.3 Yapay Zekâ

Akıllı ev sistemi için geliştirilmiş olan yapay zekâ çok farklı türdeki işlemleri gerçekleştirebilmektedir. Bunlardan ilki etki tepki işlemleridir. Akıllı ev sistemi bir hareket algıladığında kullanıcıların oda giriş çıkış işlemleri için kullandığı akıllı kartları okuyan okuyucunun etrafını otomatik olarak aydınlatmaktadır. Bu işlemin yapılan testler sonucunda sorunsuz bir şekilde sistem çalıştığı sürece çalıştığı tespit edilmiştir.

Bir sonraki yapay zekâ algoritması yetkilendirme işlemleri için geliştirilmiştir. Bu algoritma aynı anda aynı oda için işlem yapana farklı bireylerden hangisinin yaptığı işlemin uygulanacağını belirlenmesini sağlamaktadır. Algoritmanın yapılan testler sonucunda kişilerin bulunduğu odalar için doğru şekilde çalıştığı tespit edilmiştir. Ancak hiç kimsenin bulunmadığı bir oda için 2 farklı

kullanıcı işlem yaptığında en son hangi kullanıcı işlem yapmış ise o kullanıcının yaptığı işlem uygulanmaktadır. Bu sorunun çözümü olarak bir kullanıcının bir odada yetki seviyesi 0 olarak ayarlandığında o oda ilgili hiçbir işlem yapılamamasına karar verilmiş ve algoritma bu işlemi gerçekleştirecek şekilde revize edilmiştir.

Kullanıcıların hangi anda hangi konumda bulunacağını tespit etmek için Kullanıcı-Zaman-Konum algoritması geliştirilmiştir. Bu algoritmanın test edilmesi için hibrit simülasyon kullanılmıştır. Sanal aileyi oluşturan bireylerin ne zaman nerede bulunduğu senaryo ile belirlenmiştir. Ayrıca 1 yıllık sanal veriler de üretilerek kullanıcının geçmiş akıllı ev kayıtları da oluşturulmuştur. Bu algoritma test edilirken yapay zekâ algoritmasının çıkarımları ile senaryo karşılaştırılmıştır. Örneğin 1. kullanıcının(baba) Pazartesi günü saat 12.00'da senaryoya göre işyerinde bulunmaktadır. Hibrit simülasyon ekranında algoritma ile yapılan testin sonucu Şekil 4.3'te gösterilmiştir. Algoritma senaryo ile doğru sonucu bulmuştur.

Kişi-Zaman-Konum (Kim? Ne Zaman? Nerede?)

Kişi BA - Ahmet

Tarih - Saat 2016/12/05 12:00

Oda	pazartesi	Hafta İçi	Bütün Günler	Son 10 Haftanın O Günü
Salon	0	0	0	
Yatak Odası	0	0	0	
Mutfak	0	0	0	
Kız Çocuk Odası	0	0	0	
Erkek Çocuk Odası	0	0	0	
Ev	0	0	0.26086956521739	0.2
Gerçek Oda	0	0	0	
Ev Dışı	1	1	0.73913043478261	0.75

İptal Oranları Getir

Şekil 4.3: Kullanıcı konum tespit ekranı

Kullanımı konum tespiti algoritması farklı durumlar için çeşitli şekillerde test edilmiştir (Şekil 4.4). Algoritmanın senaryoya kullanıcıların tam olarak konum değiştirdikleri zamanlarda yanılabilirdiği belirlenmiştir ancak bu yanılma da hem çok normaldir hem de 5-10 dakika aralığında sistemin çalışma şekli sayesinde kendiliğinden çözülmektedir.

Kişi-Zaman-Konum (Kim? Ne Zaman? Nerede?)				
Kişi	BA - Ahmet			
Tarih - Saat	2016/12/05 17:30			
Oda	pazartesi	Hafta İçi	Bütün Günler	Son 10 Haftanın O Günü
Salon	0	0	0	
Yatak Odası	0	0	0	0.14285714285714
Mutfak	0	0	0	
Kız Çocuk Odası	0	0	0	
Erkek Çocuk Odası	0	0	0	
Ev	0.5	0.5	0.60869565217391	0.42857142857143
Gerçek Oda	0	0	0	
Ev Dışı	0.5	0.5	0.39130434782609	0.42857142857143

**Şekil 4.4:** Konum hareketinin olduğu dakikadaki konum ihtimalleri

Aygıtların zamana göre durumlarının belirlendiği algoritma kişilerin konum tespit algoritması ile çok benzer şekilde çalışmaktadır ve bunun için benzer şekillerde test edilmiştir. Başarım oranları çok yüksek olmasına karşın senaryoya göre örneğin prizinin tam kapatılacağı vakitlerde sistem doğru karar vermeyebilmektedir. Örnek olarak bir prizinin açık olma ihtimali Şekil 4.5'te gösterilmektedir.

Aygıt Kullanım Zamanları (Aygıtların Günlük Kullanım Zaman Aralıkları) ×

Kişi Salon Priz 1 - TV ▼

Tarih - Saat 2017/01/04 10:00

Durum	Genel	Haftanın Günü
Açık Olma İhtimali	0.95	1.00

Şekil 4.5: Aygıt durum tespiti

Kullanıcıların odalarda buldukları zamanlarda termostat ve lambalar için belirledikleri değerlerin tespiti için kullanıcı tercihleri algoritması geliştirilmiştir. Bu algoritmanın test işlemi yine senaryo ile karşılaştırılarak gerçekleştirilmiştir. Örneğin senaryoda birinci kullanıcının(baba) sıcaklık tercihleri: ısıtma açtığı en yüksek oda sıcaklığı gündüz 15°C, gece ise 13°C'dir. Sıcaklık açtığında belirlediği ortalama değerler ise gündüz 25°C, gece ise 22°C'dir. Aynı kişinin yine senaryoya göre lamba parlaklığı tercihleri ise 70-90 aralığındadır. Algoritmanın yapılan testler sonucunda senaryoya çok yakın değerler bulunduğu ve doğru çalıştığı tespit edilmiştir. Algoritmanın bu kullanıcı için bulunduğu sonuçlar Şekil 4.6'da gösterilmektedir.



Kişi Tercihleri (Kişinin Sıcaklık ve Aydınlık Düzeyi Tercihleri)				
Kişi		BA - Ahmet		
	Isıtma		Aydınlatma	
Gündüz	Açtığı En Yüksek Sıcaklık	27	Tercih Ettiği En Yüksek Parlaklık	90
	Açtığı En Düşük Sıcaklık	23	Tercih Ettiği En Düşük Parlaklık	70
	Açtığı Sıcaklıkların Ortalaması	24.9483	Tercih Ettiği Ortalama Parlaklık	80.2317
	Isıtma Açtığı En Yüksek Sıcaklık	15		
Gece	Açtığı En Yüksek Sıcaklık	24		
	Açtığı En Düşük Sıcaklık	20		
	Açtığı Sıcaklıkların Ortalaması	22.0256		
	Isıtma Açtığı En Yüksek Sıcaklık	13		

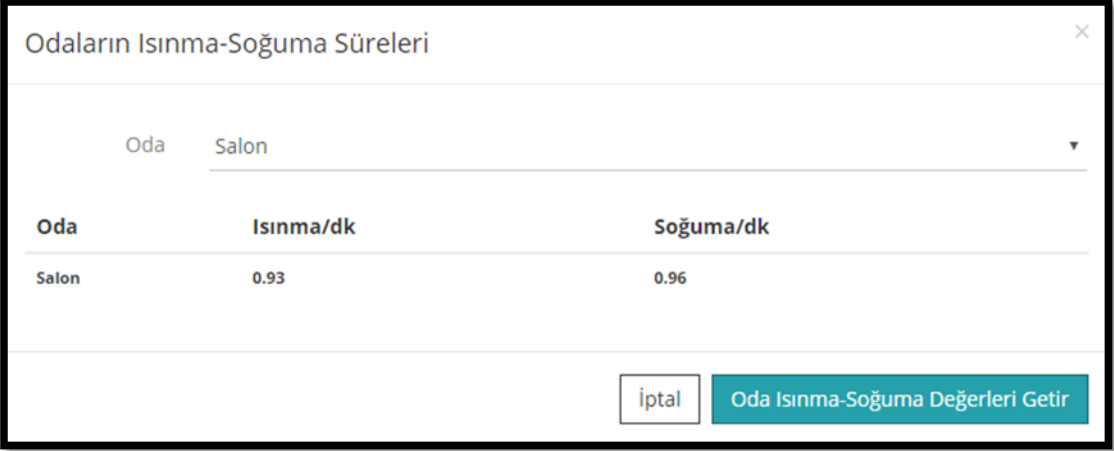
**Şekil 4.6:** Kişi tercihlerinin tespiti

Kullanıcıların bir odaya gireceklerinin tespit edilmesinin ardından kullanıcı o odaya gelmeden ne kadar önce termostatın açılması gerektiğini belirlemek için odaların dakikalık ısınma ve soğuma değerlerinin tespiti için geliştirilen algoritma hibrit simülasyonda test edilmiştir. Hibrit simülasyonda evin dış sıcaklığı Adana ilinin 2014 yılı sıcaklık verilerine göre belirlenmektedir. Evin odalarının sıcaklık değerleri de eğer termostat kapalı ise evin dış sıcaklığına açıksa termostat için belirlenen değere eşitlenmektedir. Ancak değerler eşitlenirken oda sıcaklığı her oda için belirlenen farklı değerlere göre 0,2°C sapma ile artırılmakta ya da azaltılmaktadır. Bu noktada her oda için farklı değerler verilmesinin nedeni geliştirilen algoritmanın farklı şartlar altında sınanmasını sağlamaktır.

