



T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü



**DEMİRYOLLARI İÇİN NESNELERİN İNTERNETİ İLE
UZAKTAN İZLENEBİLEN RAY SICAKLIĞI VE TREN HIZI
ÖLÇÜM SİSTEMİ TASARIMI VE METEOROLOJİ İLE SAHA
VERİSİ ÜZERİNDEN MAKİNE ÖĞRENMESİ/DERİN
ÖĞRENME İLE ZAMANA DAYALI RAY SICAKLIĞI
TAHMİNİ**

Yüksek Lisans Tezi

Eren ERDİ

Makine Mühendisliği Anabilim Dalı

İzmir
2022

T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

**DEMİRYOLLARI İÇİN NESNELERİN İNTERNETİ İLE
UZAKTAN İZLENEBİLEN RAY SICAKLIĞI VE TREN HIZI
ÖLÇÜM SİSTEMİ TASARIMI VE METEOROLOJİ İLE SAHA
VERİSİ ÜZERİNDEN MAKİNE ÖĞRENMESİ/DERİN
ÖĞRENME İLE ZAMANA DAYALI RAY SICAKLIĞI
TAHMİNİ**

Eren ERDİ

Danışman : Doç. Dr. Mustafa Turhan ÇOBAN

Makine Mühendisliği Anabilim Dalı
Makine Mühendisliği Yüksek Lisans Programı

İzmir
2022

KABUL ONAY SAYFASI

Eren Erdi tarafından yüksek lisans tezi olarak sunulan “Demiryolları İçin Nesnelerin İnterneti İle Uzaktan İzlenebilen Ray Sıcaklığı Ve Tren Hızı Ölçüm Sistemi Tasarımı ve Meteoroloji Ve Saha Verisi Üzerinden Makine Öğrenmesi/Derin Öğrenme İle Zamana Dayalı Ray Sıcaklığı Tahmini” başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesinin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 25/02/2022 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı :

Raportör Üye :

Üye :

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Demiryolları için nesnelerin interneti ile uzaktan izlenebilen ray sıcaklığı ve tren hızı ölçüm sistemi tasarımı ve meteoroloji ile saha verisi üzerinden makine öğrenmesi/derin öğrenme ile zamana dayalı ray sıcaklığı tahmini” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

25 / 02 / 2022

İmzası

Adı-Soyadı

ÖZET

DEMİRYOLLARI İÇİN NESNELERİN İNTERNETİ İLE UZAKTAN İZLENEBİLEN RAY SICAKLIĞI VE TREN HIZI ÖLÇÜM SİSTEMİ TASARIMI VE METEOROLOJİ İLE SAHA VERİSİ ÜZERİNDEN MAKİNE ÖĞRENMESİ/DERİN ÖĞRENME İLE ZAMANA DAYALI RAY SICAKLIĞI TAHMİNİ

ERDİ, Eren

Yüksek Lisans Tezi, Makine Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Mustafa Turhan ÇOBAN

Şubat 2022, 137 sayfa

Bu tezde belirlenen bir demiryolu hattında ray sıcaklıklarının değişimleri ve seyir halindeki trenlerin hızları nesnelerinin interneti ile uzaktan izlenmiş ve kayıt altına alınan veriler ile makine öğrenmesi/derin öğrenme kullanılarak zamana dayalı ray sıcaklığı tahmini yapan bir sistem geliştirilmiştir.

Arka uçlar, ön uç yazılımları ve veri tabanı bulut üzerinde çalışırken, sensör sinyallerini işleyerek ölçümleri gerçekleştiren yazılım mikro kontrolör üzerinde çalıştırılmıştır. Bulut ile mikro kontrolör arasındaki haberleşmeyi sağlayan yazılım ise uç cihaz üzerine kodlanmıştır.

Sensörler, sinyal dönüştürücüler, güç kaynağı, mikro kontrolör ve uç cihazından oluşan donanım, demiryolu hattı üzerine konuşlandırılarak saha verileri elde edilmiştir. Bu veriler, internet üzerinden alınan meteoroloji verileri ile birlikte yeni nesil haberleşme protokollerinden vasıtasıyla eş zamanlı olarak bir veri tabanına kaydedilmiş ve bir web sitesinde canlı olarak gösterilmiştir.

Bu veriler, belirlenen makine öğrenmesi/derin öğrenme algoritmalarına yüklenerek hedeflenen kriterlere göre algoritma performansları karşılaştırılmış ve uygun algoritma seçilerek web sitesi üzerinden zamana dayalı ray sıcaklığı tahmin edilmiştir.

Anahtar sözcükler: Demiryolları, ray sıcaklığı, tren hızı, nesnelerin interneti, makine öğrenmesi, derin öğrenme, uzaktan izleme, mikro servis mimarisi

ABSTRACT

DESIGNING TRAIN SPEED AND RAIL TEMPERATURE MEASUREMENT SYSTEM WITH REMOTE MONITORING BY INTERNET OF THINGS AND TIME BASED PREDICTION OF RAIL TEMPERATURE WITH MACHINE LEARNING/DEEP LEARNING

ERDİ, Eren

MSc in Mechanical Eng.

Supervisor: Assoc. Prof. Dr. Mustafa Turhan ÇOBAN

February 2022, 137 pages

In this thesis, the changes of rail temperatures and the speeds of trains were determined and monitored remotely with internet of things and an algorithm that predicts the rail temperature based on time using machine learning / deep learning was developed.

While the back-ends, the front-end software and the database was running on the cloud, the software that processes sensor signals and performs measurements was run on the microcontroller. The software that provides communication between the cloud and the micro controller is coded on the edge device.

The hardware consisting of sensors, signal converters, power supply, micro controller and edge device was deployed on the railway line and field data was obtained. These data, together with meteorological data received over the internet, were simultaneously recorded in a database through new generation communication protocols and displayed live on a website.

These data were loaded into the selected machine learning / deep learning algorithms and the algorithm performances were compared according to the targeted criteria and the rail temperature was predicted on the website by selecting the appropriate algorithm.

Keywords: Railways, rail temperature, train speed, internet of things, machine learning, deep learning, remote monitoring, microservice architecture

ÖNSÖZ

Demiryollarının ihtiyaçlarını mümkün mertebe karşılayacak dijital makineler üretme fikri ile yola çıktım. Bu fikir doğrultusunda uygulamaya koyduğum tez projemin somut hale gelişi sabır, azim, disiplin ve yüksek motivasyon gerektiren bir süreçti.

Gerek tasarım gerekse saha çalışmaları süreçlerinde karşılaşılan her sorunun çözümünün kazanılacak yeni bir tecrübe olduğunun bilinciyle hareket ederek detaylı bir yol haritası çizdim. Karşılaşılan problemlerle ilk andan itibaren mücadele ettim.

Bu proje benim için öğrenmeye olan isteği sürekli körüklemenin ve üretkenliğin yapı taşı olan planlı çalışmanın mükâfatını aldığım bir çalışma oldu. Bu projede elde edilen tecrübeler demiryollarının dijital ihtiyaçlarını karşılamak için çıkılan yolda sonraki projeler için önemli bir basamak oluşturmaktadır.

İZMİR

25/02/2022

Eren ERDİ

İÇİNDEKİLER**Sayfa**

İÇ KAPAK.....	ii
ETİK KURALLARA UYGUNLUK BEYANI	v
ÖZET	vii
ABSTRACT.....	ix
ÖNSÖZ	xi
İÇİNDEKİLER	xii
ŞEKİLLER DİZİNİ.....	xv
ŞEKİLLER DİZİNİ (devam)	xvi
ŞEKİLLER DİZİNİ (devam)	xvii
TABLolar DİZİNİ	xviii
1. GİRİŞ.....	1
2. DEMİRYOLU ÜST YAPISI.....	2
2.1. Balast.....	2
2.2. Travers.....	3
2.3. Ray.....	5
2.4. Bağlantı Elemanları.....	8
3. HIZ VE RAY SICAKLIĞINA DAİR SORUNLAR VE İHTİYAÇLAR.....	10

İÇİNDEKİLER(devam)

	Sayfa
3.1. Demiryollarında Hız Limiti	11
3.2. Ray Sıcaklığı Etkileri	12
3.3. İhtiyaçlar ve Faydalar	16
4. SİSTEMİN TASARIMI	17
4.1. Fiziksel Model ve Veri Modeli İlişkisi	17
4.1.1. Hız Tayininin Fiziksel Modeli ve Veri Modeli İlişkisi	17
4.1.2. Ray Sıcaklığı Tahminin Fiziksel Modeli ve Veri Modeli İlişkisi	19
4.2. Sistem Mimarisi	25
4.2.1. Otomasyon	25
4.2.2. Dijitalizasyon	40
4.3. Ray Sıcaklığı Tahmin Algoritması	54
4.3.1. ANN Algoritması	55
4.3.2. SVM Algoritması	63
4.3.3. XGBOOST Algoritması	65
4.3.4. Algoritmaların Karşılaştırılması	67
5. SUNUCULARIN DEVREYE ALINMASI	69
6. SONUÇ VE ÖNERİLER	70

İÇİNDEKİLER(devam)

Sayfa

KAYNAKLAR DİZİNİ	72
KAYNAKLAR DİZİNİ(devam).....	73
KAYNAKLAR DİZİNİ(devam).....	74
KAYNAKLAR DİZİNİ(devam).....	75
TEŞEKKÜR SAYFASI	76
ÖZGEÇMİŞ	77
EKLER.....	

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 2.1 Balastlı ve balastsız hat	2
Şekil 2.2 Bazalt yapıdaki balast.....	3
Şekil 2.3 Demiryolları üst yapısında kullanılan traversler	4
Şekil 2.4 Vignole tipi ray profili 60E1	6
Şekil 2.5 Yivli ray profili 59Ri2.....	6
Şekil 2.6 Muhtelif ray profilleri.....	7
Şekil 2.7 Gergi kısıkaçı ve tirfon.....	9
Şekil 2.8 Cebire ile yapılan ray bağlantısı.....	10
Şekil 3.1 Derayman	11
Şekil 3.2 Demiryolu makası	12
Şekil 3.3 Demiryolunda ray burkulması.....	15
Şekil 3.4 Demiryollarında yaygın yol kusurları	15
Şekil 4.1 Hat üzerinde endüktif sensör yerleşimi	18
Şekil 4.2 Diferansiyel kontrol hacmi.....	19
Şekil 4.3 Ray bloğu parçası	21
Şekil 4.4 Ray bloğu parçasının kesiti	23
Şekil 4.5 Otomasyon sistemi	26

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 4.6 Pano imalat sürecinden bir görüntü	28
Şekil 4.7 Otomasyon sistemi panosu	29
Şekil 4.8 Otomasyon sistemi saha dolabı.....	29
Şekil 4.9 Operatör paneli sistem ekranı	30
Şekil 4.10 Operatör paneli haberleşme ekranı	31
Şekil 4.11 PLC içerisinde çalışan hız ölçümü algoritması.....	33
Şekil 4.12 PLC içerisinde çalışan sıcaklık ölçüm algoritması	34
Şekil 4.13 PLC içerisinde çalışan haberleşme algoritması	35
Şekil 4.14 Uç cihaz algoritması	39
Şekil 4.15 Dijitalizasyon şeması	40
Şekil 4.16 IOT sunucusu algoritması	43
Şekil 4.17 REST sunucu algoritması kaydolma ve oturum açma rotaları	45
Şekil 4.18 REST sunucu algoritması panel ve hava durumu rotaları	46
Şekil 4.19 REST sunucu algoritması tahmin oturum kapatma ve parola değişim rotaları	47
Şekil 4.20 Ön uç algoritması.....	48
Şekil 4.21 Web sitesi ana sayfası	49
Şekil 4.22 Web sitesi kaydolma sayfası.....	50

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 4.23 Web sitesi giriş sayfası	50
Şekil 4.24 Web sitesi parola yenileme sayfası	51
Şekil 4.25 Web sitesi panel sayfası sistem verisi	51
Şekil 4.26 Web sitesi panel sayfası hava koşulları verisi	52
Şekil 4.27 Web sitesi panel sayfası ray sıcaklığı grafiği	52
Şekil 4.28 Web sitesi panel sayfası araç hızı grafiği	53
Şekil 4.29 Web sitesi tahmin sayfası	54
Şekil 4.30 Algılayıcının yapısı	56
Şekil 4.31 Çok katmanlı yapay sinir ağları örnek şeması.....	62
Şekil 4.32 Destek vektör mekanizması	63
Şekil 4.33 SVR algoritması örneklem hata tayini	64
Şekil 5.1 Docker konteynırlaştırma yapısı	69

TABLolar DİZİNİ

Tablo 4.1 Hız tayini veri modeli	18
Tablo 4.2 Ray sıcaklığı tayini veri modeli	24
Tablo 4.3 Otomasyon sisteminde üretilen veri yapısı	35
Tablo 4.4 OSI referans modeli	36
Tablo 4.5 İlişkisel ve ilişkisel olmayan veri tabanlarının hız karşılaştırması	41
Tablo 4.6 MongoDB ve PostgreSQL performans karşılaştırması	42
Tablo 4.7 Veri tabanına yazılan sistem verisi	44
Tablo 4.8 Yapay zeka algoritmalarına tabi tutulan veri yapısı	55
Tablo 4.9 Yaygın olarak kullanılan aktivasyon fonksiyonları	58
Tablo 4.10 Ray sıcaklığı tahmin algoritmalarının performansları	68

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
A_s	Yüzey alanı (m^2)
$A_{s,h}$	Taşınım gerçekleşen yüzey alanı (m^2)
Bi	Biot sayısı
C	SVR Regularizasyon Parametresi
C_p	Özgül ısı (J/kgK)
d_x	x yönündeki diferansiyel değişim
d_y	y yönündeki diferansiyel değişim
d_z	z yönündeki diferansiyel değişim
e	Destek vektör aralığı
$\dot{E}_g$	Birim zamanda giren enerji (Watt)
$\dot{E}_h$	Birim zamanda taşınım ile yayılan enerji (Watt)
$\dot{E}_{out}$	Birim zamanda çıkan enerji (Watt)
$\dot{E}_{st}$	Birim zamanda depolanan enerji (Watt)
$\dot{E}_r$	Birim zamanda ışınlama ile yayılan enerji (Watt)
h	Isı taşınım katsayısı (W/m^2K)
k	Isı iletim katsayısı (W/mK)

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Simgeler</u>	<u>Açıklama</u>
L	Kayıp fonksiyonu
L_c	Karakteristik uzunluk (m)
M	GD momentum fonksiyonu
Nu	Nusselt sayısı
q_x	Birim zamanda x yönünde iletilen ısı (Watt)
q_y	Birim zamanda y yönünde iletilen ısı (Watt)
q_z	Birim zamanda z yönünde iletilen ısı (Watt)
Re	Reynolds sayısı
R^2	Regresyon doğruluk değeri
t	Zaman (saniye)
T	Sıcaklık (Kelvin)
$T_{\infty,h}$	Taşınım sıcaklığı (Kelvin)
$T_{\infty,r}$	Işınım sıcaklığı (Kelvin)
V	Hacim (m ³)
w	Ağırlık katsayıları
X_i	Örneklem girdi değerleri
y_i	Örneklem çıktı değerleri

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Simgeler</u>	<u>Açıklama</u>
$\hat{y}_i$	Algoritma çıktı değerleri
x,y,z	Kartezyen koordinatlar
ϵ	Işınım yayma katsayısı
σ	Stephan-Boltzmann sabiti ($W/m^{-2}.K^{-4}$)
α	Isı yayılım katsayısı (m^2/s)
ρ	Yoğunluk (kg/m^3)
η	Algoritma öğrenme değeri
ξ	Hata değeri
λ	Regularizasyon Parametresi
γ	Dal budama parametresi
$\square$	Yaprak sayısı
<u>Kısaltmalar</u>	<u>Açıklama</u>
ACID	Bütüncüllük Tutarlılık Yalıtma Dayanım(Atomicity Consistency Isolation Durability)
ADAM	Uyarlanabilir Momentum(Adaptive Momentum)
ANN	Yapay Sinir Ağları(Artificial Neural Networks)
API	Uygulama Programlama(Arayüzü Application Programming Interface)

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
ASCII	Amerikan Standart Kodları (American Standard Code for Information Interchange)
CPU	Merkezi İşlem Birimi(Central Process Unit)
EWMA	Üstel Ağırlıklı Hareketli Ortalama(Exponetially Weighed Moving Average)
GD	Eğim İniş(Gradient Descent)
HDLC	Yüksek Seviye Veri Bağlantı Kontrolü(High Level Data Link Control)
HMI	İnsan Makine Arayüzü(Human Machine Interface)
IOT	Nesnelerin İnterneti(Internet of Things)
IP	İnternet Protokol Adresi(Internet Protocol Adress)
IPX	Ağlar Arası Paket Değişimi(Internetwork Package Exchange)
ISO	Uluslararası Standart Kurumu(Internation Standarts Organization)
JSON	Javascript Nesne Gösterimi(Javascript Object Notation)
JWT	JSON Web Anahtarı(JSON Web Token)
LLC	Mantıksal Bağlantı Kontrolü(Logical Link Control)
MAC	Medya Erişim Kontrolü(Media Access Control)
MAE	Ortalama Mutlak Hata(Mean Absolute Error)
MALE	Ortalama Mutlak Logaritmik Hata(Mean Absolute Logaritmik Error)

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
MBGD	Küçük Parçalı Eğitim İniş(Mini Batch Gradient Descent)
MPEG	Hareketli Görüntü Uzman Grubu(Moving Picture Experts Group)
MQTT	Kuyruk Mesaj Transfer Protokolü(Message Queue Telemetry Transport)
MSE	Ortalama Kareler Hatası(Mean Squared Error)
MSLE	Ortalama Kareler Logaritmik Hata(Mean Squared Logaritmik Error)
MSSQL	Microsoft Yapısal Sorgu Dili(Microsoft Structured Query Language)
OSI	Açık Sistem Bağlantıları(Open Systems Interconnection)
PLC	Programlanabilir Mantık Kontrolcüsü(Programmable Logic Controller)
PNP	Pozitif Negatif Pozitif
PPP	Noktadan Noktaya Bağlantı Protokolü(Point to Point Protocol)
QOS	Servis Kalitesi(Quality of Service)
RAM	Rastgele Erişimli Bellek(Random Access Memory)
RBF	Radyal Taban İşlevi(Radial Basis Function)
ReLU	Düzeltilmiş Doğrusal Birim(Rectified Linear Unit)
REST	Temsili Durum Aktarımı(Representational State Transfer)
RMSPROP	Ortalamaların Karekökü Yayılımı(Root Mean Square Propagation)
SGD	Rastgele Eğitim İniş(Stochastic Gradient Descent)

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
SIM	Abone Kimlik Modülü(Subscribe Identity Module)
SMPP	Eşler Arası Kısa Mesaj(Short Message Peer-to-Peer)
SPX	Sıralı Paket Değişimi(Sequenced Packet Exchange)
SQL	Yapısal Sorgu Dili(Structured Query Language)
SS	Benzerlik Skoru(Similarity Score)
SSR	Hataların Kareler Toplamı(Sum of Squared Residuals)
SVM	Destek Vektör Makineleri(Support Vector Machines)
SVR	Destek Vektör Regresyonu(Support Vector Regression)
TCP	İletim Kontrol Protokolü(Transmission Control Protocol)
UDP	Kullanıcı Veri Birimi Protokolü(User Datagram Protocol)
Wi-Fi	Kablosu Bağlantı(Wireless Fidelity)
XGBOOST	Aşırı Gradyan Arttırma (Extreme Gradient Boost)

1. GİRİŞ

Demiryollarının gelişim hızı son yıllarda artış göstermiş. 2008-2019 yılları arasında Türkiye'nin toplam demiryolu hattı uzunluğu 8,699 km'den 10,378 km'ye yükselerek %19,3 artış göstermiştir(Eurostat, 2021). Yine aynı zaman dilimi arasında şehirlerarası yolcu taşımacılığı 5,097 yolcu-km'den 11,068 yolcu-km'ye yükselerek %117,14 artış göstermiştir (Eurostat,2021).

Şehirlerarası taşımacılığın yanında şehir içi raylı sistemler de günümüzün önemli ulaşım yapıları arasındadır ve şehir ulaşım alt yapılarında geniş bir yer kaplamaktadır. 2050 yılı itibariyle dünya nüfusunun yaklaşık %66'sının, Avrupa nüfusunun ise yaklaşık %80'ninin şehirlerde yaşayacağı öngörülmektedir (Pietrzak and Pietrzak, 2019).

Bu düzlemde bakıldığında gelecekte daha sürdürülebilir, daha çevre dostu ve daha verimli bir alt yapı gerekliliği ortaya çıkmaktadır. Akıllı şehirlere olan ihtiyaç da işte tam bu noktada doğmaktadır. Akıllı şehirlere giden süreç, büyük veri, veri analitiği, makine öğrenmesi/derin öğrenme ve nesnelerin interneti gibi çekirdek teknolojiler etrafında şekillenmektedir (Nambiar et al., 2018).

Bu çalışmada demiryolu alt yapısında halihazırda var olan belirli sorunlara söz konusu teknolojiler ile çözümler üretilerek, verimlilik artışı sağlamak, daha güvenli ve daha sürdürülebilir bir alt yapıya katkı sağlamak amaçlanmıştır.

Bu hedefe yönelik olarak sahada sensörleri kontrol edecek mikro kontrolörün algoritması geliştirilip mikro kontrolör üzerine kodlanmıştır. Söz konusu sistemden sahada ayağa kaldırılmıştır. Sistem verisi mikro kontrolör üzerinden endüstriyel protokollerle Linux işletim sistemine sahip bir uç cihaz üzerinden buluta aktarılmıştır. Söz konusu veri anlık meteoroloji verileriyle birleştirilerek bir veri tabanına yazılmıştır. Bir web sitesi aracılığıyla anlık olarak kullanıcıya gösterilmiş ve uzaktan izleme işlevi yerine getirilmiştir. Bulut üzerinde bulunan bir sunucu içerisinde çalışan makine öğrenmesi/derin öğrenme algoritmasıyla aynı web sitesi üzerinden zamana dayalı ray sıcaklığı tahmini yapılmıştır.

Bu çalışma ile demiryollarının kayda değer bir ihtiyacı giderilmiş olup demiryolu hattında düzenli ve istikrarlı bir ölçüm mekanizması kurulmuştur.

2. DEMİRYOLU ÜST YAPISI

Demiryolu alt üst yapısı, demiryolu alt yapısı üzerine inşa edilen ve demiryolu araçlarının üzerinde seyir ederken bu ulaşımın taşıyıcı elemanlarını bünyesinde bulunduran bir yapıdır. Üst yapı doğası gereği farklı eksenlerde büyük yüklere maruz kaldığında içerisinde birbirinden oldukça farklı elemanlar barındırır.

Demiryolu üst yapısı genel olarak balastlı yapı ve balastsız yapı olarak ikiye ayrılır. Balastlı yapı geleneksel demiryolu yapısıdır ve günümüzün en yaygın üst yapı tipidir. Balastsız yapılar ise genellikle beton üzerine inşa edilip genellikle metro gibi şehir içi raylı sistemlerde kullanılır (Fischer et. al., 2015). Şekil 2.1 balastlı ve balastsız hat yapısını göstermektedir.



Şekil 2.1 Balastlı ve balastsız hat (Kaewunruen and Remennikov, 2016)

2.1. Balast

Demiryolu üst yapısının, alt yapının hemen üzerine serilen keskin profile sahip belirli ebatlardaki kırık taşlardır. Balast, hattın elastik bir yapıya kavuşmasını sağlar. Demiryolu hattındaki enerjiyi sönmlediği gibi farklı eksenlerde gelen yüklerin karşılayarak direnç oluşturur. Traverslerden gelen yüklerin alt yapıya homojen olarak iletilmesini sağlar (Fischer et al., 2015).

Yaygın olarak kullanılan balast tipleri bazalt, andrasit, granit veya sert kireç taşıdır. Yüksek sertlikte ve keskin profile sahip kübik kesitlerdir (Fischer et al., 2015). Şekil 2.2, yaygın balast tiplerinden bazalt yapıdaki balastı göstermektedir.



Şekil 2.2 Bazalt yapıdaki balast

Balast ayrıca demiryolu üst yapısında suyun drenajında önemli bir rol oynar. Geçirgenliği sayesinde suyu alt yapıya işleterek üst yapı elemanlarını korur.

2.2. Travers

Traversler, balast üzerine ve rayın altına genellikle 60-65 cm aralıklarla yatay olarak bağlanan elemanlardır. Demiryolu araçlarından gelen ve ray üzerinden aktarılan yatay kuvvetlere karşı mukavemet sağlar. Üst yapıdaki bağlantı elemanlarının yatakarını üzerinde bulundurur. Balastın aksine ray ile doğrudan temas halindedir.

Traversler, yatay yükleri karşılamanın yanında hattaki yüklerin homojen bir şekilde balast üzerine dağılmasını sağlar. Üzerindeki bağlantı elemanlarıyla raya bağlanarak hat geometrisini tutar ve raylar arasında elektrik izolatörü görevi görür. Ayrıca iki ray arasına eğim verilmesine yardımcı olur (Gudmundsson and Gudmundsson, 2014).

Demiryolu hatlarında ahşap, beton ve çelik traversler ile karşılaşılabilir. Bunlardan en yaygın olanları ahşap ve beton traverslerdir. Farklı malzemelerden yapılan traverslerin birbirlerine göre üstünlükleri ve eksiklikleri bulunmaktadır.

Traverslerin elektirik izolasyonu görevi, günümüz demiryollarında yaygınlaşan elektrifikasyon alt yapısı ile oldukça kritik hale gelmiştir. Bu nedenle çelik traverslerin kullanımı giderek ortadan kalkmaktadır. Özellikle izole raylarla beraber beton traversler yaygın olarak kullanılmaktadır.



(a) Beton travers



(b) Ahşap travers



(c) Çelik travers

Şekil 2.3 Demiryolları üst yapısında kullanılan traversler (Bagal et al., 2020)

Şekil 2.3 travers çeşitlerini göstermektedir. Ahşap traverslerin esneklikleri ve elektrik yalıtımları oldukça iyidir. Ayrıca üretimleri kolaydır. Ancak özellikle beton traverslere göre maliyet ve ömür kısıtları vardır. Çelik traverslerin geometrik toleransları oldukça iyidir. Kolay monte edilirler. Ancak maliyet, korozyon riski nedeniyle kısa ömür ve elektrik iletkenliği gibi eksiklikleri mevcuttur. Beton traversler ise yüksek yüklerde çok düşük deformasyona uğrar. Ömürleri uzundur. Bakıma oldukça elverişlidir. Kolay monte edilebilir ve maliyeti düşüktür. Buna karşın esneklikleri ahşap traverslere göre düşük olduğundan balasta daha yüksek gerilim yüklenmesine neden olur. Üretimi ahşap ve beton traverslere göre daha zordur. Ahşap ve çelik traverslere göre oldukça gevrek (Fischer et al., 2015).

Demiryolları üst yapısında beton traversler gün geçtikçe standart haline gelmektedir. Gerek mukavemeti gerek seri üretim kolaylığı gerekse uzun kullanım ömrü avantajlarıyla öne çıkmaktadır.

2.3. Ray

Raylar demiryolu üst yapısının demiryolu araçlarıyla kontak halindeki bileşenidir. Bu nedenle ayrı bir önem taşımaktadır.

Raylar demiryolu üst yapısına demiryolu araçlarından gelen yükü ilk karşılayan elemanlardır. Raylar demiryolu araçlarına kılavuzluk eder, tekerle belirli bir profil ile temas ederek sürtünmeyi en aza indirir, üzerine gelen farklı eksenlerdeki kuvvetleri traverslere iletir, hat geometrisini istikrarında önemli bir görev görür. Ayrıca elektrifikasyona ve sinyalizasyona sahip hatlarda devre tamamlama suretiyle trafik kontrol sistemlerinin uygulanmasına olanak sağlar (Fischer et al., 2015).

Rayların yapısını genel olarak üç temel kısımdan oluşur. Bunlar ray tabanı, ray gövdesi ve ray mantarındır. Demiryolu aracının tekeri ile rayın temas noktası ray mantarı üzerindedir. Bu nedenle yüksek hızlı tren hatlarında kullanılan rayların mantar kısımları ısıtma işlemine tabi tutularak istenen sertlik değerlerine getirilir.

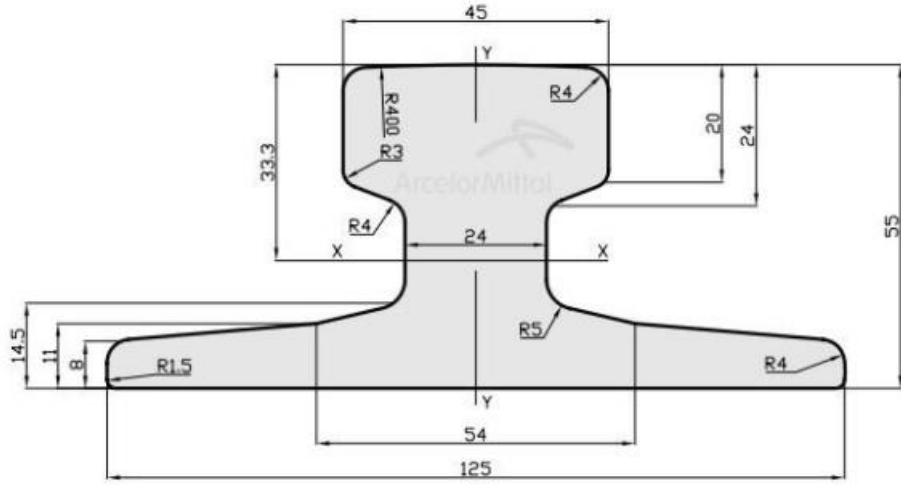
Ray profilleri geçmişten bugüne ihtiyaçlara bağlı olarak değişmiştir. Günümüzde bir profil tipi diğerlerine göre daha yaygın olarak kullanılsa da farklı ihtiyaçlara değişik profiller geliştirilmiştir.

Ray profillerini vignole profili, yivli profil ve muhtelif ray tipleri olarak gruplandırabiliriz. Bunlar içerisinde en yaygın olarak kullanılan profil vignole tip raylardır. Bunun da temel nedeni günümüzün yüksek dingil basınçlarını karşılayacak mukavemete ve atalete sahip olmasıdır (Fischer et al., 2015).

