

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

IoB Destekli Isıtma Sistemi Otomasyonu

B181210052 – Arslançan SARIKAYA

Bölüm	:	BİLGİSAYAR MÜHENDİSLİĞİ
Danışman	:	Prof. Dr. Cüneyt BAYILMIŞ

2022-2023 Güz Dönemi

ÖNSÖZ

Lisans eğitimim boyunca öğrendiğim teorik bilgiler, edindiğim tecrübeler sonucu böyle bir tasarım projesini sunabildiğim için mutlu ve gururlu hissediyorum. Beni bu aşamaya getiren yolculuğumda beni her daim destekleyen tüm öğretmenlerime minnettarım.

Ayrıca üzerimde desteklerini esirgemeyen ailem ve arkadaşlarıma sonsuz teşekkürler. Bu tasarım projesinde son kullanıcıların ısıtma sistemlerini kullanım alışkanlıkları analiz edilip, makine öğrenmesi algoritmaları kullanılarak otonom hale getirilmesi amaçlanmıştır.

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	vii
ÖZET.....	viii

BÖLÜM 1. GİRİŞ.....	1
1.1. PROBLEMİN TANIMI.....	2
1.2. DAVRANIŞLARIN İNTERNETİ (IoB).....	2
1.2.1. IoB’NİN YARARLANDIĞI TEKNOLOJİLER.....	3
1.2.2. IoB’NİN FARKLI ALANLARDAKİ UYGULAMALARI.....	3
1.2.3. IoB’DEKİ SORUNLAR	4
1.3. MAKİNE ÖĞRENMESİ.....	5
1.3.1. MAKİNE ÖĞRENMESİNİN TARİHİ	5
1.3.2. MAKİNE ÖĞRENMESİ TEKNİKLERİ	6
1.3.2.1. <i>Supervised (Gözetimli) öğrenme</i>	6
1.3.2.2. <i>Unsupervised (Gözetimsiz) öğrenme</i>	6
1.3.2.3. <i>Reinforcement (Takviyeli) öğrenme</i>	6
1.4. KULLANILAN DONANIMLAR.....	6
1.5. BAŞARI ÖLÇÜTLERİ.....	7

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER ..	8
2.1. KULLANILAN TEKNOLOJİLERİN TANITIMI	8
2.1.1. MAKİNE ÖĞRENMESİ ALGORİTMALARI	8
2.1.1.1. <i>Makine öğrenimi algoritması seçimi</i>	9
2.1.1.2. <i>Kaggle</i>	10
2.1.1.3. <i>Veri seti seçimi</i>	10
2.1.2. DHT11 SICAKLIK VE NEM SENSÖRÜ	10
2.1.3. TEK KANALLI 5V RÖLE	10
2.1.4. FIREBASE	11

BÖLÜM 3. GELİŞTİRİLEN SİSTEM.....	12
3.1. AKIŞ DİYAGRAMI.....	12
3.2. DEVRE ŞEMASI	13

3.3.	YAZILIM GEREKSİNİM LİSTESİ	13
3.4.	VERİTABANI KAVRAMSAL MODELİ.....	14
BÖLÜM 4.	ENDÜSTRİYEL BOYUT VE KISITLAR.....	15
4.1.	DONANIM MALİYET TABLOSU.....	15
4.2.	İŞ ZAMAN ÇİZELGESİ VE MALİYET ANALİZİ.....	15
4.3.	DONANIM İÇİN GEREKLİ ENERJİ ANALİZİ.....	16
4.4.	VERİTABANI MALİYET ANALİZİ	16
BÖLÜM 5.	UYGULAMA.....	17
5.1.1.	TAHMİN ALGORİTMASININ HAZIRLANMASI	17
5.2.	GEREKLİ SENSÖR VE DONANIMLARIN KURULUMU.....	18
5.2.1.	SICAKLIK SENSÖRÜNÜN PROGRAMLANMASI	19
5.2.2.	NODEMCU’NUN FİREBASE KURULUMUNUN YAPILMASI	19
5.2.3.	MAKİNE VE FİREBASE BAĞLANTISI.....	22
5.2.4.	TEK KANALLI 5V RÖLENİN PROGRAMLANMASI.....	22
BÖLÜM 6.	SONUÇLAR VE ÖNERİLER.....	24
	KAYNAKLAR.....	27
	ÖZGEÇMİŞ	28
	BSM401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI.....	29
	DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	29

SİMGELER VE KISALTMALAR LİSTESİ

IoB	: Internet of Behaviors (Davranışların İnterneti)
IoT	: Internet of Things (Nesnelerin İnterneti)

ŞEKİLLER LİSTESİ

Şekil 3.1.	Sistemin akış diyagramı.....	12
Şekil 3.2.	Deney düzeneğinin Fritzing programı ile oluşturulmuş şeması.....	13
Şekil 5.1.	Confusion matrix şeması.....	18
Şekil 5.2.	Firestore veritabanı tablosu	20
Şekil 5.3.	Firestore proje ayarları sekmesi	20
Şekil 5.4.	Proje ayarlarında doğrulama şifresinin konumu.....	21
Şekil 6.1.	Confusion matrix görseli.....	24
Şekil 6.2.	Confusion matrix hesaplama tablosu.....	24

TABLÖLAR LİSTESİ

Tablo 3.1.	Veritabanı kavramsal modeli.....	14
Tablo 4.1.	Donanım maliyet tablosu	15
Tablo 4.2.	İş Zaman Çizelgesi Maliyet Tablosu	15

ÖZET

Anahtar Kelimeler: Nesnelerin İnterneti, Davranışların İnterneti, Makine Öğrenmesi

Bu proje, insanların ısıtma sistemlerini kullandıkları zaman odalarının sıcaklık ve nem değerlerini analiz ederek, makine öğrenimi yardımıyla insanların odaları ne kadar süre kullandıklarını tahmin ederek, gereksiz ısıtma sistemi kullanımını önlemeyi amaçlamaktadır. Bu amaç doğrultusunda, insanların evlerinde kaç saat nerede bulunduğu tespit edilerek, klima veya kombi sistemlerinin yönetimi bu verilere göre otomatikleştirilmek istenmektedir. Bu proje, İnternet of Behaviors (IoB) teknolojisinin temel alındığı İnternet of Things (IoT) ağları üzerinden gerçekleştirilecektir.

BÖLÜM 1. GİRİŞ

Bu tasarım projesinde kullanıcıların ısıtma sistemlerini kullanırken kullandıkları odanın sıcaklık ve nem değerleri analiz edilerek yaşam alanlarının kullanım aralıkları makine öğrenmesi aracılığıyla tahmin edilerek gereksiz ısıtma sistemi kullanımının önüne geçilmek istenmektedir.

Sahip olduğumuz sınırlı enerji kaynaklarına rağmen dünyadaki insan nüfusu günden güne hızla artmaktadır. Bu sebeplerden enerji tasarrufu bizler için hayati önem taşımaktadır. Bu çalışmada da bu önem dikkate alınmış ve IoB teknolojisi kullanılarak insanların hayatını kolaylaştıran bir çözüm sunulmak istenmiştir.

IoB'nin temeli IoT'ye dayanır. IoT insanların günlük hayatındaki nesneler ve internet arasındaki dijital bağlantıdır. IoT son yıllarda yaşadığı dikkat çekici ivme sebebiyle davranışa dayalı verileri işleyerek yerini IoB'ye bırakmaktadır.

IoB – Davranışların İnterneti, fiziksel verilerin nesneler ve yazılımlarla ilişkilendirilmesi ile oluşan ağlardır. IoB ile nesnelerin insan davranışları esas alınarak insan hayatında var olan teknolojilerin kullanım aralıklarının ve yöntemlerinin kişiselleştirilmesi mümkündür. Son kullanıcıların fiziksel aktivitelerinde değişiklikler gözlemlendiğinde akıllı saatlerin ilgili uyarıları göndermesi buna örnektir. Son yıllarda akıllı saatlerin kullanım oranının son derece artması bizlere IoB'nin günden güne hayatlarımıza daha çok entegre olmasının insanlar tarafından tercih edildiğinin kanıtlarından biri olarak gösterilebilir.