Algoritma bütün odalar için ayrı ayrı test edilmiştir. Testler sonucunda tahminlerinin ısınma, soğuma değerlerine çok yakın olduğu tespit edilmiştir. Tüm sonuçlar Tablo 4.4'te hibrit simülasyonda salon için bulunan değerler ise Şekil 4.7'de gösterilmektedir.

**Tablo 4.4:** Oda ısıtma soğuma tahminleri.

Oda	Senaryo Isınma/dk °C	Senaryo Soğuma/dk °C	Algoritma Tahmin Isınma/dk °C	Algoritma Tahmin Soğuma/dk °C
Salon	1,1	0,9	0,93	0,96
Mutfak	2	0,5	0,48	0,51
Yatak Odası	0,7	1,8	0,75	1,7
Erkek Çocuk Odası	1	1	0,98	0,92
Kız Çocuk Odası	0,7	0,5	0,78	0,61



Oda	Isınma/dk	Soğuma/dk
Salon	0.93	0.96

**Şekil 4.7:** Salon oda ısıtma soğuma tahmin verileri

#### 4.4 Sistemin Çalışması ve Donanımlar

Sistemin merkez sunucusu işletim sistemi ve yazılımların kurulmasının ardından çalışmalar süresince sürekli olarak çalışır halde tutulmuştur. Yaklaşık olarak 3 yıl boyunca merkez sunucu sorunsuz olarak çalışmıştır. Bu süreç boyunca merkez sunucuda yer alan ve saniyede bir kere çalışarak yaklaşan olayları işleyen yazılım bazen 1 saniyelik sürede kaymalara bazen de kitlenmelere sebep olmuştur. Çözüm için saniyelik olarak olayları işleyen yazılımın 5 saniyede bir çalıştırılması sağlanmıştır. Bu değişikliğin ardından herhangi bir sorun yaşanmamıştır.

Dış sensör bileşeni kendi donanım ve yazılımının geliştirilmesinin ardından 1 ay boyunca hiç kapatılmadan dış ortamda çalıştırılmıştır. Bu süreç boyunca ortam

sıcaklığı 25°C ile -1°C aralığında deęişmiş ve dış sensör bileşeni bu süreçte sorunsuz çalışmıştır.

Diđer geliştirilen tümleşik sensör bileşeni ve ona baęlı olarak çalışan aydınlatma ve priz bileşeni donanımsal ve yazılımsal olarak geliştirilmelerinin ardından sorunsuz olarak 1 hafta boyunca çalıştırılmıştır.

Oda kontrolcüsü ve ona baęlı olarak çalışan LCD ekran yazılımlarının geliştirilmesinin ardından uzun süre test edilmiştir. Bu testler sırasında bluetooth bağlantısının dış sensör bileşenin ve tümleşik sensöre sırayla bağlanmasında donanım kaynaklı olarak kitlenmeler meydana gelmiştir. Bu sorunun çözümü için tümleşik sensör ile oda kontrolcü arasındaki bağlantı USB ile sağlanmıştır.

## 5. SONUÇ VE ÖNERİLER

Bu çalışmada kullanıcı davranışlarını takip ederek öğrenebilen, modüler, düşük enerji tüketimli ve web arayüzü ile kontrol edilebilen özgün bir akıllı ev otomasyon sistemi geliştirilmiştir. Geliştirilen sistem merkezde yer alan bir bulut sunucu ve ona bağlı olarak çalışan oda kontrolcülerini ile ısıtma sistemini kontrol eden termostat bileşeni, lambaları kontrol eden aydınlatma bileşeni, prizleri kontrol eden priz bileşeni ve sensör bileşenlerinden meydana gelmektedir. Ayrıca geliştirilen sistem mevcut akıllı evler için geliştirilmiş olan farklı firmalara ait donanımları (Örn: Google Nest Termostat) da kontrol edebilmektedir.

Sistemin genel tasarımı çoklu ajan sistemleri temelinde hazırlanmıştır. Sistem merkez sunucuya bağlı ancak özerk olarak da çalışabilen oda kontrolcülerini ile evdeki işlemleri gerçekleştirmektedir.

Çalışmada en güncel yazılımlar ve yazılım geliştirme ortamları ile son çıkan donanımlar ve en son teknolojiler bir arada kullanılmıştır. Sistemin kontrol edildiği arayüz html, css ve javascript programları dilleri ile web sitelerini çalıştırabilen farklı işletim sistemlerine, farklı yazılımlara ve farklı çözünürlüklere sahip her türlü aygıtta sorunsuz olarak çalışabilecek modern bir görünümde web tabanlı olarak hazırlanmıştır. Birimler arası iletişim kablolu ve kablosuz (wi-fi, bluetooth) olarak sağlanabilmektedir. Farklı iletişim yöntemlerinin desteklenmesi sistemin esnekliğini artırmaktadır.

Birimler arası iletişimin performansını artırmak amacıyla birimler arasında soket bağlantı kullanılmıştır. Ayrıca veriler iletilirken JSON (JavaScript Object Notation) biçimi kullanılmıştır. Böylece verilerin birimler arasında hem hiç bekleme olmadan çok hızlı iletilmesi sağlanmış hem de ihtiyaç duyulan bant genişliği düşürülmüştür.

Geliştirilen akıllı ev otomasyon sisteminin öğrenebilen özellikler için yapay zekâ çalışmaları yürütülmüştür. Bu yapay zekâ çalışmalarında kullanılmak üzere akıllı ev kullanım verileri üreten bir sanal veri üretici yazılım geliştirilmiştir. Ayrıca bu yazılımın verilerini üreteceği 5 kişilik sanal bir aile ve bu ailenin haftalık yaşam

senaryoları oluşturulmuştur. Son olarak bu yazılım ile sanal ailenin sanki 1 yıl boyunca geliştirilen akıllı evde yaşamışlar gibi kullanım kayıtları oluşturulmuştur. Bu veriler yapay zekâ algoritmalarının geliştirilmesinde ve iyileştirilmesinde kullanılmıştır. Bunların yanında yapay veri üretici yazılım web tabanlı olarak farklı kullanıcılarında kolayca kullanabileceği şekilde tasarlanmıştır. Bu alanda çalışmak isteyen farklı araştırmacılar için de sistem sanal veri üretebilmektedir. Yapılan araştırmalarda bu çalışma dışında akıllı evler için sanal veriler üreten başka bir yazılıma rastlanmamıştır.

Yapay verilerin yanında yapay zekâ algoritmalarının gerçek zamanlı olarak farklı şartlarda kolayca test edilebilmesi için hibrit bir akıllı ev simülasyonu geliştirilmiştir. Bu simülasyon senaryosu oluşturulan 5 kişilik sanal ailenin yaşadığı 5 odalı sanal bir evde yaşamını kişilerin senaryolarına göre işletmekte ve geliştirilen akıllı ev sisteminin kurulu olduğu gerçek bir oda ve bu sistemi kullanan gerçek bir kişiyi bu sanal aile ile aynı evde yaşıyormuş gibi birbirine entegre bir şekilde sunabilmektedir. Bu yazılım ile bilgisayar ortamında çok farklı şartlar oluşturularak yapay zekâ algoritmalarının bu şartlarda testleri yapılmış ve iteratif olarak başarımları artırılmıştır. Yapılan literatür taramasında bu alanda yapılan çalışmalarda çeşitli akıllı ev simülasyonları (eCclipse SmartHome, 2016; Comfortclick, 2016; Home IO Smart Home Simulation, 2016) bulunmasına karşın hibrit olarak gerçek ve sanal bireylerin bir arada entegre olarak sunulduğu herhangi bir çalışmaya rastlanmamıştır.