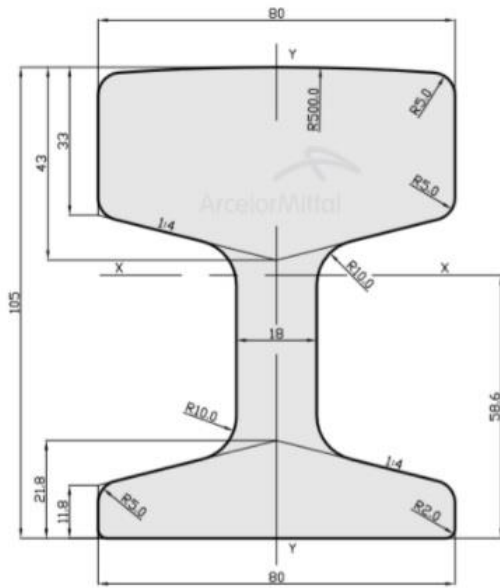
Demiryollarında yüksek yolcu ve yük ulaşımı ihtiyacını karşılamak için yaygın olarak 60E1 ve 49E1 vignole ray profilleri kullanılmaktadır. Şekil 2.4'te görülen bu ray profilleri hem konvansiyonel demiryolu hatlarında hem şehir içi raylı sistemlerde yaygın olarak tercih edilmektedir. Tramvay gibi bazı hafif raylı ulaşım sistemlerinde ise Şekil 2.5'te görülen yivli ray profilleri tercih edilmektedir (Voestalpine, 2021).

[illegible]

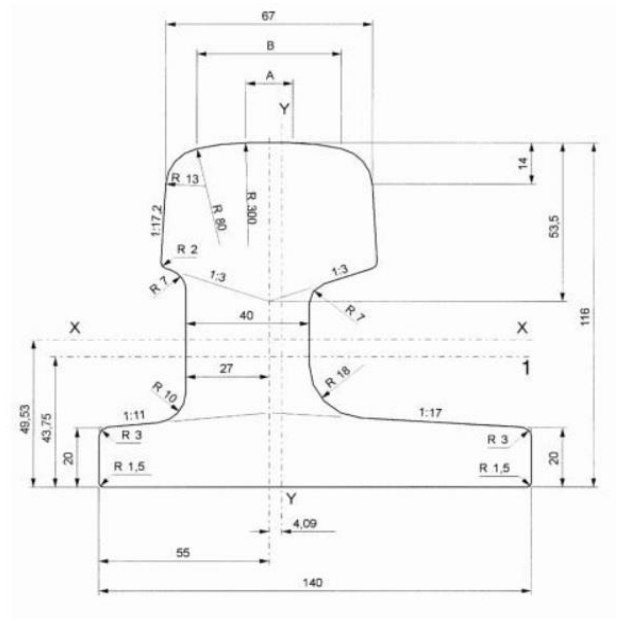
Şekil 2.5 Yivli ray profili 59Ri2(Voestalpine, 2021)



(a) vinç ray profili A45



(b) Enerji iletim rayı profili STR 40



(c) Makas dili ray profili 49E1A1

Şekil 2.6 Muhtelif ray profilleri (ArcelorMittal, 2021)

Demiryollarında birçok muhtelif ray tipi bulunmaktadır. Bunlardan yaygın olarak kullanılanları Şekil 2.6 ile gösterilmiştir (ArcelorMittal, 2021). Mantar kısmı geniş olan vinç ray profili yüksek yük basıncının olduğu durumlarda kullanıma uygundur. Enerji iletim ray

profilleri genellikle metro sistemleri başta olmak üzere şehir içi raylı sistemlerde üçüncü ray olarak karşımıza çıkar. Katener hattı olmayan raylı sistemlerde üçüncü ray üzerinden yüksek gerilim iletimi sağlanır. Makas dili profili ise demiryolunda trafik akışını yönlendiren makasların oynar kısımlarının en ucunda bulunan ray profilidir.

Ray profillerinin belirlenmesinde birçok gereksinim rol oynamaktadır. Ray mantarının teker ile temas ettiği yüzeyinin mümkün olduğunca az temas gerilmesi ve aşınma oluşturması, ray geometrisinin hem düşey hem yatay ekseninde yüksek atalet momentine sahip olması, ray gövdesinin bükülmelere karşı mukavim olması, ağırlık merkezinin ray yüksekliğinin tam merkezinde bulunması ve ray geometrisinde keskin geçişlerden kaçınılması bu gereksinimlerden başlıcalarıdır (Fischer et al., 2015).

Raylar çelik malzemelerdir. Kullanım alanlarına göre ısıtılma maruz kalırlar. Hattın dingil basıncı ve hattaki izin verilen hız sınırı arttıkça başka bir deyişle konvansiyonel hatlardan hızlı demiryolu taşımacılığına geçiliyorsa, kullanılacak raya olan ısıtılma işlemleri de farklılaşır. Ülkemiz yüksek hızlı tren işletmeciliği doğrultusuna mantarı sertleştirilmiş raylar tercih edilmeye başlanmıştır.

Ray çelikleri, gerekli kimyasal direnç, uniform bir bileşime, yüksek aşınma dayanımına, gevrek kırılmalara ve yorulmalara karşı yüksek dayanıma, düşük artık gerilmelere, yüksek sertliğe, yüksek çeki ve bası mukavemetine, iyi bir yüzey kalitesine ve yüksek korozyon direncine sahip olmalıdır. Tüm bunların yanına ısıtılma kararlılığı, kaynak edilebilirliği ve elektrik iletkenliği iyileştirilmiş olmalıdır. Özellikle sıcaklıkların gerek mevsimler arasında gerekse gün içinde kaydedeğer biçimde değiştiği iklim koşullarında ve termit kaynağı gibi çok yüksek sıcaklıklarda yapılan işlemlerde rayların ısıtılma özellikle kritik önem taşır. İç gerilme oluşturan ısıtılma yüklerine karşı rayların üretimden kalan artık gerilmelerinin mümkün olduğunca düşük seviyede kalması beklenir. Ray çeliklerinin tüm bu özelliklerin ölçüleri, muhtelif kullanım alanlarına ve coğrafi bölgelere göre farklılaşabilir (Fischer et al., 2015).

2.4. Bağlantı Elemanları

Demiryollarında bağlantılar raylar arası ve raylar ile traversler arası bağlantılar olarak gruplanabilir. Raylar ile traversler arası bağlantılar gergi kısıkaçları ve tirfonlar ile sağlanırken raylar arası bağlantılar ise yaygın olarak alın kaynağı ya da alümina termit kaynağı ile yapıldığı gibi cebire adı verilen bağlantı elemanları ile de yapılabilir.

Bağlantı elemanları, demiryoluna gelen dikey ve yatay yüklerin tüm demiryolu boyunca dağıtılmasını yani demiryollarında yük dağılımının sürekliliğini sağlar. Böylece demiryollarının güvenliği de garanti altına alınmış olur. Genleşmeden dolayı oluşan artık ısı gerilmeler neticesinde oluşacak hasarları en aza indirmeyi sağlar. Ayrıca sinyalizasyona sahip hatlarda raylar arası yalıtımları sağlayarak trafik güvenliğine ve yönetimine katkı sağlar (Fischer et al., 2015).



Şekil 2.7 Gergi kısıkaçı ve tirfon

Ray ile travers arasındaki bağlantılar günümüzde yaygın olarak Şekil 2.7’de görülen gergi kısıkaçı ve tirfon aracılığıyla yapılmaktadır. Gergi kısıkaçı esneme kabiliyeti ile demiryoluna binen dikey yükleri ve titreşimleri sönmlemede rol oynar. Tirfon ise ray tabanının traverse bağlanmasını sağlayan bağlantı elemanlarının genel adıdır. Traversler genellikle 63 cm aralık ile dizilir ve traverslerin her iki tarafında bu bağlantılar mevcuttur.

Travers çeşidine göre ray ile travers arasındaki bağlantıyı sağlamak için kullanılan elemanların geometrisi ve yapısı değişebilir. Günümüzde sık kullanılan beton traverslerde yaygın olarak Şekil 2.7’de görülen bağlantı elemanları kullanılır.



Şekil 2.8 Cebire ile yapılan ray bağlantısı

Cebireler ve kaynaklı bağlantılar raylar arasındaki bağlantıları sağlar. Şekil 2.8 cebireli bir ray bağlantısını göstermektedir. Alın kaynağı ve alümina termit kaynağı demiryollarında en yaygın kullanılan kaynak yöntemleridir. Kaynaklı bağlantılarda ısı gerilemelere azami derecede dikkat etmek gerekmektedir. Cebirelerde ise genleşmeye dair paylar genellikle mevcuttur.

3. HIZ VE RAY SICAKLIĞINA DAİR SORUNLAR VE İHTİYAÇLAR

Demiryollarında seyreden demiryolu araçlarının hızları ve üst yapının en kritik elemanı olan rayların sürekli sıcaklık değişimleri belirli sorunlara yol açmakta ve bu sorunlar zaman içerisinde demiryollarında bazı riskler oluşturmaktadır.

Bu risklerin önüne geçme ihtiyacı demiryollarının güvenliği ve istikrarı açısından önem arz etmektedir. Hem ray sıcaklığı hem de seyir halindeki demiryolu araçlarının hızları ayrı ayrı inceleme ve eylem planı oluşturma gereksinimi yaratan olgulardır.

3.1. Demiryollarında Hız Limiti

Demiryollarında hattın yapısına göre hız limitleri değişmektedir. Ayrıca hat üzerindeki üst yapısının anlık durumları da hız limitlerini belirlemede önemli bir etmendir. Hat üzerinde yapılan çalışmalar haricinde bu iki durum hat üzerindeki hızları sınırlar. Güney Kore'nin hızlı demiryolu sistemi KTX için 55-60 C derece aralığındaki ray sıcaklıklarında hız limiti genellikle saatte 230 km iken ray sıcaklığı 60-64 C aralığına çıktığında hız limiti saatte 70 km'ye düşmektedir (Hong et al., 2019).

Eğer hat geometrisi üzerinde çok sayıda kurp varsa başka bir deyişle çok sayıda kıvrım mevcutsa ve bu kurpların yarıçapları küçükse hat hızı düşer. Dar kurplar yani düşük yarıçapa sahip kurplar (örneğin 300 m) üzerinde hız limiti verilmezse yüksek hızlarda hat üzerinde seyreden demiryolu aracı deray etme tehlikesiyle başka bir deyişle raydan çıkma tehlikesi ile karşı karşıya kalır. Derayman olayı yüksek hızlarda hem yolcu hem yük trenleri için demiryollarında görülebilecek en büyük tehlikelerden biridir. Şekil 3.1'de derayman hadisesi görülmektedir.



Şekil 3.1 Derayman (de Aguiar et al., 2018)

Dar kurplarda yüksek hızla seyreden demiryolu araçları deray tehlikesi yaşayacağı gibi izin verilen hız sınırları dışına çıkan araçlar boden-ray baskısını arttırarak raylarda olağan dışı sürtünmelere neden olarak raylardan talaş kaldırır. Bunun sonucunda kurp yönüne göre üzerinde aşırı sürtünme oluşan rayın profili sürtünmeye uğrayamayan karşıt rayın profiline kıyasla bozulma gösterir. Bu durum raylar üzerinde profilleri dengeleyecek taşlama gibi ilave bakım hizmeti doğurur.

Öte yandan demiryolu hattındaki makas bölgeleri de hız sınırlarının varlığında önemli bir unsurdur. Makaslar demiryolu aracının seyir halinde bulunduğu hattın diğer hatta geçişini

sağlayan hareketli ray parçalarıdır. Yine derayman ve aşınma tehlikelerinden burada da bahsedilebilir. Hattın hız sınırlarına hat üzerindeki makasların sayısı, konumu ve yapısı da etkindir. Şekil 3.2’de bir demiryolu makası ve makas motoru gösterilmiştir.



Şekil 3.2 Demiryolu makası(de Aguiar et al., 2018)

Bunlara ek olarak karayolu ile demiryolunun kesiştiği hemzemin geçitlerde bu denkleme eklenebilir. Bunlar demiryolu kazalarının sık görüldüğü kesişimlerdir. Hemzemin geçitlerin bulunduğu hatlardaki hız sınırları idari belgelerde düzenlenmektedir. Hemzemin geçitlerdeki karayolu trafiğine göre hız sınırlamaları değişmekle beraber yüksek hızlara izin verilmemektedir.

Tüm bu etmenler nedeniyle demiryolu araçlarının demiryollarındaki hız sınırlarına uyup uymadığının saptanması ve hatlarda gerektiğinde dinamik hız sınırlarının belirlenmesiyle demiryollarında güvenliğin sağlanması oldukça önem arz etmektedir.

3.2. Ray Sıcaklığı Etkileri

Demiryollarında mevsimler boyunca özellikle de yaz ve kış aylarında sıcaklık, nem vb. hava koşullarının demiryolu üst yapısını oluşturan mekanik elemanlara olan etkisi oldukça farklıdır. Yaz ve kış aylarında oluşan uç sıcaklıklar başta raylar olmak üzere yol yatağını balast hacmini olumsuz etkiler.

Demiryolu araçlarının hat üzerinde hızlanması ve yavaşlaması başka bir deyişle ivmelenme kabiliyeti ray-teker teması ile doğrudan ilişkilidir. Kış koşullarının sert geçtiği

bölgelerde ray üzerinde oluşan buzlanma, çamurlanma ya da ıslak yüzey oluşumu ray-teker temasına etki ederek sürtünmeyi düşürücü rol oynar. Bu durum fren mesafelerine etki ederek demiryolu araçlarının duruş mesafelerini uzatır, zaman zaman sinyal bölgelerinin dur koşulu gerektiren ilgili kısımlarının aşılmasına neden olarak kaza riskini artırır (Tahvili, 2016). Bu durumlarda hat seyir hızlarının taşımacılık sistemi elverdiğince düşürülmesi gerekir.

Gizli buzlanma durumları söz konusu ray-teker temasının azalmasındaki en önemli etkindir. Oldukça düşük sıcaklıkların görüldüğü kış koşullarında ray yüzeyinde biriken suyun veya nemin donmasıyla ray yüzeyinin üzerinde buzdan bir film tabakası oluşabilir. Bu tabaka kalın ise eğer basınç ile sertleşebilir ve deraymanlara sebep olabilir.

Öte yandan buzlanma, makaslar üzerinde de etkilere sebep olur. Makas dili hareketinin zorlaşması, ray-dil temasının tam olarak sağlanamaması makas sıkışmalarına neden olabilir. Bu durumlarda ısıtıcıların devreye sokulması gerekebilir (Tahvili, 2016).

Kış aylarındaki soğuk hava koşullarının ve düşük sıcaklıkların getirdiği en önemli sorunlardan bir tanesi de ray kırılmalarıdır. Düşük sıcaklıklar özellikle sürekli kaynaklı rayların bulunduğu hatta raylarda yüksek iç gerilmelere neden olur. Bu ray kırılmaları için yeterli bir neden iken tekerlerinde apleti oluşan yani düzleşme meydana gelen demiryolu araçları da iç gerilmesi artan bu raylar üzerinde periyodik darbeler oluşturarak ray kırılması riskini artırır. Ray kırıkları deraymanların en önemli sebeplerinin başında gelir. Ancak ray kırıklarının oluşturduğu tehdit sadece derayman tehlikesi değildir. Ray devresi kullanan sinyalizasyon sistemlerinde belirli sinyal bölgelerine ayrılan hatlarda demiryolu aracının algılayan sinyal devresinin ray üzerinden kapalı devre çalışması nedeniyle ray kırıkları sinyalde aksamalara neden olarak trafiğin aksamasına hatta kazalara sebep olabilir (Tahvili, 2016).

Donma olayı sadece raylar ve makasları değil, üst yapının diğer bir parçası olan balast için de bir problemdir. Eğer balastın yapısında donma kabarması olursa diğer bir deyişle balast hacmi içerisinde bulunan nem veya su donarsa yol yatağını kaydırarak yol geometrisini bozabilir. Bu da derayman riski doğurabilir. Ayrıca balast tabakasında oluşan bu durum balast tabakasının yüksekliğini artırarak demiryolu aracı tarafından balast saçılmasına neden oluşturabilir. Bu da kısa vadede dingil çatlakları oluşumuna orta ve uzun vadede de dingil kırılmalarına neden olabilir (Tahvili, 2016).

Düşük sıcaklıkların diğer bir olumsuz etkisi de yol bakım onarım çalışmalarında gözlenir. Kurp adı verilen demiryolu hattının kıvrımlı bölümlerinin geometrisi zamanla değişimler gösterebilir. Yol yatağının tekrar istenen geometriye çekilebilmesi için buraj adı

verilen bakım çalışması uygulanır. Bir iş makinası yardımı ile yol adım adım kaydırılarak istenen geometriye getirilir. Donma olaylarının görüldüğü durumlarda yapılan buraj işlemi iç gerilmeleri artmış raylarda ray mekaniğine zarar verebilir. Balast kabarması oluşmuş kurplarda ise balast saçılmasına neden olabilir. Bu nedenle özellikler ray sıcaklıklarının izlenerek dinamik bir bakım süreci oluşturulması önem taşır.

Şehir içi raylı sistemlerde katener yerine enerji iletim hattı olarak üçüncü bir ray kullanılır. Bu ray elektrik enerjisinin demiryolu aracına iletilmesini sağlar. Bu raylardaki buzlanma da elektrik iletimine zarar verebilir (Tahvili, 2016).

Kış aylarının tüm bu olumsuz etkilerinin bertaraf edebilmek ya da en aza indirebilmek için demiryolu üst yapısına dair güncel veri akışına sahip olmak oldukça önem arz etmektedir. Bu veri akışı sayesinde olası tehditlere anlık olarak tepki verilebilir., hızlı eylem planları oluşturulup aktivasyona geçilebilir ve dinamik bakım-onarım eylem planları ortaya konabilir. Böylece demiryolu taşımacılığı yoğun kış şartları altında sürdürülebilir kılınabilir.

Yaz aylarının getirdiği sorunlar ise yüksek sıcaklık kaynaklıdır. Yüksek sıcaklıklar yüksek iç gerilmelere yüksek iç gerilmeler ise raylarda burkulmaya neden olur. Burkulmanın en önemli parametresi yüksek sıcaklık farklarıdır.

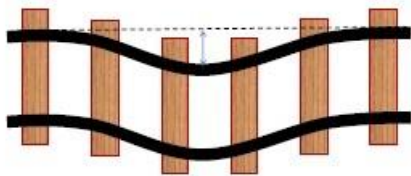
Raylarda görülen burkulma yaz sıcaklıkları yüksek seyreden coğrafyalarda demiryolu taşımacılığının önemli sorunlarından biri olarak karşımıza çıkmaktadır. Raylarda görülen burkulmanın nedenlerinin en başında yüksek sıcaklık farkları dolayısıyla oluşan iç gerilmeler ve en belirginine enine dislokasyonlar olan yol kusurları gelir. Bunlar dışında balast direnci ve bağlantı elemanlarına yüklenen yüksek tork değerleri de burkulmaya neden olan etmenlerdir(Kish and Samavedam, 2013).



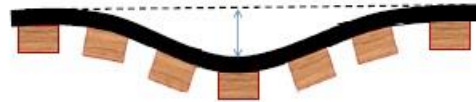
Şekil 3.3 Demiryolunda ray burkulması (Kish ve Samavedam, 2013).

Sürekli kaynaklı raylarda, rayın hatta monte edildiği sıcaklığa nötr sıcaklık adı verilir. Bu sıcaklıkta eksenel yükler sıfır kabul edilir. Nötr sıcaklık ile herhangi bir sıcaklık arasındaki fark eksenel yükler doğurur. Her 1 C derecelik sıcaklık fark ray üzerinde yaklaşık 2,5 MPa gerilim oluşturur. Bu da UIC 60 ray profilleri için 1 C derece başına 18,6 kN eksenel yüke eşit kabul edilmektedir. Mevsimler arası sıcaklık farkları 30 C dereceyi aşan coğrafi bölgelerde bu durum demiryolu taşımacılığının sürekliliğine ve sürdürülebilirliğine tehdit oluşturur (Vangi and Virga, 2007).

Sürekli kaynaklı raylara sahip demiryolu hatlarındaki kısa ray blokları fazla sayıda kaynak alanına işaret eder. Ray kaynaklarının oldukça yüksek sıcaklıklarda gerçekleştirilmesi nedeniyle raylarda kalıcı gerilmeler oluşur bu da ray burkulmaları için gerekli iç gerilmelerin oluşumunda önemli bir etmen olarak karşımıza çıkar. Şekil 3.3 ray burkulmasını göstermektedir. Ray kaynağının fazla olduğu hatlarda burkulma oluşumunu tetikleyen kritik sıcaklık sınırı düşer (Yang ve Bradford, 2016). Buna benzer bir duruma neden olan diğer bir unsur yol kusurlarıdır. Bunların başında da enine ve düşey ray dislokasyonları gelir.



(a) Enine dislokasyon



(b) Düşey dislokasyon

Şekil 3.4 Demiryollarında yaygın yol kusurları

Şekil 3.4 yaygın görülen yol kusurlarını göstermektedir. Yüksek sıcaklık farkı sonucu raylarda oluşan iç gerilmeler, demiryolu hattındaki belirgin yol kusurları ile birleşirse enine deplasmanı yaklaşık 75 cm uzunluğa kadar ulaşabilen burkulmalara neden olabilir. Kritik sıcaklıklarda oluşan iç gerilmelerin etkisiyle ray bloğu elastik şekil değiştirme sınırlarını aşarak kalıcı deformasyona uğrar. Yol kusurları mevcutsa kritik sıcaklık değeri düşer. Yoldaki kusurun genliği büyüdükçe burkulmaya neden olacak kritik sıcaklık ile nötr ray sıcaklığı arasındaki fark da azalır (Kish and Samavedam, 2013).

Burkulma problemi demiryollarındaki kritik sorunlardan biri olup deraymanların oluşumundaki birincil nedenlerdendir.

3.3. İhtiyaçlar ve Faydalar

İklim koşulları sonucunda ortaya çıkan düzensiz sıcaklıkların getirdiği kritik mekanik problemler, hatta seyreden araçların hız limitlerinin aşması sonucu oluşabilecek aşınma sorunları, hat güvenliği ile derayman riskleri demiryollarında ray sıcaklığının ve hat üzerindeki araçların hızlarının sürekli olarak izlenmesi ihtiyacını doğurmaktadır. Bakım planlarının ve bakıma ayrılan işgücünün verimi, geleneksel bakım programlarından öngörücü bakım uygulamalarına geçiş ihtiyacı da demiryollarında özellikle üst yapı elemanlarının düzenli olarak izlenmesi gerekliliğini ortaya çıkarmaktadır. Bu nedenle demiryollarında ray sıcaklığının ve hat üzerinde seyir halinde bulunan araçların hızlarının sürekli olarak izlenip kayıt altına alınması önem taşımaktadır.

Demiryolları doğası gereği çok geniş coğrafyalara yayılan alt ve üst yapılara sahiptir. Farklı coğrafi bölgeler farklı iklimler farklı iklimler de farklı sıcaklık alanları anlamına gelmektedir. Gerek yüksek gerekse düşük sıcaklıkların üst yapıdaki etkileri göz önüne alındığında demiryolu hatlarının özellikle kritik elemanlarını içerecek şekilde belirli bölümlere ayrılarak ray sıcaklıklarının izlenmesi ve hat üzerinde seyir halindeki araçların hızlarının takip edilmesi başta derayman ve ray kırılmaları olmak üzere birçok sorunun önüne geçebilir. Kaza riskini azaltarak demiryolu taşımacılığının sürekliliğine ve sürdürülebilirliğine katkı sağlar.

Ray sıcaklıklarının izlenmesi üst yapıdaki tehlike arz edecek mekanik problemlerin önceden tespit edilebilmesine olanak sağlarken demiryolu araçlarının seyir hızlarının izlenmesi ise demiryolunun mekanik durumlarına ve hat trafiğine göre belirlenen hız limitlerinin aşılıp aşılmadığının kontrolüne aynı zaman da hatların hız limitlerinin dinamik olarak belirlenmesine olanak sağlar.

Düzenli olarak izlenen ray sıcaklıkları ve demiryolu araçlarının hızları, demiryolu üst yapısı bakımlarının dinamik olarak sürekli olarak güncellenmesine, bakımlarda öncelik verilmesi gereken hat bölümlerinin belirlenmesine, hat bakımının planlandığı zaman diliminde hattın bakıma müsait olup olmadığının tespitini sağlar.

Makine öğrenmesi ya da derin öğrenme gibi yapay zeka modelleriyle hat bakımının planlandığı tarih ve saat içerisinde ray sıcaklıkları tahmin edebilme kabiliyeti sürekli güncellenen bir bakım programını olanaklı kılar. Böylece çok daha verimli bir bakım sistemi oluşturularak işgücü kaybı en aza indirilebilir. İleri bir tarihte üst yapıda çıkabilecek mekanik problemler öngörülerek demiryollarının güvenliğine katkı sağlanabilir.

4. SİSTEMİN TASARIMI

Ray sıcaklığının ve hat üzerinde seyir halindeki araçların hızlarını izleyecek olan sistemin nesnelerin interneti ile entegre olması uzaktan izlenebilirliğini mümkün kılacaktır. Verilerin sahada konuşlanan sensörlerin bağlı olduğu mikro kontrolör içerisinden alınıp internete üzerinden kullanıcıya açılması sistemin ana işlevi olarak tanımlanır.

4.1. Fiziksel Model ve Veri Modeli İlişkisi

Ray sıcaklığının tahmini ve hat üzerinde seyir halindeki araçların hızlarının tayini yapılırken öngörülen fiziksel model bir veri modeliyle eşleştirilmiş ve bu sayede dijitalize edilmiştir.

4.1.1. Hız Tayininin Fiziksel Modeli ve Veri Modeli İlişkisi

Hız ölçümü iki adet endüktif sensör ile yapılmıştır. Endüktif sensörlerin hat üzerindeki karşılıklı raylardan birine arada belirli bir mesafe bırakarak yerleştirilmesi ile seyir halindeki demiryolu araçlarının tekerlerini boden adı verilen dış çaplarının algılanması sağlanmıştır. Endüktif sensörlerin hat üzerindeki yerleşimi Şekil 4.1’de gösterilmiştir. Burada sayılan her bir dingil mikrokontrolöre sinyal olarak gönderilmiştir. Mikrokontrolör içinde çalışan algoritma bu sinyaller vasıtasıyla demiryolu aracının hızını hesaplayacak ve dingil sayıları ile beraber bu bilgiyi, saha verisini sanal ortama iletmekle görevli olan uç cihaza TCP/IP haberleşme protokolü soketleriyle iletecek şekilde tasarlanmıştır.



Şekil 4.1 Hat üzerinde endüktif sensör yerleşimi

Endüktif sensörler yüksek frekanslı geçişleri algılayacak kadar hassas frekans değeri öngörülerek seçilmiştir. Günümüz demiryolları sistemleri üzerinde seyir halinde olan tüm demiryolu araçlarının hızları tayin edebilecek kapasiteye sahiptir. İki endüktif sensör arasındaki mesafe sistemin yerleştirileceği bölgede hattın sınırlayıcı şartları dikkate alınarak belirlenmiş bu mesafe içerisine giriş yapan demiryolu aracının dingilleri sayılarak zaman değeri oluşturulmuştur. Söz konusu mesafe saptanan zaman değerine bölünerek hız değeri saptanmıştır. Bu değer saptanırken seyir halindeki demiryolu aracının endüktif sensörlerin önünde ve arkasındaki ivmelenme değerleri göz önünde tutulmaktadır. İvmelenme değerlerinin belirlenen mesafede hıza olan etkisinin ihmal edilebilir durumdadır. Saptanan hız değeri m/s biriminde olup km/h birime çevrilerek hem insan-makine ara yüzünde gösterilmiş hem de uç cihaz vasıtasıyla veri tabanına kaydedilerek bir web sitesi üzerinde izlenebilir kılınmıştır.

Hız ölçümü sırasında hız değerinin yanı sıra endüktif sensörlerin saydığı dingil sayıları da yine uç cihaz üzerinde veri tabanına kaydedilmekte ve hem web sitesinde hem de insan-makine arayüzü üzerinde kullanıcıya sunulmaktadır.

Veri	Veri Tipi	Veri Tanımı
Dingil Sayıcı_1	Integer	Birinci endüktif sensörün saydığı dingil sayısı
Dingil Sayıcı_2	Integer	İkinci endüktif sensörün saydığı dingil sayısı
Hız	Double	Hız değeri

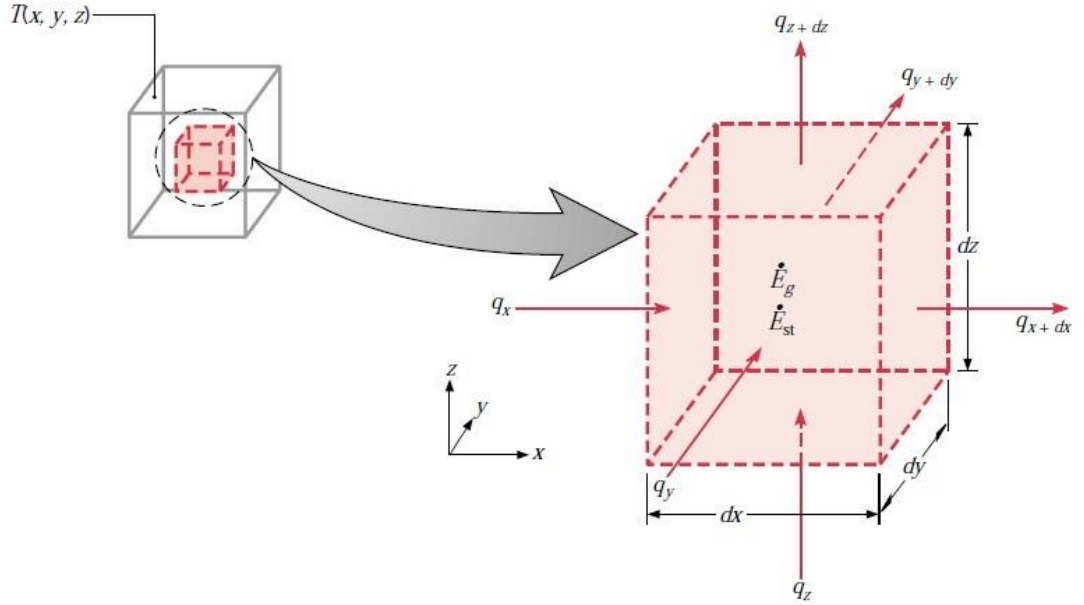
Tablo 4.1 Hız tayini veri modeli

Hız ölçümü sonucunda veri tabanına 3 farklı veri gönderilmektedir. Bu veri dingil sayıcı değerleri ve hız değerini kapsar. Dingil sayıcı değerleri 32 bit uzunluğunda tamsayı (Integer), hız değeri ise 32 bit uzunluğunda gerçel sayı (Double) veri tipiyle ifade edilmektedir. Hız değerinin hesabına dair veri modelinin özeti Tablo 4.1 üzerinde görülmektedir.