Bu tasarım çalışması ile son kullanıcıların günlük hayatlarındaki kullanımları analiz edilerek ısıtma sistemlerinin yönetiminin otonom hale getirilmesi amaçlanmıştır. Örnek vererek açıklamak gerekirse, insanların gün içerisinde evlerinde hangi odada kaç saat bulunduğu tespit edilip klima veya kombi sistemlerinin yönetiminin bu davranışların analizi esas alınarak otonom hale getirilecektir.

1.1. Problemin Tanımı

İnsanların kullandığı ofis, oda, toplantı salonu vb. yaşam alanlarının kullanılmadığı halde sıcaklığının optimum tutulmasından dolayı ortaya çıkan enerji israfı

1.2. Davranışların İnterneti (IoB)

IoB'yi anlatırken IoT ile arasındaki düşünsel yaklaşım farkına değinirsek konunun biraz daha kolay anlaşılacağını düşünüyorum.

IoT, doğrusal bir şekilde çalışır. Evinize gitmeden önce dijital asistanınızdan evinizin sıcaklığını ayarlamasını istediğinizde IoT bunu herhangi bir veri analizi olmadan doğrudan gerçekleştirecektir. IoB ise kullanıcının sıcaklık tercihlerini ve hangi saat aralıklarında evde olduğunu anlayarak güç tüketimini azaltmak ile ilgilenecektir. [7]

Internet of behaviors, insan davranışlarının internet aracılığıyla toplandığı, analiz edildiği ve kullanıldığı bir teknolojidir. Bu teknoloji, insanların internet kullanımı, arama terimleri, web sitelerini ziyaret ettiği süreler, sosyal medya gönderileri gibi birçok farklı veriyi toplar ve bu verileri kullanarak insanların davranışlarını anlamaya çalışır. [7]

Bu teknolojinin amacı, insanların davranışlarını anlamak ve bu davranışlarının nedenlerini araştırmaktır. Örneğin, bir kişinin internette arama yaptığı kelime ve terimler, o kişinin ne hakkında bilgi aradığını gösterebilir. Bu bilgi, o kişinin ilgi alanlarını, müşteri potansiyelini ve diğer özelliklerini belirleyebilir. [7]

Internet of behaviors teknolojisi, birçok farklı alanda kullanılabilir. Örneğin, bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını öğrenerek, müşteri taleplerine daha iyi cevap verebilir. Ayrıca, bir sosyal medya şirketi, kullanıcılarının ne tür içerikleri paylaştıklarını ve hangi tür içerikleri daha fazla ilgi gördüğünü öğrenerek, kullanıcılarına daha iyi hizmet sunabilir.

1.2.1. IoB'nin yararlandığı teknolojiler

Veri toplama: IoB teknolojisi insanların davranışlarını anlamak için insanların verilerine ihtiyaç duyar ve insanların arama terimleri, web sitelerini ziyaret ettiği süreler, sosyal medya gönderileri gibi birçok farklı veriyi toplar.

Veri analizi: Toplanan verilerin anlamlandırılması için veri analizine başvurulur. Bu işlem, yapay zeka ve makine öğrenimi gibi teknolojilerle gerçekleştirilebilir.

Veri güvenliği: Internet of behaviors teknolojisi kullanırken, toplanan verilerin güvenliği önemlidir. Bu nedenle, verilerin korunması için güvenlik duvarı, şifreleme gibi önlemler alınabilir.

Veri depolama: Toplanan veriler, internet of behaviors teknolojisi çalışırken, veri depolama sistemlerine kaydedilir. Bu sistemler, verilerin daha sonra analiz edilmesine ve kullanılmasına olanak sağlar.

1.2.2. IoB'nin farklı alanlardaki uygulamaları

Pazarlama: Bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını ve hangi web sitelerini ziyaret ettiklerini öğrenerek, müşteri taleplerine daha iyi cevap verebilir. Ayrıca, şirket, müşterilerinin hangi tür içerikleri daha fazla ilgi gördüğünü öğrenerek, bu tür içerikleri üretebilir ve müşterilere daha iyi hizmet sunabilir.

Sağlık: Internet of behaviors teknolojisi, insanların sağlıkla ilgili davranışlarını inceleyerek, sağlık risklerini azaltmaya yardımcı olabilir. Örneğin, bir kişinin internette ne hakkında arama yaptığı, hangi web sitelerini ziyaret ettiği ve hangi sağlık uygulamalarını indirdiği öğrenilerek, o kişinin sağlık durumunun takip edilebilir ve olası sağlık sorunları önceden tespit edilebilir.

Eğitim: Öğrencilerin internette ne hakkında arama yaptıkları, hangi web sitelerini ziyaret ettikleri ve hangi öğrenme kaynaklarını kullandıkları öğrenilerek, eğitim

sistemlerine daha iyi hizmet verilebilir. Örneğin, bir öğrencinin hangi konuları daha fazla ilgi gösterdiği öğrenilerek, o öğrenciye daha fazla ilgi göstereceği konular hakkında dersler verilebilir. [9]

Sosyal medya: IoB, sosyal medya şirketlerinin kullanıcılarının ne tür içerikleri paylaştıklarını ve hangi tür içeriklerin daha fazla ilgi gördüğünü öğrenmeye yardımcı olur. Bu bilgiler, sosyal medya şirketlerinin kullanıcılarına daha iyi hizmet sunabilmelerine yardımcı olur. [9]

Müşteri profili ve hedefleme: Bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını ve hangi web sitelerini ziyaret ettiklerini öğrenerek, müşteri profillerini oluşturabilir ve müşteri hedefleme çalışmalarında kullanabilir. [9]

1.2.3. IoB'deki Sorunlar

Güvenlik: IoT cihazları, Bilgisayar korsanlarının en büyük hedeflerinden biridir. Ağımızdaki IoT cihazların varlığı, ağa sızmak için potansiyel alanların sayısını artırır. Bu tür güvenlik açıkları, IoT cihazlarının kullanımını ciddi şekilde kısıtlamaktadır. Bu, ağdaki cihazlar tarafından toplanan verilerin bütünlüğünü azaltır. Verilerin bütünlüğünden ödün verilmesi nedeniyle verilerin IoB analizi tehlikeye girebilir ve üzerinde oynanmış bir sonuca yol açabilir.[10][9]

Bant genişliği: IoT cihazlar her zaman internete bağlı olmak zorundadır ve bağlı oldukları sürece gerçek zamanlı upload ve download işlemi yaparlar. IoT cihazlarında veri depolayıp toplu olarak yükleyebilecekleri bir bölüm yoktur. Bu yüzden düşük bant genişliği IoT cihazının gerçek zamanlı veri yüklemesini engelleyebilir. [10][9]

Uyumsuzluk: IoB cihazları arasında herhangi bir standart yoktur. Bu yüzden her cihaz aynı şekilde veri toplamayabilir. Cihazların farklı veri organizasyon türü olacaktır ve bu IoB ağının analizini yaparken oluşan veri tutarsızlıkları can sıkıcı olabilir.[9]

Donanım Sorunları: IoB cihazları [8]de yazdığı gibi her türlü veriyi toplamak için kullanılabilir; ama bu cihazlar çevresel faktörlerin etkisine karşı savunmasızdır. Oda

sıcaklığını ölçen bir sistemde sıcaklık sensörünün yanına koyulan bir kibrit parçası verileriniz doğruluğunu ortadan kaldıracaktır. [9]

Gizlilik: IoB insan davranışlarını anlamak için veri madenciliği kullanır ve kullanıcı deneyimini iyileştirmek için daha fazla veriye ihtiyaç duyar ve bu da mahremiyet sorununu gündeme getiriyor. [9]

1.3. Makine Öğrenmesi

İnsan davranışları analiz edilip davranışların tahmin edilmesinde makine öğrenmesinden faydalanılır.