Bu çalışma kapsamında akıllı evlere yönelik çeşitli yapay zekâ algoritmaları geliştirilmiştir. Bunlardan ilki Markov modeli ile karınca koloni algoritmalarından esinlenilerek geliştirilen kişilerin gelecek bir zamanda hangi konumda bulunacağını tespit eden algoritmadır. Bu algoritma kişilerin geçmişte kişiler ile ilgili konum kayıtlarını işleyerek gelecek bir zamanda istenilen kişinin hangi konumda olabileceğini tahmin etmektedir. Algoritma rutin olarak hareket eden kişilerde yüksek başarımlar elde ederken, hareket şekillerini zamanla değiştiren kişilerin yeni hareketlerine de kısa sürede uyum sağlamaktadır. Bu algoritmanın yanında kişilerin ev içerisinde akıllı ev sistemini kullanırken yapmış oldukları tercihleri sistem öğrenerek zaman içerisinde bu tercihlere uygun olarak kişilerin yaptıkları işlemleri otonom olarak yapabilmektedir. Örneğin termostatı kişinin belirlediği sıcaklığa, lambayı ise kişinin belirlediği aydınlık seviyesine sistem zaman içerisinde otonom

olarak ayarlayabilmektedir. Bu algoritma kullanıcının geçmiş tercihlerini istatistiksel yöntemlerle çıkarsayarak gelecek işlemlerde kullanmaktadır. Geliştirilen yapay zekâ algoritmaların bir diğeri de odaların ısınma ve soğuma sürelerini takip ederek istatistiksel yöntemlerle bir odanın dakikada ortalama kaç santigrat derece ısınıp soğuduğunu belirlemektedir. Bu 3 algoritma tasarlanan bir karar ağacı ile geliştirilen akıllı ev otomasyon sistemine entegre edilmiştir. Sonuç olarak öğrenebilen algoritmalara sahip otonom hareket edebilen bir akıllı ev otomasyonu ortaya çıkarılmıştır.

Geliştirilen sistemin yapay zekâsı diğeri sistemlerde yer alan yöntemlere göre çeşitli üstünlükler sağlamıştır. Genel olarak akademik çalışmalarda akıllı evin geneline kapsayan çalışmalar yerine tek bir alana yönelmiş yapay zekâ çalışmaları yürütülmüştür. Bu sistemde ise akıllı evin geneline yönelik birden fazla işlemi gerçekleştiren bir yapay zekâ yapısı ortaya çıkarılmıştır. Kullanıcı konum tespitine göre evin sıcaklığını ayarlayan sistemler incelendiğinde genellikle kullanıcıları cep telefonlarından aldıkları GPS sinyalleri ile takip edilmekte olduğu, tüm evi bir kişi kullanıyormuş gibi varsayıldığı ve ısıtma için evin tamamı bir bütün olarak düşünüldüğü görülmüştür (Cosa Evinizin Akıllı Asistanı, 2016) (Google Nest Thermostat, 2016). Geliştirilen sistemde ise kullanıcıların her biri ayrı ayrı takip edilmekte ve sistem kişilere göre özelleştirilmiş şekilde ayarlanmaktadır. Bu çalışmanın diğeri akıllı ev sistemlerinden en büyük farkı da bu noktadır. Sistemi kullanan her kullanıcıya özel çözümler üretebilmesidir. Ayrıca kişinin sadece eve gelişi değil ev içerisinde hangi odaya gireceği de tahmin edilmektedir. Böylece tüm ev değil sadece gerekli oda ısıtılarak enerji tasarrufu sağlanabilmektedir. Bunun yanında enerji tasarrufu termostatın otonom olarak kapatılması işleminde de geçerlidir.

Akıllı ev sistemi bu çalışmada web arayüzü ile kontrol edilmektedir. Farklı çalışmalarda telefon, bilgisayar, uzaktan kumanda ya da kontrol panelleri ile kontrol edilmektedir. Son zamanların en gelişmiş ticari ürünü olarak gösterilebilecek Amazon Echo'da ise sistem ses ile kontrol edilebilmektedir (Amazon Echo Alexa, 2016). Gelecekte yapılacak çalışmalarda ise beyin dalgalarını algılayarak insanların düşüncelerini algılayabilen cihazlar ile kontrol edilebilen akıllı ev sistemleri geliştirilebilir.

Bu alıřma ile bir evin tm odaları ayrı ayrı kontrol edilebilmekte ve tm bu kontroller ile elde edilen veriler merkezi bir sunucuda toplanmaktadır. Birok evin toplanan verileri zaman ierisinde deęerlendirilerek evlere ynelik genel enerji tasarruf politikaları oluřturulabilir.

Son olarak dnyanın en byk teknoloji řirketlerinin (Google, Apple, Amazon, Samsung, Philips) byk yatırımlar yaparak, řirketler satın alarak (Google – Nest, Samsung –Smarthings) alıřmalar yrttę, rekabet ettięi bu alanda lkemizin geri kalmaması iin bu tr alıřmaların artarak devam etmesi gerekmektedir.

## 6. KAYNAKLAR

*Adafruit 5" and 7" 800x480 TFT HDMI Backpack.* (2016, 05 28). Adafruit: <https://learn.adafruit.com/adafruit-5-800x480-tft-hdmi-monitor-touchscreen-backpack/overview> adresinden alındı

Alam, M., Reaz, M., & Mohd Ali, M. (2012). A Review of Smart Homes - Past, Present, and Future. *IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews*, 1190-2001.

Albert, T. (2015, 06 23). *A Timeline of Houses In History.* A World History Encyclopedia: <http://www.localhistories.org/housestime.html> adresinden alındı

Alpaydın, E. (2013). *Yapay Öğrenme.* İstanbul: BÜTEK.

*Amazon Echo.* (2016, 12 06). Amazon Echo: <https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E> adresinden alındı

*Amazon Echo Alexa.* (2016, 12 05). Amazon: <https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E> adresinden alındı

*Ant Applying Technologies.* (2015, 02 15). Ant Applying Technologies: <http://www.ant-automation.net/> adresinden alındı

Arasu, A., Kaushik, R., & Li, J. (2011). DataSynth: Generating Synthetic Data using Declarative Constraints. *The Proceedings of the VLDB Endowment*, 1418-1421.

*Arduino MEGA 2560.* (2016, 05 28). Arduino: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560> adresinden alındı



Babakura, A., Sulaiman, M., Mustapha, N., & Perumal, T. (2014). HMM-Based Decision Model for Smart Home Environment. *International Journal of Smart Home*, 129-138.

*Bluetooth*. (2015, 07 12). Bluetooth: <http://www.bluetooth.com/> adresinden alındı

*Bluetooth Low Energy*. (2015, 07 12). Bluetooth: <http://www.bluetooth.com/Pages/low-energy-tech-info.aspx> adresinden alındı

*Bluetooth Shield*. (2016, 05 28). SeedStudio: [https://www.seedstudio.com/item\\_detail.html?p\\_id=866](https://www.seedstudio.com/item_detail.html?p_id=866) adresinden alındı

Bruno, N., & Chaudhuri, S. (2005). Flexible Data Generator. *Proceedings of the 31st VLDB Conference* (s. 1097-1107). Trondheim: Very Large Data Bases Endowment Inc.

Burkhard, S., & Bouée, C.-E. (2013). *Home Automation Solutions are Becoming Mature, With Convincing Products Already Being Marketed*. Hamburg: Roland Berger Strategy Consultants.

*CA Test Data Manager*. (2016, 12 10). CA Technologies: <https://www.ca.com/us/products/ca-test-data-manager.html> adresinden alındı

*Candela*. (2016, 05 28). Elektrikçe: <http://www.elektrikce.com/tekniksozluk/candela/> adresinden alındı

Chan, M., Campo, E., Esteve, D., & Fourniols, J.-Y. (2009). Smart homes — Current features and future perspectives. *Matirutas*, 90-97.

Chan, M., Est`eve, D., Escriba, C., & Campo, E. (2008). A review of smart homes—Present state and future challenges. *Computer Methods and Programs in Biomedicine*, 55-81.

Chan, M., Hariton, C., Ringard, P., & Campo, E. (1995). Smart House Automation System for the Elderly and the Disabled. *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century* (s. 1586-1589). IEEE.