4.1.2. Ray Sıcaklığı Tahminin Fiziksel Modeli ve Veri Modeli İlişkisi

Demiryolunda ray sıcaklığı tahmini için biri analitik yöntem diğeri ise veriye dayalı istatistiksel yöntem olmak üzere iki model izlenebilir. Analitik yöntem, ısı yayılım denkleminin üzerinde gerekli varsayımlar ve kabuller yapıldıktan oluşturulan fiziksel modelin diferansiyel denkleminin doğrudan çözümüne dayanırken veriye dayalı istatistiksel yöntem ise yine aynı fiziksel model üzerinden uygun bir makine öğrenmesi ya da derin öğrenme algoritmasıyla kayıt altına alınmış veriden çıkarım yapılmasıdır.

Bu nedenle öncelikle fiziksel modelin ısı yayılım denklemi çıkarılmalıdır. Kartezyen koordinatlarda ısı yayılım denkleminin elde edilmesini için bir kontrol hacmi ele alınır.



Şekil 4.2 Diferansiyel kontrol hacmi (Incropera et al., 2007).

Isı yayılım denklemi çıkarımı Şekil 4.2’de gösterilen diferansiyel bir kontrol hacmi üzerinde enerji korunumu eşitliği ile elde edilir. Bu dx , dy ve dz boyutlarında sonsuz küçüklükte bir hacimdir. Sıcaklık gradyanlerinin varlığı durumunda yüzeylerde ısı akısı oluşur. Bu ısı akıları q_x , q_y ve q_z olarak simgelenebilir, x , y ve z eksenlerinde bu yüzeylere

diktir. Bu durumda her bir yüzeyden çıkan ısı akısı ikinci ve daha yüksek dereceden terimlerin atıldığı Taylor serisi olarak ifade edilir (Incropera et al., 2007).

$$q_{x+dx} = q_x + \frac{\partial q}{\partial x} dx \quad (4.1a)$$

$$q_{y+dy} = q_y + \frac{\partial q}{\partial y} dy \quad (4.1b)$$

$$q_{z+dz} = q_z + \frac{\partial q}{\partial z} dz \quad (4.1c)$$

Enerji korunumu eşitliğinde yer alacak sonsuz küçük hacimdeki ısı üretimi

$$\dot{E}_g = \dot{q} dx dy dz \quad (4.2)$$

eşitliği ile ifade edilir (Incropera et al., 2007).

Diferansiyel kontrol hacminde depolanan ısı zamana bağlıdır ve faz değişimi yoksa

$$\dot{E}_{st} = \rho c_p \frac{\partial T}{\partial t} dx dy dz \quad (4.3)$$

denklemleri ile ifade edilir . Bu noktada herhangi bir anda diferansiyel kontrol hacmi içerisinde enerji korunum denklemi yazılabilir (Incropera et al., 2007).

$$\dot{E}_{in} + \dot{E}_g - \dot{E}_{out} = \dot{E}_{st} \quad (4.4)$$

Enerji korunumu denklemi açılarak kartezyen koordinatlarda diferansiyel kontrol hacminde sıcaklık gradyeninden kaynaklanan ısı akıları eşitliğe dâhil olur.

$$q_x + q_y + q_z + \dot{q} dx dy dz - q_{x+dx} - q_{y+dy} - q_{z+dz} = \rho c_p \frac{\partial T}{\partial t} dx dy dz \quad (4.5)$$

Bu noktada Denklem 4.1 yerine yazılarak gerekli sadeleştirmeler yapılır.

$$-\frac{\partial q_x}{\partial x} dx - \frac{\partial q_y}{\partial y} dy - \frac{\partial q_z}{\partial z} dz + \dot{q} dx dy dz = \rho c_p \frac{\partial T}{\partial t} dx dy dz \quad (4.6)$$

Diferansiyel kontrol hacmi içerisinde ısı iletimi Fourier yasasından elde edilir.

$$q_x = -k dy dz \frac{\partial T}{\partial x} \quad (4.7a)$$

$$q_y = -k \, dx \, dz \, \frac{\partial T}{\partial y} \quad (4.7b)$$

$$q_z = -k \, dx \, dy \, \frac{\partial T}{\partial z} \quad (4.7c)$$

Denklem 4.7 , Denklem 4.6'da yerine konulursa

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) + \dot{q} = \rho \, c_p \, \frac{\partial T}{\partial t} \quad (4.8)$$

ifadesi elde edilir.

Bu noktada diferansiyel kontrol hacminin sabit ısı iletim katsayısına sahip olduğu kabul edilirse denklem

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{q}}{k} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (4.9)$$

ifadesi meydana gelir. Bu denklemde $\alpha = k/\rho \, C_p$ terimi ısı yayılım katsayısıdır (Incropera et al., 2007).



Şekil 4.3 Ray bloğu parçası

Denklem 4.9, ısı iletim katsayısı ve yoğunluğu kendi içerisinde sabit ve içerisinde ısı üretimi olan bir katıda ısıнын zamana bağlı yayılımını ifade eder. Bu denklem çözüldüğü takdirde katı içerisindeki zaman bağlı sıcaklık dağılımı elde edilir. Ancak demiryolu hattında ray sıcaklığını tayin ederken ray blokları uzun bir hat boyunca kaynaklı ve bütünleşik olduğundan sıcaklık dağılımı çözümlmek yerine Şekil 4.3 ile gösterilen ray bloğunun bir parçasını ele alarak toplam kütle yaklaşımı uygulamak uygun olacaktır. Ray üzerinde birbirine yakın iki noktadan sıcaklık ölçümü yapılmasından ve ölçümlerde çok yüksek bir hassasiyete ulaşma gereği duyulmadığından sıcaklık dağılımından feragat ederek toplam kütle yaklaşımı kabulü yapmak uygundur.

Toplam kütle yaklaşımı bir katı içerisindeki sıcaklığın herhangi bir zaman diliminde katının her noktasında aynı olduğunu kabul eder. Diğer bir deyişle katı içerisindeki sıcaklık gradyanı ihmal edilir (Incropera et al., 2007).

Toplam kütle yaklaşımının uygulanabilirliğini Bi sayısı belirler.

$$Bi = \frac{hL_c}{k} \quad (4.10)$$

Denklem 4.10 ile gösterilen eşitlik Bi sayısını ifade eder. Bi sayısı taşınım sınır koşulu bulunan bir katı üzerindeki taşınım direncinin katı içerisindeki iletim direncine oranıdır. Bu oran 0,1'den küçük ise katı içerisindeki sıcaklık gradyanı ihmal edilerek sıcaklık sabit kabul edilir. Böylece toplam kütle yaklaşımı uygulanabilir duruma gelir (Incropera et al., 2007).

Denklem 4.10 ile gösterilen eşitlikte h havanın taşınım katsayısıdır ve ray bloğu parçası için 10 m/s rüzgar hızına ve 255 K sıcaklığa göre yaklaşık değeri olan 180 W/m²K alınmıştır (Shitzer, 2006). Yine aynı eşitlikte k ısı iletim katsayısı konvansiyonel hatlarda yaygın olarak kullanılan karbon-mangan alaşımlı R260 ray çeliği için 60,5 W/mK alınmıştır (Incropera et al., 2007).

$$L_c = \frac{V}{A_s} \quad (4.11)$$

Bi sayısı içerisindeki diğer bir değişken olan karakteristik uzunluk, sıcaklık ölçümü yapılacak ray bloğu parçası için hesaplanmalıdır. Karakteristik uzunluk Denklem 4.11 ile ifade edilir (Incropera et al., 2007).



Şekil 4.4 Ray bloğu parçasının kesiti

Karakteristik uzunluk hacim ve toplam yüzey alanının oranı olarak ifade edilir. Şekil 4.4 ile gösterilen 60E1(UIC60) profiline sahip ray kesiti için toplam kesit alanı $76,70 \text{ cm}^2$ olarak alınır (ArcelorMittal, 2021). Şekil 4.3 ile gösterilen ray bloğu parçasının uzunluğu yaklaşık 65 cm 'dir. Bu durumda yüzey alanı A_s $4675,67 \text{ cm}^2$ hesaplanmıştır. Hacim V ise $4985,5 \text{ cm}^3$ bulunmuştur. Biot sayısı bu durumda $0,0317$ bulunur ve bu sonuç toplam kütle yaklaşımına uygundur.

$$\dot{E}_h + \dot{E}_r = \dot{E}_{st} \quad (4.12)$$

Denklem 4.12 ile ifade edilen eşitlik, ray bloğu parçasının enerji denklemidir. Ele alınan ray parçasının belirlenen uzunluk içerisinde sağ ve sol kesitlerinin adyabatik olduğu diğer bir deyişle bu kesitlerde sıcaklık gradyanının sıfır olduğu varsayımı ve ısı üretimi olmadığı kabulü yapıldığında

$$h (T - T_{\infty,h}) A_{s,h} + \varepsilon \sigma (T^4 - T_{\infty,r}^4) = \rho c_p \frac{dT}{dt} \quad (4.13)$$

eşitliği ortaya çıkar. Bu eşitlikte dikkat edilmesi gereken nokta taşınımın gerçekleştiği $A_{s,h}$ yüzey alanı ile ışıınıma maruz kalan $A_{s,r}$ yüzey alanının farklı alanlar olduğudur.

Denklem 4.13 vasıtasıyla ray sıcaklığının zamana bağlı değerinin analitik olarak hesaplanması oldukça zordur. Ray sıcaklığının zamana dayalı tahminini yapabilmek için özellikle ısı taşınımı ve ısı ışıınımı etkilerini göz önüne alan zamana dayalı bu diferansiyel denklemin tam çözümü mümkündür. Ancak parametre düzeyine inildiğinde çözüm karmaşık bir hale bürünmektedir.

Öncelikle ısı taşınımı bu denklemde önemli bir yer tutmaktadır. Taşınım etkisinin denkleme dahil edilebilmesi için anlık hava koşulları bilgileri alınarak rayın yüzeylerinde taşınım katsayısının hesaplanması gereklidir. Bunun için de ray için bir akış modeli belirlenip belli bir zaman diliminde Nu sayısı tespit edilmelidir. Dolayısıyla Re ve Pr sayıları bilinmelidir. Taşınım hesabı için tespiti için rüzgar hızı ve hava sıcaklığının bilinmesi önem arz edecektir çünkü akış modeli belirli koşullar altında laminar ya da türbülanslı akış özelliği gösterebilir. Bunlar dışında yağışlı havalarda bu modelin alternatif bir model ile ikame edilmesi gerekecektir.

Öte yandan denklemdeki ısınım etkisi taşınımına göre daha fazla belirsizlik içerecektir. Azimut açısı, ray yüzeyinin ışıma maruz kalan alanının tespiti, solar radyasyon değeri gibi parametrelerin anlık olarak tayini gerekir. Bu parametreler arasında tayin edilmesi en zor olanlardan biri rayın ışıma maruz kalan yüzey alanı olacaktır. Belirli varsayımlar ile sadeleştirmeler yapıldıktan sonra anlık olarak ray yüzeyinin ışıma maruz kalan alanını, azimut açısını, dünyanın güneşe göre konumunu ve hattın bulunduğu bölgenin enlem ve boylamını, solar radyasyon değerini tespit ederek yapılan ray sıcaklığı analitik hesabı literatürde mevcuttur (Hong et al., 2019). Söz konusu hesaplarda sadece ısınım dikkate alınmış olup buna rağmen kompleks bir sistem ve hesap yükü ortaya konmuştur.

Tüm bu etmenler dikkate alındığında belirli zaman diliminde ray sıcaklığının hesaplanması gerekli bu parametrelerin tayini, yüksek sistem donanımı yatırımı ve kullanıcı tarafından gönderilen her istekte parametrelerin tayini için gereken denklemlerle birlikte diferansiyel denklem gerektirir. Bu durum işlemci yükü anlamına gelir. Tüm bunlar göz önüne alındığında hem ilk yatırım hem de bakım maliyeti artacaktır.

Veri	Veri Tipi
Hava Sıcaklığı	Double
Hissedilen Hava Sıcaklığı	Integer
En Yüksek Hava Sıcaklığı	Double
En Düşük Hava Sıcaklığı	Double
Havadaki Nem Yüzdesi	Integer
Rüzgar Hızı	Double
Rüzgar Açısı	Integer
Gökyüzü Durumu	String
Ay	String
Gün	Integer
Saat	Integer
Dakika	Integer

Tablo 4.2 Ray sıcaklığı tayini veri modeli

Analitik çözüm yerine makine öğrenmesi ya da derin öğrenme gibi istatistiksel algoritmaların uygun bir veri modeliyle beraber kullanılması uygun olacaktır. Böyle bir algoritma için belirlenen bir veri modeli ile toplanan verilerin algoritmaya öğretilmiştir. Bu sayede Tablo 4.2 ile gösterilen veri modeline uygun olmak koşuluyla gönderilen her veri için ray sıcaklığı tahmini yapılmıştır. Söz konusu veri modeli hava sıcaklığı, hissedilen hava sıcaklığı, en yüksek ve en düşük hava sıcaklıkları, havadaki nem yüzdesi, rüzgar açısı ve hızı ile gökyüzünün anlık durumunu kapsamaktadır. Ayrıca ay, saat ve dakika değerleri verinin üretildiği anı kayıt altına alan bilgilerdir. Bu veri modeli 32 bit uzunluğunda tamsayı (Integer), 64 bit uzunluğunda gerçel sayı (Double) ve metin (String) veri tiplerinden oluşmaktadır.

Bu veri modelinde hava sıcaklığı, hissedilen hava sıcaklığı, en yüksek ve en düşük hava sıcaklıkları, gökyüzü durumu ile ay, saat ve dakika hem taşınım hem ısıtım parametrelerine, havadaki nem yüzdesi, rüzgar hızı ve rüzgar açısı ise yalnızca taşınım parametrelerine dair bilgileri üzerinde taşır.

Söz konusu bu veri modelini uygun bir yapay zeka algoritmasına öğretmek ray sıcaklığını belirli bir yakınsama ile tahmin etmek mümkündür.

4.2. Sistem Mimarisi

Ray sıcaklığı ve demiryollarında seyir halindeki araçların hızlarının tespit etmek için oluşturulan sistemin mimarisi otomasyon, ön uç ve arka uç olmak üzere üç ana kısım halinde irdelenebilir. Bu bölümler hem donanım hem de yazılım teknolojileri içerisinde barındırmaktadır. Ölçüleri gerçekleştirmek üzere sahaya yerleştirilen, ölçüm ve haberleşme algoritmasının içerisinde yazıldığı donanım yapısı, otomasyon olarak adlandırıldı. Sistemin kullanıcı ile iletişimini sağlayan web sitesi yazılımı ön uç olarak isimlendirildi. Sahadan ve kullanıcıdan aldığı verileri işleyen, veri tabanı ile etkileşimde bulunan ve yapay zeka algoritmasını üzerinde bulunduran sunucu ise arka uç olarak adlandırıldı.

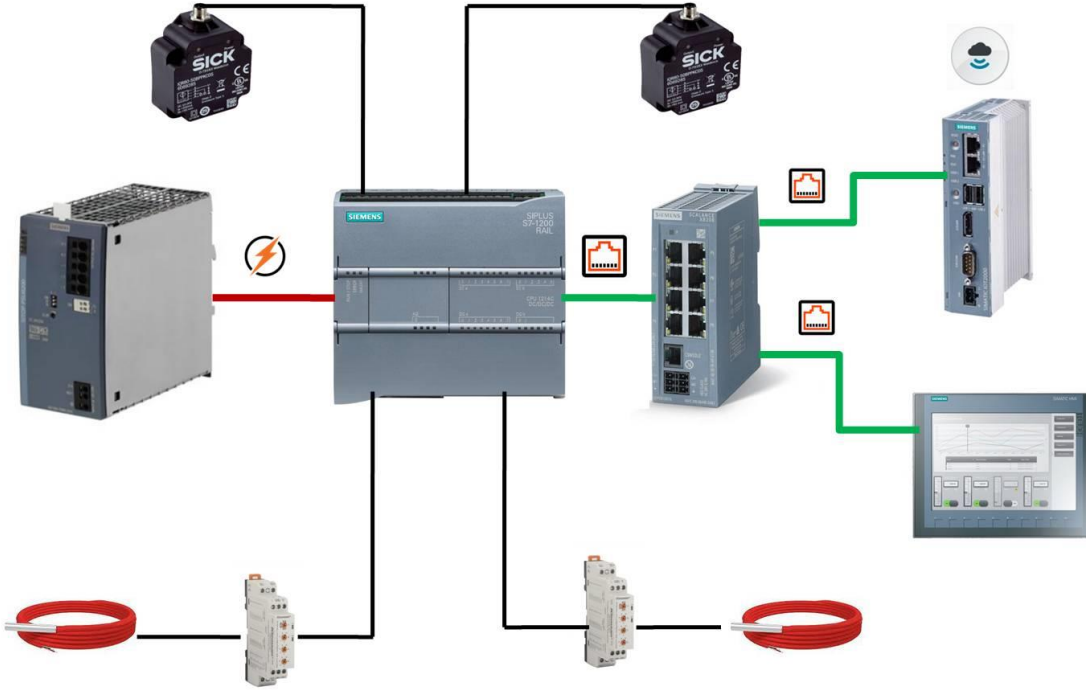
4.2.1. Otomasyon

Ölçüm yapan algoritmanın kodlandığı mikro kontrolör olarak endüstride yaygın olarak kullanılan donanımlardan biri olan PLC cihazı seçilmiştir. Endüktif sensörler ve sıcaklık sensörleri girişleri bu cihaza bağlanmıştır.

4.2.1.1. Donanım

Endüktif sensörler dijital sinyal, sıcaklık sensörleri ise analog sinyal göndermektedir. Bu sinyaller PLC içerisindeki algoritma tarafında işlenip bilgiye dönüştürülmektedir. Otomasyon mimarisinin tamamı 24 Volt gerilim ile çalışmakta olup koruma sınıfları IP65 ile IP68 arasında değişmektedir. Demiryollarının zorlu koşulları öngörülerek uzun zaman boyunca sorunsuz çalışma hedeflenmiş bu nedenle endüstriyel bir mikro kontrolör olan PLC cihazında karar kılınmıştır.

Hat üzerinden 220 V gerilim alınmış ve bir sigortadan geçirilerek 220 VAC/24 VDC 10 A monofaze bir doğrultucuya verilmiştir. Bu doğrultucudan çıkan 24 V gerilim pano içerisindeki tüm cihazlara dağıtılmıştır. Doğrultucudan sonra PLC, uç cihaz, ethernet cihazı, HMI, endüktif sensörler ve sinyal dönüştürücülerin önünde cam sigortalar kullanılarak pano içi olası kısa devre senaryolarına karşı önlem alınmıştır.



Şekil 4.5 Otomasyon sistemi (Siemens, 2021)(Sick, 2021).

Şekil 4.5 sahaya yerleştirilen otomasyon mimarisini göstermektedir. Otomasyon sistemi; AC/DC doğrultucu, PLC, endüktif sensörler, sinyal dönüştürücüler, PT100 sıcaklık sensörleri, ethernet cihazı, HMI ve uç cihazdan oluşmaktadır.

PLC cihazı içerisinde iki adet analog giriş, 14 adet dijital giriş ve 10 adet dijital çıkışa sahiptir. İlk 6 dijital giriş hızlı sayıcı girişidir. Hızlı sayıcı girişlerinin 100 kHz algılama kapasitesi mevcuttur. Endüktif sensörler bu girişlere bağlanmıştır. Cihazın program hafızası 50 KB, dijital çıkışların akımı 2 A değerindedir. Dijital girişleri μ s mertebesinde filtre edilebilmektedir. Analog çözünürlüğü 10 bit değerindedir. TCP/IP haberleşme portu bulunmaktadır (Siemens, 2021).

Endüktif sensörler algılama mesafesi içerisinde bulunan metalleri algıladığı anda dijital sinyal üreten elektronik cihazlardır. Seyir halindeki demiryolu araçlarının boden adı verilen tekerlerindeki çıkıntıyı algılamak üzere ray mantarının hemen altına yerleştirilen endüktif sensörler ile tren dingil sayıları tespit edilmektedir.

Kullanım için seçilen endüktif sensörler 50 mm lagılama mesafesine sahip PNP düz kafa sensörlerdir. Normalde açık ve normalde kapalı anahtarlama seçenekleri içerir. Normalde açık sinyal kullanılmıştır. 250 Hz anahtarlama frekansı mevcuttur. IP68 koruma sınıfına sahiptir. 80mm x 40mm x 114mm ölçülerindedir (Sick, 2021).

Endüktif sensörler, hızlı sayıcı dijital girişlerine bağlanmıştır. Sensörler 250 Hz frekansa sahiptir, hızlı sayıcı girişleride 100 kHz frekans değerine kadar anahtarlama yapılabilmektedir. Dolayısı ile otomasyon sistemi yüksek hızlı tren hatları da dahil olmak üzere tüm demiryolu hatlarında dingil sayımı yapabilecek kapasitededir.

Ray sıcaklığını algılamak için PT100 sensörler kullanılmıştır. Direnç artışına dayalı olarak tepki veren sensörlerdir. Kabloları istenen ölçülerde uzatılabilir. Bu durum otomasyonu sisteminin hattın farklı noktalarına konuşlandırmada oldukça kolaylık sağlar. PT100 sensörler iki, üç ve dört telli tiplerdedir. Ray sıcaklığı ölçümünde yüksek hassasiyet aranmadığından maliyet açısından daha uygun olan iki telli PT100 sensörler seçilmiştir.

PT100 sinyalini PLC cihazı doğrudan işleyemediği için sensörler ile PLC arasında sinyal dönüştürücü kullanılmıştır. -150 ile 800 °C sıcaklıkları işleme kabiliyetindedir. 0-5 V, 0-10 V, 0-20 mA, 4-20 mA analog sinyal seçenekleri bulunmaktadır. Bu seçenekler arasından 0-10 V analog sinyal tercih edilmiştir (Klemsan, 2021).

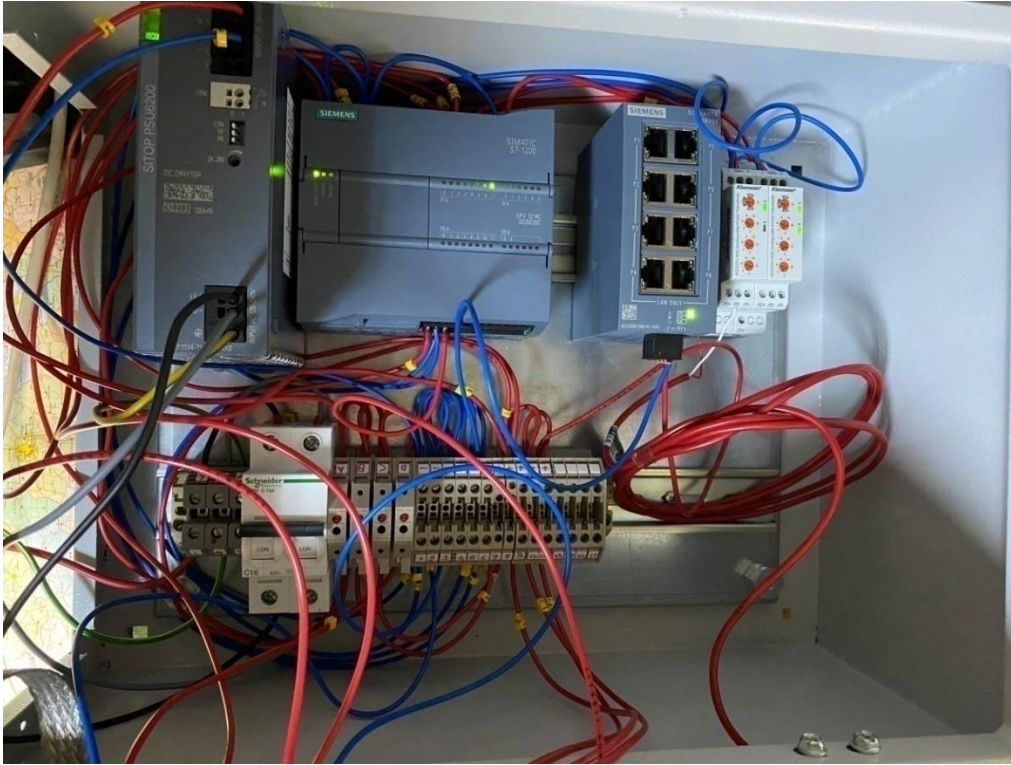
Otomasyon sistemi internet üzerinden kullanıcıya açılmak üzere programlanmıştır. Ancak sadece internet üzerinden değil doğrudan bir operatör paneli ile de kontrol edilebilecek şekilde geliştirilmiştir. 7 inç boyutunda bir HMI panel ile pano üzerinde kontrol sağlanmıştır.

HMI panel 7 inç büyüklüğünde IP65 koruma sınıfında 800 x 480 piksel çözünürlükte dokunmatik bir ekrandır (Siemens, 2021). PLC cihazından bu panele açılan parametreler kontrol edilerek PLC cihazına direktif verilmektedir. Panel ile PLC cihazı ethernet üzerinden haberleşmektedir.

Uç cihaz Linux işletim sistemi çekirdeğine sahip olup Debian tabalı bir Linux dağıtımı kullanan endüstriyel tip bir bilgisayardır. 1 GB işlemciye ve RAM belleğe sahiptir. Sabit bellek ile hafıza alanı genişletilebilir. SIM kart yuvası mevcuttur. Ethernet girişlerine sahip ile TCP/IP protokolüyle haberleşmeye uygundur (Siemens, 2021).

Eternet cihazı ise PLC, uç cihaz ve HMI panelin birbiriyle haberleşmesi sağlayan bir dağıtıcı görevi görmektedir. Bu cihaz 8 adet eternet girişine sahip olup aynı IP alanında bulunan 8 cihazı aynı anda birbirine bağlayabilecek kapasitededir.

Sensörler sahada konuşlandırılırken elektromanyetik gürültünden korunabilmek için ekranlı kablolar kullanılmış ve bu kabloların ekranları topraklanmıştır. Böylece sinyal güvenliği güvence altına alınmıştır.



Şekil 4.6 Pano imalat sürecinden bir görüntü



Şekil 4.7 Otomasyon sistemi panosu

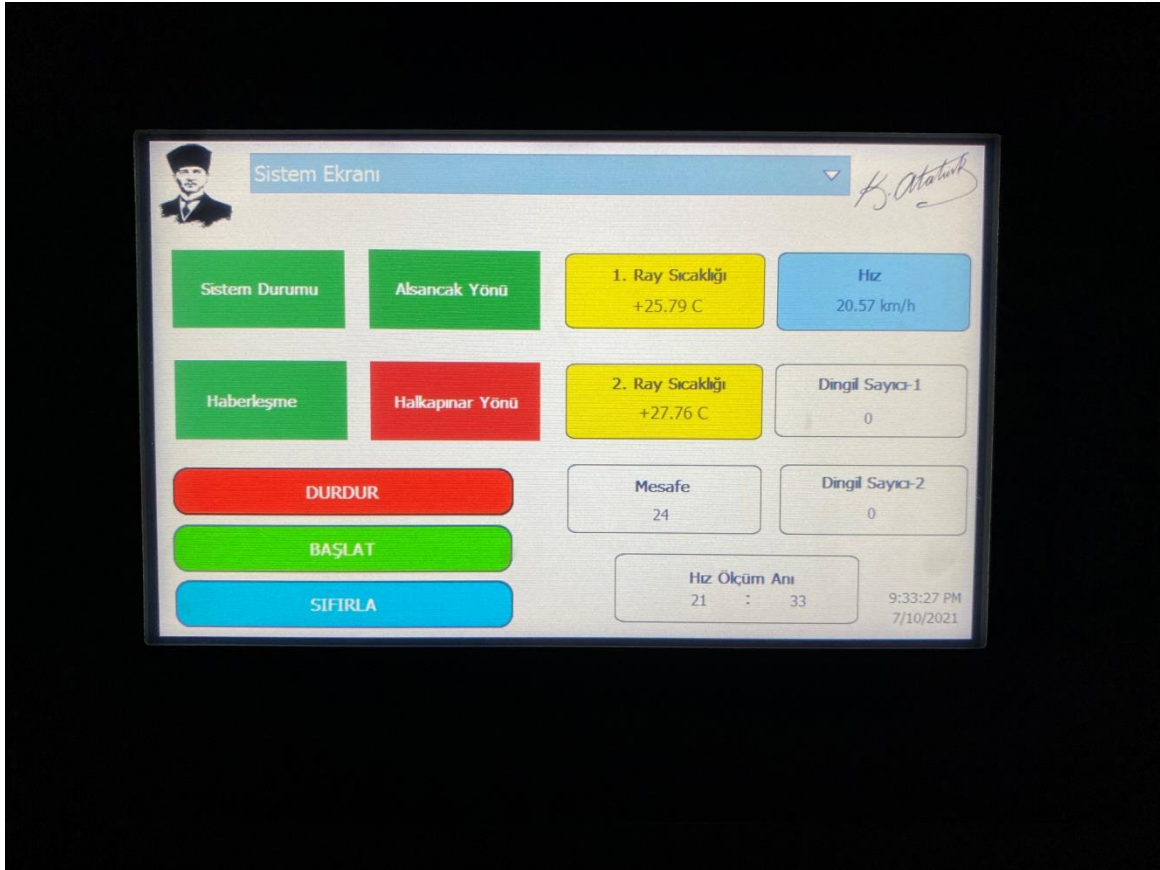
Şekil 4.6 pano imalat sürecini gösteren bir görüntüdür. Otomasyon sistemi ve HMI panel tek bir pano üzerine yerleştirilmiştir. Şekil 4.7 ise panonun devreye alınmış halini göstermektedir. Başlat düğmesi sistemi devreye alır. Durdur düğmesi ise sistem içerisindeki bütün fonksiyonları durdurur. Sıfırla butonu kullanıcıya, endüktif sensörlerin bağlı oldu hızlı sayıcı değerlerini sıfırlama imkanı sunar. Sistem ışığı ise sistemin devrede olduğunu gösteren bir ikazdır. Haberleşme anahtarı sistemi sistemin verilerinin internet ortamına iletiminin kararını kullanıcıya bırakır.