1.3.1. Makine öğrenmesinin tarihi

Makine öğrenimi terimi 1959'da bilgisayar oyunları ve yapay zeka alanında öncü ve IBM çalışanı olan Amerikalı Arthur Samuel tarafından icat edildi. 1960'larda makine öğrenimi araştırmasının temsili bir kitabı, Nilsson'un Öğrenme Makineleri hakkındaki kitabıydı ve çoğunlukla örüntü sınıflandırması için makine öğrenimi ile ilgiliydi. Model tanıma ile ilgili ilgi, 1973'te Duda ve Hart tarafından tanımlandığı gibi 1970'lerde de devam etti. 1981'de, bir sinir ağı 'nın bir bilgisayar terminalinden 40 karakteri (26 harf, 10 rakam ve 4 özel sembol) tanımayı öğrenmesi için öğretme stratejilerinin kullanımına ilişkin bir rapor verildi. [1]

Tom M. Mitchell, makine öğrenimi alanında incelenen algoritmaların geniş ölçüde alıntılanan daha resmi bir tanımını yaptı: "Bir bilgisayar programının performans ölçüsü "P" ve bazı "T" görev sınıflarıyla ilgili olarak "T" görevlerindeki performansı "E" deneyimiyle iyileşiyorsa "P" ile ölçüldüğü gibi E deneyiminden öğrendiği söylenir. Makine öğreniminin söz konusu olduğu görevlerin bu tanımı, alanı bilişsel terimlerle tanımlamak yerine temelde operasyonel tanım sunar.[1]

1.3.2. Makine öğrenmesi teknikleri

Makine öğrenmesi, öğrenme yöntemine göre üç gruba ayrılır; Supervised (Gözetimli), Unsupervised (Gözetimsiz) ve Reinforcement (Takviyeli).

1.3.2.1. Supervised (Gözetimli) öğrenme

Bu öğrenme tekniğinde verilen X girdi kümesinden istenen Y çıktı kümesinin elde edilmesi için bir fonksiyon öğrenilir.

1.3.2.2. Unsupervised (Gözetimsiz) öğrenme

Bu yöntem gözetimli öğrenmeden farklı olarak, verileri sebep-sonuç ya da giriş-çıkış şeklinde etiketlemeden, veri içerisinde var olan ilişkilerin ve yapılarının öğrenilmesidir. Veri örneklerinin birbirine olan uzaklıklarını, komşuluk ilişkilerini ve yoğunluklarını kullanarak veriyle ilgili çıkarımlar yapılmasını sağlar. Bu tekniğin önemli yaklaşımı boyut indirgeme ve kümelemedir.

1.3.2.3. Reinforcement (Takviyeli) öğrenme

Bu Teknik, öznelerin bir ortamda en yüksek ödül miktarına ulaşabilmesi için hangi eylemleri yapması gerektiğiyle ilgilenen bir makine öğrenmesi yaklaşımıdır. Takviyeli öğrenme, doğru girdi/çıkı eşleşmelerinin verilmemesi ve optimal olmayan eylemlerin dışarıdan düzeltilmemesi yönleriyle gözetimli öğrenmeden ayrışır. Dahası, pekiştirmeli öğrenmede bilinmeyen uzayda keşif ile mevcut bilgiden istifade arasında bir denge kurma söz konusudur.

1.4. Kullanılan Donanımlar

Bu tasarım çalışmasını gerçeklemek için bir tane devre kartına, sıcaklık sensörüne ve ısıtma sistemimizi açıp kapatabilmek için röleye ihtiyaç duyuyoruz.

Devre kartı olarak NodeMCU V3 tercih ettim. Bunun nedeni karta entegre wifi modülünün bulunması ve kartın seri üretimde olmasından dolayı maliyetlerinin özel üretim kartlara göre daha düşük fiyat bandında olması diyebilirim.

Sıcaklık sensörü olarak DHT11 sıcaklık sensörünü kullanmayı tercih ettim. Bu tercihi yapmamdaki sebep DHT11'in diğer sıcaklık sensörlerine göre daha çok tercih ediliyor olması ve dolayısıyla erişebileceğim kaynakların sayısının fazla olmasıydı. Elbette diğer sensör fiyatlarına göre fiyatının makul olması da sebeplerimden biriydi.

Röle olarak Tek Kanallı 5V bir röle tercih ettim. NodeMCU'nun her röleyi kaldırmadığını bildiğimden sınırlı sayıda seçeneğim vardı. NodeMCU ve Tek Kanallı 5V rölenin birlikte kullanıldığı projeler hakkında bilgi sahibi olduğum için bu tercihi yaptım.

1.5. Başarı Ölçütleri

Bu tasarım çalışmasında kullanıcıların kullanım verilerini analiz ederek kullanım alışkanlıklarını tahmin etmeye çalışıyorum. Isıtma sistemlerinin gereksiz kullanımını ortadan kaldırarak kullanıcı konforundan ödün vermeden ortamları kullanılan vakitlerde istenilen sıcaklığa getirebilmek, bu tasarım projesini başarılı olarak nitelendirebilecek maddelerdir.

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER

Bu bölümde tasarım projesinin uygulama aşamasında hangi yöntemlerin kullanıldığı anlatılıyor olacaktır.

Projenin işleyişi şu şekilde gerçekleşiyor. NodeMCU'ya bağlı DHT11 sensörü sıcaklık ve nem bilgisini firebase platformuna gerçek zamanlı bir şekilde yüklüyor. Yüklenen bu değerleri makine alıyor ve o an odanın dolu olup olmadığını tahmin etmek için kullanıyor. Oda'nın dolu olup olmadığını bilgisi firebase'e geri yollanıyor. Az önce sensör okumak için kullanılan NodeMCU bu sefer oda'nın dolu olup olmadığını bilgisini firebase'den çekerek NodeMCU'ya bağlı olan tek kanallı röleyi oda'nın boş olup olmama durumuna göre açıp kapatıyor.

2.1. Kullanılan Teknolojilerin Tanıtımı

2.1.1. Makine öğrenmesi algoritmaları

Doğrusal regresyon: İki değişkenin kesiştikleri noktalar arasında doğrusal bir ilişki kurarak tahminde bulunur. ($y=ax+b$) denkleminde x 'i değiştirdiğimizde y 'nin değeri nedir sorusunu cevaplar.

Lojistik regresyon: Bir sınıflandırma algoritmasıdır. Lineer regresyonda bir tahmin sonucu 0.2 gibi float bir değer elde edilebilir; fakat bu durum lojistik regresyon için mümkün değildir. Lojistik regresyonda var-yok, canlı-ölü, veri iletişimi dersini geçti-kaldı gibi cevapları 0 ve 1 olarak şifreleyerek cevap ararız.

Polinomial regresyon: Bu model lineer olmayan ve polinom olarak tanımlanabilen veri kümeleri için kullanılır. İşleyişi doğrusal regresyon ile aynıdır.

Karar ağacı: Aynı ağaçlar gibi dallanarak kırılımlar üzerinden ilerleyen bir yapıya sahiptir. İnsan düşünce yapısına benzer bir yapıdadır. Birbirini takip eden if-else yapıları olarak düşünülürse zihinde canlandırmak daha kolay olacaktır.

2.1.1.1. Makine öğrenimi algoritması seçimi

Lojistik regresyon ve diğer makine öğrenimi algoritmaları, veri kümelerinden öğrenme ve tahmin yapma amaçlı kullanılırlar. Ancak, bu algoritmaların uygunluk durumları ve çalışma prensipleri arasında farklılıklar vardır.