Choperena, M. (2015, 07 12). *RFID-powered Sensors Can Play a Big Role in the Internet of Things*. RFID Journal: <http://www.rfidjournal.com/articles/view?11062> adresinden alındı

Cinsdikici, M. (2016, 01 19). *Yapay Sinir ağları*. <http://ube.ege.edu.tr/~cinsdiki/UBI521/Chapter-1/cinsdikici-neural-net-giris.pdf> adresinden alındı

*Comfortclick*. (2016, 12 05). Comfortclick: [https://www.comfortclick.com/Features/Details/ComfortClick.Tasks\\_PresenceSimulator](https://www.comfortclick.com/Features/Details/ComfortClick.Tasks_PresenceSimulator) adresinden alındı

*Cornel CIS Computer Science*. (2016, 12 06). Decision Tree: [http://www.cs.cornell.edu/courses/cs4700/2011fa/lectures/09\\_decision\\_trees.pdf](http://www.cs.cornell.edu/courses/cs4700/2011fa/lectures/09_decision_trees.pdf) adresinden alındı

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 273-297.

*Cosa Evinizin Akıllı Asistanı*. (2016, 12 05). Cosa: <https://cosa.com.tr> adresinden alındı

Çayıroğlu, İ. (2016, 01 19). *Yapay Sinir Ağları*. İleri Algoritma Analizi: <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf> adresinden alındı

Çomak, E. (2008). *Destek Vektör Makinalarının Etkin Eğitimi İçin Yeni Yaklaşımlar*. Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Das, S., Cook, D., Bhattacharya, A., Heierman III, E., & Lin, T. (2003). The Role of Prediction Algorithms in Mavhome Smart Home Architecture. *Wireless Communications*, 77-84.

Dash, S., Mohanty, G., & Mohanty, A. (2012). Intelligent Air Conditioning System using Fuzzy Logic. *International Journal of Scientific & Engineering Research*, 1-6.

De Luca, G., Lillo, P., Mainetti, L., Mighali, V., Patrono, L., & Sergi, I. (2013). The use of NFC and Android technologies to enable a KNX-based Smart

Home. *Software, Telecommunications and Computer Networks (SoftCOM)* (s. 1-7). Primosten: IEEE.

Dementyev, A., Hodges, S., Taylor, S., & Smith, J. (tarih yok). *Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario*. Seattle: Microsoft Research.

Deokar, P., & Nagmode, M. (2014). A Survey on Home Automation using Cloud Network and Mobile Devices. *International Journal of Latest Trends in Engineering and Technology*, 129-132.

*DHT11 - Humidity and Temperature Sensor* . (2016, 05 29). Robocraft: <http://robocraft.ru/files/datasheet/DHT11.pdf> adresinden alındı

*eclipse SmartHome*. (2016, 12 05). eclipse smarthome: <http://www.eclipse.org/smarthome/> adresinden alındı

Elmas, Ç. (2010). *Yapay Zeka Uygulamaları*. Ankara: Seçkin Yayıncılık.

ElShafee, A., & Hamed, K. (2012). Design and Implementation of a WiFi Based Home Automation System. *World Academy of Science, Engineering and Technology*, 1852-1858.

Fahad, L., Khan, A., & Rajarajan, M. (2015). Activity Recognition in Smart Homes With Self Verification of Assignments. *Neurocomputing*, 1286-1298.

Fang, H., He, L., Si, H., Liu, P., & Xie, X. (2014). Human activity recognition based on feature selection in smart home using back propagation algorithm. *ISATransactions*, 1629-1638.

Fernandez, J., Losada, D., & Domonte, E. (2014). An Integrated and Low Cost Home Automation System with Flexible Task Scheduling. *XV Workshop of Physical Agents*. Leon.

*Flame Sensor Technical Data Sheet*. (2016, 02 30). Robotistan: [http://maker.robotistan.com/download/usr\\_attachment-flamesensor.pdf](http://maker.robotistan.com/download/usr_attachment-flamesensor.pdf) adresinden alındı

Fleury, A., Vacher, M., & Noury, N. (2010). SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results. *IEEE Transactions on Information Tehnology in Biomedicine*, 274-283.

Frenzel, L. (2015, 07 12). *What's The Difference Between Bluetooth Low Energy And ANT?* Electronic Design: <http://electronicdesign.com/mobile/what-s-difference-between-bluetooth-low-energy-and-ant> adresinden alındı

Gao, Z., & Zhao, L. (2010). Research and Design of Smart Home System Based on Zigbee Technology. *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on* (s. 290 - 293). Sanya: IEEE.

*Generatedata.* (2016, 12 10). [generatedata.com:](http://www.generatedata.com/) <http://www.generatedata.com/> adresinden alındı

*Google Nest Thermostat.* (2016, 12 05). Google Nest: <https://nest.com/> adresinden alındı

Harper, R. (2003). *Inside the Smart Home*. London: Springer.

*HC-SR501 PIR Motion Detector.* (2016, 05 30). Marlin P. Jones & Assoc. Inc.: <https://www.mpja.com/download/31227sc.pdf> adresinden alındı

Heierman, E., & Cook, D. (2003). Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns. *Proceedings of the Third IEEE International Conference on Data Mining* (s. 537-540). Washington, DC: IEEE Computer Society.

Hill, M., & Parker, S. (2002). *McGraw-Hill Dictionary of Scientific and Technical Terms*. McGraw-Hill Education.

Hoag, J., & Thompson, C. (2007). A Parallel General-Purpose Synthetic Data Generator. *ACM Sigmod Record*, 19-24.

*Home IO Smart Home Simulation.* (2016, 12 05). Home IO: <https://realgames.co/home-io/> adresinden alındı

*Homekit*. (2016, 12 06). Apple Homekit: <http://www.apple.com/tr/ios/home/> adresinden alındı

*IEEE 802.3™-2012 – IEEE Standard for Ethernet*. (2015, 07 11). IEEE Standards Association: <http://standards.ieee.org/about/get/802/802.3.html> adresinden alındı

*Internet Live Stats : Internet Users*. (2015, 07 11). Internet Live Stats: <http://www.internetlivestats.com/internet-users/> adresinden alındı

Jadhav, P., Chaudhari, A., & Vavale, S. (2014). Home Automation using ZigBee Protocol. *International Journal of Computer Science and Information Technologies*, 1778-1780.

Jarmin, R., Louis, T., & Miranda, J. (2014). Expanding the Role of Synthetic Data at the U.S. Census Bureau. *U.S. Census Bureau Center for Economic Studies*, 3-16.

*JSON'a Giriş*. (2016, 11 01). JSON: <http://www.json.org/json-tr.html> adresinden alındı

Kiyafar, M., Lotfibonab, M., & Lotfibonab, P. (2012). Automated Room-Light Controller Using Fuzzy Logic. *Journal of Artificial Intelligence in Electrical Engineering*, 78-81.

*KNX Association*. (2015, 07 11). KNX: <http://www.knx.org/knx-en/index.php> adresinden alındı

Kobersi, I., Finaev, V., Almasani, S., & Abdo, K. (2013). Control of the Heating System with Fuzzy Logic. *World Applied Sciences Journal*, 1441-1447.

Korel, B. (1990). Automated Software Test Data Generation. *IEEE Transactions on Software Engineering*, 870-879.

Kriesel, D. (2016, 12 10). *A Brief Introduction to Neural Networks*. David Kriesel: [http://www.dkriesel.com/en/science/neural\\_networks](http://www.dkriesel.com/en/science/neural_networks) adresinden alındı

Lee, W. S., & Hong, S. H. (2009). Implementation of a KNX-ZigBee Gateway for Home Automation. *The 13th IEEE International Symposium on Consumer Electronics* (s. 545-549). Kyoto: IEEE.

Lesser, V., Atighetchi, M., Benyo, B., Horling, B., Raja, A., Vincent, R., . . . Zhang, S. (1999). Amherst: Computer Science Department University of Massachusetts at Amherst.

Lian, K.-Y., Hsiao, S.-J., & Sung, W.-T. (2013). Intelligent Multi-Sensor Control System Based on Innovative Technology Integration via ZigBee and Wi-Fi Networks. *Journal of Network and Computer Applications*, 756-767.

Lutolf, R. (1992). *Smart Home Concept And The Integration Of Energy Meters Into A Home Based System*. Switzerland: Landis & Gyr Energy Management Corp.