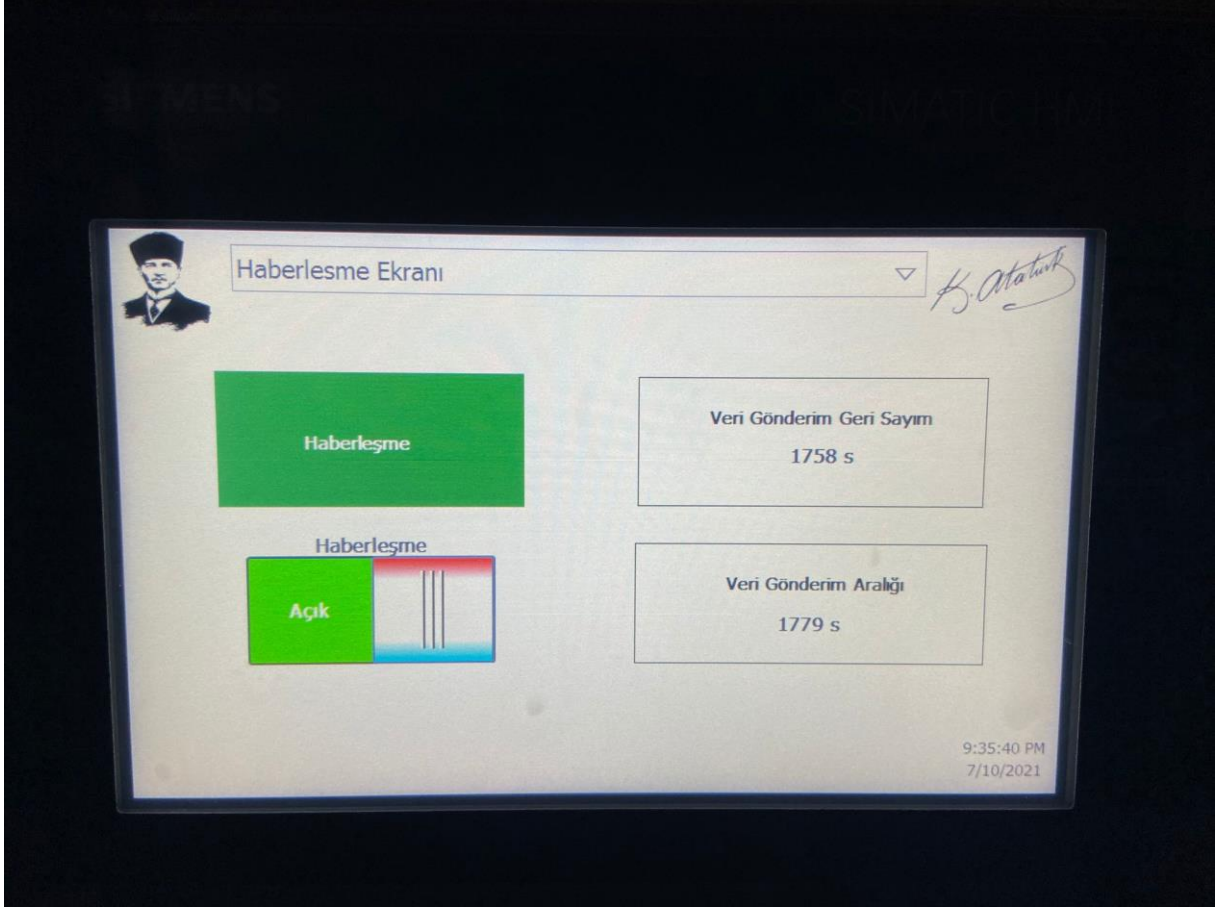
Şekil 4.8 Otomasyon sistemi saha dolabı

Daha sonra imalatı tamamlanan bu pano demiryolu hattının yaklaşık 5 metre kenarında bir beton kaide üzerine monte edilen ve kendi içerisinde fan sistemi bulunan Şekil 4.8’de gösterilmiş bir dolap içerisine yerleştirilmiştir. Dolap içerisindeki fan sistemi sıcaklık ayarına sahip olduğundan sistem bileşenlerinin soğutulması için PLC cihazı içerisine bir algoritma yazılmasına gerek görülmemiştir.

Diğer taraftan söz konusu dolap topraklama işlerinde bu dolap şase olarak kullanılmış olup sinyal güvenliğine katkıda bulunmuştur. Dolaba giden kablolar spiral kılıflarla örtülerek dış ortamdan yalıtılmıştır.



Şekil 4.9 Operatör paneli sistem ekranı



Şekil 4.10 Operatör paneli haberleşme ekranı

Operator paneli için tasarlanan ekranlar Şekil 4.9 ve Şekil 4.10 ile gösterilmiştir. Kullanıcı dostu olması açısından sistemin kullanıcıya açılan değişkenleri sistem ekranı ve haberleşme ekranı adıyla iki ayrı alanda gruplandırılmıştır.

Sistem ekranı üzerindeki başlat, durdur ve sıfırla düğmeleri, pano üzerine konumlandırılmış başlat, durdur, sıfırla düğmeleriyle aynı işlevi görmekte ve paralel çalışmaktadır. Pano üzerindeki düğmelerin çeşitli nedenlerle işlevlerini kaybetmesi durumunda sistem operatör panel üzerinden idame ettirilebilir. Sistem durumu göstergesi pano üzerindeki sistem ışığı ile aynı görevi görür. Başlat düğmesine basıldığında sistem durumu göstergesi aktif olur. Sistem durumu göstergesinin hemen altındaki haberleşme göstergesi, PLC cihazı içerisindeki verilerin internet ortamına aktarılmasını sağlayan işlevin devrede olup olmadığını gösterir. Hemen yan taraftaki iki gösterge ise demiryolu hattında seyir halinde bulunan ve endüktif sensörlerin sınırlandığı alana giriş yapan araçların yönlerini gösterir. Bunlar dışında sistem ekranında, ray sıcaklıklarını ölçen iki sensörün değerleri, endüktif sensörlerin algıladığı dingil sayıları, endüktif sensörler arasındaki mesafe,

endüktif sensörler üzerinden geçmiş olan son aracın hızı, ve hız ölçümünün yapıldığı an gösterilmektedir.

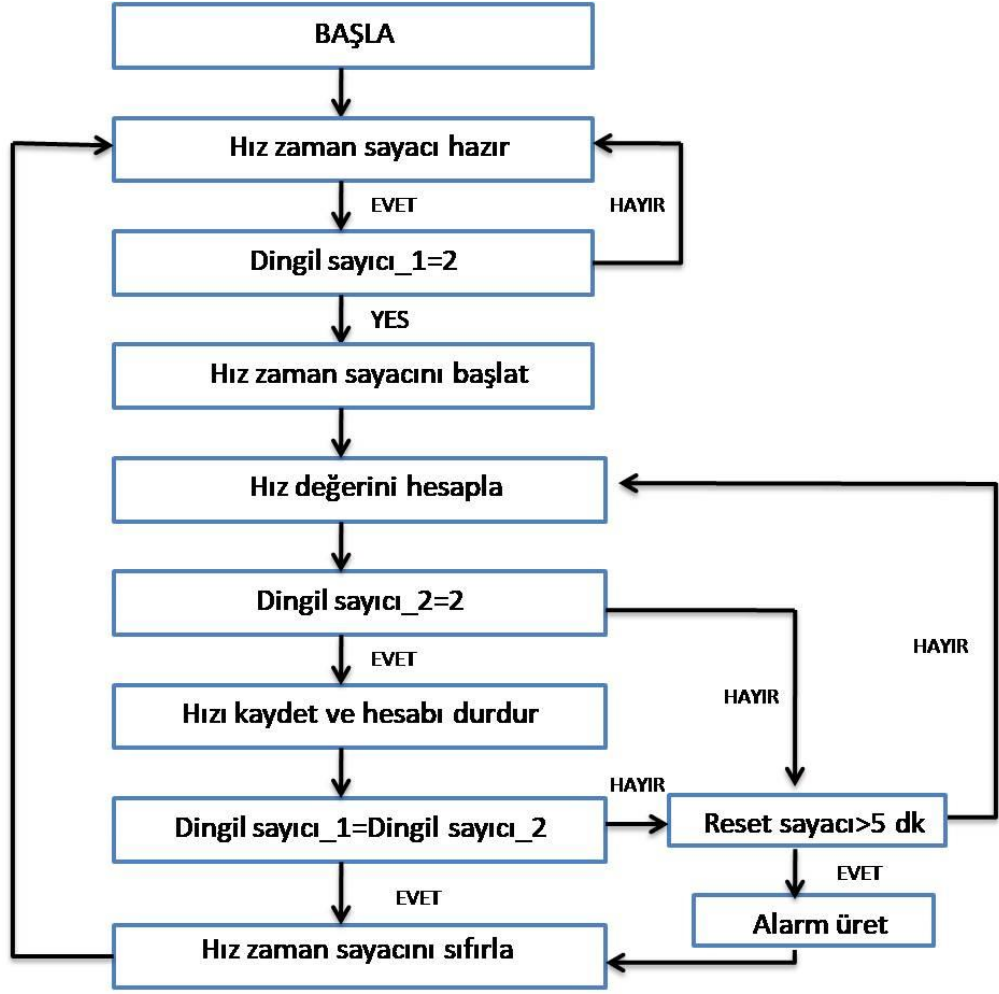
Operatör panelindeki diğer bir ekran ise haberleşme ekranıdır. Bu ekran üzerinde sistem ekranında yer alan haberleşme göstergesi bulunmaktadır. Aynı zamanda pano üzerinde bulunan haberleşme anahtarının aynı işlevi gören ve paralel çalışan bir haberleşme düğmesi bulunmaktadır. Bu düğme sayesinde pano üzerindeki haberleşme düğmesi çeşitli nedenlerle işlevini yerine getiremez hale geldiğinde veri gönderimi panel üzerinden yürütülebilir. Veri gönderim aralığı göstergesi ise bir giriş-çıkış alanı olup kullanıcıya veri gönderim aralığı ayarlama yetkisi verir. Veri geri sayım göstergesi ise bir sonraki veri gönderimi için kalan süreyi gösterir.

Operatör paneli sayesinde kullanıcıya sistem üzerinde bazı yetkiler verilerek sistemin rijid yapısı esnetilmiş ve daha verimli hale getirilmiştir. Aynı veri erişiminin kullanıcıya kapalı olması sistemin güvenilirliğini garanti altına alınmıştır.

4.2.1.2. Algoritma

PLC içerisine TIA portal ara yüzü kodlanan algoritma mümkün olduğu kadar esnek ve işlevsel tasarlanmış olup uç cihaz ile TCP soket üzerinden haberleşmektedir. Endüktif sensörlerden gelen dijital sinyalleri, ve sıcaklık sensörlerinden gelen analog sinyalleri işleyerek, sistem durumu ile birlikte bir veri bloğu haline getirmekte ve TCP/IP protoklü ile uç cihaza iletmektedir.

PLC üzerinde çalışan algoritmalar tarafından üretilen verinin uç cihaza ve buradan internet ortamına aktarılması dijitalizasyonun bir parçasıdır. Ancak uç cihaz aynı zamanda otomasyon sisteminin de fiziki bir bileşeni olduğundan otomasyon kısmında ele alınmıştır. PLC içinde çalışan algoritmalar ölçüm yapıp elde ettikleri veriyi işleyerek uç cihaza iletir.

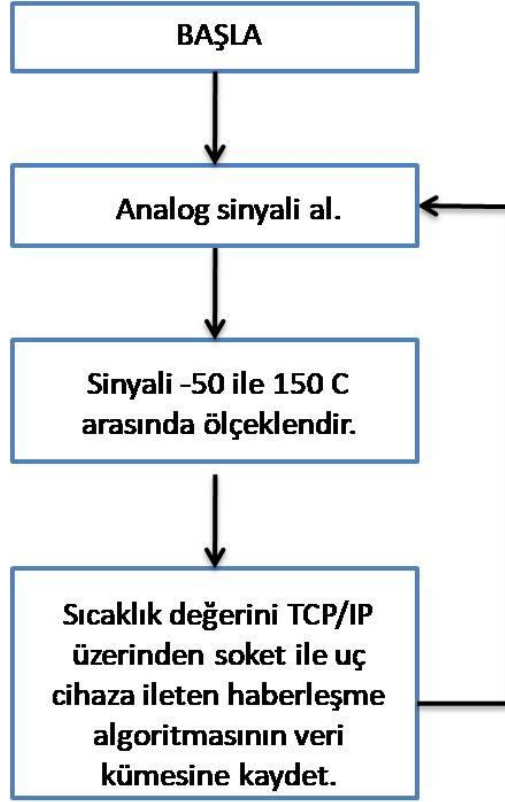


Şekil 4.11 PLC içerisinde çalışan hız ölçümü algoritması

PLC içerisinde çalışan hız ölçümü algoritması Şekil 4.11 ile gösterilmiştir. Endüktif sensörleri bağlı olduğu hızlı sayıcılar dingil sayıcı olarak adlandırılmıştır. Sistem başlat düğmesi ile aktif edildikten sonra demiryolu aracının ilk iki dingili algılandığında zaman sayacı çalışmaya başlar. Demiryolu aracının yönüne göre ilk dingil sayıcı 2 değerine ulaştığı anda hız ölçümü yapan o endüktif sensörün yönüne ait fonksiyon bloğu aktif olur zaman sayacı çalışmaya başlar. Diğer endüktif sensöre ait dingil sayıcı değeri de 2 olduğu anda bir zaman sayacı durur zaman değeri kayıt altına alınır ve zaman sayacı sıfırlanır. Endüktif sensörler arasında 24 m olarak belirlenen mesafe kayıt altına alınan zaman değerine bölünmektedir. Bulunan değer km/h birimine çevrilerek TCP soket ile uç cihaza iletilmekte ve aynı zamanda operator panelinde (HMI) gösterilmektedir.

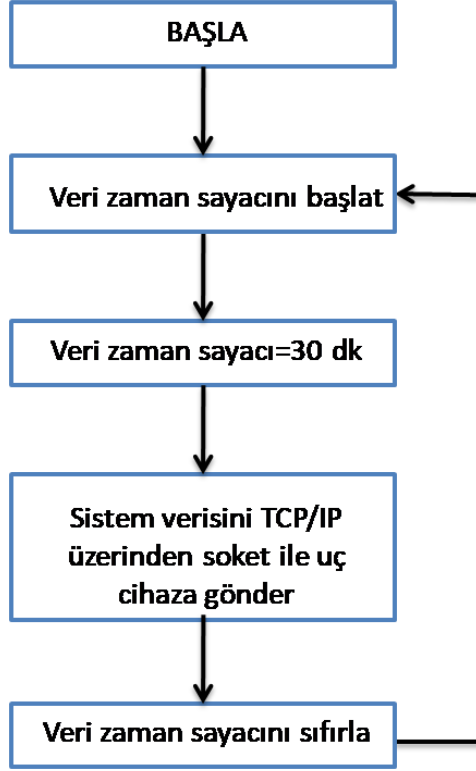
Hız ölçümü algoritmasının dingil sayıcıların 2 değerine ulaştığında tetiklenmesi nedeni sensörler demiryolu aracı tekeri dışında olağan olmayan bir cisim algılandığında hız ölçümü yapılmasını engellemektir. Demiryolu araçlarının tamamı en az iki dingil içerdiğinden hat üzerinde bu algoritmayla ölçülen hız değerinin bir demiryolu aracına ait olduğu kesindir.

Endüktif sensörlerin ayırdığı 24 m uzunluğundaki bölgede demiryolu aracı duruşu gerçekleşirse iki dingil sayacı değerleri birbirine eşit olmadığı için dingil sayıcı sıfırlaması gerçekleşmeyecektir. Bu durumda algoritma içerisinde bulunan bir zamanlayıcı 5 dk bekledikten sonra dingil sayıcıları sıfırlayıp isteğe bağlı alarm üretir. Duran araç 5 dk içerisinde ilerlemeye devam edip söz konusu bölgeden çıkarsa algoritma kaldığı yerden çalışmaya devam eder.



Şekil 4.12 PLC içerisinde çalışan sıcaklık ölçüm algoritması

Şekil 4.12 ile gösterilen ray sıcaklığı ölçüm algoritması ise ray sıcaklığı değerlerini ölçen sensörlerden gelen analog sinyaller sürekli olarak işler ve -50 ile 150 °C arasında ölçekler ve bu değerleri, TCP soket ile uç cihaza ileten haberleşme algoritmasının veri kümesine kaydeder. Ölçülen sıcaklık verileri arasındaki fark 2234 veri alındıktan sonra incelendiğinde iki ölçüm arasındaki farkın 1 ile 2.25 °C arasında değiştiği görülmüştür. Bu fark, demiryolu hattında tehlike oluşturabilecek maksimum ve minimum sıcaklık değerleri göz önüne alındığında oldukça küçüktür. Dolayısıyla tolerans sınırları içerisinde kalır.



Şekil 4.13 PLC içerisinde çalışan haberleşme algoritması

Şekil 4.13 ile gösterilen haberleşme algoritması ölçümü yapılan ray sıcaklığı değerlerini, hesaplanan hız ile anlık dingil sayıcı değerlerini ve demiryolu aracının yönünü, sistemin bazı diğer değerlerini TCP/IP soketi açarak uç cihaza iletir. Veri iletim aralığı pano üzerinde bulunan operatör paneli (HMI) ile ayarlanabilir. Algoritmada gösterilen 30 dakikalık aralık cihaz devreye alındığı andaki başlangıç değeridir. Her veri iletiminden sonra haberleşme zaman sayacı sıfırlanır ve böylece haberleşme çevrimi devam eder.

Veri	Veri Tipi
Başlat	Boolean
Yön	Boolean
Ray Sıcaklığı 1	Double
Ray Sıcaklığı 2	Double
Hız	Double
Dingil Sayıcı 1	Integer
Dingil Sayıcı 2	Integer

Tablo 4.3 Otomasyon sisteminde üretilen veri yapısı

Otomasyon sisteminde üretilen veri Tablo 4.3 ile gösterilmiştir. Sistemin çalışır durumda olduğu ve hızı ölçülen demiryolu aracının geldiği yön bir bitlik değer tutan boolean

veri tipiyle, ray sıcaklıkları 32 bitlik reel sayı değeri tutan double veri tipiyle, dingil sayıları ise 32 bitlik tamsayı değeri tutan integer veri tipiyle eşlenmiştir.

Uç cihaz ise PLC cihazından TCP soketiyle aldığı Tablo 4.3 ile gösterilen veriyi TCP protokolü üzerinde çalışan MQTT protokolü ile internete yayımlar. MQTT de TCP soketleri TCP üzerinde çalışır. Ancak ulaşım katmanından sonrafarklılaşma meydana gelir. MQTT ile TCP soketlerinin ISO standartlarının sunduğu OSI modelinde referans modeli ele alındığında ulaşım katmanından sonra farklılaştığı görülür.

Katman Sırası	Katman	Açıklama
7	Uygulama	Kullanıcı ara yüzü katmanıdır. HTTP, MQTT gibi yapılarla çalışır.
6	Sunum	Veriyi dönüştüren katmandır. ASCII, MPEG gibi yapılarla çalışır.
5	Oturum	Ağ üzerindeki cihazların birbiriyle iletişim halinde kalmasını sağlayan katmandır. Soketler, SMPP vb. yapılarla çalışır.
4	Ulaşım	Gönderilecek veriyi parçalara bölen ya da paketleyen katmandır. TCP, UDP, SPX vb. yapılarla çalışır.
3	Ağ	Farklı bir ağa gönderilecek olan veriye yönlendiricilerin tanıyacağı veri paketi bilgisini ekleyen katmandır. TCP/IP, IPX vb. yapılarla çalışır.
2	Veri	Ağ katmanından donanım katmanındaki cihazlara MAC ve LLC ile erişimi belirleyen katmandır. Ethernet, Wi-Fi, HDLC, PPP vb. erişim yapısı ile çalışır.
1	Donanım	Sinyalin taşındığı fiziksel yapıdır. (Ethernet kablosu vb.)

Tablo 4.4 OSI referans modeli (Muşkinja et al., 2003)

OSI referans modeline göre kullanıcının ürettiği veri 7. katmandan itibaren aşağı doğru iletilirken kullanıcının cihazına gelen veri 1. katmandan itibaren yukarı doğru iletilir. Burada ilk katman donanım katmanıdır ve tamamen fiziksel bir yapıdan oluşup sinyal taşır. Bir sonraki katman veri katmanıdır. Burada donanım katmanından gelen sinyali MAC ve LLC ile cihazlara eşleyen katmandır. Ağ katmanı farklı bir ağa gönderilecek olan veriye yönlendiricilerin tanıyacağı veri paketi bilgisini ekleyen katmandır. Ulaşım katmanı ise veriyi parçalara bölen veya paketleyen katmandır. Oturum katmanı bir ağ üzerindeki cihazların birbiriyle iletişim halinde kalması sağlar. Sunum katmanı ise veri byte dizilerinden daha anlaşılır yapılara dönüştüren katmandır. Son olarak uygulama katmanı kullanıcıya en yakın ve kullanıcının veriyi üretip gözlemlediği katmandır (Muşkinja et al., 2003).

OSI referans modeli ele alındığında uç cihazın PLC cihazıyla haberleşmesini sağlayan TCP soketlerin oturum katmanı üzerinde çalıştığı uç cihazın internete veri yayımladığı protokol olan MQTT protokolünün ise uygulama katmanında çalıştığı görülür.

MQTT, nesnelerin internetiyle bağlı cihazların haberleşmesinde fiili bir standart haline gelmiş bir haberleşme protokolüdür. ISO tarafından standardize edilmiştir. Geleneksel protokol yapılarının aksine yayımcı/abone yapısında çalışır. Bir başlığa abone olan tüm cihazlar aynı başlığa yayın yapan tüm cihazların ilettiği verileri aynı anda alabilir ve bir cihaz aynı anda hem yayımcı hem abone olabilir. Bu sistemli ölçeklendirilebilir kılar ve büyük miktarda veri aktarımını eş zamanlı olarak mümkün hale getirir. Oldukça hafif bir protokoldür ve veri aktarım bakımından güvenilirdir (HiveMQ, 2021).

MQTT, cihazlar ile aracı arasında sürekli bir oturum açar. Söz konusu bu oturumun güvenliği bir kullanıcı adı ve parola ile sağlanır. Ağ bağlantısı kopması oturum bilgileri saklı kalır. Ağa yeniden bağlandıktan sonra, cihazların aracıya yeniden bağlanır böylece oturum devam eder (HiveMQ, 2021).

MQTT protokolü yayımcılar, aboneler ve bir aracıdan oluşur. Yayımcılar aracı üzerindeki bir başlığa veri gönderir bu veri aracı tarafından tüm abonelere iletilir. İletim sürecinde veri iletim güvenliğini QOS değeri belirler. Bu değer 0 olduğunda veri bir kez gönderilir ve iletilip iletilmediğine bakılmaz. Değer 1 olduğunda verinin en az bir kez dağıtımı yapılır. Bu veri gönderimi kesinleştirirken aynı verinin birden fazla kez iletilmesine neden olabilir. Değer 2 olduğunda ise veri kesinlikle sadece bir kez iletilmiş olur. Bu veri iletim güvenliği açısından son derece önemlidir. Burada dikkat edilmesi gereken husus uygun QOS seçimidir. QOS seviyesi yükselirken gecikmenin de yükseldiği ve hızın bir miktar düştüğü göz önüne alınmalıdır (HiveMQ, 2021).

Veri iletim hızı ile CPU ve hafıza kullanımı ele aldığından MQTT protokolü diğer haberleşme protokollerine göre çok daha düşük gecikmeleriminimum hafıza ve CPU kullanımı ile sağlamaktadır (Naik, 2017).

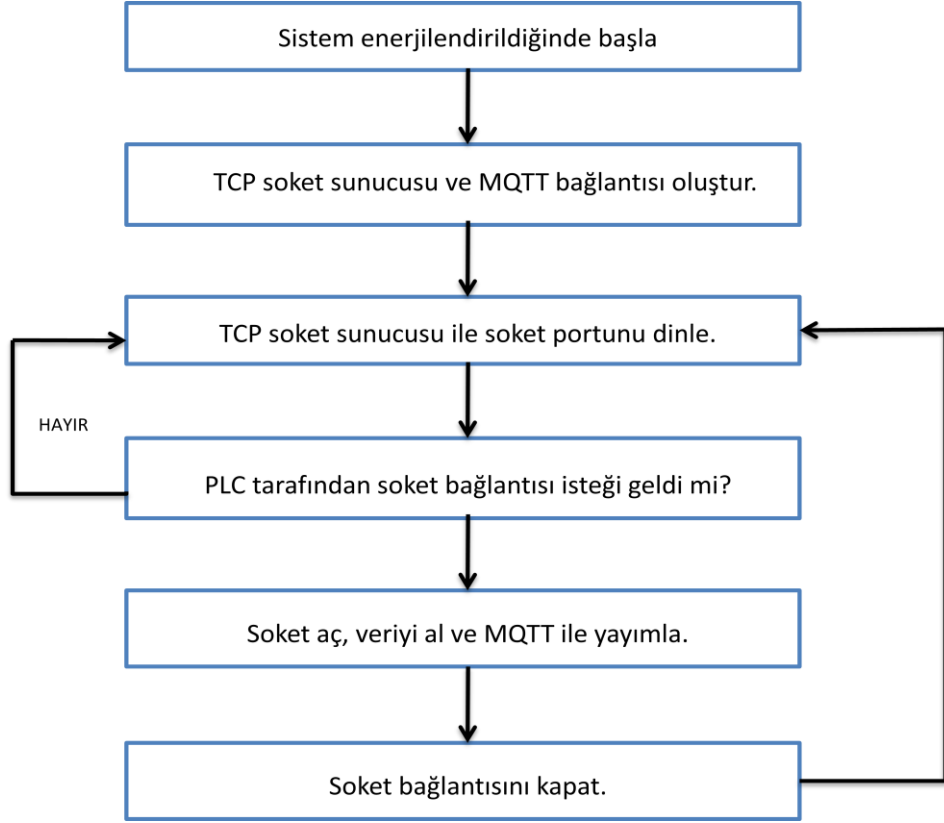
Otomasyon sisteminden internete veri aktarmak için güvenilirliği, hızı ve ölçeklenebilirliği nedeniyle MQTT protokolü tercih edilmiştir. QOS değeri ise 0 olarak ayarlanmıştır. QOS 2 mutlaka iletilmesi gereken kritik veriler için kullanılmaktadır ve bağlantı kopup geri geldiğinde iletilmeyen veri iletilmiş olur. Bu teze konu olan sistemde iletilen veri kritik değildir. Sunucu tarafında hava koşulları verileri, çekildiği andaki zaman değeriyle birlikte sahada bulunan otomasyon sisteminden gelen verilerle birleştirilerek veri tabanına yazılır. Dolayısıyla olası bağlantı kopmalarından sonra iletilenmemiş verinin tekrar

gönderilmesi uygun olmayacaktır. Bu nedenle QOS 2 kullanılmamıştır. QOS 1 değerinin tercih edilmemesinin nedeni ise aynı verinin mükerrer olarak veritabanına yazılmasının istenmemesidir. Bu veri iletim aralığı düştüğünde veri tabanının mükerrer veriler ile genişlemesine neden olur ve uygulanan yapay zeka algoritmasının hızını düşürür. QOS 0 ile veri yalnızca bir kez ve daha hızlı gönderilmesi tercih edilmiştir.

Bu tez mimarisi içerisinde hem ön uç hem arka uç hem de uç cihaz yazılımı geliştirilmiştir. Arka uç yazılımı ile uç cihaz yazılımı temelde birbirine benzer yapıda yani sunucu görevi ifa eden yazılımlardır. Uç cihaz, PLC için sunucu görevi görürken arka uç ön uç için sunucu görevi görür. Ön uç için javascript dili en yaygın kullanılan programlama dili olduğundan hızlı geliştirme yapabilmek için arka uçta da javascript üzerinde çalışan bir çalışma zamanı ortamı olan node.js teknolojisi tercih edilmiştir. Javascript programlama dili yorumlanan betik bir dildir. Node.js ortamının tercih edilmesinin diğer bir nedeni nesnelerin interneti teknolojilerine olan uygunluğudur.

Node.js sunucu tarafında çalışan bir javascript ortamıdır. Google tarafından geliştirilen V8 motoru üzerinde çalışır. Hızlı ve düşük hafıza kullanıma sahip bir teknolojidir. Diğer programlama dillerinin aksine çok iş parçacıklı bir yapı yerine tek iş parçacıklı asenkron giriş-çıkış olay odaklı bir yapıya sahiptir. Tek iş parçacıklı olması tek CPU çekirdeği kullanır ancak kümelenme kullanılarak veya alt prosesler eklenerek yüksek hacimli CPU işlerinde paralel çalıştırma yapılabilir. Elbette bu paralel çalıştırma yine asenkron olacağından verim artırılırken eş zamanlılık sağlanır. Node.js bu özellikleri sayesinde yüksek eşzamanlılık performansı sağlar. Üzerine gelen istekleri kuyruğa alır ve asenkron olarak işler. Tek iş parçacıklı ve asenkron çalıştığından bir istek üzerindeki işlem bir diğerini engellemez. Bu nedenle istek bloklamayan bir yapısı mevcuttur. Bu durum TCP, HTTP, MQTT gibi yüksek giriş çıkış yükleri içeren haberleşmeler için idealdir. Yüksek performans ve ölçeklenebilirlik sağlar. Özellikle ilişkisel olmayan veri tabanlarıyla kullanıldığında nesnelerin interneti için oldukça yüksek performans gösterir. Bu nedenle oldukça yaygın olarak kullanılır. En büyük eksikliği ise yüksek CPU yükleri gerektiren işlemlerde ortaya çıkar. Bu tür işlemler gecikmeleri artırır. Bu nedenle yüksek CPU yükü gerektiren işlemler için çok sık kullanılmaz. Yine aynı nedenle ilişkisel veri tabanlarından çok ilişkisel olmayan veri tabanlarıyla kullanılır. Söz konusu bu eksiklik kümelenme ya da alt süreç ayarları ile çoklu CPU çekirdeği kullanılarak aşılabilir (Tilkov and Vinoski, 2010).