Lojistik regresyon, sınıflandırma problemlerine çözüm getirme amacıyla kullanılır. Örneğin, bir şirketin müşterilerini satın alma ihtimalleri yüksek olan ve düşük olan müşteri gruplarına ayırmak istediğinde lojistik regresyon algoritması kullanılabilir. Bu algoritma, verilen girdi değerlerine göre çıktı değerlerini 0 veya 1 olarak tahmin eder. Bu tahminler, müşterilerin satın alma ihtimallerinin yüksek veya düşük olarak sınıflandırılmasına yardımcı olur. Lojistik regresyon, veri kümesindeki örneklerin sınıflandırılması için kullanılırken, diğer makine öğrenimi algoritmaları ise çoğunlukla tahmin yapma amacıyla kullanılır. Örneğin, bir perakende mağazasının gelecekteki satış miktarlarını tahmin etmek istediğinde regresyon algoritmaları kullanılabilir.

Lojistik regresyon ve diğer makine öğrenimi algoritmaları arasında en belirgin fark, çıktı değerlerinin türüdür. Lojistik regresyon algoritması, sınıflandırma problemlerine çözüm getirirken, diğer algoritmalar ise tahmin yapmaya yönelik olarak kullanılır. Lojistik regresyon algoritması, çıktı değerlerini 0 veya 1 olarak tahmin ederken, diğer algoritmalar ise sayısal değerler tahmin eder. Bu nedenle, hangi algoritmanın kullanılacağı problemlerden ve hedeflerden bağımsız olarak seçilmelidir. Bu tasarım projesinde çıktı değeri 0 ve 1 ile ifade edildiğinden bu projede lojistik regresyon tercih edilmiştir.

2.1.1.2. Kaggle

Kaggle, Google tarafından üstlenilen ve yapay zeka, makine öğrenimi ve veri bilimi alanlarında çalışan profesyonel ve hobistler için bir platformdur. Kaggle, işbirliği, öğrenme ve veri bilimi projelerinin oluşturulmasını ve paylaşılmasını hedefleyen bir topluluktur. Platform, çeşitli veri setleri, öğrenme setleri ve modelleri barındıran bir veri deposudur ve bunların kullanımı ve paylaşımı için çeşitli araçlar sunar. Kaggle, ayrıca çeşitli veri bilimi yarışmaları düzenler ve bu yarışmalara katılarak ödül kazanmak mümkündür. [13]

2.1.1.3. Veri seti seçimi

Veri seti seçimi, projenin hedefine ulaşmasındaki önemli adımlardan biridir çünkü makineyi kullandığımız ortamın verilerine ne kadar yakın bir veri setiyle eğitirsek, makinemizden alacağımız tahminler de o kadar gerçeğe yakın ve kararlı olacaktır. Veri seti elde etmek için birden fazla yol vardır. Birincisi, kullandığımız odanın verilerini kaydeden bir devre kurup bu devre zaman tanımaktır. Diğer bir yöntem ise, kaggle gibi hazır veri setleri içeren siteleri inceleyip odamızın sıcaklık değerlerine en yakın veri setlerini bulup, ihtiyacımız olan verilere göre düzenleyerek kullanmaktır. Bu projede, pratiklik ve hızlı çıktı verme amacıyla ikinci yöntem tercih edilmiştir.

2.1.2. DHT11 sıcaklık ve nem sensörü

DHT11 Adafruit firmasının geliştirmiş olduğu sıcaklık ve nem sensörüdür. Bu sensör ile 2°C'lik hata payıyla 0°C-50°C sıcaklık ölçümü ve %5RH'lik hata payıyla %20 – %90 RH nem ölçümü yapılabilir. Çalışma voltajı 3V-5V'dur. Maliyeti düşük sıcaklık sensörlerinin başında gelir. [6]

2.1.3. Tek kanallı 5V röle

Röle, basit bir açma/kapama anahtarıdır. Giriş voltajı 5 Volt olduğunda çalışır ve 0 Volt olduğunda kapanır. Aynı zamanda, röle izolasyon sağlayan elektromekanik bir komponenttir. Kontrol (geliştirme) kartları ve shield'ın bağlı olduğu düşük güç

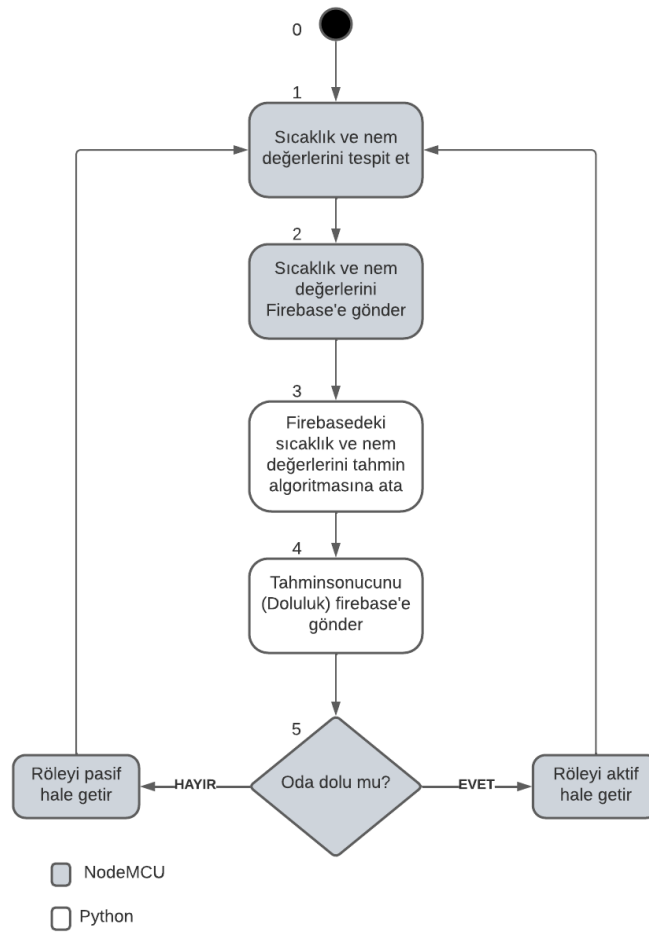
tarafıyla, röle modülünün çalıştığı yüksek güç tarafında röle yoluyla bağlantı koparılmıştır. Bu şekilde yüksek voltaj/akım koruması sağlanmış olur. Röle lojik 0 (0V) ile tetiklenir. Ürün üzerinde 3 bağlantı terminali mevcuttur: COM (Ortak uç), NC (Normalde Kapalı), NO (Normalde Açık). Röle, 30VDC veya 220VAC gerilimde 10A'a kadar akımı anahtarlayabilmektedir. [5]

2.1.4. Firebase

Firebase Google tarafından geliştirilen bir NoSQL veritabanıdır. Kullanımı belli bir veri kullanım limitine kadar ücretsizdir. En önemli özelliklerinden birisi Realtime Database'e sahip olmasıdır. Diğer veritabanlarına göre hız konusunda öne geçmektedir.