Lüks (birim). (2016, 05 28). Wikipedia: [https://tr.wikipedia.org/wiki/L%C3%BCks\\_\(birim\)](https://tr.wikipedia.org/wiki/L%C3%BCks_(birim)) adresinden alındı

Manufacturers. (2015, 07 11). KNX: <http://www.knx.org/knx-en/manufacturers/list/index.php> adresinden alındı

Medjahed, H., Istrate, D., Boudy, J., & Dorizzi, B. (2009). Human Activities of Daily Living Recognition Using Fuzzy Logic. *IEEE International Conference on Fuzzy Systems* (s. 2001-2006). Jeju Island: IEEE.

Meet Hue. (2016, 12 06). Philips Hue: [http://www2.meethue.com/tr-tr/?&origin=|mckv|sjhaK4a0V\\_dc&pccid=117667884307|plid|](http://www2.meethue.com/tr-tr/?&origin=|mckv|sjhaK4a0V_dc&pccid=117667884307|plid|) adresinden alındı

Mennicken, S., Vermeulen, J., & Huang, E. (2014). From Today's Augmented Houses to Tomorrow's Smart Homes: New Directions for Home Automation Research. *UbiComp 2014: the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (s. 105-115). Seattle.

Mol, P. R., Chinthamani, B., Kamini, K., & Sugashini, K. (2012). Intelligent Home Heating System. *International Journal of Research in Engineering and Technology*, 102-107.

Mozer, M. (1998). *The Neural Network House: An Environment that Adapts to its Inhabitants*. Colorado: AAAI.

*MQ-9 Semiconductor Sensor for CO/Combustible Gas* . (2016, 05 30). Henan Hanwei Electronics: [https://www.pololu.com/file/download/MQ9.pdf?file\\_id=0J314](https://www.pololu.com/file/download/MQ9.pdf?file_id=0J314) adresinden alındı

*MySQL Nedir?* (2015, 12 20). MySQL: <http://www.mysql.com.tr/tr/KonuDetay.php?BKey=34&AKey=1> adresinden alındı

Nabiyev, V. V. (2012). *Yapay Zeka*. Ankara: Seçkin Yayıncılık.

*Nest*. (2016, 12 06). Google Nest: <https://nest.com/> adresinden alındı

Özden, Y. (2003). *Öğrenme ve Öğretme*. Ankara: Pegem A Yayıncılık.

Pande, S., & Sen, P. (2014). Review On: Home Automation System For Disabled People Using BCI. *IOSR Journal of Computer Science*, 76-80.

Piyare, R., & Tazil, M. (2011). Bluetooth Based Home Automation System Using Cell Phone. *International Symposium on Consumer Electronics* (s. 291-195). Singapore: IEEE.

Pothuganti, K., & Chitneni, A. (2014). A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. *Advance in Electronic and Electric Engineering*, 655-662.

Prindle, D. (2016, 09 01). *What The Heck Are Zigbee, Z-Wave, And Insteon? Home Automation Standards Explained*. Digital Trends: <http://www.digitaltrends.com/home/zigbee-vs-zwave-vs-insteon-home-automation-protocols-explained/> adresinden alındı

Prospero, M. (2016, 11 30). *Best Smart Thermostats 2016*. Tom's Guide: <http://www.tomsguide.com/us/best-smart-thermostats,review-2751.html> adresinden alındı

*RAID Modlar*. (2015, 12 20). Seagate: <http://www.seagate.com/tr/tr/manuals/network-storage/business-storage-nas-os/raid-modes/> adresinden alındı

*Raspberry Pi 3 Model B*. (2016, 05 28). Raspberry Pi: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> adresinden alındı

Reaz, M. (2013). Artificial Intelligence Techniques For Advanced Smart Home Implementation. *Acta Technica Corviniensis - Bulletin of Engineering*, 51-57.

*Riverside Concept Houses*. (2014, 02 15). Riverside Concept Houses: <http://www.riversideconcept.com/akilli-ev-urun.php?urun=akilli-ev-nedir-nasildir-> adresinden alındı

Samosuyev, V. (2010, 05 24). Bluetooth Low Energy compared to ZigBee and Bluetooth Classic. Mikkeli University of Applied Sciences.

Schwenker, B., Bouée, C.-E., & Oltmanns, T. (2013). Home automation – The next big move in the utilities and telecom industries. *Roland Berger Strategy Consultants*.

Sehili, M., Istrate, D., Dorizzi, B., & Boudy, J. (2012). Daily Sound Recognition Using A Combination Of Gmm And Svm For Home Automation. *20th European Signal Processing Conference* (s. 1673-1677). Bucharest: EURASIP.

Shwehdi, M., & Khan, A. (1996). A Power Line Data Communication Interface Using Spread Spectrum Technology In Home Automation. *IEEE Transactions on Power Delivery*, 1232-1237.

Singh, M., & Huhns, M. (tarih yok). Multiagent Systems A Theoretical Framework for Intentions, Know-How, and Communications. *Lecture Notes in Computer Science* (s. 1-14). içinde North Carolina: Springer-Verlag.

*Smartthings*. (2016, 12 06). Samsung Smartthings: <https://shop.smartthings.com/> adresinden alındı

*socket.io*. (2015, 12 20). socket.io: <http://socket.io/> adresinden alındı

*SQL Data Generator*. (2016, 12 10). Redgate: <http://www.redgate.com/products/sql-development/sql-data-generator/> adresinden alındı

Sriskanthan, N., Tan, F., & Karande, A. (2002). Bluetooth based home automation system. *Microprocessors and Microsystems*, 281-289.



Stefanov, D., & Bien, Z. (2004). The Smart House for Older Persons and Persons With Physical Disabilities: Structure, Technology Arrangments, and Perspectives. *IEEE Transactions On Neural Systems And Rehabilitation Engineering*, 228-250.

Taş, E. (2015, 12 20). *NodeJS Nedir?* Bilgisayar Mühendisleri Odası: <http://dergi.bmo.org.tr/sayi-3/nodejs-nedir> adresinden alındı

Teoh, C., & Tan, C. (2010). A Neural Network Approach towards Reinforcing Smart Home Security. *Information and Telecommunication Technologies* (s. 1-5). Kuching: IEEE.

Thinagaran, P., Sulaiman, M., Yatim Sharif, K., Ramli, A., & Leong, C. (2013). Development of an Embedded Smart Home Management Scheme. *International Journal of Smart Home*, 15-26.

*Türk Dil Kurumu*. (2016, 03 25). <http://www.tdk.gov.tr/> adresinden alındı

Türkiye Zeka Vakfı. (2015, 09 03). *Zeka Nedir?* Türkiye Zeka Vakfı: <http://tzv.org.tr/zeka/zeka-nedir> adresinden alındı

*Ubuntu*. (2015, 12 20). Wikipedi: <http://wiki.ubuntu-tr.net/index.php?title=Ubuntu> adresinden alındı

Ulutaş, E., & Özer, M. (2000). Markov Modeli Kullanılarak Çukurova Bölgesinin Deprem Tehlikesinin Belirlenmesi. *Jeofizik*, 103-112.

*Usage of web servers for websites*. (2015, 12 20). W3 Techs: [http://w3techs.com/technologies/overview/web\\_server/all](http://w3techs.com/technologies/overview/web_server/all) adresinden alındı

Vainio, A.-M., Valtonen, M., & Vanhala, J. (2008). Proactive Fuzzy Control and Adaptation Methods for Smart Homes. *IEEE Intelligent Systems*, 42-49.

Vintan, L., Gellert, A., Petzold, J., & Ungerer, T. (2004). *Person Movement Prediction Using Neural Networks*. Augsburg: University of Augsburg.

Warren, S., & Pitts, W. (1990). Bulletin of Mothemnticnl. *Bulletin of Mathematical Biology*, 99-115.

*What is ppm?* (2016, 05 30). RapidTables: <http://www.rapidtables.com/math/number/PPM.htm#definition> adresinden alındı

Wilson, D., Wolfe, B., & Atkeson, C. (2008). *Using Many Simple Sensors for Automatic Monitoring in the Home*. PA: Robotics Institute Carnegie Mellon University.

*X10 Basics*. (2015, 07 11). X10: <http://www.x10.com/x10-basics.html> adresinden alındı

Yas, T., & Das, Y. (2013). Design of A Room Temperature And Humidity Controller Using Fuzzy Logic. *American Journal of Engineering Research* , 86-97.

Yılmaz, H. (2014). Akıllı Ev'in Dünyadaki ve Türkiye'deki Yeri-1. *Best Dergisi*.

Zadeh, L. (1965). Fuzzy Sets. *Information and Control*, 338-353.