Node.js ortamında javascript dili ile geliştirilmiş uç cihaz algoritması PLC için bir sunucu olarak çalışır. PLC içerisinden gelecek veriyi devamlı olarak bir TCP soket yardımı ile dinler. Soket isteği geldiğinde soket açar veriyi çektikten sonra ise soketi kapatıp belirlenen portu dinlemeye devam eder.



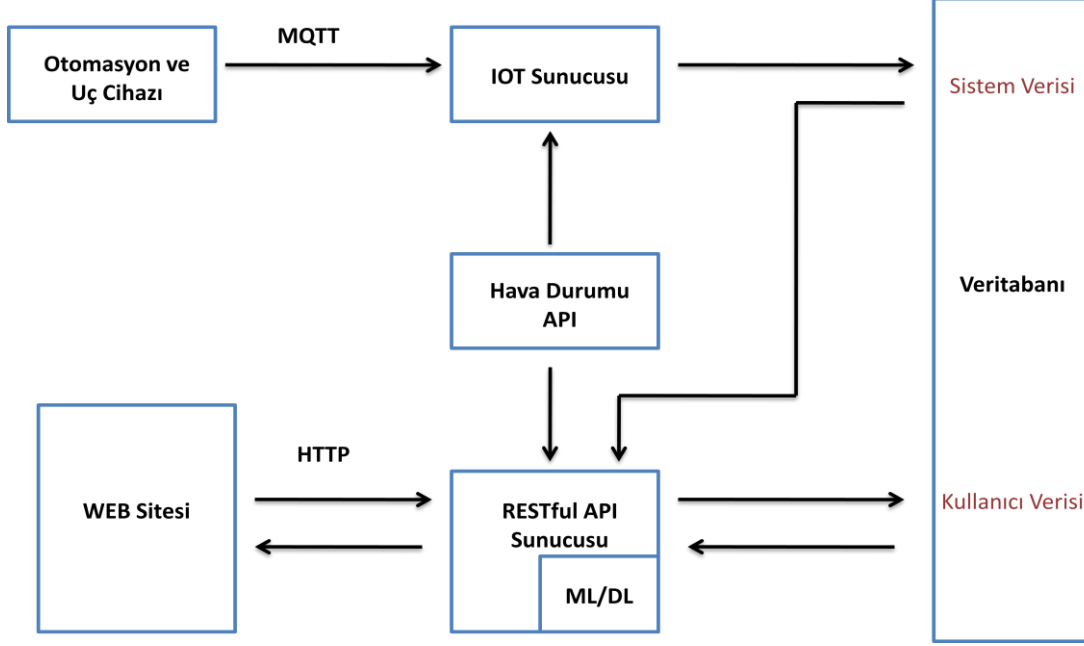
Şekil 4.14 Uç cihaz algoritması

Uç cihazın görevi sahadaki otomasyon sisteminin ürettiği veriyi arka uçta çalışan sunucuya iletmektir. Bunun için PLC içerisindeki veriyi TCP socket vasıtasıyla alır ve MQTT protokolüyle yayımlar. Şekil 4.14 uç cihaz içerisinde bulunan algoritmayı betimlemektedir. Uç cihaz enerjilendiği anda cihaz içerisinde gömülü olan program çalışmaya başlar ve bir TCP socket sunucusu oluşturur. Hemen ardından MQTT protokolü ile aracıya bağlanarak oturum açar. Sonrasında TCP socket, belirlenen portu dinlemeye başlar. PLC tarafından socket bağlantısı isteği geldiğinde socket açar ve gelen veriyi yakalar. Bu veriyi MQTT ile yayımladıktan sonra socket bağlantısını kapatır ve belirlenen portu dinlemeye devam eder.

Sahaya kurulan otomasyon sisteminin ürettiği verinin internete çıkışı ile dijitalizasyonun ilk kısmı tamamlanmış olur. Uç cihaz hem sahadaki otomasyon sisteminin hem de dijitalizasyonun bir parçasıdır. Bu noktadan sonraki geliştirmeler sanal ortama ait olacaktır. Dolayısıyla tamamiyle dijitalizasyonu temsil eder.

4.2.2. Dijitalizasyon

Dijitalizasyona ilk adım sahada konumlanmış otomasyon sisteminin bir parçası olan uç cihaz ile başlar. Uç cihazın internete taşıdığı veriyi arka uçta çalışan sunucular işleyerek ön ucu oluşturan bir web sitesinde kullanıcıya açar.



Şekil 4.15 Dijitalizasyon şeması

Şekil 4.15, sahada konuşlanmış otomasyon sisteminin ürettiği verinin dijital haritasını sunmaktadır. Uç cihazdan MQTT protokolüyle yayımlanan sistem verisinin aracı üzerindeki başlığına IOT sunucu abone durumdadır. Dolayısıyla bu veriyi sürekli dinlemektedir. Sistem verisi yayımlandığında bu veriyi yakalar ve aynı anda hava koşullarıyla ilgili bilgileri de bir dış uygulamadan alır. Bu iki veri grubunu birleştirip işleyerek ilişkisel olmayan bir veri tabanına kaydeder. Bu işlem belirli aralıklarla sürekli olarak gerçekleşir.

Diğer taraftan veri tabanıyla bağlantı kuran REST sunucusu ise ön uçta çalışan bir web sitesinden gelen istekleri yanıtlar. Aynı zamanda veri tabanına kullanıcı kayıtlarını yazar; oturum açma, oturum doğrulama gibi işlemleri yapar ve parola hareketlerini kontrol eder. Web sitesinden gelen kullanıcı isteklerine cevap vererek veri alış verişi sağlar. Ayrıca içerisinde bir alt process olarak çalışan yapay zeka algoritması ile zamana dayalı ray sıcaklığı tahmini yapar. Uç cihaz ile IOT sunucusu arasında MQTT protokolü, web sitesi ile REST sunucu arasında http protokolü ile iletişim sağlanır.

4.2.2.1. Arka Uç

Arka uç, sunucuları ve veritabanını kapsayan mimaridir. Arka uçta üç farklı sunucu çalışmaktadır. Bunlardan IOT ve REST sunucuları javascript programlama dili ile node.js ortamında geliştirilen özgün sunucular iken hava durumu bilgilerinin alındığı sunucu ise kayıt olduğu takdirde kullanılabilen ve herkese açık olan hazır bir sunucudur. Geliştirilen IOT ve REST sunucular ortak bir veritabanına bağlanır. Bu, ilişkisel olmayan ve MongoDB olarak adlandırılan bir veri tabanıdır.

MongoDB ilişkisel olmayan bir veri tabanıdır. İlişkisel veri tabanlarıyla ilişkisel olmayan veri tabanları arasında belirgin farklar mevcuttur. İlişkisel veri tabanların katı bir şema yapısı vardır. Bu şema yapıları belirgin tablolar oluşturur. Veri tabanına yazılan kayıtlar bu tablo yapısının dışına çıkamaz bu nedenle dikey ölçeklenebilirliği vardır. ACID kurallarıyla çalışmasından dolayı kritik verilerin bulunduğu sistemler için idealdir. İlişkisel olmayan veri tabanlarına kıyasla yavaştır. İlişkisel olmayan veri tabanları ise anahtar-değer, dokuman gibi şemadan bağımsız esnek yapıya sahip sistemlerdir. Her dokuman kendi içerisinde farklı bir yapı tutabilir. Dikey ve yatay ölçeklenebilirlikleri mevcuttur. İlişkisel veri tabanlarına kıyasla önemli derecede hızlıdır. Bu nedenle nesnelerin interneti ve büyük veri uygulamalarında yaygın olarak kullanılır. Operasyonlarda ACID kurallarını kullanmadığından kritik verilerin bulunduğu yerlerden çok kritik olmayan büyük verinin bulunduğu sistemlere daha uygundur (Čerešňák and Kvet, 2019).

VERİ TABANLARI PERFORMANSI (ms)									
İŞLEM SAYISI	MongoDB			Cassandra			MSSQL		
	Okuma	Yazma	Silme	Okuma	Yazma	Silme	Okuma	Yazma	Silme
10	8	61	4	115	117	33	13	30	11
50	14	75	15	230	160	95	23	94	32
100	23	84	29	354	212	130	46	129	57
1000	138	387	235	2385	1200	1061	277	1790	360
10000	1085	2693	2115	19758	9801	9230	1968	15588	3571
100000	10201	23354	18688	228096	88197	83694	17214	216479	32741

Tablo 4.5 İlişkisel ve ilişkisel olmayan veri tabanlarının hız karşılaştırması (Li and Manoharan, 2013)

Tablo 4.5, yaygın olarak kullanılan ilişkisel ve ilişkisel olmayan veri tabanlarının performansını göstermektedir. MSSQL ilişkisel bir veri tabanı iken Cassandra ve MongoDB ilişkisel olmayan veri tabanlarıdır. MongoDB diğerlerine göre gözle görülür bir performans farkı sergilemektedir (Li and Manoharan, 2013).

VERİ TABANLARI PERFORMANSI (s)								
İŞLEM SAYISI	MongoDB				PostgreSQL			
	Okuma	Yazma	Güncelleme	Silme	Okuma	Yazma	Güncelleme	Silme
30000	0,81	6,24	0,2	0,96	1,02	100,51	0,4	6,92
90000	2,58	20,65	0,54	3,51	2,82	301,04	1,15	20,74
150000	4,18	35,96	1,34	5,78	4,86	502,38	2,08	59,46
210000	5,95	50,34	1,22	7,16	6,59	705,87	2,69	123,41
3000000	8,55	69,8	2,66	11,65	8,78	1003,51	3,28	342,85

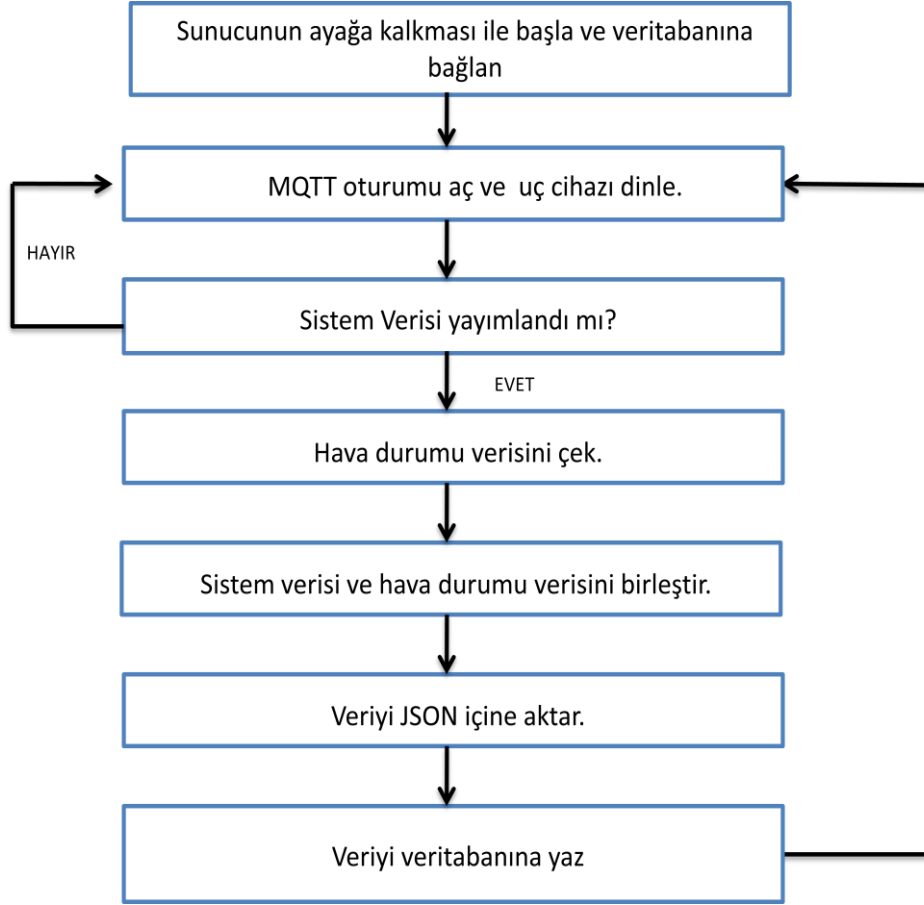
Tablo 4.6 MongoDB ve PostgreSQL performans karşılaştırması (Jung et al., 2016)

Tablo 4.6 ise ücretsiz ve oldukça yaygın olarak kullanılan MongoDB ve PostgreSQL veritabanlarının performanslarını gösterir. PostgreSQL ilişkisel, MongoDB ilişkisel olmayan bir veri tabanıdır. Giriş-çıkış işlemlerinin sayılarına göre yapılmış testte ilişkisel olmayan veri tabanlarından MongoDB daha üstün bir performans göstermiştir (Jung et al., 2016).

Bu teze konu olan sistemin veri yapısı kritik veri olarak görülmediğinden, veri yapısının esnek olması istendiğinden ve sistemin uzaktan izlenmesi için hızlı bir veri iletim yapısına ihtiyaç duyulduğundan ilişkisel olmayan ve hız açısından diğer veri tabanlarına göre öne çıkan mongoDB tercih edilmiştir. MongoDB doküman bazlı JSON nesneleriyle çalışan bir veri tabanıdır. Node.js ortamında Mongoose adında bir paket ile eklemlendirilmiştir. Sistem mimarisi içerisinde MongoDB uzak bir bulut sunucu içerisinde çalışmaktadır.

Arka uç mimarisi içerisindeki REST ve IOT sunucuları tek bir MongoDB veri tabanına bağlıdır. Bu sunuculardan REST kullanıcıya açık iken IOT sunucusu kullanıcıya kapalıdır. Hava durumu sunucusunun ise veri tabanı ile bağlantısı yoktur, doğrudan sunucular ile haberleşir.

IOT sunucusu sahadaki otomasyon sisteminden üretilen ve uç cihaz tarafından yayımlanan verileri veri tabanına yazmakla görevliken REST sunucusu bu verileri sadece okumaktadır.



Şekil 4.16 IOT sunucusu algoritması

IOT sunucusu ayağa kalkıp koşturmaya başladıktan sonra MongoDB veri tabanına bağlanır. Hemen sonra MQTT oturumu açarak aracıya bağlanır ve belirlenen başlığı dinlemeye başlar. Uç cihazdan sistem verisi yayımlandığı anda bu veriyi yakalar. Sistem verisi yakalandıktan sonra IOT sunucusu, hava durumu sunucusundan hava koşulu ile ilgili bilgilerini çeker. Hava koşulu verileri ile sahadaki otomasyon sistemi verileri birleştirilerek bir JSON nesnesi oluşturur. Bu nesne içerisindeki bilgileri veri tabanının ilgili bölümüne yazar. Böylece verinin sahadan veri tabanına aktarımı gerçekleştirilmiş olur.

Veri tabanında sistem verileri ile kullanıcı verileri birbirinden tamamen ayrılmış ve yalıtılmıştır. IOT sunucusu kullanıcı bilgilerine erişemez, kullanıcı bilgileri sadece REST sunucusunun erişimine açıktır. Sistem verileri ile kullanıcı verilerinin yapıları birbirinden tamamen farklıdır.

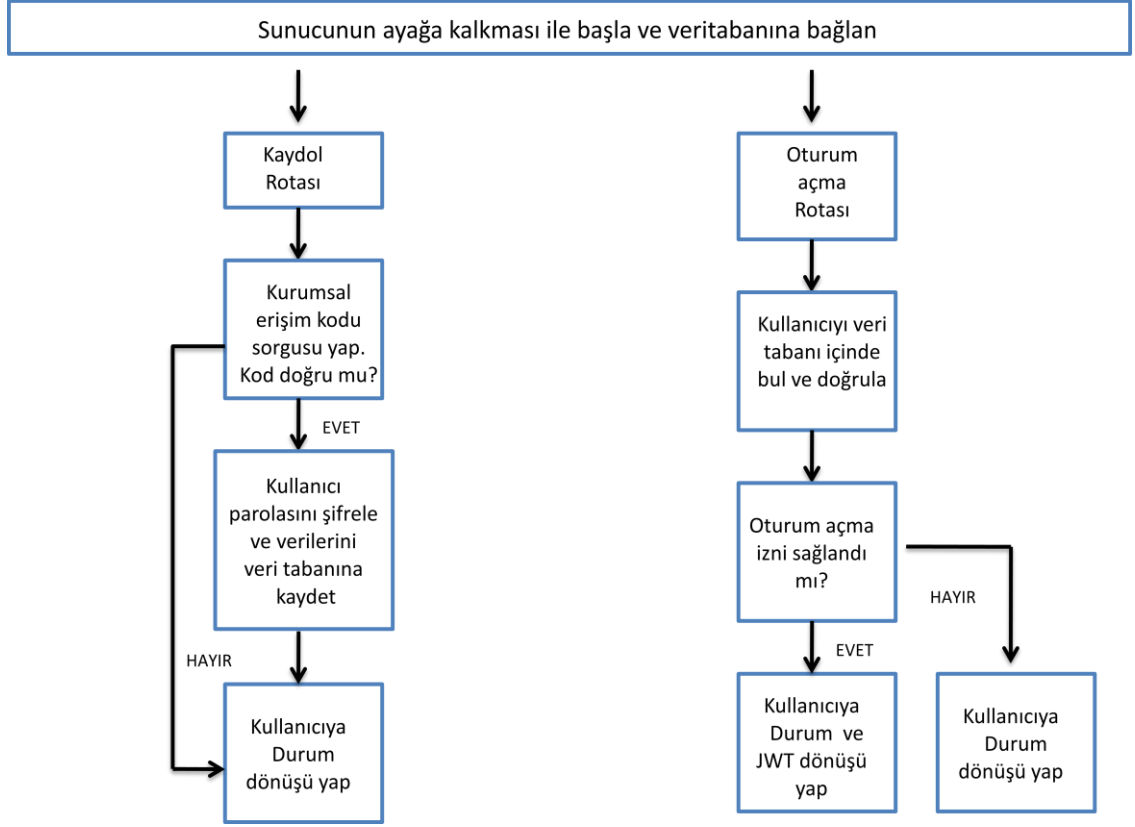
Sistem verileri üzerinde IOT sunucusu sadece yazma yetkisi, REST sunucusunun ise sadece okuma yetkisi vardır.

Veri	Veri Tipi
Başlat	Boolean
Yön	Boolean
Ray Sıcaklığı 1	Double
Ray Sıcaklığı 2	Double
Hız	Double
Dingil Sayıcı 1	Integer
Dingil Sayıcı 2	Integer
Ay	String
Gün	Integer
Saat	Integer
Dakika	Integer
Hava Sıcaklığı	Double
Hissedilen Hava Sıcaklığı	Integer
En Yüksek Hava Sıcaklığı	Double
En Düşük Hava Sıcaklığı	Double
Havadaki Nem Yüzdesi	Integer
Rüzgar Hızı	Double
Rüzgar Açısı	Integer
Gökyüzü Durumu	String

Tablo 4.7 Veri tabanına yazılan sistem verisi

Tablo 4.7 IOT sunucusu tarafından MongoDB veri tabanına yazılan sistem verisinin yapısını göstermektedir. Bu verilerin bütünü mongoDB içerisinde bir doküman oluşturur. Dokümanlar da bir araya gelerek koleksiyonları oluşturur. MongoDB veri tabanında sistem verileri ayrı, kullanıcı verileri ayrı koleksiyonlara yazılmaktadır.

REST sunucusu ise sistem verilerinin kullanıcıya açılan kapısı konumundadır. Ön yüzde kullanıcının kontrolünde olan bir web sitesiyle iletişim halindedir. Bu iletişim HTTP protokolü ile sağlanır. REST sunucusunu veri tabanındaki sistem verilerini gelen istekler doğrultusunda okur. Kullanıcı verileri üzerinde ise tüm yetkilere sahiptir. Bu sunucu rotalar ile çalışır her rotanın kendine özgü işlevleri mevcuttur. Bu rotalara web sitesinden gelen isteklere göre işlemler gerçekleştirilir. REST sunucusu üzerinde her birinin görevi farklı olan 7 rota bulunur. Bu rotalara tamamen kullanıcıya açıktır.

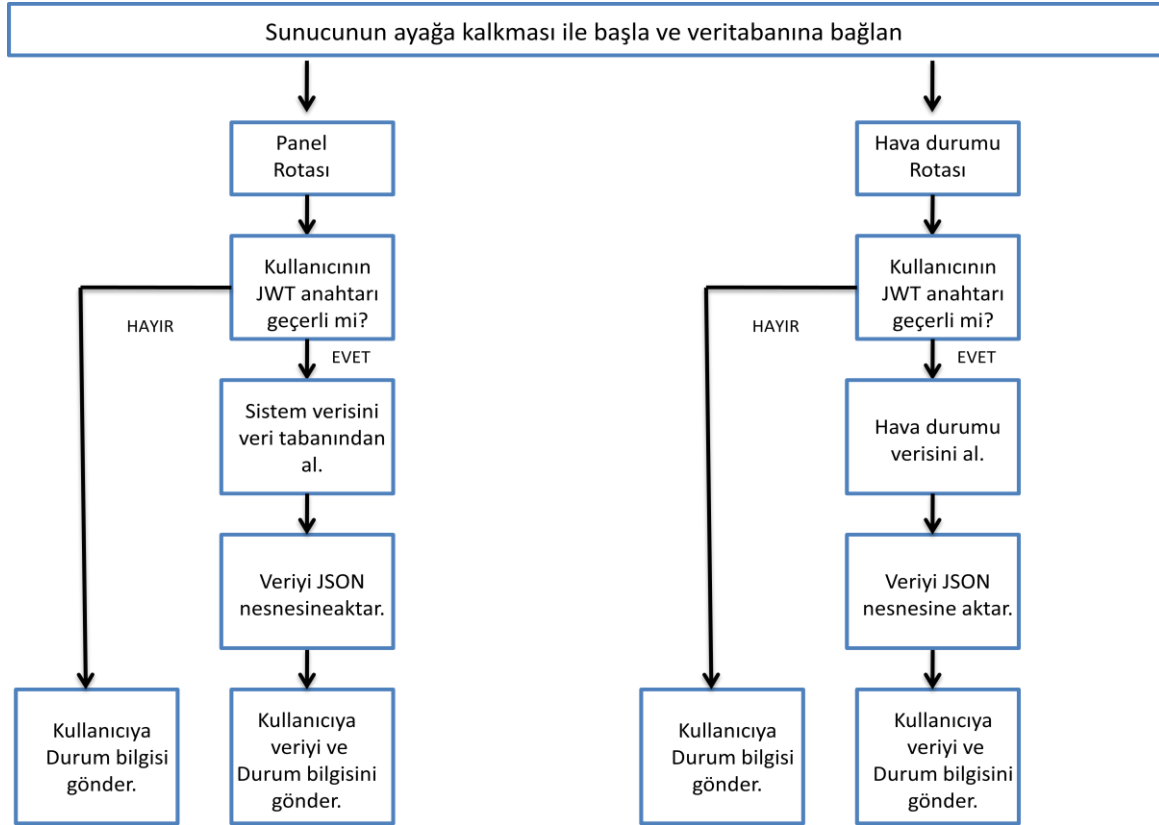


Şekil 4.17 REST sunucu algoritması kaydolma ve oturum açma rotaları

Şekil 4.17, REST sunucusu algoritmasının kaydolma ve oturum açma rotalarını göstermektedir. Kullanıcı kaydolduktan sonra oturum açıp sisteme erişim sağlayabilir. Bunun için kullanıcı tarafından kaydolma rotasına web sitesi üzerinden bir istek gönderilir. Bu istek kullanıcı bilgilerini içerir. Kullanıcı elektronik posta, kendi belirlediği bir parola, çalışan birim tarafından verilmiş kurumsal onay anahtarı ve personel numarası bilgilerini girerek kaydolma işlemini web sitesi üzerinden gerçekleştirir. REST sunucusunun kaydolma rotasına gelen bu bilgiler veri tabanına kaydedilir. Kullanıcı oturum açma işlemini elektronik posta adresi ve parola ile yapacağından, bu kaydetme işlemi yapılırken kullanıcının parolası şifrelenir, Böylece kullanıcı bilgilerinin güvenliği sağlanmış olur. Eğer onay anahtarı bilgisi doğru değilse kaydolma işlemi başarısız olacaktır.

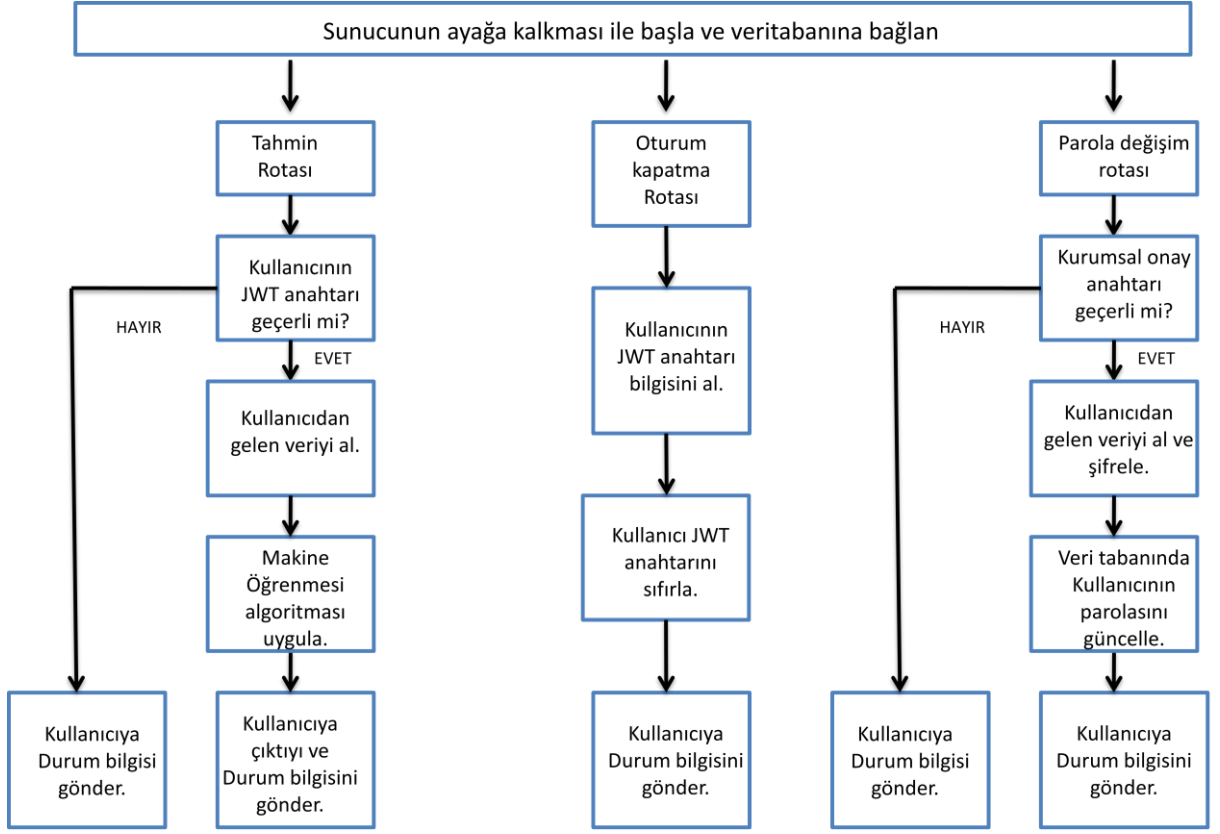
Oturum açma işleminde web sitesinden gelen istek kullanıcının elektronik posta ve parola bilgilerini içerir. Bu istek, web sitesi tarafından oturum açma rotasına gönderilir. REST sunucusu gelen bilgiler doğrultusunda kimlik doğrulaması yapar. Kullanıcı veri tabanında kayıtlı ise sunucu ön uçta çalışan web sitesine bir JWT anahtarı gönderir. Böylece oturum açma işlemi gerçekleşmiş olur. Eğer kullanıcı veri tabanında kayıtlı değilse oturum açma başarısız olur ve kullanıcıya kayıt olması gerektiği bilgisi gönderilir.

JWT her oturuma özgü olarak üretilen baş, yük ve imza içeren şifrelenmiş ve süre kısıtlı bir oturum anahtarıdır. Belirlenen süre sonunda zaman aşımına uğrar ve böylece kullanıcının yeniden oturum açması istenir. Bu oturum güvenliği açısından önemli bir işlevdir. JWT içerisinde baş kısım hangi şifreleme algoritmasının kullanıldığını ve anahtar tipini taşır. Yük kısmında kullanıcıdan gelen bilgiler bulunur. İmza kısmı ise belirlenen gizli bir söz öbeği üzerinden yük ve baş bilgilerini de içine alarak üretilen bir kod dizisidir. Tüm bu kısımlar bir araya gelerek JWT anahtarını oluşturur ve bu anahtarın taşıdığı imza kısmı üzerinden oturum doğrulaması yapılır. Oturum açma sırasında üretilen JWT ile kullanıcı diğer tüm rotalara erişim yetkisi elde eder. Her rotaya gelen istekte söz konusu rota bu JWT anahtarını doğruladıktan erişime izin verir. Sunucudan kullanıcıya giden durum bilgileri ise HTTP durumları ve web sitesinin kullanacağı o rotaya özgü bildirimlerdir.



Şekil 4.18 REST sunucu algoritması panel ve hava durumu rotaları

Şekil 4.18, REST sunucusu algoritmasının panel ve hava durumu rotalarını göstermektedir. Sunucu, JWT anahtarı doğrulaması yaptıktan sonra veri tabanındaki son beş sistem verisini ve hava durumu sunucusundan aldığı anlık hava durumu bilgisini web sitesine gönderir. Oturum açıldıktan sonra web sitesinden gelen her istek üzerinde JWT anahtarını taşır. Bu anahtar REST sunucusu tarafından doğrulanmazsa kullanıcıya işlemin başarısız olduğuna dair bir durum bilgisi gönderilir ve kullanıcı web sitesinde giriş ekranına yönlendirilir.