BÖLÜM 3. GELİŞTİRİLEN SİSTEM

3.1. Akış Diyagramı

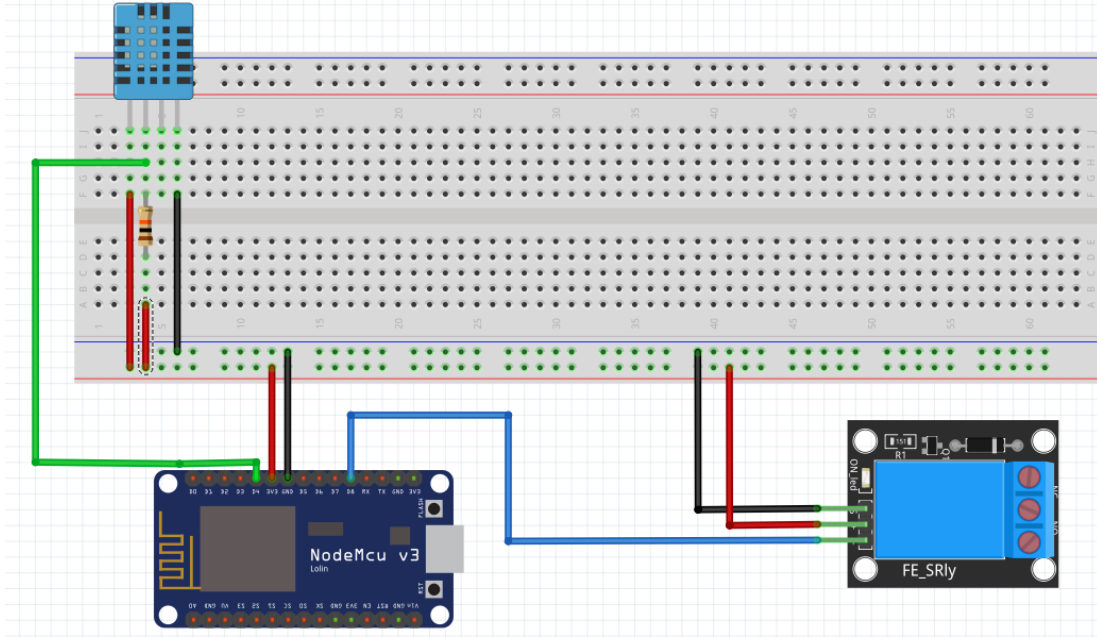


Şekil 3.1 Sistemin akış diyagramı

Bu düzenekte, adım 1'de DHT11 sensörü kullanılarak sıcaklık ve nem ölçülür. Adım 2'de, NodeMCU kullanılarak ölçülen sıcaklık ve nem değerleri firebase veritabanına aktarılır. Adım 3'te, eğitilmiş makineye oda sıcaklığı değeri atanır. Adım 4'te, makine verilen değerler doğrultusunda oda doluluk durumunu tahmin edip firebase'e gönderir.

Adım 5'te ise, NodeMCU tarafından firebase'den çekilen doluluk değeri kullanılarak röle durumu belirlenir ve oda doluluk veya boşluk durumuna göre ayarlanır.

3.2. Devre Şeması



Şekil 3.2: Devre düzeneğinin Fritzing programı ile oluşturulmuş şeması

Devrede, röle ve DHT11 sensörünün VCC ve GND ayakları ortak bir pin üzerinden bağlıdır. Rölenin output pini D8 portuna, DHT11 sensörünün input pini D4 portuna bağlanmıştır. DHT11 sensörünün çalışması sırasında, pull-up direnç gereklidir. Bu nedenle, 10KOHM direnç input ve VCC arasına bağlanarak bu işlem gerçekleştirilir.

3.3. Yazılım Gereksinim Listesi

Projeyi geliştirirken sahip olunması gereken yazılımlar ve kütüphaneler şunlardır;

Python v3.9.0

Pandas kütüphanesi v1.2.4

Numpy kütüphanesi v1.23.4

Matplotlib kütüphanesi v3.6.2

Arduino IDE v2.3.0

NodeMCU ESP8266 sürücüsü v2.7.4

Adafruit DHT sensör kütüphanesi v1.4.3

Adafruit JSON kütüphanesi v5.13.1

FirebaseArduino kütüphanesi [4]

Firebase NoSQL Veritabanı

3.4. Veritabanı Kavramsal Modeli

Room
Humadity: Int
Temperature: Int
Occupancy: Int

Tablo 3.1: Veritabanı Kavramsal Modeli

Projede kullanılan üç değişkenden Humadity ve Temperature DHT11 sensörü tarafından ölçülüp veri tabanına gönderiliyor. Bu verileri makine alıp bir tahminde bulunuyor ve tahmin sonucunu Occupancy değişkenine atıyor. Son olarak röle Occupancy değerine göre kendini açıp kapatır.

BÖLÜM 4. ENDÜSTRİYEL BOYUT VE KISITLAR

4.1. Donanım Maliyet Tablosu

Tablo 4.1: Donanım Maliyet Tablosu

Maliyet Kalemi	Tutar (TL)
NodeMCU V3 ESP8266 ESP-12E Geliştirme Kartı	73,36
DHT11 Sıcaklık ve Nem Sensörü	26,45
Tek Kanallı 5V Röle Modülü	22,42
Bağlantı Kabloları	10,00
Breadboard	35,00
10 KOHM Direnç	1,00
TOPLAM	168,23

4.2. İş Zaman Çizelgesi ve Maliyet Analizi

Tablo 4.2 İş Zaman Çizelgesi Maliyet Analizi

Etkinlikler	Süre (hafta)	Maliyet (TL)
Proje Öncesi Araştırma	3	6000
Ön Tasarım	1	2000
Detaylı Tasarım	2	4000
Prototip Oluşturma	2	4168,23
Testler	2	4000
TOPLAM	10	20168.23

4.3. Donanım İçin Gerekli Enerji Analizi

Oluşturduğumuz sistem 1 saniyede 96 mA akım tüketiyor ve gerilim değeri 3.3V, günde 24 saat çalıştığında, günde 86400 saniye çalışır ve bu sürede $86400 \times 96 = 8294400$ mA güç tüketir. Bu değer aylık olarak hesaplandığında, 30 günlük bir ayda $8294400 \times 30 = 248832000$ mA güç tüketir. Bu değeri amper (A) cinsinden ifade etmek için, $248832000/1000 = 248.832$ A güç tüketir. Bu değer watt cinsinden ifadesi ise $P=V \times I$ şeklinde hesaplanır. Bu bilgiye göre, 248.832 A güç tüketen ve 3.3 Volt olan bir sistemin güç tüketimi $P=3.3 \times 248.832 = 819.986$ watt'tır. Bu değer aylık olarak hesaplandığında, 30 günlük bir ayda $819.986 \times 30 = 24.59958$ watt-saat güç tüketir. Bu değeri kilowatt-saat (kWh) cinsinden ifade etmek istersek, $24.59958/1000 = 0.02460058$ kWh güç tüketir. Türkiyede 01.01.2023'te elektrik enerjisi fiyatı vergiler dahil 1.7346 TL/kWh'dir. [11] Kullanılan güç değeri ile fiyat değerini çarparsak devremizin 1 ayda tükettiği elektrik maliyetini bulabiliriz ve bu değer yaklaşık 0.042 TL'dir.

4.4. Veritabanı Maliyet Analizi

Sistemimiz her saniye veri gönderip çekmekte ve bu değer 1 dk'da yaklaşık olarak 1 megabyte'a ulaşmaktadır. Bu verilerden yola çıkarak 1 günlük veri tüketimi 1.4 Gigabyte olur. 1 ayda tüketilen veri boyutu ise $1.4 \times 30 = 42$ Gigabyte'tır. Firebase veri tabanı aylık 10 Gigabyte'a kadar ücretsiz kullanım sunmaktadır ve sistemin kullandığı veri boyutu bu sınırı geçmektedir. Firebase 10 Gigabyte'ı geçen indirme işlemlerinden 1\$ almaktadır. $42 \times 1\$ = 42\$$ 42\$ dolar kurunu 18,72 olarak hesaplırsak 785 TL yapmaktadır. [3]

BÖLÜM 5. UYGULAMA

Bu bölümde, bölüm 2 ve bölüm 3'de anlatılan teorik ve pratik bilgiler kullanılarak gerçekleştirilecek olan düzeneğimiz oluşturulacaktır.

5.1.1. Tahmin Algoritmasının Hazırlanması

Makine öğrenmesine başlamadan önce, elimizdeki verileri ve bu verilerle tahmin etmek istediğimiz şeyi tanımlamamız gerekiyor. Örneğin, Kaggle üzerinde bulunan bir veri setinde odanın doluluk durumunu ve odada çeşitli sensörler aracılığıyla elde edilen diğer verileri içeren bir veri seti buldum. Ancak, bu veri setinde yer alan bütün verileri uygulama ortamında elde edemiyordum, bu nedenle de sağlıklı bir tahmin yapamıyordum. Bu tasarım projesi için kurduğum devrede, DHT11 sensörü aracılığıyla sadece sıcaklık ve nem verilerine ulaşabiliyordum. Bu bilgileri değerlendirdikten sonra, elimdeki veri setini 'Occupancy', 'Temperature' ve 'Humidity' gibi sütunlar içerecek şekilde yeniden düzenledim. Bu düzenlemenin amacı, odanın doluluk durumunu tespit ederken sıcaklık, nem ve doluluk arasındaki ilişkiyi kullanmak istememdi.