Zhang, C., & Zhang, L. (2013). Activity Recognition in Smart Homes Based on Second-Order Hidden Markov Model. *International Journal of Smart Home*, 7(6), 237-244.

*ZigBee Alliance*. (2015, 02 15). ZigBee Alliance: <http://zigbee.org/> adresinden alındı

Zuo, F., & H. N., P. (2005). Real-time Embedded Face Recognition for Smart Home. *IEEE Transactions on Consumer Electronics*, 183-190.

*Z-Wave*. (2015, 02 15). Z-Wave: [http://www.z-wave.com/what\\_is\\_z-wave](http://www.z-wave.com/what_is_z-wave) adresinden alındı

*Z-Wave vs Zigbee*. (2015, 02 15). ZWave Guide: <http://zwaveguide.com/z-wave-vs-zigbee/> adresinden alındı

# **EKLER**



## EK B. Veritabanı tabloları

ad	tanım	kod	sıcaklık	nem	aydınlık	hareket	gaz	alev
Salon	Tüm aile bireylerinin tv seyretmek, sohbet etmek i...	os	34	0	10	0	0	0
Yatak Odası	Anne ve babanın yatak odası.	oy	34	0	11	0	0	0
Mutfak	Genellikle annenin vakit geçirdiği ve ailede herke...	om	35	0	12	0	0	0
Kız Çocuk Odası	Kız çocuğun evde genel olarak vakit geçirdiği oda.	ok	36	0	13	0	0	0
Erkek Çocuk Odası	Erkek çocuğun evde genel olarak vakit geçirdiği od...	oe	37	0	14	0	0	0
Ev	Evin girişi. Buradan çıktığında evin dışına çıkmış...	og	38	0	15	0	0	0
Gerçek Oda	Gerçek çalışmaların yapıldığı oda	go	39	39	22	0	1	0
Ev Dışı	Evin dışı	ed	28	8	91	0	0	0

Şekil B. 1: Oda bilgileri tablosu

idaygit	tip	ad	acik_kapali	deger
1	1	Salon Termostat	0	0
2	1	Yatak Odası Termostat	1	20
3	1	Mutfak Termostat	1	27
4	1	Kız Çocuk Odası Termostat	1	30
5	1	Erkek Çocuk Odası Termostat	1	29
6	2	Salon Lambası	1	50
7	2	Mutfak Lambası	0	0
8	2	Yatak Odası Lambası	0	0
9	2	Erkek Çocuk Odası Lambası	0	0
10	2	Kız Çocuk Odası Lambası	0	0
11	3	Salon Priz 1 - TV	0	0
12	3	Salon Priz 2 - Çok Amaçlı	1	0
13	3	Yatak Odası Priz 1	0	0
14	3	Yatak Odası Priz 2 - Çok Amaçlı	0	0
15	3	Mutfak Priz 1 - Bulaşık Makinesi	0	0
16	3	Mutfak Priz 2 - Çok Amaçlı	0	0
17	3	Kız Çocuk Odası Priz 1 - Bilgisayar	0	0
18	3	Kız Çocuk Odası Priz 2 - Çok Amaçlı	1	0
19	3	Erkek Çocuk Odası Priz 1 - Bilgisayar	1	0
20	3	Erkek Çocuk Odası Priz 2 - Çok Amaçlı	0	1

**Şekil B. 2:** Aygıt bilgileri tablosu

idaygit_tipi	aygit_tipi
1	Termostat
2	Lamba
3	Priz

**Şekil B. 3:** Aygıt tipi tablosu

idkullanici	isim	bulundugu_oda	yetki	kullanici_adi	sifre	goruntuleme_kodu
1	Ahmet YILMAZ	3	10	aliyilmaz	1234	BA
4	Ayşe YILMAZ	4	10	ayseyilmaz	1234	AN
5	Zeynep YILMAZ	6	5	zeynepyilmaz	1234	KÇ
6	Mehmet YILMAZ	7	4	mehmetyilmaz	1234	EÇ1
7	Mustafa YILMAZ	5	1	mustafayilmaz	1234	EÇ2
8	Hüseyin GÜNEŞ	8	9	hgunes	1234	HG

Şekil B. 4: Kullanıcı bilgileri tablosu



idkayit	tarih_saat	kullanici	oda	aygit	islem_tipi	islem_degeri	oda_sicaklik	oda_aydinlik	oda_nem
1	2015-01-03 17:05:38	1	0	19	7	0	2	98	0
2	2015-01-04 00:03:55	4	0	19	7	0	0	100	0
3	2015-01-04 12:56:19	1	0	9	7	0	2	98	0
4	2015-01-10 17:06:47	1	0	19	7	0	-4	98	0
5	2015-01-10 23:58:00	4	0	19	7	0	1	100	0
6	2015-01-11 12:59:27	1	0	9	7	0	0	98	0
7	2015-01-17 16:56:12	1	0	19	7	0	12	97	0
8	2015-01-18 00:08:27	4	0	19	7	0	-1	100	0
9	2015-01-18 13:05:58	1	0	9	7	0	11	98	0
10	2015-01-24 17:09:36	1	0	19	7	0	13	98	0
11	2015-01-24 23:50:29	4	0	19	7	0	-1	100	0
12	2015-01-25 13:05:32	1	0	9	7	0	14	98	0
13	2015-02-01 13:09:17	1	0	9	7	0	9	98	0
14	2015-02-07 17:06:45	1	0	19	7	0	5	98	0
15	2015-02-08 00:10:17	4	0	19	7	0	1	100	0
16	2015-02-08 13:03:54	1	0	9	7	0	14	98	0
17	2015-02-08 13:03:49	1	0	9	7	0	9	98	0

Şekil B. 5: İşlem bilgileri tablosu

idsensor_kayit	tarih_saat	oda	sicaklik	nem	aydinlik	hareket	gaz	ates	dis_sicaklik	dis_nem	dis_aydinlik
1	2016-07-23 15:42:30	10	27	17	65	0	1	1	28	8	91
2	2016-07-23 15:42:31	10	27	17	65	0	1	1	28	8	91
3	2016-07-23 15:42:32	10	27	17	65	0	1	1	28	8	91
4	2016-07-23 15:42:33	10	27	17	63	0	1	1	28	8	91
5	2016-07-23 15:42:35	10	27	17	60	0	1	1	28	8	91
6	2016-07-23 15:42:36	10	27	17	60	0	1	1	28	8	91
7	2016-07-23 15:42:37	10	27	17	60	0	1	1	28	8	91
8	2016-07-23 15:42:38	10	27	17	59	0	1	1	28	8	92
9	2016-07-23 15:42:39	10	27	17	60	0	1	1	28	8	91
10	2016-07-23 15:42:40	10	27	17	60	0	1	1	28	8	91
11	2016-07-23 15:42:42	10	27	17	59	0	1	1	28	8	91
12	2016-07-23 15:42:43	10	27	17	59	0	1	1	28	8	92
13	2016-07-23 15:42:44	10	25	18	60	0	1	1	28	8	91
14	2016-07-23 15:42:45	10	27	17	65	0	1	1	28	8	91
15	2016-07-23 15:42:46	10	27	17	67	0	1	1	28	8	91
16	2016-07-23 15:42:47	10	27	17	60	0	1	1	28	8	91
17	2016-07-23 15:42:49	10	27	17	57	1	1	1	28	8	91

Şekil B. 6: Sensör verileri tablosu

idkayit	tarih_saat	kullanici	oda	aygit	islem_tipi	islem_degeri	oda_sicaklik	oda_aydinlik	oda_nem
1	2015-01-01 06:37:26	1	8	1	2	0	10	50	0
2	2015-01-01 06:38:20	1	4	2	5	0	10	3	0
3	2015-01-01 06:38:50	1	4	8	9	0	10	3	0
4	2015-01-01 07:02:34	1	5	1	1	0	10	45	0
5	2015-01-01 07:02:45	1	5	3	3	19	10	45	0
6	2015-01-01 07:03:07	1	5	7	8	1	35	12	0
7	2015-01-01 07:17:01	1	8	1	2	0	10	100	0