Şekil 4.19 REST sunucu algoritması tahmin oturum kapatma ve parola değişim rotaları

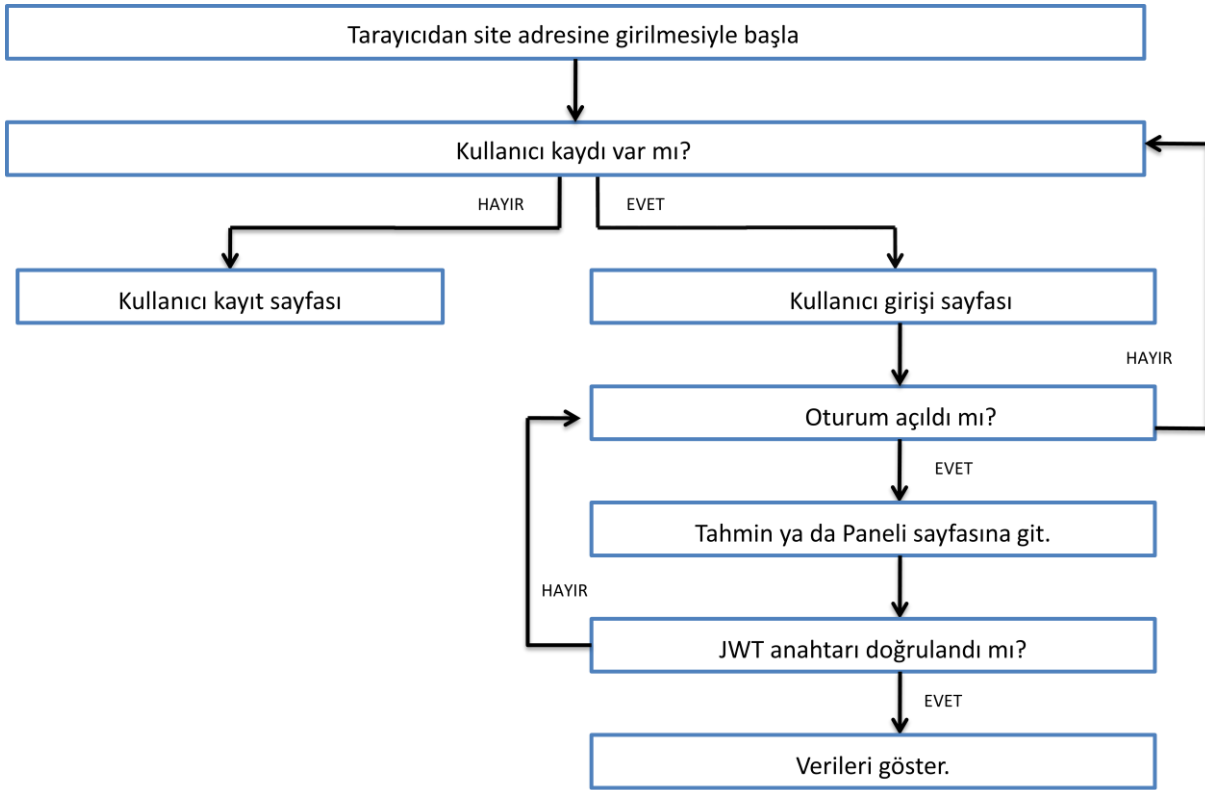
Şekil 4.19, REST sunucusu algoritması içerisindeki tahmin, oturum kapatma ve parola değiştirme rotalarının işlemlerini açıklamaktadır. Tahmin rotası alt iş parçacığı olarak bir yapay zeka algoritması çalıştırır. Kullanıcı web sitesi üzerinden istekte bulunur ve bu istek hava sıcaklığı, hissedilen hava sıcaklığı, en yüksek hava sıcaklığı, en düşük hava sıcaklığı, nem yüzdesi, rüzgar hızı, rüzgar açısı, gökyüzü durumu, ay ve saat bilgileri içerir. Bu bilgileri web sitesine kullanıcının kendisi girmektedir. Bu bilgiler REST sunucusu tarafından ray sıcaklığı tahmini yapmak için kullanılır. Sunucu bu bilgileri almadan önce istek üzerinde JWT anahtar doğrulaması yapar, anahtar onaylanırsa sunucu bu bilgileri alır ve alt iş parçacığına iletir. Tahminlemeyi yapan alt iş parçacığı python programlama diliyle yazılmıştır. Alt parçacığı sonucu ana ipliğe iletir ve sunucu bu değeri kullanıcıya döner. Böylece tahminleme işlemi tamamlanmış olur.

Oturum kapatma işleminde JWT anahtar doğrulaması yapıp kullanıcının anahtarı sıfırlanarak oturum kapatılır. Oturum kapalı ise kullanıcı web sitesi üzerinden panel ve tahminleme sayfalarına ulaşamaz ve giriş sayfasına yönlendirilir. Parola değiştirme işleminde ise oturum açma şartı aranmaz JWT kontrolü söz konusu değildir. Kurumsal onay anahtarı doğrulaması yapılır. Eğer kurumsal onay anahtarı geçerli ise kullanıcının yeni parolası şifrelenerek veri tabanında o kullanıcıya ait bilgiler güncellenir. Kurumsal onay anahtarı geçerli değilse kullanıcıya hata bildirisi yapılır.

4.2.2.2. Ön Uç

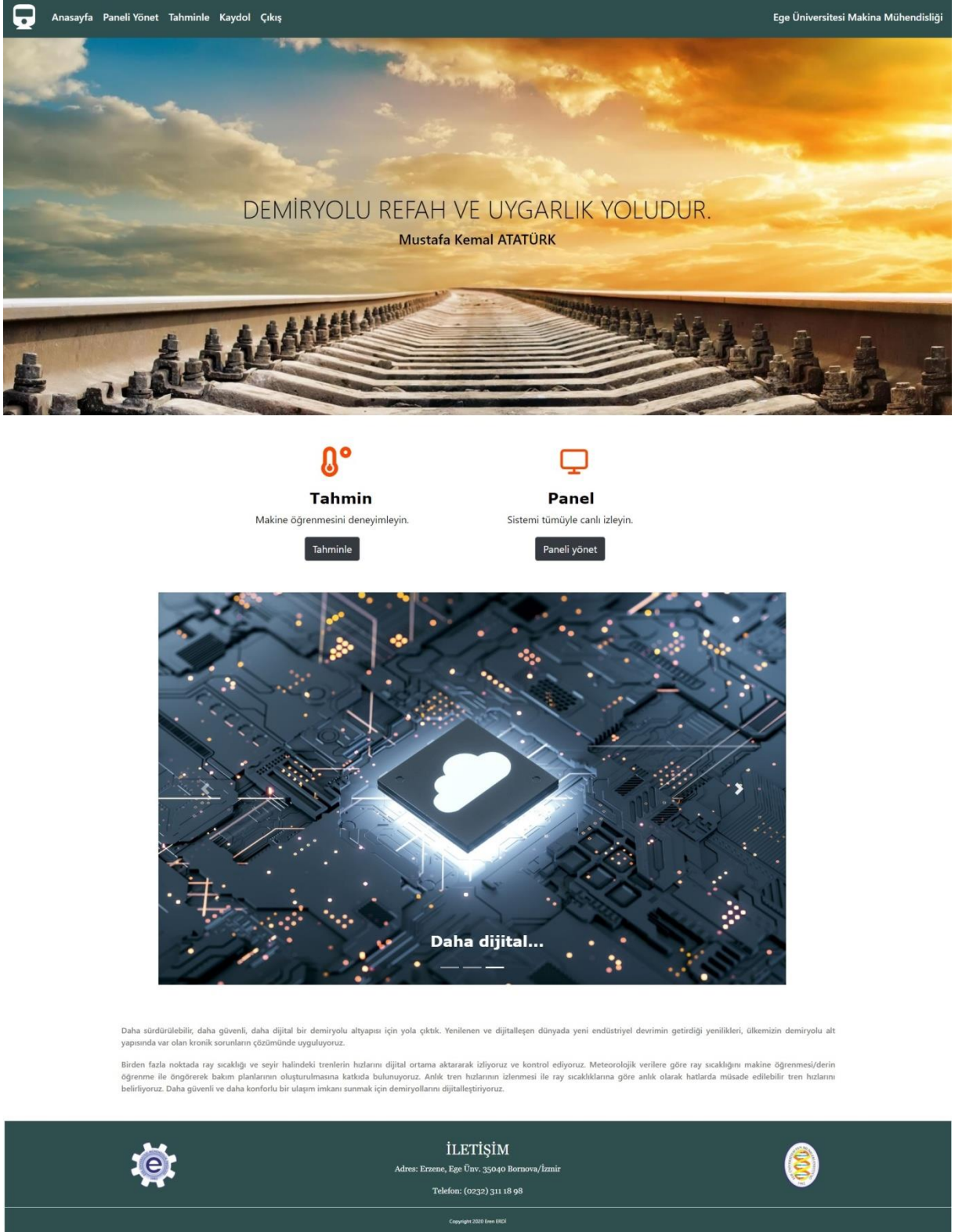
Dijitalizasyon mimarisinin diğer bileşeni ve kullanıcıya açılan kapısı ön uçtur. Bir web sitesinden oluşur ve arka uç ile HTTP protokolü vasıtasıyla haberleşir. Ön uç, arka uc algoritmasının ihtiyaç duyduğu her isteği karşılayacak şekilde geliştirilmiştir. Kullanıcı kaydolma, oturum açma, oturum kapatma, parola değiştirme, panel sayfasında uzaktan izleme, ray sıcaklığı tahminle işlerinin tümünü web sitesi üzerinden yapar.

Söz konusu web sitesi HTML, CSS ve javascript dilleri ile react.js ve redux.js kütüphaneleri üzerinde inşa edilmiş bir tekil sayfa uygulamasıdır.



Şekil 4.20 Ön uç algoritması

Şekil 4.20, ön uçta çalışan web sitesinin algoritmasını göstermektedir. Kullanıcının sistemde kaydı yok ise web sitesi kullanıcıyı kayıt sayfasına yönlendirir ve kaydolma işlemi gerçekleştirilir. Eğer kullanıcının kaydı mevcut ise kullanıcı doğrudan giriş sayfasına giderek oturum açabilir. Oturum açmayan bir kullanıcı panel ve tahmin sayfalarına erişemez ve web sitesi tarafından giriş sayfasına yönlendirilir. Kullanıcı oturum açtıktan sonra kullanıcıya REST sunucusu bir JWT anahtarı gönderilir. Bu sayede kullanıcı tahmin ve panel sayfalarına giderek sistem verilerini izleyip ray sıcaklığı tahmini yapabilir. JWT anahtarı zaman aşımına uğradığında doğrulama işlemi başarısız olursa kullanıcının oturumu kapatılır ve kullanıcı giriş sayfasına yönlendirilir.



Şekil 4.21 Web sitesi ana sayfası

Şekil 4.21, web sitesinin kullanıcıya temel açıklamalar sunan ana sayfasını gösterir. Üst kısımdaki gezinme çubuğundan sayfalar arasında geçişler yapılabilmektedir. Aynı şekilde orta kısımda bulunan düğmelere vasıtasıyla tahmin ve panel sayfalarına geçiş yapılabilir.

Şekil 4.22 Web sitesi kaydolma sayfası

Şekil 4.22, web sitesinin kullanıcı kayıt sayfasını göstermektedir. Bu sayfada kullanıcı; elektronik posta, parola, personel numarası ve kurumsal onay anahtarı bilgilerini girerek kayıt işlemi gerçekleştirir.

Şekil 4.23 Web sitesi giriş sayfası

Şekil 4.23, web sitesinin kullanıcı giriş sayfasını göstermektedir. Bu sayfada kullanıcı; elektronik posta, parola bilgilerini girerek oturum açabilir. Oturum açıldığında REST

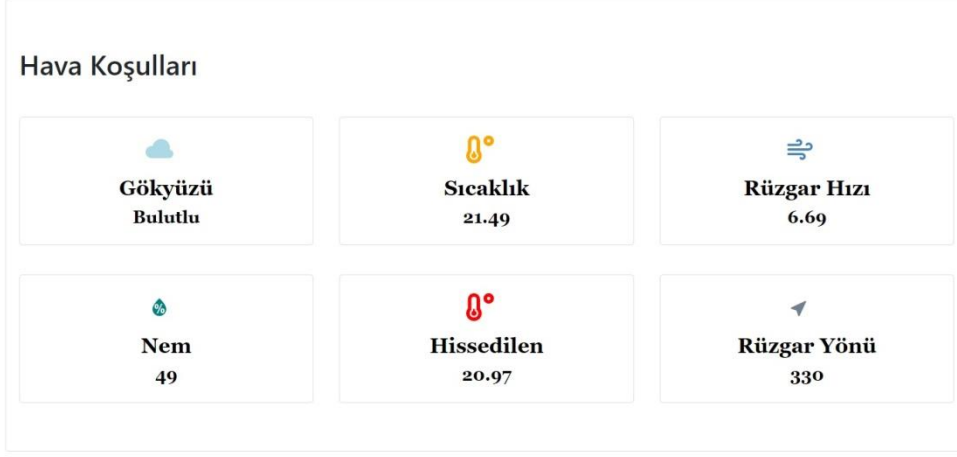
sunucusundan bir JWT anahtarı gönderilir. Bu anahtar sayesinde kullanıcı panel ve tahmin sayfalarına erişim yetkisi elde eder. Kullanıcı oturum açmadan bu sayfalara erişemez.

Şekil 4.24 Web sitesi parola yenileme sayfası

Şekil 4.24, web sitesinin parola yenileme sayfasını göstermektedir. Bu sayfada kullanıcı; elektronik posta, kurum onay anahtarı ve yeni parola bilgilerini parola yenileyebilir. Kullanıcı oturum açmamış iken sayfanın üst kısmındaki gezinme çubuğu üzerinde giriş düğmesi görünmektedir. Kullanıcı oturum açtığı anda bu düğmenin yerini çıkış düğmesi alır. Kullanıcı oturumunu bu düğme yardımıyla doğrudan kapatabilir.

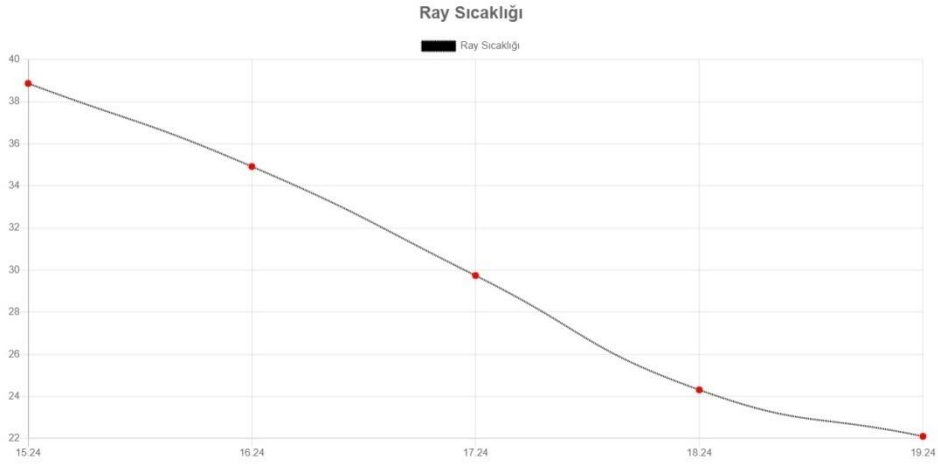
Şekil 4.25 Web sitesi panel sayfası sistem verisi

Şekil 4.25, web sitesinin panel sayfası içerisinde bulunan sistem verisi bölümünü göstermektedir. Buradaki değerler sahada konuşlandırılmış otomasyon sistemin anlık durumunu bildirmektedir.



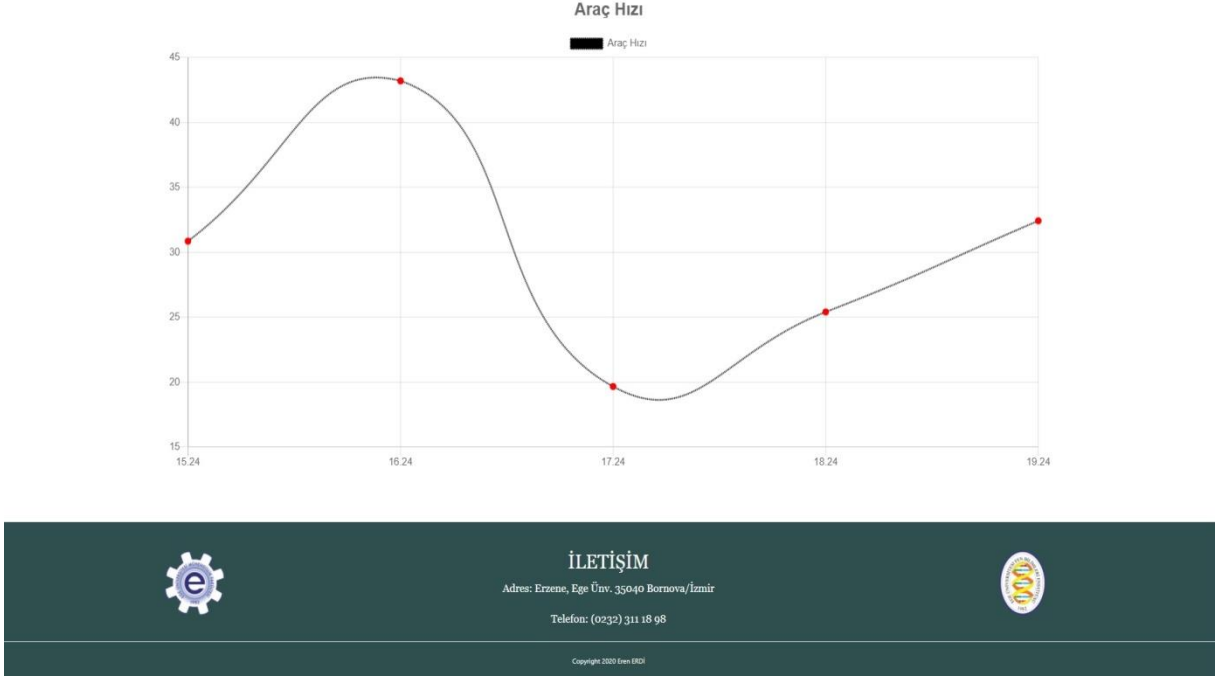
Şekil 4.26 Web sitesi panel sayfası hava koşulları verisi

Şekil 4.26, web sitesinin panel sayfası içerisinde bulunan hava koşulları verisi bölümünü göstermektedir. Buradaki değerler hava durumu sunucusundan anlık olarak alınmaktadır.



Şekil 4.27 Web sitesi panel sayfası ray sıcaklığı grafiği

Şekil 4.27, web sitesinin panel sayfası içerisinde bulunan ray sıcaklığı grafiğini göstermektedir. Veri tabanında bulunan son beş sistem verisi çekilerek oluşturulmuştur.



Şekil 4.28 Web sitesi panel sayfası araç hızı grafiği

Şekil 4.28, web sitesinin panel sayfası içerisinde bulunan ray sıcaklığı grafiğini göstermektedir. Veri tabanında bulunan son beş sistem verisi çekilerek oluşturulmuştur.

Panel sayfasında gösterilen ray sıcaklığı ve araç hızı grafikleri chart.js kütüphanesi kullanılarak oluşturulmuştur.

Web sitesinin geriye kalan son sayfası tahmin sayfasıdır. Kullanıcıya zamana dayalı ray sıcaklığı tahmini yapma imkanı sunar. Kullanıcıdan hava sıcaklığı, hissedilen sıcaklık, en yüksek sıcaklık, en düşük sıcaklık, nem yüzdesi, rüzgar hızı, rüzgar açısı, gökyüzü durumu, ay ve saat bilgilerini ister. Kullanıcı bu bilgileri girip gönder düğmesi yardımıyla istekte bulunduğu bu istek REST sunucusunun tahminleme rotasına iletilir. Sunucu önce gelen isteğin JWT anahtarını sorgular. Eğer anahtar sorgulaması başarılı olursa kullanıcıdan gelen bilgileri alt parçacık olarak çalışan yapay zeka algoritmasına iletir. Alt iş parçacığı olarak çalışan bu algoritma bilgileri işleyerek sonucu ana iş parçacığına döner ve ana iş parçacığı bu bilgiyi ön uca iletir.

REST sunucusunun ilettiği bu bilgi tahminleme sayfası ekranında bulunan ray sıcaklığı bölümünde kullanıcıya gösterilir. Böylece ray sıcaklığı tahminleme işlemi tamamlanmış olur.

REST sunucusunun ana iş parçacığında javascript dilinin yorumlayıcısı çalışırken, tahminleme rotasına bir istek geldiğinde alt iş parçacığında python programlama dilinin yorumlayıcısı çalışır.

The screenshot shows a web interface for predicting ray temperature. At the top, there is a navigation bar with links: Anasayfa, Paneli Yönet, Tahminle, Kaydol, Çıkış. The main content area features a large yellow box at the top displaying 'Öngörülen Ray Sıcaklığı' (Predicted Ray Temperature) as 0°C. Below this, there is a form with various input fields and dropdown menus. The form includes fields for 'Hava Sıcaklığı' (Air Temperature), 'Hissedilen Sıcaklığı' (Perceived Temperature), 'Nem' (Humidity), 'En Düşük Sıcaklık' (Minimum Temperature), 'En Yüksek Sıcaklık' (Maximum Temperature), 'Rüzgar Hızı' (Wind Speed), 'Rüzgar Açısı' (Wind Direction), 'Gökyüzü' (Sky Condition), 'Ay' (Month), and 'Saat' (Hour). Each field has a placeholder text indicating the unit or type of input. A green 'Gönder' button is located at the bottom of the form. Below the button, there is a small text: 'İleri tarihli bir ray sıcaklığı tahminleyin.' (Predict a ray temperature for a later date).

Şekil 4.29 Web sitesi tahmin sayfası

Şekil 4.29, web sitesinin tahmin sayfasını göstermektedir. Kullanıcının ekrandaki formu doldurarak hızlı bir şekilde zaman dayalı ray sıcaklığı tahmini yapmasına olanak sağlar.

4.3. Ray Sıcaklığı Tahmin Algoritması

Ray sıcaklığının zaman dayalı olarak tahminini, toplam kütle yaklaşımı ile geliştirilen analitik modeli yerine bu analitik modele karşılık gelen veri modeliyle ilişkilendirilerek uygun yapay zeka algoritmasıyla gerçekleştirmek oldukça hızlı ve maliyetsizdir.

Yapay zeka kavramı içerisinde makine öğrenmesi ve derin öğrenmeyi de alan geniş bir kavramdır. Bu kavramlar dışından yapay zeka algoritmaları doğası gereği gözetimli ve gözetimsiz olmak üzere ayrışır. Gözetimli öğrenmede ulaşılması istenen bir hedef değer mevcut iken gözetimsiz öğrenmede ilişkisel kümeleme amaçlanır. Geliştirilen veri modeli ve fiziksel model ele alındığında algoritmanın gözetimli öğrenme kapsamına girdiği görülmektedir. Gözetimli öğrenme algoritmaları arasında yaygın olarak kullanılan ve oldukça verimli olan XGBOOST, SVM ve ANN algoritmaları seçilmiş ve kullanılmıştır. XGBOOST ve SVM makine öğrenmesi algoritmaları, ANN ise derin öğrenme algoritması olarak kabul görmektedir. Söz konusu bu algoritmaların çıktıları değerlendirilmiş ve en uygun olan REST sunucu içerisinde bir alt iş parçacığı olarak çalışacak şekilde gömülmüştür.

Gözetimli yapay zeka algoritması uygulanacak veri yapısında hedef değer başka bir deyişle tahmin edilmek istenen değer ray sıcaklığıdır. Yapay zeka algoritmalarının performansı, geliştirme süreci boyunca kayıt altına alınan 2234 veri üzerinde uygulanarak ölçülmüştür.

Veri	Veri Tipi
Ray Sıcaklığı 1	Double
Saat	Integer
Ay	String
Hava Sıcaklığı	Double
Hissedilen Hava Sıcaklığı	Integer
En Yüksek Hava Sıcaklığı	Double
En Düşük Hava Sıcaklığı	Double
Havadaki Nem Yüzdesi	Integer
Rüzgar Hızı	Double
Rüzgar Açısı	Integer
Gökyüzü Durumu	String

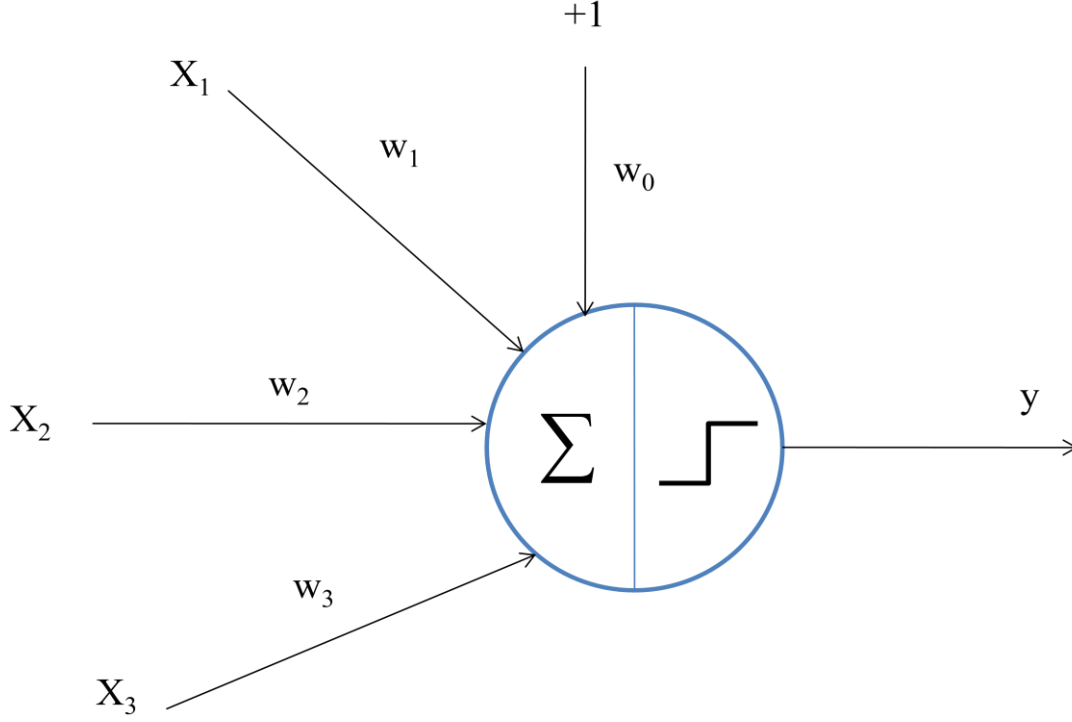
Tablo 4.8 Yapay zeka algoritmalarına tabi tutulan veri yapısı

Tablo 4.8 , yapay zeka algoritmalarının değerlendirilmesinde kullanılacak veri yapısını göstermektedir. Ulaşılması gereken hedef yani bağımlı değişken ray sıcaklığı iken geri kalanlar ise bağımsız değişkenlerdir.

4.3.1. ANN Algoritması

ANN algoritması yapay sinir ağları olarak adlandırılır. İnsan beyninin çalışma prensipleri taklit edilerek oluşturulmuş bir algoritmadır.

İnsan fizyolojisindeki nöronların yerini bu algorithmada algılayıcılar(perceptron) alır. İleri ve geri yayılım ile bir optimizasyon methodu kullanılarak öğrenme gerçekleştirilir (Aggarwal, 2018).



Şekil 4.30 Algılayıcının yapısı

ANN algoritması, nöron olarak da bilinen algılayıcılardan oluşur. Şekil 4.30 bir algılayıcının yapısını göstermektedir. Algılayıcının temel olarak iki ana işlevi yerine getirir. Öncelikle girdi olarak kabul ettiği x değerleri ile sapma değerini oluşturmakla görevli $+1$ değerini w ağırlıklarıyla çarpıp toplar ve bir matematiksel ifade üretir. Daha sonra bu ifade bir aktivasyon fonksiyonuna tabi tutularak hedeflenen y çıktısı elde edilir. Algılayıcılar bir araya gelerek katmanları katmanlar da bir araya gelerek yapay sinir ağı olarak tanımlanan ANN algoritmasını temelini oluşturur.

ANN algoritmasındaki temel hedef, bilinen x girdilerini ve hata fonksiyonunu temel alarak w ağırlıklarını iterasyona tabi tutup optimazyon metodları ile bu ağırlıkları saptamaktır. Böylece öğrenme gerçekleştirilir ve yeni yüklenen girdi değerlerine karşılık belirli bir olasılıkla hedef değer tahmin edilir.

$$f = \sum_{i=1}^n X_i w_i + w_0 \quad (4.14)$$

$$X = [1, X_1, X_2, \dots, X_n]^T \quad (4.15a)$$

$$w = [w_0, w_1, w_2, \dots, w_n]^T \quad (4.15b)$$

$$f = w^T X \quad (4.15c)$$

$$y = z(f) \quad (4.16a)$$

$$y = z(w^T X) \quad (4.16b)$$

Denklem 4.14, algılayıcıya giren sapma değeri ve girdi değerlerinin kendi ağırlıklarıyla çarpılarak toplanması sonucu ortaya çıkar. Başlangıçta söz konusu bu ağırlık değerleri bilinmediğinden rastgele değerler ile hesaba başlanır. Denklem 4.15a, 4.15b ve 4.15c, 4.14'te verilen ifadeyi daha sade bir görünüme getirmek için kullanılır. Denklem 4.16a ve 4.16b içerisinde bulunan z fonksiyonu ise aktivasyon fonksiyonudur. Denklem 4.14 ile hesaplanan f değeri bu fonksiyon içerisinde yerleştirilerek y çıktı değeri hesaplanır. Eğer tek bir algılayıcı yerine çok sayıda algılayıcıdan oluşan bir ağ var ise burada aktivasyon fonksiyonunun verdiği değer başka algılayıcıların girdi değerini oluşturur (Aggarwal, 2018). Böyle ileri doğru bir yayılım gerçekleşir. İleri yayılımın en önemli bileşeni aktivasyon fonksiyonlarıdır.

Aktivasyon fonksiyonlarının çok sayıda tipi bulunmaktadır. Bu tipler arasında en uygun olanın seçilmesi gerekmektedir. Bir aktivasyon fonksiyonunda belirli beklentiler bulunmaktadır. Bu beklentilere uygun bir aktivasyon fonksiyonu optimizasyonun başarısında oldukça önemli bir rol oynar.