Veri setindeki gereksiz sütunları ortadan kaldırdıktan sonra, veri setimi min-max normalizasyonu aracılığıyla ölçeklendirdim. Min-max normalizasyonu, her özelliğin en küçük değerini 0, en büyük değerini 1 ve arada kalan değerleri de 0 ile 1 arasında bulunan ondalık sayı değerlerine dönüştürür. Veri setini normalize etmek makine öğrenme algoritmalarına sokmadan da mümkün olabilir, ancak normalize edilmemiş bir veri setinde daha geniş aralıklara sahip olunur ve bu durum verileri yakınsamada zorluk çıkartabilir.[1]

Normalizasyon işleminden sonra, veri setimizi denetimli bir şekilde eğiteceğimiz için train ve test olarak ikiye ayırmamız gerekiyor. Test ve train verilerini ayırmak için herhangi bir standart yoktur ve elinizdeki veriye ve sonuçlara göre en uygun oranı

bulmak için deneyim yapılabilir. Ayırdıktan sonra, kullanacağımız makine öğrenme algoritmasını belirlemek ve ona uygun kütüphaneleri çağırmak gerekiyor. Bu problem için ben karar ağacı sınıflandırmasını tercih ettim. Train setlerimizi fit methodu kullanarak makinemizi eğitiyoruz. Burada dikkat edilmesi gereken bir husus var: fit methodu sadece train için kullanılmalıdır, çünkü test kısmı için de kullanılırsa, makinemize istenen sonuçlar hakkında ipucu vermiş olur ve bu da tahminlerimizin tutarlılığını etkiler.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Şekil 5.1: Confusion matrix şeması

Makine öğrenmesi bittiğinde, tahminlerimizin tutarlılığını kontrol etmek için ayırdığımız test veri setini kullanarak tahminlerimizi kontrol edebiliriz. Ben tahminlerimin doğruluğunu kontrol etmek için confusion matrix kullandım. Confusion matrix, bize verilerimiz hakkında bilgi verir. Örneğin, [(+)-değeri (+)-tahmin, (-)-değeri (+)-tahmin] ve [(+)-değeri (-)-tahmin, (-)-değeri (+)-tahmin] gibi. Bizim verilerimizin sonuçları ise şu şekildeydi: [[13317, 2347], [1536, 3155]]. [2]

Bir diğer yöntem ise, x_{test} 'i kullanarak y_{test} 'teki verileri tahmin etmesini istemek ve çıkan sonuçlar ile gerçek sonuçlar arasındaki orana göre bir score oluşturmaktır. Bu tahmin algoritmasının skor değeri 0.80 çıktı. Bu sayede, makinemize verdiğimiz nem ve sıcaklık değerlerini kullanarak odanın boş mu dolu mu olduğunu tahmin edebiliriz.

5.2. Gerekli Sensör ve Donanımların Kurulumu

Bir önceki başlıkta makinemizi eğittik, ancak makinemizin tahmin yapabilmesi ve tahmin sonucuna göre röleyi açık ya da kapalı konuma getirebilmesi için bir düzeneğe

ihtiyacımız var. Devreyi nasıl oluşturduğumuzu Şekil 3.2'de görebilirsiniz. Devre elemanlarımızı şemadaki gibi birbirine bağladıktan sonra, programlama kısmına geçebiliriz. Programlamaya başlamadan önce, NodeMCU'nun 2.7.4 sürümünü kullandığımızı belirtmek istiyoruz. Bunun nedeni, firebase ve DHT11 için kullandığımız kütüphanelerin NodeMCU'nun yeni sürümlerinde hata vermesidir. Arduino IDE'de, ESP8266WIFI isimli Adafruit kütüphanesini include ederek, bu işlemleri tamamladıktan sonra, bir sonraki başlıklara geçerek gerekli sensörlerin ve donanımların kurulumuna başlayabiliriz.

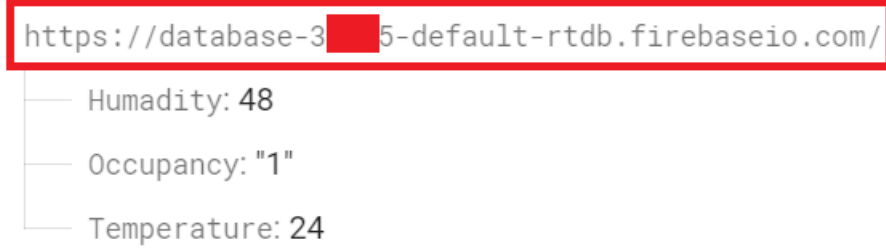
5.2.1. Sıcaklık sensörünün programlanması

Programlama kısmına sıcaklık sensörü DHT11 ile başlamaktayız. Bu sensör için Adafruit'in DHT sensor library 1.4.4 kütüphanesini kullanıyoruz. Kütüphanemizi include ettikten sonra DHT11 üzerindeki veri portunu NodeMCU üzerinde D4 pinine bağlıyor ve DHT11 sensörümüzü tanımlıyoruz. Tanımlamamızı DHT dht (DHTPin, DHTTYPE) şeklinde yapıyoruz. Tanımlamalarımız tamamlandıktan sonra setup fonksiyonunun içinde dht.begin() komutu ile sensörümüzü başlatıyoruz. Bu şekilde setup işlemlerimiz bitiyor. Loop fonksiyonunun içine girdiğimizde dht.readTemperature() ve dht.readHumidity() komutlarıyla sıcaklık ve nem değerlerine ulaşabiliyoruz. Sıcaklık sensörümüzün programlaması bu şekildedir. Bu işlemler tamamlandıktan sonra sensörümüz nem ve sıcaklık değerlerini okumaya hazır hale gelir.[12]

5.2.2. NodeMCU'nun firebase kurulumunun yapılması

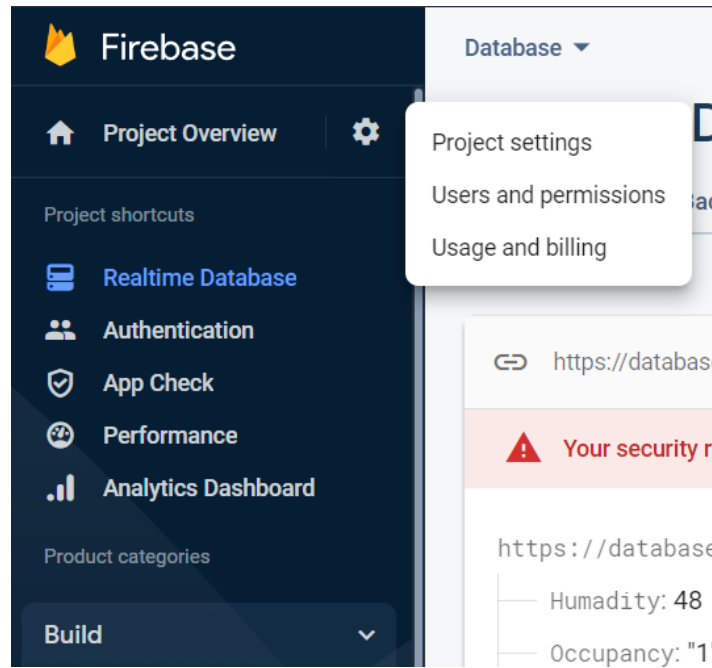
Firebase bağlantısını yapmak için tercih ettiğim kütüphane FirebaseArduino'dur. [4] FirebaseArduino kütüphanesini tercih etmemin nedenleri, diğer kütüphanelerde kullanıcı doğrulama ve kod karmaşası gibi sorunların ortaya çıkma riskinin olmaması ve NodeMCU ile birlikte kullanımda sorun yaşanma riskinin az olmasıdır. Bu kütüphane, kullanımı kolay ve açıklayıcı dökümantasyona sahiptir. Kütüphaneyi include ettikten sonra, Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH) fonksiyonunu kullanarak Firebase bağlantısını başlatabilirsiniz. Firebase bağlantısını başlatmak için Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH)

fonksiyonunu kullanın. Bağlantıyı başlatabilmek için FIREBASE_HOST ve FIREBASE_AUTH parametrelerine ihtiyacınız olacak.