Şekil B. 7: Yapay kayıtlar tablosu

## EK C. Sanal ailenin senaryosu

**Tablo C. 1:** Sanal ailenin senaryosu.

Ailenin Genel Özellikleri	<ul style="list-style-type: none"><li>• Aile, akşam yemeklerini her zaman 19.30 da beraber yer.</li><li>• Çocuklar hafta içi her sabah aynı vakitte uyanır ve anne ile beraber saat 9.00'da kahvaltı yapar.</li><li>• Pazar günleri 13.00 – 17.00 arası ailecek dışarı çıkarlar.</li><li>• Cumartesi günleri 17.00 – 19.00 arası dışarıda akşam yemeği yerler.</li></ul>
Ahmet YILMAZ	<ul style="list-style-type: none"><li>• Hafta içi her gün 08.00 – 19.00 arası çalışmaktadır. İş yeri ile evi arasındaki mesafe, 30 dakikadır.</li><li>• Salı, Perşembe ve Pazar günleri 20.00 – 22.00 arası spora gitmektedir. Evi ile spor salonu arasındaki mesafe, 10 dakikadır.</li><li>• Pazartesi ve Çarşamba günleri 21.00 – 23.00 arası eşi ile birlikte dizi izlemektedir (tv salondadır).</li><li>• Cumartesi günleri 00.00 – 01.00 arası halı saha maçlarına gitmektedir. Maçın yapıldığı yer ile evi arasında 10 dakikalık bir mesafe vardır.</li><li>• Pazar günleri 20.00 – 23.00 arası spor programı seyretmektedir.</li><li>• Belirtilmeyen diğer günlerde ise 21.00 – 23.00 arası eşi ile beraber salonda vakit geçirmektedir.</li><li>• Hafta içi 06.30 – 07.00 arası, hafta sonu ise 10.00 civarı uyanmaktadır.</li><li>• Cuma günleri 00.00 – 01.00 arası, Cumartesi günleri 02.00 civarı, haftanın geri kalan günlerinde ise 23.00 civarı uyumaktadır.</li></ul>
Ayşe YILMAZ	<ul style="list-style-type: none"><li>• Hafta içi her gün sabah 8.15-8.45 arası uyanır. Hafta sonları 10.00 civarı uyanır.</li><li>• Her sabah uyandıktan 10 dakika sonra mutfakta kahvaltı hazırlamaya gider. 15 dakikada kahvaltı hazırladıktan sonra tekrardan uyumaya gider. 8.40-9.10 arası tekrardan uyumaya gider ve 10.00'da uyanır.</li></ul>

**Tablo C. 1:** Sanal ailenin senaryosu (devam).

	<ul style="list-style-type: none"><li>• Hafta içi her gün 10.00'dan sonra 30'ar dakika sürecek şekilde her odada temizlik yapar. (12.30'da bu işlem biter.)</li><li>• 12.30-13.30 arası hafta içi her gün salonda televizyon seyrederek.</li><li>• Salı ve Perşembe günleri evin haftalık alışverişini yapar. Bunun için 30 dakika uzaklıktaki alışveriş merkezine gider, 14.00-16.45 arası alışveriş yapar.</li><li>• Pazartesi, Çarşamba ve Cuma günleri 20 dakika uzaklıktaki spor salonuna gider. 13.50-16.50 arası spor yapar.</li><li>• Pazartesi, Çarşamba ve Cuma günleri 17.10'da; Salı ve Perşembe günleri 17.15'te evde olur. 30 dakika boyunca mutfakta akşam yemeğini hazırlar.</li><li>• Pazartesi ve Çarşamba günleri 21.00-23.00 arası eşi ile beraber televizyon seyrederek. Pazar akşamı hariç eşiyile beraber 21.00-23.00 arası salonda oturur.</li><li>• Cumartesi günleri 20.00-21.00 arası oğluna (erkek çocuk odasında), 21.00-22.00 arası kızına (kız çocuk odasında) ödevlerinde yardım eder. 22.00-23.00 kendi odasında bilgisayarında vakit geçirir.</li><li>• Hafta içi (Cuma hariç) 23.00 civarı, Cuma günü ve hafta sonu 00.00-01.00 civarı uyur.</li></ul>
Mustafa YILMAZ	<ul style="list-style-type: none"><li>• Hafta içi her gün 9.30- 17.30 arası okulda ya da dışardadır.</li><li>• Hafta içi her gün saat 8.30'da uyanır. 9.30'da evden çıkar ve 30 dakika uzaklıktaki okuluna gider. Hafta sonları ise 10.00 civarı uyanır.</li><li>• Perşembe ve Cuma akşamları 00.00'da eve dönmekte ve yarım saat yalnız yemek yer.</li><li>• Cumartesi sabaha karşı 03.00-05.00 arası basketbol maçları seyretmektedir.</li><li>• Cumartesi sabah 11.00'da evden çıktıktan sonra Pazar akşam 10.00'da döner.</li><li>• Hafta içi eve erken geldiğinde 00.00'da geç geldiğinde ise 01.00'da uyur.</li></ul>

**Tablo C. 1:** Sanal ailenin senaryosu (devam).

Zeynep YILMAZ	<ul style="list-style-type: none"><li>• Hafta içi her gün 9.30- 14.30 arası okuldadır.</li><li>• Hafta içi her gün saat 8.30'da uyanır. 9.00'da evden çıkar ve 30 dakika uzaklıktaki okuluna gider. Hafta sonları ise 10.00 civarı uyanır.</li><li>• Pazartesi, Çarşamba ve Cuma günleri okuldan sonra 1 saat 30 dakika boyunca ek çalışma yapar. 16.30'da evde olur. Salı ve Perşembe günleri 15.00'da evde olur.</li><li>• Cumartesi günleri 12.00-14.00 arası ebru kursuna gider daha sonra 15.00-17.00 dışarıya arkadaşlarıyla eğlenmeye gider. 17.00-19.00 arası ailesi ile dışarda yemek yer. 19.00-21.00 arası odasında vakit geçirir. 21.00-22.00 arası odasında ders çalışır. Annesi dersine yardım eder.</li><li>• Pazar günü 13.00-17.00 arası ailesi ile beraber dışarı çıkar.</li><li>• Hafta içi (Cuma hariç) 23.00 civarı, Cuma günü ve hafta sonu 00.00 civarı uyur.</li></ul>
Mehmet YILMAZ	<ul style="list-style-type: none"><li>• Hafta içi her gün 9.30- 14.30 arası okuldadır.</li><li>• Hafta içi her gün saat 8.30'da uyanır. 9.00'da evden çıkar ve 30 dakika uzaklıktaki okuluna gider. Hafta sonları ise 10.00 civarı uyanır.</li><li>• Pazartesi ve Çarşamba günleri okuldan sonra 2 saatliğine futbol kursuna gider. 17.00'da evde olur.</li><li>• Cuma günleri 14.30-16.30 arası müzik kursuna gider. 17.00'da evde olur.</li><li>• Cumartesi günleri 17.00-19.00 arası ailesi ile dışarda yemek yer. 19.00-20.00 arası odasında vakit geçirir. 20.00-21.00 arası odasında ders çalışır. Annesi dersine yardım eder.</li><li>• Pazar günü 13.00-17.00 arası ailesi ile beraber dışarı çıkar.</li><li>• Hafta içi (Cuma hariç) 22.30 civarı, Cuma günü ve hafta sonu 23.00 civarı uyur.</li></ul>

## EK D. Akıllı Ev Simülasyonu (Hibrit Simülasyon)

The screenshot displays a hybrid simulation interface for a smart home. The top section shows a 3D floor plan of a house with various rooms labeled in Turkish: Yatak Odası (Bedroom), Kız Çocuk Odası (Daughter Room), Erkek Çocuk Odası (Boy Room), Salon (Living Room), Mutfak (Kitchen), Gerçek Oda (Real Room), and EVDİŞİ (Smart Home Hub). The floor plan includes colored circles representing smart devices: KÇ AN (Red), HG (Yellow), EC2 (Green), BA (Blue), and EC1 (Green).

The bottom section shows a list of smart devices under the heading "Yaklaşan Olaylar" (Upcoming Events). The list includes:

Smart Device	Location	Value
2038829 KÇ - Zeynep	Aydınlatma Ayarı	89
2038813 KÇ - Zeynep	Kız Çocuk Odası	27
2034226 KÇ - Zeynep	Oda Giriş	0
2022554 BA - Ahmet	Mutfak	0
2022349 BA - Ahmet	Lamba Kapat	0
2022330 BA - Ahmet	Mutfak Temizlik	0

Below this, there is a section for "Emirler" (Commands) with the following data:

Command	Location	Value
1020000 BA - Ahmet	Salon Temizlik	25
2312100 BA - Ahmet	Yatak Odası	0
2300000 ANI - Ayşe	Yatak Odası	0
1139000 KÇ - Zeynep	Ev	0

The interface also features a navigation menu on the right side with options like "Olay Ekle", "Ayarlar", "Kişi-Zaman-Konum", "Konumdan Konuma", "Kişi Tercihleri", "Aygıt Kullanım Zamanları", "İlişili Olaylar", and "İsminin Sağına Sürükle". At the bottom right, the "Simülasyon Sani: 1:1438:57" is displayed.