Aktivasyon fonksiyonun en önemli özelliklerinden biri veri için doğrusal olmayan bir uzayın oluşumuna katkıda bulunmaktır. Böylece verinin sınıflandırılmasını ya da regrese edilmesini kolaylaştırır. Diğer bir önemli özellik ise türevlenebilir olmasıdır. Böylece geri yayılıma imkan tanır. Bu noktada aktivasyon fonksiyonun kompleks olmaması da önem kazanır. Aksi halde hesaplama maliyeti yüksek olacaktır. Geri yayılım için aktivasyon fonksiyonunda bulunması gereken diğer nitelik ise sınırlarının var olmasıdır. Sınırları olmayan bir aktivasyon fonksiyonu ile geri yayılımda sonuç alma ihtimali zayıftır. Türevler çok büyük ya da çok küçük çıkabilir. Bunlara ek olarak aktivasyon fonksiyonu düzenli artan veya azalan olmalıdır. Böylece gradyan kaybolmasının önüne geçilir ve optimizasyonun başarı olasılığını güçlendirir (Bircanoglu ve Arica, 2018).

Aktivasyon Fonksiyonunun Adı	Aktivasyon Fonksiyonunun Denklemi
Doğrusal	X
Sigmoid	$\frac{\exp(X)}{1 + \exp(X)}$
Hiperbolik Tanjant	$\frac{2}{1 + \exp(-2X)} - 1$
SoftSign	$\frac{X}{1 + X }$
ReLU	$\begin{cases} X & \text{eğer } X > 0 \\ 0 & \text{eğer } X \leq 0 \end{cases}$
SoftPlus	$\ln(1 + \exp(X))$
Swish	$X \text{sigmoid}(X)$
Kare	X^2

Tablo 4.9 Yaygın olarak kullanılan aktivasyon fonksiyonları (Bircanoglu ve Arica, 2018)

Tablo 4.9, yaygın olarak kullanılan aktivasyon fonksiyonlarını ve matematiksel ifadelerini göstermektedir. Bu fonksiyonlar arasında ReLU, başarılı bir aktivasyon fonksiyonu olarak öne çıkmaktadır. Birçok veri setinde ReLU fonksiyonu kullanılarak daha düşük hata ve değerleri daha yüksek başarı yüzdeleri elde edilmiştir (Bircanoglu ve Arica, 2018). Bu nedenlerle ray sıcaklığı tahmini için geliştirilen ANN algoritmasında ReLU aktivasyon fonksiyonu tercih edilmiştir.

Denklem 4.16b ile belirtilen çıktı değeri, başlangıçta verilmiş rastgele w ağırlıklarıyla elde edilmiştir. Bir algılayıcıda öğrenme bu çıktı değeri ile gerçek çıktı değeri arasındaki hataların hesaplanması ve bu hata değerlerine göre ağırlıkların güncellenmesi ile gerçekleşir.

Ağırlıkların güncellenmesi hata fonksiyonuna uygulanan geri yayılım ile gerçekleşir (Aggarwal, 2018).

Bu noktada hata fonksiyonu ön plana çıkar. Yaygın olarak kullanılan hata fonksiyonları şunlardır:

- MSE : $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
- MAE : $\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$
- MSLE : $\frac{1}{n} \sum_{i=1}^n (\log(y_i + 1) - \log(\hat{y}_i + 1))^2$
- MALE : $\frac{1}{n} \sum_{i=1}^n |\log(y_i + 1) - \log(\hat{y}_i + 1)|$

Bu fonksiyonlardan MSE ve MAE regresyon problemleri için kullanılırken MSLE ve MALE sınıflandırma problemleri için kullanılmaktadır. Ray sıcaklığı tahmini için geliştirilen algoritma için MSE hata fonksiyonu kullanılmıştır.

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.17)$$

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial f} \frac{\partial f}{\partial w} \quad (4.18)$$

Denklem 4.17, hata fonksiyonunu göstermektedir. Hata fonksiyonu kayıp fonksiyonu olarak da adlandırılır. Denklem 4.18 ise geri yayılım denklemidir. İleri yayılım yapılarak Denklem 4.16b ile elde edilen çıktı hata fonksiyonu içerisine gönderilir ve hata değeri hesaplanır. Daha sonra Denklem 4.18 ile gösterildiği gibi geriye doğru zincir türev vasıtasıyla hatanın ağırlığa göre değişim değeri hesaplanır. Amaç hatayı en aza indirmek (Aggarwal, 2018).

Bu noktada hata fonksiyonuna uygulanacak geri yayılım için bir optimizasyon algoritması seçilmelidir. Günümüzde bu optimizasyon algoritmaları birinci ya da ikinci dereceden algoritmalar. Birinci derece algoritmalar en sık kullanılanlar GD, SGD, ADAM gibi algoritmalar. İkinci derece algoritmalar yaygın olarak kullanılanlar ise BFGS, Levenberg-Marquardt gibi algoritmalar. İkinci derece algoritmalar optimizasyon için Hessian matris kullanır. Bu nedenle birinci dereceden algoritmalar göre daha az iterasyon ile daha iyi bir yaklaşım gösterirler. Buna karşın hesaplama maliyeti ve süresi yüksektir (Henriques et al., 2019).

Ray sıcaklığını tahmin eden algoritma web üzerinde bir sunucu içerisinde çalışacağından hesaplama maliyeti ve süresi oldukça önem arz etmektedir. Bu nedenle GD tabanlı algoritmalarından biri kullanılmıştır.

$$w^{(k+1)} = w^{(k)} - \eta \frac{\partial L}{\partial w} \quad (4.19)$$

Denklem 4.19, ağırlıkların her iterasyonda güncellendiği matematiksel ifadedir. Geri yayılım uygulandıktan sonra elde edilen değişim değerleri belirlenen bir η öğrenme değeri ile çarpılarak bir önceki ağırlık değerlerine eklenir. Böylece yeni ağırlıklar bulunur. Hata değeri belirlenen bir değer altına inene kadar bu işlem devam eder (Aggarwal, 2018). Bu en yalın GD algoritmasıdır.

GD algoritması en kaba tabirle adım adım ilerleyerek w ağırlıklarından oluşan çok boyutlu bir fonksiyonun global minimum noktasını bulmaya çalışır. Ancak GD algoritmasının yalın hali bir takım dezavantajlar içerir. Dolayısıyla giderek geliştirilmesine ihtiyaç duyulmuştur. Yalın GD algoritması veri setinin tamamını kullanır. Bu nedenle yüksek bir kaynak gereksinimi vardır. Diğer GD tabanlı algoritmalarla göre oldukça yavaş kalır. Öğrenme katsayısı kullanıcı tarafında optimize edilmek zorundadır. Çok düşük öğrenme katsayıları çok uzun bir optimizasyon zamanı ve yerel minimumlarda takılma riski açığa çıkarırken yüksek öğrenme katsayıları ise global minimum noktasının atlanmasına yol açabilir (Ruder, 2017).

Tüm veri setinin kullanımı optimizasyonda yüksek maliyete yol açmaktadır. Bu durum optimizasyon için her seferinde rastgele bir veri alınarak çözülmeye çalışılmış ve SGD algoritması ortaya atılmıştır. Ancak yalın GD algoritmasının getirdiği bir çok dezavantajı hala ortadan kaldırılamamıştır. SGD algoritmasından yola çıkılarak geliştirilen MBGD algoritması bu sorunları azaltmada önemli rol oynamıştır. MBGD algoritmasında veri seçimi yapılırken rastgelelik korunur ancak her adımda veri içerisinde tek bir örneklem değil bir grup örneklem seçilerek optimizasyon yapılır. Bellek 8, 16, 32 vb. veri sayılarından oluşan paketler kullanılır. Paketler için seçilen yaygın veri sayıları ise 32, 64, 128'dir (Ruder, 2017).

MBGD algoritması yalın GD algoritmasının bir çok dezavantajını tam olarak çözme de azaltmış ve optimizasyonu hızlandırmıştır. Ancak günümüzde daha az kaynak tüketimine ve daha fazla hıza olan ihtiyaç günden güne artmaktadır. Bu nedenle MBGD algoritması üzerinden geliştirilen yeni algoritmalar bu ihtiyaçlara cevap vermeyi amaçlamaktadır. Bunlardan biri de momentuma sahip GD algoritmasıdır.

$$M_t = \beta M_{t-1} + (1 - \beta)dw \quad (4.20)$$

$$w^{(k+1)} = w^{(k)} - \eta M_t \quad (4.21)$$

Momentuma sahip GD algoritması, MBGD algoritması gibi rastgele seçilen veri paketleriyle işlem yapar. Denklem 4.20, EWMA bağıntısıdır. Ağırlıkların hareketli üstel ortalamasını ifade eder. Denklem 4.21, ağırlıkları güncellemek için doğrudan ağırlık gradyanını kullanmak yerine EWMA bağıntısından gelen momentum değerini kullanır. EWMA bağıntısındaki β parametresi hareketli ortalamayı manipüle eder. Genellikle 0,9 alınır. Bu, hareketli ortalamanın son 10 gradyan değerine göre hesaplanması anlamına gelir. Momentuma sahip GD, MBGD algoritmasından oldukça hızlıdır ve minimuma ulaşma sürecindeki gürültüleri azaltır. Hesaba MBGD algoritmasından farklı olarak sadece EWMA denklemi eklendiğinde yüksek kaynak kullanımına da neden olmaz. Ancak hala öğrenme katsayısının kullanıcı tarafından belirlenmesi gerekmektedir. Öte yandan yanlış β değeri seçimleri sonucu global minimum atlanabilir veya yerel minimumlarda öğrenme durabilir (Ruder, 2017).

$$S_t = \beta M_{t-1} + (1 - \beta)dw^2 \quad (4.22)$$

$$w^{(k+1)} = w^{(k)} - \eta \frac{dw}{\sqrt{S_t + \epsilon}} \quad (4.23)$$

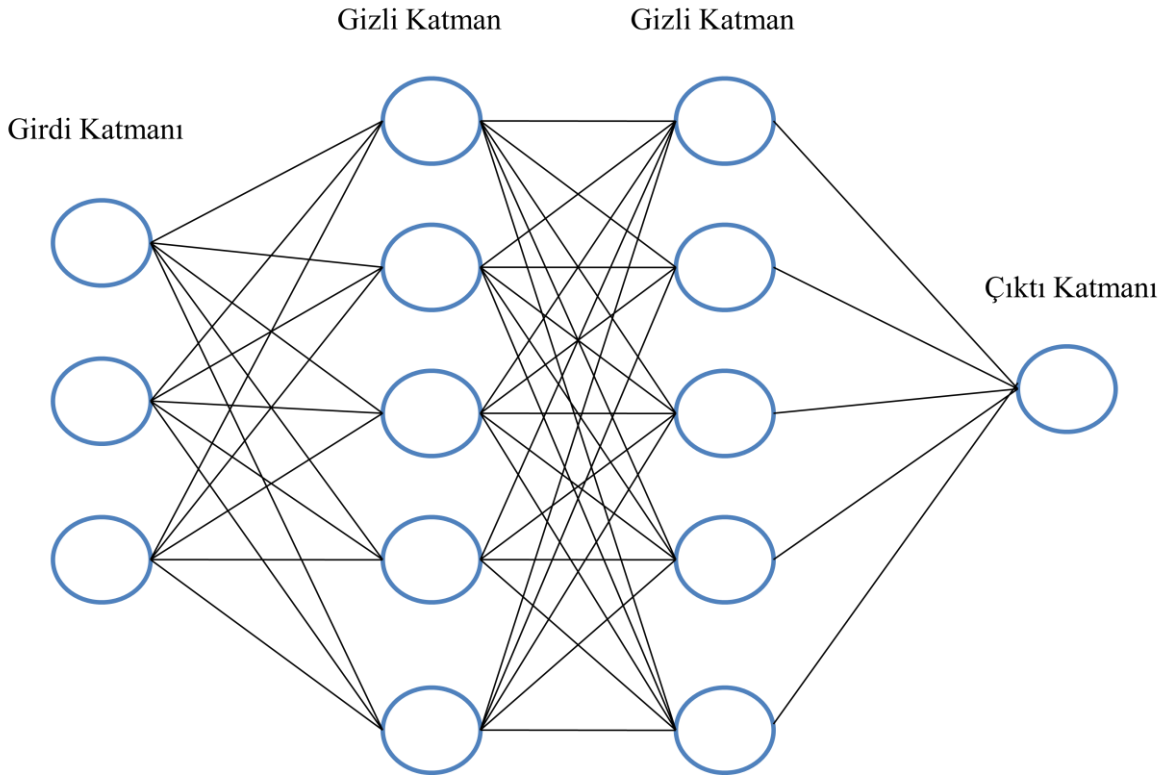
Öğrenme katsayısının kullanıcı tarafından belirlenmesi ve bu katsayısının öğrenme boyunca sabit kalması global minimum noktasına ulaşmada önemli bir sorun teşkil eder. Bu sorunu aşmak için RMSPROP algoritması geliştirilmiştir. Denklem 4.22’de görüldüğü üzere RMSPROP algoritması da EWMA kullanır. Ancak momentuma sahip GD algoritmasından farklı olarak ağırlık gradyanlarının karesinin hareketli ortalamasını alır ve Denklem 4.23’te gösterilen ifadede öğrenme katsayısını bu ortalama değerin kareköküne böler. Böylece başlangıçta bir kez tayin edilen öğrenme katsayısı değeri her adımda kendini güncellemiş olur. Bu denklemdeki ϵ değeri ise paydanın sıfıra gitmesini engellemek için kullanılan bir sabittir. RMSPROP algoritması sayesinde momentuma sahip GD algoritmasından daha hızlı bir şekilde sonuca ulaşılır. RMSPROP algoritması da momentuma sahip GD algoritması gibi minimuma ulaşırken oluşabilecek gürültüleri en aza indirir. Her adımda öğrenme katsayısını günceller (Ruder, 2017).

$$w^{(k+1)} = w^{(k)} - \eta \frac{M_t}{\sqrt{S_t + \epsilon}} \quad (4.24)$$

Momentuma sahip GD algoritması gradyanların üstel hareketli ortalamasını kullanırken ve RMSPROP gradyanların karelerinin üstel hareketli ortalamasını alarak öğrenme katsayısını günceller. Bu iki algoritmanın özelliklerini kullanabilmek için de ADAM algoritması geliştirilmiştir.

Denklem 4.24, ADAM algoritmasının ağırlıkları güncellemek için kullandığı eşitliği göstermektedir. ADAM algoritması, diğer algortimalara kıyasla daha hızlı, düşük bellek tüketen, konveks olmayan problemlerde dahi optimizasyon gücünü koruyan çok boyutlu büyük veri setlerinde oldukça verimli çalışan bir algoritmadır (Kingma and Ba, 2015).

Bu üstünlükleri sebebiyle zamana dayalı ray sıcaklığı tahmini için geliştirilen ANN modelinde ADAM optimizasyon algoritması kullanılmıştır.



Şekil 4.31 Çok katmanlı yapay sinir ağıları örnek şeması

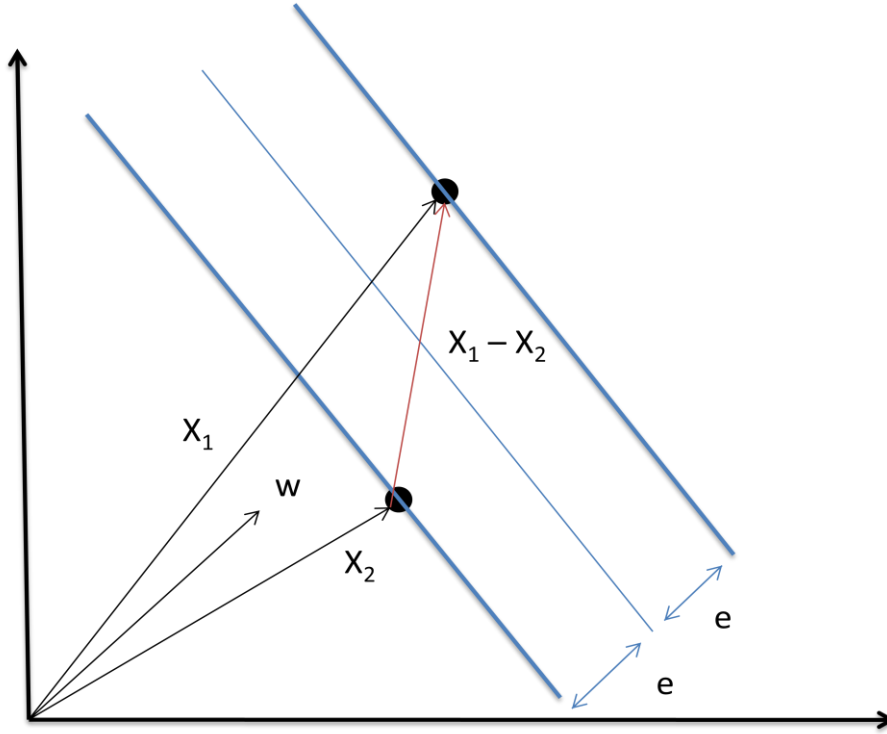
Tek bir algılayıcı gerçek hayatta karşılaşılan problemlerin çözümü için çok yetersiz kalır. Bu nedenle çok sayıda algılayıcı barındıran bir den fazla katman ve bu katmanlardan oluşan bir ağ kullanılır. Bu Böylece derin öğrenme gerçekleştirilir. Derin öğrenmeyi ortaya çıkaran bu ağı oluşturan gizli katmanlardır. Şekil 4.31, böyle bir ağ örneğini göstermektedir.

Ray sıcaklığı tahmini için geliştirilen ANN modeli de çok katmanlı olarak tasarlanmıştır.

4.3.2. SVM Algoritması

SVM algoritması hem sınıflandırma hem regresyon problemleri için kullanılan ve hata fonksiyonunun minimize edilmesi yerine veri集中的i örneklemeler arasında sınırlar oluşturarak bu sınırları maksimize eden bir algoritmadır. Regresyon için kullanıldığında SVR adını alır.

SVR, diğer yapay zeka algoritmalarından farkı vektörel çözüm üretmesidir. Veri içerisindeki örneklemelerden destek vektörler oluşturup bu vektörler arasındaki uzaklığı maksimuma getirmeye çalışır. Böylece gerek sınıflandırma gerekse regresyon problemlerinde bir genelleştirmeye ulaşılır.



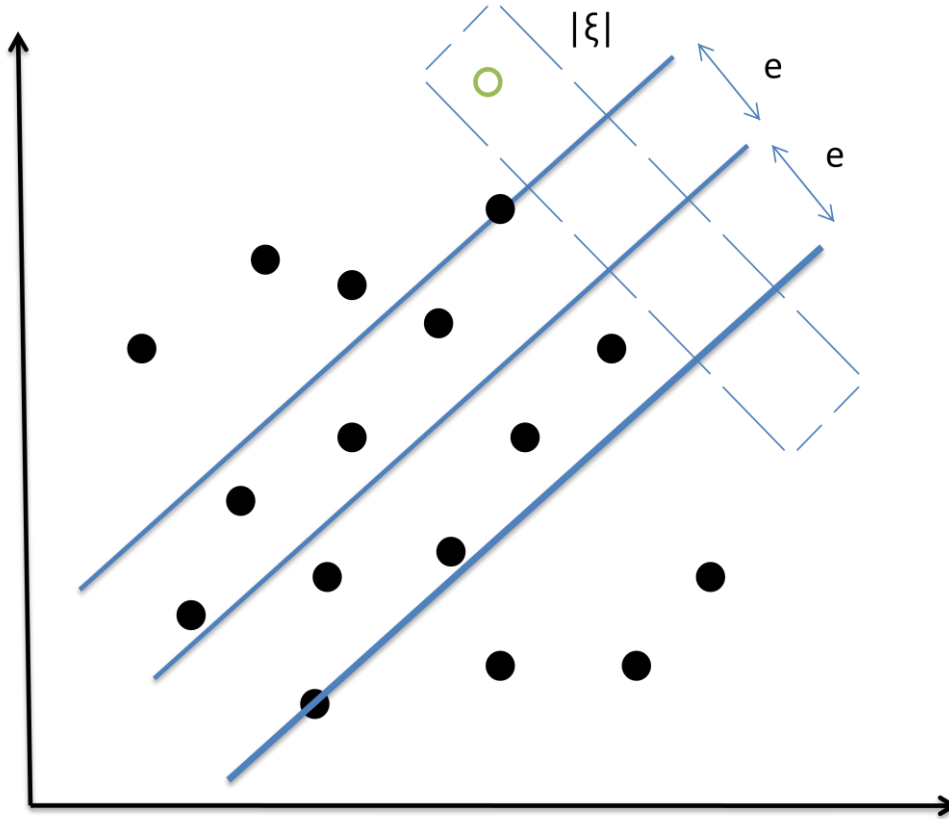
Şekil 4.32 Destek vektör mekanizması

Şekil 4.32 destek vektör mekanizmasını ifade etmektedir. Veri seti içerisindeki örneklemelerin üzerinden geçen iki doğrusal destek vektörü gösterilmiştir. Bu örneklemelere ait vektörler X_1 ve X_2 vektörleridir. $X_1 - X_2$ vektörü ise vektörlerin farkını belirtir. İki destek vektörü arasındaki doğru ise problemin türüne göre regresyon ya da sınıflandırma doğrusudur. Bu Doğru, w ağırlık vektörü ile X örneklem vektörlerinin skaler çarpımına eklenen bir sabit ile bulunur. Destek vektörler bu doğruya eşit uzaklıktadır. w vektörü bu destek vektörlere dik bir vektördür (Cortes and Vapnik, 1995).

$$2e = (X_1 - X_2) \frac{w}{\|w\|} \quad (4.25)$$

$$\min \frac{1}{2} \|w\|^2 \quad (4.26)$$

Destek vektörler arasında kalan örneklem için ξ hata değeri göz ardı edilir. Böylece $y_i - (\langle wX_i \rangle + b) \leq e$ ve $(\langle wX_i \rangle + b) - y_i \leq e$ sınır koşulları oluşur. Denklem 4.25 ise destek vektörler arasında kalan mesafeyi ifade eder. Bu mesafe $X_1 - X_2$ vektörünün w ağırlık vektörü üzerine ortogonal iz düşümü ile ifade edilir. Bu denklemi maksimize etmek için $\|w\|$ ağırlık vektörü normunu minimize etmek yeterlidir. Matematiksel kolaylık elde etmek amacıyla Denklem 4.26’da gösterilen ifade kullanılır (Cortes and Vapnik, 1995) (Basak et al., 2007).



Şekil 4.33 SVR algoritması örneklem hata tayini (Debasish Basak et al., 2007)

SVR algoritmasında destek vektörler vasıtasıyla oluşturulan tüp içerisinde kalan alanda bulunan örneklemelerde ξ hata değeri sıfır kabul edilir. Şekil 4.33’te görülen destek vektörler dışarısında kalan bir her örneklemin ise $|\xi|$ kadar bir hata değeri olacaktır (Basak et al., 2007).

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i + \xi_i^* \quad (4.27)$$

$$\begin{cases} y_i - ((wX_i) + b) \leq e + \xi_i \\ ((wX_i) + b) - y_i \leq e + \xi_i^* \end{cases} \quad (4.28)$$

Denklem 4.27'deki ifade 4.28 ile gösterilen koşullara dayanılarak Lagranjyan çarpanlar kullanılarak optimize edilir. Böylece SVR algoritması destek vektörler yardımıyla veri seti için bir genelleştirme sağlamış olur. C parametresi, regularizasyon için kullanılan bir hata cezalandırma katsayısıdır ve her zaman pozitif alınır (Basak et al., 2007).

Karşılaşılan problemlerin çözümünde bu doğrusal SVR yaklaşımını zaman zaman yetersiz kalabilir. Bu gibi durumlarda doğrusal olmayan çekirdekler kullanılarak veri uzayı manipule edilir. Bu sayede doğrusal olmayan bir SVR algoritması ortaya çıkar. Bu çekirdeklerden biri olan RBF çekirdeği oldukça verimli, hızlı ve yaygın olarak kullanılan bir çekirdektir (Van Gestel et al., 2004). Bu nedenle RBF çekirdeği tercih edilmiştir.

4.3.3. XGBOOST Algoritması

XGBOOST ağaç bazlı karar mekanizmalarına sahip algorithmadan biridir. Diğer ağaç bazlı algoritmalarından farklı olarak tek bir ağaç kullanır bu nedenle diğer algoritmalarla göre oldukça hızlı çalışır (Chen and Guestrin, 2016).

$$\mathcal{L}^{(t)} \cong \sum_{i=1}^n [l(y_i, \hat{y}^{(t-1)} + f_t(X_i))] + \Omega(f_t) \quad (4.29)$$

Denklem 4.29 ile gösterilen ifade XGBOOST algoritmasının hata fonksiyonudur. $\sum_{i=1}^n l(y_i, \hat{y}^{(t-1)} + f_t(X_i))$ hata fonksiyonunun merkezini oluşturur. SSR metodunu kullanır. Her örneklem için bir önceki adımın hesaplanan çıktısı ile bulunulan adımın hesaplanan çıktısını toplar ve o örneklemin çıktısı ile arasındaki farkın karesini alır. Ortaya çıkan bu değerleri toplayarak hata SSR değerini bulur. $\Omega(f_t)$ ise aşırı öğrenmeyi engelleyen regularizasyon ve branş budama ifadesidir (Chen and Guestrin, 2016).

$$\mathcal{L}^{(t)} \cong \sum_{i=1}^n [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(X_i) + h_i f_t^2(X_i)] + \Omega(f_t) \quad (4.30)$$

Denklem 4.29'da $\sum_{i=1}^n l(y_i, \hat{y}^{(t-1)} + f_t(X_i))$ ifadesine ikinci dereceden Taylor açılımı uygulandığında Denklem 4.30 açığa çıkar. Burada g ve h sırasıyla birinci ve ikinci derece türevlerdir (Chen and Guestrin, 2016).

$$\bar{\mathcal{L}}^{(t)} = \sum_{i=1}^n [(g_i f_t(X_i) + h_i f_t^2(X_i))] + \Omega(f_t) \quad (4.31)$$

$$\bar{\mathcal{L}}^{(t)} = \sum_{j=1}^J [(\sum_{i \in I_j} g_i) \omega_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) \omega_j^2] + \gamma J \quad (4.32)$$

Denklem 4.30'dan sabit kısım atılırsa Denklem 4.31 elde edilir. Bu denklemde $\Omega(f_t)$ ifadesi açılır ve $f_t(X_i)$ ifadesi yerine de ω yazılırsa Denklem 4.32 oluşur. $\mathbb{J}$ sayısı, XGBOOST ağacındaki yaprak sayısıdır. γ parametresi aşırı öğrenmeyi engellemek için belirlenen dal budama parametresidir. Yaprak sayısını azaltır. λ ise hata değerlerini cezalandıran regularizasyon parametresidir (Chen and Guestrin, 2016).

$$\omega_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (4.33)$$

$$\bar{\mathcal{L}}^{(t)} = -\frac{1}{2} \sum_{j=1}^{\mathbb{J}} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma \mathbb{J} \quad (4.34)$$

Veri setine uygulanacak XGBOOST regresyonunda öncelikle bir ön çıktı değeri değeri belirlenir. Bu, örneklemelerin hedeflenen çıktı parametrelerinin ortalamasıdır. Algoritmanın karar süreci, her bir örneklemin çıktı değerinin, belirlenen bu ön değer ile farkının alınmasıyla başlar. Böylece hatalar hesaplanır ve bu hatalar oluşturulan ağacın yapraklarına yazılır. Denklem 4.33 ile her yaprak için ω_j^* hata değeri hesaplanır. $(\sum_{i \in I_j} g_i)^2$ ifadesi SSR iken $\sum_{i \in I_j} h_i$ ifadesi ise hatalar toplamıdır. Denklem 4.34 ile de yine her yaprak için SS değeri hesaplanır. Bu SS değerleri vasıtasıyla kök yaprakların kazancı, nihai olarak tüm ağacın kazancı hesaplanır. Her kök yaprak için kazancı en yüksek olan kıstas seçilir. Böylece tek bir ağaç meydana gelir (Chen and Guestrin, 2016).

Ağaçtaki fazla dallanma çoğu zaman aşırı öğrenmeye neden olabilir. Bu nedenle belirlenen bir γ değeri ile aşırı öğrenmeyi engellemek için budama yapılır. Her kök yaprak için hesaplanan kazanç değeri belirlenen γ değerinden küçükse o dal budanır. λ parametresi de çıktı ve SS değerlerini düşürürerek algoritmayı aşırı öğrenme sorunundan kurtarır.

Ağaç oluşturulduktan sonra ağacın tüm çıktı değerleri η öğrenme katsayısıyla çarpılarak bir önceki adımdaki çıktı değerlerine eklenir. Bu iterasyonlar hata değerleri minimuma inen kadar devam eder. Her yeni iterasyonda yeni ve tek bir ağaç oluşturulur.

XGBOOST algoritması hızı ve doğruluğu ile ön plana çıkmaktadır. Bu nedenle zamana dayalı ray sıcaklığı tahmini için elde edilen veri setine ANN ve SVR algoritmalarıyla birlikte uygulanmasına karar verilmiştir. Elde edilen sonuçlar karşılaştırılmıştır.

4.3.4. Algoritmaların Karşılaştırılması

Zaman dayalı ray sıcaklığı tahmini için ANN, SVR ve XGBOOST algoritmaları kullanılmıştır. Bu algoritmaların uygulanmasında python programlama dili ile tensorflow, scikit-learn ve xgboost kütüphaneleri kullanılmıştır. VS Code, Jupyter ve Pycharm geliştirme ortamlarından yararlanılmıştır.

Veri tabanında bulunan veriler bir excel dosyası içerisine aktarılmış ve bu dosya üzerinde yerelde geliştirme yapılmıştır. Algoritmalarından uygun olanı doğruluk ve hız kriterlerine göre seçildikten sonra sunucu tarafındaki geliştirme tamamlanmıştır. Her bir algoritma için ayrı betik yazılmıştır. Bu betikler ekte sunulmuştur.

Algoritma parametrelerinin seçimi, o algoritmanın performansını doğrudan etkilemektedir. Dolayısıyla bu parametrelerin seçimi algoritmalarının optimize edilmesinde birincil etkindir.