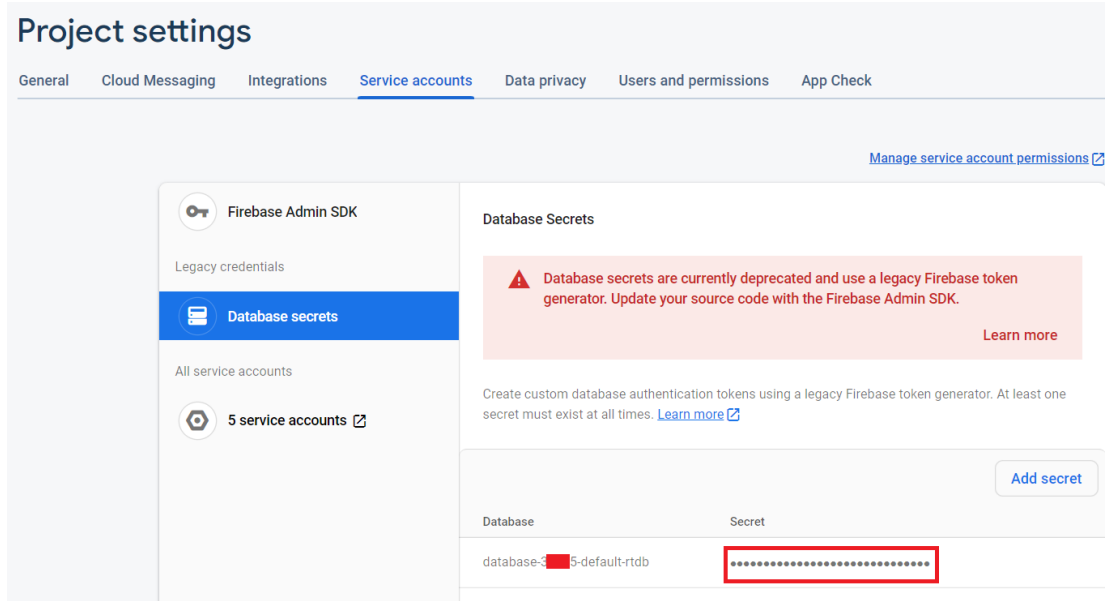


Şekil 5.2: Firebase veritabanı tablosu

FIREBASE_HOST parametresi, Firebase Realtime Database'inizi görüntülerken gördüğünüz URL adresinin (`https://` dahil değil) yazılmış halidir.



Şekil 5.3: Firebase Proje ayarları sekmesi



Şekil 5.4: Proje ayarlarında doğrulama şifresinin konumu

FIREBASE_AUTH parametresine ulaşmak için, Firebase üzerinde sol üstteki Ayarlar sekmesinden "Project Settings'e gidin. Açılan sayfada, "Service Accounts"a tıklayın ve sol tarafta çıkan "Database secrets" seçeneğine tıklayın. Açılan sayfada bulunan tablonun sol tarafında veri tabanı isimlerinizi, sağ tarafında da FIREBASE_AUTH parametresini göreceksiniz. Bu adımları tamamladıktan sonra, Arduino IDE içerisinde gerekli yerlere bu parametreleri yazın.

Firebase bağlantımız tamamlandıktan sonra, sensörlerimizden okuduğumuz verileri Firebase'e post etmemiz gerekiyor. Sensörlerimizin kurulumunda kullandığımız `dht.readTemperature()` ve `dht.readHumidity()` komutlarından gelen verileri `t` ve `h` değişkenlerine atayın. Son olarak, `Firebase.setFloat("Temperature", t)` ve `Firebase.setFloat("Humadity", h)` komutlarını kullanarak Firebase'e veri post işlemini yapın. Bu iki komut, Firebase içerisinde "Temperature" ve "Humadity" değişkenlerini oluşturup içlerine `t` ve `h` değerlerini atayarak sensör verilerimizin Firebase'e atanmasını sağlar. [3] [4]

5.2.3. Makine ve firebase bağlantısı

Bu aşamada yapmayı amaçladığımız şey, sensör tarafından Firebase'e yollanan verileri kullanarak bir tahmin yapmak ve tahmin sonucunu Firebase'e geri göndermektir. Bunun için, `from firebase import firebase` [4] diyerek kütüphanemizi ekleyin. Kütüphaneyi ekledikten sonra, `FirebaseApplication` fonksiyonu ile Firebase'e bağlanın. Fonksiyonun ilk parametresi Firebase host'udur, ikinci parametresi ise doğrulama tipidir. İkinci parametreyi `None` yapın ve bu fonksiyonu bir değişkene atayın. Böylece Firebase bağlantımızı tamamlamış olursunuz. Bu aşamadan sonraki işlemlerde, fonksiyonlara değişkene atadığımız fonksiyon üzerinden ulaşacağız.

Verilerimizi veri tabanından çekmek için `get` fonksiyonunu kullanın. `get` fonksiyonunun ilk parametresine, parametre yolunuzu yazın (örneğin: `/humadity`). İkinci parametreye ise `None` girin. Bunun nedeni, başlık veya bağlantı ile ilgili bir ihtiyacınızın olmamasıdır. Bu yöntemle, sıcaklık ve nem verilerini çekerek değişkenlere atayın.

Makinemize tahmin yaptırabilmemiz için, verdiğimiz inputun iki boyutlu bir dataframe olması gerekiyor. Bu yüzden, aldığımız verileri bu şekilde düzenleyin. Tahmin işlemini yapmadan önce, yapmamız gereken son şey, veri tabanından çektiğimiz verileri ölçeklemek olacak. Ölçekleme işlemini, elimizdeki data setini esas alarak yapın. Artık input verimiz tamamen hazır. Verilerimizi makineye verin ve bir sonuç elde edin. Sonucunuzu `fb_app.put("/", "Occupancy", str(sonuc[0]))` kullanarak veri tabanımıza post edin. Böylece makine tarafındaki Firebase işlemlerimizi tamamlamış olursunuz. [3][4]

5.2.4. Tek kanallı 5V rölenin programlanması

Rölenin programlamasına başlamak için, öncelikle röleyi bağladığımız pinin tanımlanması gerekmektedir. Ben, röleyi D8 piniyle bağladım. Tanımlama işlemini, sıcaklık sensöründe olduğu gibi `pinMode(role, OUTPUT)` komutunu kullanarak ya da `#define role D8` şeklinde de yapabilirsiniz. Tanımlama işlemini tamamladıktan sonra, `setup` içerisinde `pinMode(role, OUTPUT)` komutunu kullanarak pinin nasıl

kullanılacağını belirtin. Bu işlemlerden sonra, röleyi açıp kapatmak için `digitalWrite(role, LOW)` ve `digitalWrite(role, HIGH)` komutlarını kullanın. Bu işlemler tamamlandıktan sonra, röleniz hazır olacaktır.