Şekil D.1: Hibrit simülasyon arayüzü

## EK E. Odada Daha Yetkili Biri Var mı? Program Kodu

```
function o_an_odada_daha_yetkili_biri_var_mi($kisi, $oda)
{
    $CI = get_instance();

    $yetki = $CI->vt_model-
>bul_ve_alan_getir("person_logger_information", "person", $kisi,
"oda_yetki");

    $kod = $CI->vt_model->bul_ve_alan_getir("oda_", "idoda", $oda,
"kod");

    //echo "Kod : ".$kod."<br>";

    $yetki_seviyesi = 0; $yetkiler = json_decode($yetki);
    foreach ($yetkiler as $yetki){
        if (!empty($yetki->{"$kod"}))
            $yetki_seviyesi = $yetki->{"$kod"};
    }

    $odadakilerin_yetkisi = 0;

    $odadakiler = $CI->vt_model->coklu_bul_ve_getir("person_",
array("place"=>$oda));

    foreach($odadakiler as $odadaki){
        $odadaki_yetkiler = $CI->vt_model-
>bul_ve_alan_getir("person_logger_information", "person",
$odadaki["idperson"], "oda_yetki");

        $odadaki_yetkiler = json_decode($odadaki_yetkiler);
        foreach ($odadaki_yetkiler as $yetkicik){
            if (!empty($yetkicik->{"$kod"})){
                $yetki_karsilastir = $yetkicik->{"$kod"} *
1;

                if($yetki_karsilastir > $odadakilerin_yetkisi)
                    {
                        $odadakilerin_yetkisi = $yetki_karsilastir;
                    }
            }
        }

        if ($odadakilerin_yetkisi > $yetki_seviyesi)
            return true;
        else return false; }
}
```



## EK F. Kullanıcı Konum Tespit İşlemi Kodları

```
function bu_saatte_bu_kisi_nerde($kisi, $saat){

    $date = date('H:i:00', strtotime($saat));

    $CI = get_instance();

    $CI->db->where("saat",$date);

    $CI->db->where("kisi",$kisi);

    $konumlar = $CI->db->get("yz_kisi_konum")->result_array();

    $toplama_bulunulan_konum = 0;

    $date = date('H:i:00', time());

    $haftanin_gunu = date("w", strtotime($saat));

    $gun="";

    $hafta_ici_sonu = "gunluk_haftaici";

    switch($haftanin_gunu){

        case 1: $gun = "pazartesi"; break;

        case 2: $gun = "sali";          break;

        case 3: $gun = "carsamba";  break;

        case 4: $gun = "persembe";  break;

        case 5: $gun = "cuma";          break;

        case 6: $gun = "cumartesi"; $hafta_ici_sonu = "gunluk_haftasonu"; break;

        case 7: $gun = "pazar";      $hafta_ici_sonu = "gunluk_haftasonu"; break;

        case 0: $gun = "pazar";      $hafta_ici_sonu = "gunluk_haftasonu"; break;

    }$gunluk_toplam = 0; $hafta_ici_sonu_toplam = 0;

    $haftanin_gunu_toplam = 0; $konum_bilgileri = new stdClass();

    $i=0;

    foreach($konumlar as $konum){

        $gunluk_toplam += $konum["gunluk"];

        $hafta_ici_sonu_toplam += $konum[$hafta_ici_sonu];

    }

}
```

```

        $haftanin_gunu_toplam += $konum[$gun];}

$date_h = date('H', strtotime($saat));

$date_m = date('i', strtotime($saat));

$date2 = date('Y:m:d', strtotime($saat));

$CI->db->where("HOUR(zaman)", $date_h);

$CI->db->where("MINUTE(zaman)", $date_m);

$CI->db->select("count(*) as toplam");

$CI->db->select("kisi_". $kisi. "_konum as konum");

$CI->db->order_by("zaman", "DESC");

$CI->db->group_by("kisi_1_konum");

$konumlar_son_10_gun = $CI->db->get("yz_dakika_kayit", 10, 0)->result_array();

$son_10_gun_toplam = 0;

foreach($konumlar_son_10_gun as $konum){

        $son_10_gun_toplam += $konum["toplam"];}

// Min-Max normalizasyonu yapıyor.

foreach($konumlar as $konum){

        $konumlar[$i]["p_olasilik_gunluk"] = $konum["gunluk"]/$gunluk_toplam;

        $konumlar[$i]["p_olasilik_hafta_ici_sonu"] = $konum["hafta_ici_sonu"]/$konum["hafta_ici_sonu"]/$hafta_ici_sonu_toplam;

        $konumlar[$i]["p_olasilik_haftanin_gunu"] = $konum[$gun]/$haftanin_gunu_toplam;

        foreach($konumlar_son_10_gun as $konum2){

                if ($konum2["konum"]==$konum["bulundugu_oda"])

$konumlar[$i]["p_olasilik_son_10_gun"] = $konum2["toplam"]/$son_10_gun_toplam;

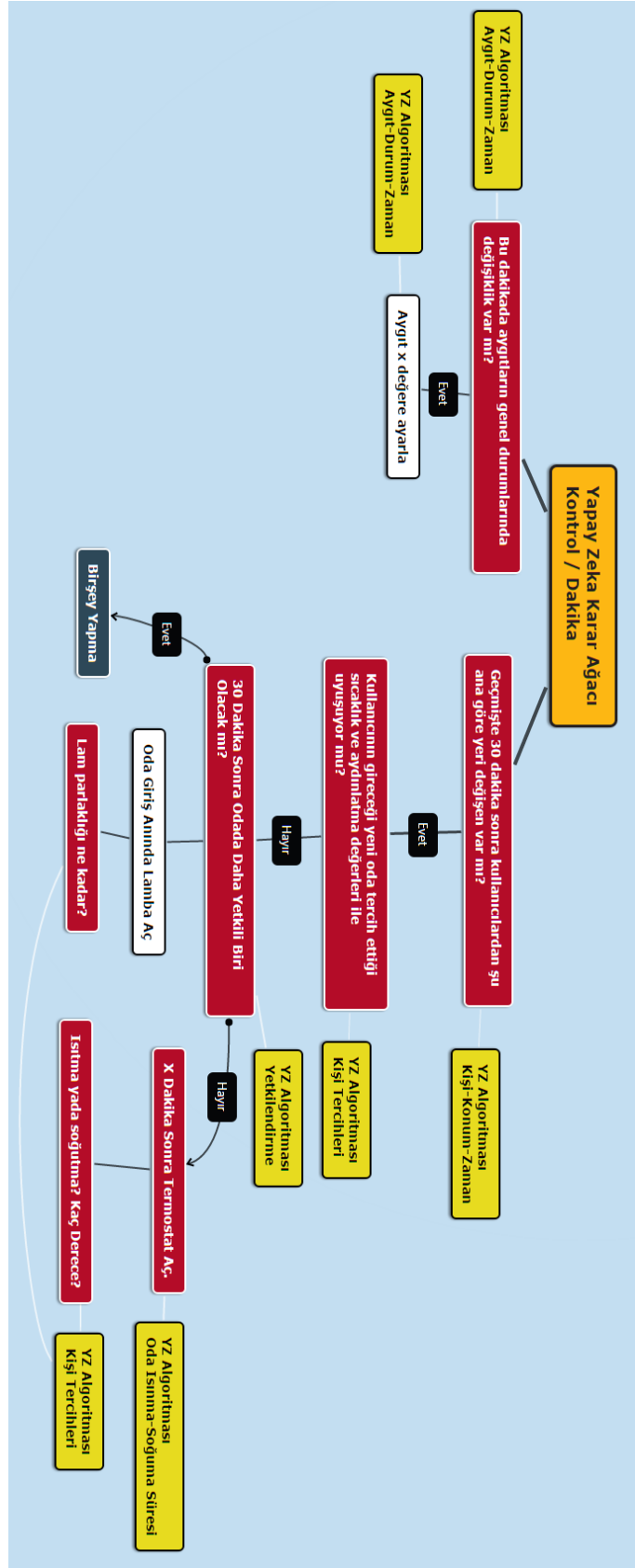
        }

        $i++;}

return $konumlar;}

```

## EK G. Karar Ağacı



Şekil G. 1: Yapay zekâ karar ağacı