Tüm algoritmalarda veri setinin %75'i öğrenme için, %25'i ise doğrulama için ayrılmıştır. Bu yüzdelik dilimlere karşılık gelen veri örneklemeleri veri seti içerisinden tamamen rastlantısal seçilmiştir. Veri seti içerisinde varyans heterojenliği azaltmak amacıyla veriye standardizasyon uygulanmıştır.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (4.35)$$

Algoritmalar sonuçlarının değerlendirilmesi aşamasında Denklem 4.35 ile gösterilen R^2 skorundan yararlanılmıştır. Bu değer istatistikte yaygın olarak kullanılır ve 0 ile 1 arasında değer alır. Ayrıca öğrenme ve test veri setleri arasında RMSE değeri karşılaştırması yapılmıştır.

Kullanılan her yapay zeka algoritmasının öğrenme ve doğrulama işlemlerinin bütünü için geçen süre ms bazında tespit edilmiş ve gecikme süreleri bulunmuştur. İşlemler tekil iş parçacığında yürütülmüştür.

ANN algoritmasının optimizasyonunda, girdi ve çıktı katmanı dışında 4 adet gizli katman kullanılmıştır. Katmanlarda sırasıyla 10, 100, 500, 100, 10, 1 adet algılayıcı kullanılmıştır. Optimizasyon algoritması olarak ADAM algoritması tercih edilmiş ve başlangıç öğrenme katsayısı 0,001 olarak verilmiştir. İterasyon sayısı 300, bellek için paket veri sayısı 256 olarak alınmıştır.

SVR algoritmasının optimizasyonunda RBF çekirdeği kullanılmış, regularizasyon ise parametresi 120 olarak belirlenmiştir. XGBOOST algoritmasının optimizasyonunda öğrenme katsayısı 0,05, maksimum branşlaşma 7, regularizasyon parametresi ise 10 olarak belirlenmiştir.

Algoritma	Öğrenme R^2	Test R^2	Öğrenme RMSE	Test RMSE	Gecikme (ms)
ANN	0,96948	0,93356	1,90309	2,73386	14734,87451
SVR	0,96961	0,9494	1,89918	2,38499	926,94116
XGBOOST	0,95759	0,94274	2,24345	2,47185	763,16675

Tablo 4.10 Ray sıcaklığı tahmin algoritmalarının performansları

Yapay zeka algoritmaları genellikle veri setini öğrenme ve test aşamaları için ikiye ayırır. ANN, SVR ve XGBOOST algoritmaları çalıştırılırken veri setinin öğrenme için ayrılan kısmı tüm veri setinin %75'i olarak belirlenmiştir. Tablo 4.10 ray sıcaklığı tahmin algoritmalarının performanslarını göstermektedir. Performans göstergelerinde veri setinin öğrenme ve test için ayrılan kısımlarının ayrı ayrı R^2 ve RMSE değerleri incelenmiş, bunlara ek olarak her algoritma için gecikme süresi ms cinsinden saptanmıştır.

ANN, SVR ve XGBOOST algoritmaları kendi içerisinde ayrı ayrı incelendiğinde tümünde test ile öğrenme R^2 ve RMSE değerleri arasında belirgin bir fark görülmemektedir. Bu veri modeli için uygulanan regresyonların tümünde fazla ya da zayıf öğrenme probleminin olmadığını göstermektedir. ANN algoritması SVR ve XGBOOST algoritmalarına çok daha yüksek bir gecikmeye sahiptir. SVR ve XGBOOST performanslarına bakıldığında R^2 ve RMSE açısından belirgin bir farklılaşmanın mevcut olmadığı görülmektedir. Bu iki algoritmanın gecikme değerleri de birbirine çok yakındır.

Tüm bu bilgiler göz önüne alındığında gecikme süresi nedeniyle ANN algoritması dışarılanmıştır. XGBOOST algoritması da xgboost kütüphanesi güvenli alt iş parçacığı modelini desteklemediği için tercih edilmemiştir. Söz konusu algoritma REST sunucu içerisinde alt iş parçacığı şeklinde çalışacağından SVR algoritmasının kullanılması daha uygun bulunmuştur. XGBOOST algoritması, ayrık bir sunucu geliştirilerek REST sunucusu ile haberleştirilmesi halinde kullanılabilir. ANN algoritması ise veri seti çok büyüdüğünde XGBOOST ve SVR algoritmasına kıyasla daha uygun kalabilir.

REST sunucusunda ray sıcaklığı tahmini SVR algoritmasıyla gerçekleştirilmiştir. Bu algoritma kullanıcıdan her tahmin isteği geldiğinde bir alt iş parçacığı olarak çalışmakta ve çıktı değerini tekrar kullanıcıya dönmektedir.

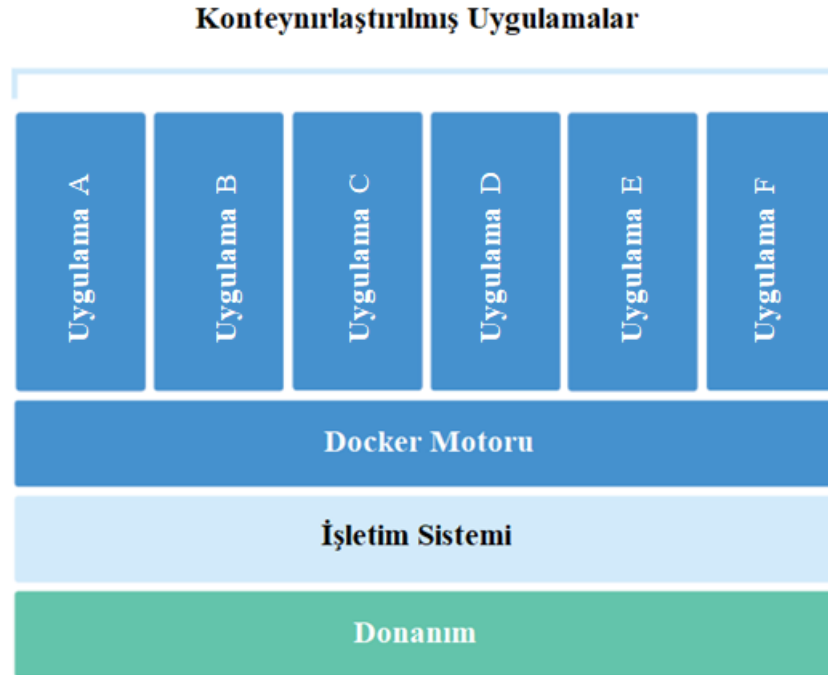
5. SUNUCULARIN DEVREYE ALINMASI

Sistemde bulunan üç sunucudan TCP soket sunucusu sahadaki uç cihaz üzerinde bulunurken REST ve IOT sunucuları belirlenmiş bir donanım üzerine kurulmuştur. Sunucuların ve PLC'nin kod bütünleri açık kaynak olarak github üzerinde paylaşılmıştır.

Uç cihaz içerisinde çalışan sunucu, pm2 kütüphanesi yardımıyla devreye alınmış olup uç cihaz çalışmaya başlar başlamaz kullanıma açılmaktadır. Herhangi bir enerji kesintisinden sonra uç cihazla tekrar enerjilendiği anda donanımın ayağa kalkmasıyla devreye girer. Sürekli olarak PLC'yi dinleyerek veriyi MQTT protokolü ile aracıya iletir.

IOT sunucusu ise kullanıcıya açık değildir. REST sunucusuyla aynı veri tabanını kullanmasına rağmen bağımsız çalıştıkları için ayrı ayrı devreye alınabilirler. Hem IOT hem REST sunucusunun devreye alınması için konteynırlaştırma uygulanmıştır. Konteynırlaştırma işlemi için Docker teknoloji kullanılmıştır.

Docker motoru kod bütünleri için konteynırlaştırma sağlayan Linux çekirdeği üzerinde çalışan bir teknolojidir. Docker, geliştirilen yazılımın işletim sisteminden bağımsız bir şekilde çalışmasını sağlar (Docker, 2021).



Şekil 5.1 Docker konteynırlaştırma yapısı (Docker , 2021)

Docker motorunun olduđu her yerde Linux çekirdeđi üzerinden sanallaştırma yapılabilir. Sanallaştırma yazılımı izole ederek bağımsız hale getirir. Böylece platformdan bağımsız çalışabilen tek bir obje elde edilmiş olur. Bu obje konteynır haline getirilir. Şekil 5.1 Docker teknolojisinin kullandığı konteynırlaştırma yapısını göstermektedir. Konteynırlar Docker motorunun sağladığı Linux çekirdeğini ortak kullanırlar. Bu konteynırlar hem Linux hem Windows işletim sistemlerinde rahatlıkla çalıştırılabilir. Ayrıca Docker için geliştirmede bulunan güçlü bir topluluğun var olması kaynak bulmada yazılım geliştiricilerin işini kolaylaştırmaktadır (Docker, 2021).

Docker kullanırken birden fazla konteynırı yönetmek ihtiyacı doğmuştur. Bu operasyonların daha kolay ve hızlı yönetilmesi için Docker-compose kullanılmıştır. Docker-compose çoklu Docker konteynırları oluşturmaya ve çalıştırmaya olanak tanıyan bir operasyon aracıdır. Uygulamalar için Docker-compose dosyası hazırlandıktan sonra komut satırı ara yüzünde tek bir komut ile bu uygulamaların tümü ayağa kaldırılabilir. Bunun için her uygulamanın Docker dosyası hazırlanır ve bu Docker-compose dosyası içinde bu Docker dosyalarına referans verilir (Docker, 2021).

REST sunucusunun önünde ters vekil sunucu olarak nginx kullanılmıştır. Ters vekil sunucu olarak kullanılan nginx sunucuyu ileriki zamanlarda ihtiyaç duyulabilecek yük dengeleme ve ölçekleme ihtiyaçlarına karşı hazır tutmak için kullanılmıştır. REST sunucusu kurumsal yapı dışarısında kullanılmayacağından nginx, pilot uygulama süresince REST sunucusuyla aynı donanım üzerinde çalıştırılmıştır. Farklı donanımda çalıştırılıp REST sunucusunun DNS bilgisini saklama ihtiyacı duyulmamıştır. Bu nedenle REST ve nginx sunucuları için aynı donanım içerisinde konteynırlaştırma uygulanmıştır.

6. SONUÇ VE ÖNERİLER

Demiryollarında belirlenen hatlar üzerinde seyreden demiryolu araçlarının hızlarının anlık izlenmesi sayesinde hat güvenliği açısından işletmenin elini güçlendiren bir unsur olmuştur. Belirlenen noktalarda ölçülen ve takip edilen hızlar, hat üzerinde seyreden araçların belirlenen hız limitlerine uyup uymadığının kontrolünü sağlar. Böylece aynı zamanda yolcu güvenliğine de katkıda bulunulmuş olur. Hat güvenliğini oluşturan temel unsurlar ise özellikle kısa yarı çaplı kurp bölgelerinde hız limitlerine uyum kontrolü vasıtasıyla ray ömrünün uzatılmasıdır. Kurp bölgelerinde hıza bağlı olarak ray-teker sürtünmesi sonucu oluşan ray aşınması, ray ömrünü önemli ölçüde azalttığından düzenli hız takibi ile bu etki en aza indirilmektedir. Bu sayede ray ömrü nedeniyle ortaya çıkacak üst yapı yenileme maliyetleri geciktirilmiş olur. Aynı zamanda ray yağlama maliyetlerinden de kaçınılmış olur.

Ray sıcaklığı bilgilerinin düzenli olarak toplanması uzun vadede rayın mekanik ömrünün hesap edilebilmesine ve olası ray kırıklarının önüne geçilmesine yardımcı olur. Ray sıcaklığının takip edilmesi ise kritik sıcaklıklar sonucu oluşabilecek mekanik sorunların anlık olarak öngörülüp hat trafiğinin düzenlenmesine yardımcı olur.

Zamana dayalı ray sıcaklığı tahmini, ileri bir tarihteki bir zaman diliminde belirlenen hat üzerindeki ray sıcaklığını tahmin ederek özellikle düşük sıcaklıkların ve buzlanmanın görüldüğü kış koşulları ile yüksek sıcaklıkların görüldüğü yaz koşullarında bakım programlarının dinamik hale getirilmesini sağlar. Hat burajı gibi bakım çalışmalarına çok düşük ve çok yüksek sıcaklıklar hat mekaniği için tehlike arz ettiğinden ray sıcaklığının tahmin edilmesi ile hat burajı sırasında oluşabilecek mekanik risk ortadan kaldırılmış olur. Öte yandan esnek ve dinamik bakım programı oluşturulmasına olanak tanıdığından işgücü kaybı önlenerek işletme maliyetleri optimize edilmiş olur.

Ray sıcaklığı tahmin mekanizması için varsayılan analitik model, oluşturulan bir veri modeli ile ikame edildiğinden analitik model için gereken yüksek sistem donanımı maliyeti ortadan kaldırılmıştır. Analitik modeli ikame eden veri modeli için uygulanan yapay zeka algoritmaları (ANN, SVR ve XGBOOST) ortalama % 94 seviyesinde bir doğruluk ile çalışmıştır. Bu sonuçlar geliştirilen veri modelinin, analitik model yerine başarıyla kullanılabileceğini göstermiştir.

Sistemin tasarımı esnasında sürekli geliştirmeye uygunluk göz önünde bulundurulmuştur. İleriki aşamalarda kullanıcı sayısının artması halinde REST sunucusu üzerinde kullanıcı kaydı, kurumsal onay kodu yerine kullanıcının elektronik posta adresine yönlendirilen bir elektronik posta ile yapılabilir. Bu kullanıcı güvenliğini daha üst bir seviyeye çıkaracaktır. Öte yandan ölçüm istasyonlarının sayısı arttıkça oluşturulmuş mikro servis mimarisinin de sayısı arttırılıp bulut üzerinde konuşlandırılabilir. Saha personelinin mobil cihazlarda tarayıcı üzerinden sistemi kullanımı nispeten verimsiz olacağından bir mobil uygulama geliştirilmesi uygun olabilir.

KAYNAKLAR DİZİNİ

- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning. Neural Networks and Deep Learning*. doi:10.1007/978-3-319-94463-0
- Arnor Mar Gudmundsson ve Freyr Gudmundsson. (2014). *Prestress losses in railway sleeper production with long bed systems*. CHALMERS UNIVERSITY OF TECHNOLOGY, Göteborg.
- Bircanoglu, C. ve Arica, N. (2018). A comparison of activation functions in artificial neural networks. *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018* içinde . doi:10.1109/SIU.2018.8404724
- Čerešňák, R. ve Kvet, M. (2019). Comparison of query performance in relational a non-relation databases. *Transportation Research Procedia* içinde (C. 40). doi:10.1016/j.trpro.2019.07.027
- Chen, T. ve Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* içinde (C. 13-17-August-2016). doi:10.1145/2939672.2939785
- Cortes, C. ve Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3). doi:10.1023/A:1022627411411
- de Aguiar, E. P., Amaral, R. P. F., Vellasco, M. M. B. R. ve Ribeiro, M. V. (2018). An enhanced singleton type-2 fuzzy logic system for fault classification in a railroad switch machine. *Electric Power Systems Research*, 158. doi:10.1016/j.epsr.2017.12.018
- Debasish Basak, Srimanta Pal ve Dipak Chandra Patranabis. (2007). Support Vector Regression.
- Dr. Szabolcs Fischer, Balazs Eller, Zoltan Kada ve Attila Nemeth. (2015). *railway construction*. Győr: Universitas-Győr Nonprofit Kft.
- Henriques, J., Ehrhardt, S., Albanie, S. ve Vedaldi, A. (2019). Small steps and giant leaps: Minimal newton solvers for deep learning. *Proceedings of the IEEE International Conference on Computer Vision* içinde (C. 2019-October). doi:10.1109/ICCV.2019.00486

KAYNAKLAR DİZİNİ(devam)

- Hong, S. U., Kim, H. U., Lim, N. H., Kim, K. H., Kim, H. ve Cho, S. J. (2019). A Rail-Temperature-Prediction Model Considering Meteorological Conditions and the Position of the Sun. *International Journal of Precision Engineering and Manufacturing*, 20(3). doi:10.1007/s12541-019-00015-1
- <https://www.docker.com/>. (2021).
- <https://www.klemsan.com.tr/>. (y.y.).
- Incropera, F. P., DeWitt, D. P., Bergman, T. L. ve Lavine, A. S. (2007). heat and mass transfer - Incropera 6e. *Fundamentals of Heat and Mass Transfer*.
- Jung, M. G., Youn, S. A., Bae, J. ve Choi, Y. L. (2016). A study on data input and output performance comparison of MongoDB and PostgreSQL in the big data environment. *Proceedings - 8th International Conference on Database Theory and Application, DTA 2015* içinde . doi:10.1109/DTA.2015.14
- Kingma, D. P. ve Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* içinde .
- Kish, A. ve Samavedam, G. (2013). Track Buckling Prevention: Theory, Safety Concepts, and Applications. *U.S. Department of Transportation, DOT/FRA/OR*(March).
- Li, Y. ve Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings* içinde . doi:10.1109/PACRIM.2013.6625441
- Muškinja, N., Tovornik, B. ve Terbuc, M. (2003). Use of TCP/IP protocol in industrial environment. *Proceedings of the IEEE International Conference on Industrial Technology* içinde (C. 2). doi:10.1109/icit.2003.1290778
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings* içinde . doi:10.1109/SysEng.2017.8088251
- Nambiar, R., Shroff, R. ve Handy, S. (2018). Smart cities: Challenges and opportunities. *2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018* içinde (C. 2018-January). doi:10.1109/COMSNETS.2018.8328204

KAYNAKLAR DİZİNİ(devam)

- Nima Tahvili. (2016). *Winterization of Railways Issues and Effects*. Norwegian University of Science and Technology, Trondheim.
- Pietrzak, O. ve Pietrzak, K. (2019). The role of railway in handling transport services of cities and agglomerations. *Transportation Research Procedia* içinde (C. 39).
doi:10.1016/j.trpro.2019.06.043
- Railway Infrastructure Products - voestalpine Railway Systems. (y.y.). 23 Mart 2021 tarihinde [https://www.voestalpine.com/railway-systems/en/products/#!view=products&product_category_ids=%2F.categories%2Fproduct%2Fcategory%2Frails%2F](https://www.voestalpine.com/railway-systems/en/products/#!view=products&product_category_ids=%2Fcategories%2Fproduct%2Fcategory%2Frails%2F) adresinden erişildi.
- Ruder, S. (2017). Overview Optimization Gradients. *arXiv preprint arXiv:1609.04747*.
- Shitzer, A. (2006). Wind-chill-equivalent temperatures: Regarding the impact due to the variability of the environmental convective heat transfer coefficient. *International Journal of Biometeorology*, 50(4). doi:10.1007/s00484-005-0011-x
- Statistics | Eurostat. (y.y.-a). 17 Mart 2021 tarihinde <https://ec.europa.eu/eurostat/databrowser/view/ttr00003/default/table?lang=en> adresinden erişildi.
- Statistics | Eurostat. (y.y.-b). 17 Mart 2021 tarihinde <https://ec.europa.eu/eurostat/databrowser/view/ttr00015/default/table?lang=en> adresinden erişildi.
- support.industry.siemens.com. (y.y.).
- Tilkov, S. ve Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6). doi:10.1109/MIC.2010.145
- Types of rails - ArcelorMittal. (y.y.). 23 Mart 2021 tarihinde <https://rails.arcelormittal.com/types-rails/> adresinden erişildi.
- Van Gestel, T., Suykens, J. A. K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., ... Vandewalle, J. (2004). Benchmarking Least Squares Support Vector Machine Classifiers. *Machine Learning*. doi:10.1023/B:MACH.0000008082.80494.e0

KAYNAKLAR DİZİNİ(devam)

Vangi, D. ve Virga, A. (2007). A Practical Application of Ultrasonic Thermal Stress Monitoring in Continuous Welded Rails. *Experimental Mechanics*, 47(5).
doi:10.1007/s11340-006-9016-6

www.hivemq.com. (2021).

www.sick.com. (y.y.).

Yang, G. ve Bradford, M. A. (2016). Thermal-induced buckling and postbuckling analysis of continuous railway tracks. *International Journal of Solids and Structures*, 97_98.
doi:10.1016/j.ijsolstr.2016.04.037

TEŞEKKÜR

Bu çalışma süresince başta ailem olmak üzere derin bilgi birikimi ve deneyimleri ile bana yol gösteren kıymetli hocam Doç. Dr. Mustafa Turhan ÇOBAN'a, yardımlarını eksik etmeyen demiryollarının değerli ustaları Ali UĞUR ile Yaşar Ernoyan HACIOSMANOĞLU'na ve Türkiye Cumhuriyeti Devlet Demiryollarına teşekkür ederim.

Ve bizlere aydınlanmanın, pozitif bilimlerin yolunu açan büyük önder Mustafa Kemal ATATÜRK'e şükranlarımı sunarım.

.

25 / 02 / 2022

İmzası

Adı-Soyadı

ÖZGEÇMİŞ

1987 yılında İzmir’de doğdu. İlk, orta ve lise öğretimini İzmir’de tamamladıktan sonra 2005 yılında mühendislik lisans eğitimine başladı. 2010 yılında Ege Üniversitesi Makine Mühendisliği bölümünden mezun olduktan sonra bir süre özel sektörde çalıştı. 2015 yılının başında TCDD kamu iktisadi teşekkülünde Makine Mühendisi unvanıyla göreve başladı. TCDD İzmir 3. Bölge Müdürlüğü’nde demiryolu hatlarında ve makinelerinde otomasyon, IOT, dijitalleşme ve yazılım çalışmaları yapmaktadır.

EKLER

EK1: Yapay Zeka Algoritmaları Betikleri

EK2: Docker ve Docker-compose Dosyaları

EK1: Yapay Zeka Algoritmaları Betikleri

ANN betiği

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import r2_score

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.optimizers import Adam

import matplotlib.pyplot as plt

import time

from sklearn.metrics import mean_squared_error as mse

from ann_visualizer.visualize import ann_viz

veri=pd.read_excel('veri.xlsx')

veri.head()
```



```
veri_girdiler=veri.drop(["Rail Temperature_1","Day","Minute"],axis=1)

veri_cikti=veri["Rail Temperature_1"]

veri_girdiler["Sky"]=LabelEncoder().fit_transform(veri_girdiler["Sky"])

veri_girdiler["Month"]=LabelEncoder().fit_transform(veri_girdiler["Month"])

veri_girdiler.head()

X_train, X_test, y_train, y_test=train_test_split(veri_girdiler, veri_cikti,

                                                    test_size=0.25, random_state=42)

X_train=StandardScaler().fit_transform(X_train)

X_test=StandardScaler().fit_transform(X_test)

model=Sequential()

model.add(Dense(10, activation="relu", input_dim=X_train.shape[1]))

model.add(Dense(100, activation="relu"))

model.add(Dense(500, activation="relu"))

model.add(Dense(100, activation="relu"))

model.add(Dense(10, activation="relu"))

model.add(Dense(1, activation="linear"))
```



```
opt = Adam(learning_rate=0.001)

model.compile(optimizer=opt, loss='mse',

              metrics=[tf.keras.metrics.RootMeanSquaredError()])

baslangic_an=time.time()*1000

history=model.fit(X_train, y_train, epochs=300, batch_size=256,

                 validation_split=0.25, verbose=1)

tahmin_y_test=model.predict(X_test)

print(round(r2_score(tahmin_y_test, y_test),5))

bitis_an=time.time()*1000

print(round((bitis_an-baslangic_an),5))

loss=history.history["loss"]

loss_validation=history.history["val_loss"]

plt.figure(figsize=(8, 6), dpi=100)

plt.plot(loss, "b", label="Öğrenmedeki Hata")

plt.plot(loss_validation, "r", label="Doğrulamadaki Hata")

plt.xlabel("İterasyon")

plt.ylabel("Hata")

print(y_test)
```



```
print(tahmin_y_test)

tahmin_y_train=model.predict(X_train)

rmse_test = round(mse(y_test, tahmin_y_test, squared=False),5)

rmse_train = round(mse(y_train, tahmin_y_train, squared=False),5)

r2_train=round(r2_score(y_train, tahmin_y_train),5)

print(rmse_test)

print(rmse_train)

print(r2_train)

ann_viz(model, view=True, filename="ann.pdf", title="ann")
```


SVR betiği

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn import preprocessing

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model_selection import GridSearchCV

from sklearn.svm import SVR

from sklearn.metrics import r2_score

import matplotlib.pyplot as plt

import time

from sklearn.metrics import mean_squared_error as mse

veri=pd.read_excel('veri.xlsx')

veri.head()

veri_girdiler=veri.drop(["Rail Temperature_1","Day","Minute"],axis=1)

veri_cikti=veri["Rail Temperature_1"]

veri_girdiler["Sky"]=LabelEncoder().fit_transform(veri_girdiler["Sky"])

veri_girdiler["Month"]=LabelEncoder().fit_transform(veri_girdiler["Month"])
```



```
veri_girdiler.head()

X_train, X_test, y_train, y_test=train_test_split(veri_girdiler, veri_cikti,

                                                    test_size=0.25, random_state=42)

X_train=StandardScaler().fit_transform(X_train)

X_test=StandardScaler().fit_transform(X_test)

parametreler={

    'C': [1, 10, 75, 80, 90, 100, 120],

    'epsilon': [ 0.001, 0.01, 0.05, 0.1]

}

optimizasyon=GridSearchCV(estimator=SVR(kernel="rbf"), param_grid=parametreler,

                           scoring="r2", n_jobs=8, cv=5)

optimizasyon.fit(X_train,y_train)

print(optimizasyon.best_params_)

baslangic_an=time.time()*1000

svr = SVR(kernel='rbf', C=120, gamma="scale", epsilon=0.1)

svr.fit(X_train,y_train)

tahmin_y_test=svr.predict(X_test)
```



```
print(round(r2_score(tahmin_y_test, y_test),5))

bitis_an=time.time()*1000

print(round((bitis_an-baslangic_an),5))

data=pd.DataFrame({'Output': y_test, "Predict":tahmin_y_test})

pd.set_option('display.max_rows', None)

print(data)

tahmin_y_train=svr.predict(X_train)

rmse_test = round(mse(y_test, tahmin_y_test, squared=False),5)

rmse_train = round(mse(y_train, tahmin_y_train, squared=False),5)

r2_train=round(r2_score(y_train, tahmin_y_train),5)

print(rmse_test)

print(rmse_train)

print(r2_train)
```


XGBOOST betiği

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model_selection import GridSearchCV

import xgboost as xgb

from sklearn.metrics import r2_score

import matplotlib.pyplot as plt

import time

from sklearn.metrics import mean_squared_error as mse

veri=pd.read_excel('veri.xlsx')

veri.head()

veri_girdiler=veri.drop(["Rail Temperature_1","Day","Minute"],axis=1)

veri_cikti=veri["Rail Temperature_1"]

veri_girdiler["Sky"]=LabelEncoder().fit_transform(veri_girdiler["Sky"])

veri_girdiler["Month"]=LabelEncoder().fit_transform(veri_girdiler["Month"])

veri_girdiler.head()
```



```
X_train, X_test, y_train, y_test=train_test_split(veri_girdiler, veri_cikti,

                                                    test_size=0.25, random_state=42)

X_train=StandardScaler().fit_transform(X_train)

X_test=StandardScaler().fit_transform(X_test)

parametreler={

    "learning_rate":[0.05, 0.1, 0.3],

    'max_depth': [ 3, 7, 10],

    'colsample_bytree': [ 0.5, 0.8, 1],

    'reg_lambda': [ 0.01, 0 , 1, 10],

}

optimizasyon=GridSearchCV(estimator=xgb.XGBRegressor( base_score = 0.5 , booster =
"gbtree", n_estimators=500, subsample=0.8,

                                objective = 'reg:squarederror', verbosity = 1 ),

                            param_grid=parametreler, scoring="r2", n_jobs=8, cv=5)

optimizasyon.fit(X_train,y_train)

print(optimizasyon.best_params_)

baslangic_an=time.time()*1000
```



```
veri_xgb=xgb.XGBRegressor( base_score = 0.5 , booster = "gbtree" ,

                           colsample_bytree = 0.8 ,

                           gamma = 100 , learning_rate = 0.05 , max_depth = 7 ,

                           missing = 1 , n_estimators = 300 , nthread=None,

                           objective = 'reg:squarederror' , reg_lambda = 10 ,

                           seed=42 , silent = None , subsample = 1 , verbosity = 1 )

veri_xgb.fit(X_train,y_train,verbose=True,eval_metric="rmse")

tahmin_y_test = veri_xgb.predict(X_test)

print(round(r2_score(tahmin_y_test, y_test),5))

bitis_an=time.time()*1000

print(round((bitis_an-baslangic_an),5))

data=pd.DataFrame({'Output': y_test, "Predict":tahmin_y_test})

pd.set_option('display.max_rows', None)

print(data)

tahmin_y_train=veri_xgb.predict(X_train)

rmse_test = round(mse(y_test, tahmin_y_test, squared=False),5)
```



```
rmse_train = round(mse(y_train, tahmin_y_train, squared=False),5)
```

```
r2_train=round(r2_score(y_train, tahmin_y_train),5)
```

```
print(rmse_test)
```

```
print(rmse_train)
```

```
print(r2_train)
```


EK2: Docker ve Docker-compose Dosyaları:

REST sunucusu Docker dosyası

```
FROM nikolaik/python-nodejs

RUN apt-get update

RUN pip install -U scikit-learn

RUN pip install pandas

COPY . .

RUN npm cache clean --force && npm install --production

CMD npm run start

EXPOSE 3578
```

NGINX Docker dosyası

```
FROM nginx

WORKDIR /Desktop/restapi/nginx

COPY default.conf /etc/nginx/conf.d/
```


IOT sunucusu Docker dosyası

```
FROM node:8-alpine
```

```
COPY . .
```

```
RUN npm cache clean --force && npm install --production
```

```
CMD npm run start
```

Docker-compose dosyası

```
version: "3.3"
```

```
services:
```

```
  app:
```

```
    container_name: restapi-container
```

```
    image: restapi
```

```
    restart: always
```

```
    env_file:
```

```
      - ./restapi/config/config.env
```

```
    depends_on:
```

```
      - hosting
```


hosting:

container_name: nginx-container

image: nginx

restart: always

ports:

- "3578: 3578"

iot:

container_name: iot-container

image: iot

restart: always

env_file:

- ./iot/config/config.env