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Bu çalışmada, Internet of Things (IoT) ve makine öğrenimi tabanlı bir Internet of Behaviors (IoB) sistemi tasarlandı. Bu tasarımın amacı, bir odanın dakika dakika boş olup olmadığının, sıcaklık ve nem değerlerinin tutulan bir veri seti üzerinden makine öğrenimi algoritmaları aracılığıyla makineye öğretilerek gelecekteki kullanımlarda odanın dolu veya boş olup olmadığının tahmin edilmesidir. Tahmin durumuna göre, odanın ısıtılması için kullanılan klima veya kombinin bir röle ile kapatılıp açılması mantığıyla, ısıtma sisteminin gereksiz kullanımı önlenmeye çalışılmıştır.

Projede kullanılan DHT11 sensörü, düşük maliyeti ve kullanım kolaylığı nedeniyle tercih edilirken, bu sensörün rakiplerine göre daha yüksek hata payına sahip olması nedeniyle dezavantaj sağlamaktadır. NoSQL veritabanı kullanımı ve JSON formatı üzerinden iletişim kurulması, projenin oluşturulmasını kolaylaştırmıştır. NodeMCU ESP8266 devre kartının içinde wifi modülü bulundurması nedeniyle ekonomik bir seçim olurken, 3.3V çıkış desteği nedeniyle sensör seçiminde özgürlüğü kısıtlamaktadır. Devre kartının fazla pin desteği ise çeşitli sensörlerin bağlanmasını kolaylaştırmaktadır.

	True Positive	True Negative
Predicted Positive	13317	2347
Predicted Negative	1536	3155

Şekil 6.1: Confusion matrix görseli

Measure	Value	Derivations
Sensitivity	0.8966	$TPR = TP / (TP + FN)$
Specificity	0.5734	$SPC = TN / (FP + TN)$
Precision	0.8502	$PPV = TP / (TP + FP)$
Negative Predictive Value	0.6726	$NPV = TN / (TN + FN)$
False Positive Rate	0.4266	$FPR = FP / (FP + TN)$
False Discovery Rate	0.1498	$FDR = FP / (FP + TP)$
False Negative Rate	0.1034	$FNR = FN / (FN + TP)$
Accuracy	0.8092	$ACC = (TP + TN) / (P + N)$
F1 Score	0.8728	$F1 = 2TP / (2TP + FP + FN)$
Matthews Correlation Coefficient	0.4957	$TP*TN - FP*FN / \sqrt{((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))}$

Şekil 6.2: Confusion matrix hesaplamaları

Bu projede, eğitilen makinenin tahmin skoru %80.92 olarak belirlenmiş ve bu tahminlerin doğruluğu hakkında bir Confusion matrix oluşturulmuştur. Confusion matrix, sistemin doğru ve yanlış sınıflandırma oranlarını gösterir ve Sensitivity, Accuracy, Negative Predictive Value, False Positive Rate, False Discovery Rate, False Negative Rate, ve F1 Score gibi ölçütleri kullanarak sistemin performansını ölçer. Bu çalışmada, makine öğreniminin gerçekleştiği ve ezber yapılmadığı sonucuna varılmıştır.

Projede kullanılan veri setinin hazır alınmış olması projenin pratikte uygulanabilirliğini zorlaştırabilir. Bu, çünkü veri seti seçilirken kendi odanızın değerlerine yakın veriler seçmeye çalışsanız bile, makine önceden hazır veri setine göre eğitildiğinden, kendi odanızın değerlerine göre tahmin yapmaya çalıştığınızda tahminlerin tutarsızlık göstermesine neden olabilir. Bu nedenle, tahminlerinizin odamıza yüksek uyumluluk sağlaması için, verileri kendiniz oluşturmanız daha sağlıklı olacaktır. Bu sayede, veri seti tamamen kendi odanızın değerlerine göre oluşturulmuş olacak ve makine de bu veriler üzerinden eğitilerek, odanızın değerlerine göre daha doğru tahminler yapabilecektir.

Internet of Behaviors (IoB) teknolojisi, insanların internet kullanımındaki davranışlarını analiz ederek, bu verileri kullanarak kişiselleştirilmiş hizmetler sunmayı amaçlar. Bu teknolojinin kullanım alanları arasında pazarlama, sağlık, eğitim ve sosyal medya gibi alanlar bulunmaktadır. Ancak, IoB projelerinin de bazı zaafları vardır. Bunların başında gizlilik konusu gelir. Bu projenin çalışabilmesi için kullanıcıların odalarındaki sıcaklık ve nem değerleri kayıt altına alınır ve odalarının boş olup olmadığı tespit edilir. Bu veriler, eğer bir hırsız tarafından ele geçirilirse, evinizin soyulması için gerekli bilgileri sağlamaya yardımcı olabilir. Bir diğer zaaf ise, kullanıcıların reklamlara maruz kalabileceğidir. Örneğin, proje sayesinde tespit edilebilecek verilere göre, kullanıcıların hangi tarih aralığında hangi odalarda ne tür koşulları tercih ettiği biliniyorsa, kişiselleştirilmiş reklamlar sunulabilir. Bu durumda, ihtiyacınız olmadığı halde, ısıtma çözümleri satın almayı tercih edebilirsiniz. Bir diğer zaaf ise, verilerin güvenilirliğidir. Bu projenin başarılı olabilmesi için, kullanılan verilerin tutarlılığı çok önemlidir, ancak IoT sistemlerinde kullanılan sensörler ortam

řartlarından etkilenmektedir. sensörler hava akışının daha yoğun olduđu yerlere yerleştirilirse sıcaklık ve nem değeri odanın ortalama değeriinden farklı olabilir. Sensörler en optimum konumda yerleştirildiğinde bile, sensörlerin sıcaklığı devre kartının sıcaklığından etkilenebilir.

Bu konuları çözebilmek için verilerin şifrelenmesi, yetkilendirme sistemlerinin kullanımı ve sensörlerin doğru kalibrasyonu gibi yöntemler kullanılabilir. Bu sayede, insanların günlük hayatlarındaki davranışlarını inceleyen IoB projelerinin güvenilirliği artar ve müşterilerin gizliliği korunur.

KAYNAKLAR

- [1] https://tr.wikipedia.org/wiki/Makine_%C3%B6%C4%9Frenimi Erişim Tarihi: 10.12.2022
- [2] https://en.wikipedia.org/wiki/Confusion_matrix, Erişim Tarihi: 17.12.2022
- [3] <https://firebase.google.com/>, Erişim Tarihi: 02.12.2022
- [4] <https://github.com/ozgur/python-firebase>, Erişim Tarihi 02.12.2022
- [5] <https://maker.robotistan.com/role-nedir/>, Erişim Tarihi 03.12.2022
- [6] <https://components101.com/sensors/dht11-temperature-sensor>, Erişim Tarihi: 6.12.2022
- [7]<https://www.techfunnel.com/information-technology/internet-of-behaviors/>, Erişim Tarihi: 20.12.2022
- [8] “What is IoT Data Collection?” <https://www.enertiv.com/resources/faq/what-is-iot-data-collection> Erişim Tarihi: 18.12.2022
- [9]<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9702450> Erişim Tarihi 20.12.2022
- [10] <https://softtek.eu/en/tech-magazine-en/user-experience-en/what-is-the-internet-of-behaviour-iob-and-why-is-it-the-future/> Erişim Tarihi 20.12.2022
- [11] <https://enerji.gov.tr/bilgi-merkezi-enerji> Erişim Tarihi :01.09.2022
- [12] <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/> Erişim Tarihi 20.12.2022
- [13] kaggle.com Erişim Tarihi: 20.12.2022

ÖZGEÇMİŞ

Arslanca Sarıkaya, 16 Ağustos 1999 yılında Ordu Merkez’de doğmuştur. İlk, Orta ve lise eğitimini Ordu’da tamamlamıştır. Üniversite eğitimini, Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nde devam ettirmektedir.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : IoB Destekli Isıtma Sistemi Otomasyonu

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): B181210052/ ARSLANCAN SARIKAYA

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problem tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN : PROF. DR. CÜNEYT BAYILMIŞ

DANIŞMAN İMZASI: