



NESNELERİN İNTERNETİ (IOT)

MARS ROVER ÜZERİNDE EDGE COMPUTING (UÇ BİLİŞİM) DESTEKLİ KESTİRİMCİ BAKIM VE YEREL TELEMETRİ SİSTEMİ

Hazırlayanlar:

- 1)2023123003 – Ebrar KILIÇ**
- 2)2023141311 – Mehmet YENER**
- 3)2024123289 – Fatma Merve KILIÇARSLAN**
- 4)2023141002 – Beyda TURA**
- 5)2023123340 – Beyza DEMİRTAŞ**
- 6)2024123308 – Gizem ÖZBEK**
- 7)2022123044 – Onur MEŞE**
- 8)2024123291 – Tuğba ERTENGİ**
- 9)2023123051 – Elif Su PERİŞAN**

1. EKİP BİLGİLERİ VE GÖREV DAĞILIMI

Bu proje, farklı uzmanlık alanlarına sahip 9 kişilik mühendislik ekibi tarafından "Agile" (Çevik) proje yönetim prensipleriyle yürütülmüştür. Her ekip üyesinin üstlendiği teknik görevler ve sorumluluklar aşağıda detaylandırılmıştır:

1. Ebrar KILIÇ (2023123003) – Proje Yöneticisi & Gömülü Sistemler Lideri

- **Görevi:** Projenin elektromekanik montajını, güç dağıtım mimarisini ve donanım entegrasyonunu yönetmiştir.
- **Teknik Katkısı:** 6 adet DC motorun yük altındaki akım ihtiyacını analiz ederek, başlangıçta planlanan L298N sürücüsünün yetersizliğini (2A sınırını) tespit etmiş ve sistemi **BTS7960B** (43A) endüstriyel sürücülerle revize etmiştir. Güç stabilitesi için **XL4015** regülatör devresini kurmuş ve sensörlerin parazitsiz çalışması için ortak toprak (Common Ground) hattı tasarımını gerçekleştirmiştir.

2. Mehmet YENER (2023141311) – Gömülü Yazılım Geliştirici & Hibrit Entegrasyon

- **Görevi:** Raspberry Pi ve Arduino Nano üzerindeki tüm gömülü yazılımların geliştirilmesi.
- **Teknik Katkısı:** Raspberry Pi'nin (Master) Arduino'yu (Slave) yönetmesini sağlayan **USB Serial** haberleşme protokolünü yazmıştır. Sensör verilerini okuyup JSON formatında paketleyen ve Wi-Fi üzerinden sunucuya ileten ana döngüyü, motorları kilitlememesi için **Asenkron (Asyncio)** yapıda kodlamıştır.

3. Onur MEŞE (2022123044) – Backend Lideri & Yerel Sunucu Mimarı

- **Görevi:** Sistemin yerel ağ üzerinde çalışan sunucu ve veritabanı altyapısının kurulması.
- **Teknik Katkısı:** Raspberry Pi üzerinde **Python FastAPI** framework'ü ile yüksek performanslı bir REST API ve WebSocket sunucusu geliştirmiştir. Sensör verilerinin güvenli ve sorgulanabilir olması için yerel **PostgreSQL** veritabanı mimarisini tasarlamış ve SensorData tablosunu optimize etmiştir.

4. Elif Su PERİŞAN (2023123051) – İletişim Mimarı & Ağ Yöneticisi

- **Görevi:** Robot ile Kontrol İstasyonu arasındaki veri trafiğinin ve bağlantı güvenliğinin yönetilmesi.
- **Teknik Katkısı:** ConnectionManager sınıfını geliştirerek WebSocket trafiğini yönetmiştir. Bağlantı kopması (Disconnect) durumunda sistemin bunu algılayıp arayüze anında "OFFLINE" bilgisini iletmesini sağlayan **Heartbeat** mekanizmasını ve veri yayınlama (Broadcast) algoritmalarını kurmuştur.

5. Gizem ÖZBEK (2024123308) – Frontend Lideri & UI/UX Tasarımcı

- **Görevi:** Operatörün sistemi izlediği ve yönettiği web tabanlı kontrol panelinin (Dashboard) geliştirilmesi.
- **Teknik Katkısı:** **React.js** kullanarak modüler ve ölçeklenebilir bir arayüz mimarisi kurmuştur. SocketContext yapısı ile sunucudan gelen canlı verilerin tüm bileşenlere (Grafikler, Uyarılar) gecikmesiz dağıtılmasını sağlayan merkezi durum yönetimini (State Management) kodlamıştır.

6. Beyza DEMİRTAŞ (2023123340) – Veri Görselleştirme Uzmanı

- **Görevi:** Sensörlerden gelen sayısal verilerin, operatörün anlayabileceği grafiklere dönüştürülmesi.
- **Teknik Katkısı:** Recharts kütüphanesini kullanarak Akım, Sıcaklık ve Titreşim verilerini "Kayan Pencere" (Sliding Window) yöntemiyle canlı akan grafiklere dönüştürmüştür. Anomali durumunda grafiklerin renk değiştirilerek operatörü uyarmasını sağlayan dinamik görselleştirme mantığını kurmuştur.

7. Beyda TURA (2023141002) – Kontrol Sistemleri Geliştirici

- **Görevi:** Kullanıcı komutlarının robota iletilmesi ve güvenlik mekanizmalarının arayüze entegrasyonu.
- **Teknik Katkısı:** Klavye (WASD) ve ekran butonlarını dinleyerek, bunları robotun anlayacağı standart JSON komutlarına ({"komut": "ileri"}) dönüştüren modülü yazmıştır. Kritik durumlarda sistemi milisaniyeler içinde durduran "**ACİL STOP**" butonunu sisteme entegre etmiştir.

8. Fatma Merve KILIÇARSLAN (2024123289) – Yapay Zeka Model Geliştirici

- **Görevi:** Kestirimci bakım için kullanılacak Yapay Zeka modelinin seçimi, eğitimi ve entegrasyonu.
- **Teknik Katkısı:** Arıza verisinin azlığı (Dengesiz Veri Seti) problemini çözmek için Denetimsiz Öğrenme tabanlı **Isolation Forest** algoritmasını seçmiştir. Modeli yerel bilgisayarda eğitip, serileştirilmiş (.pkl) formatta Raspberry Pi üzerine deploy ederek **Edge AI** (Uçta Yapay Zeka) vizyonunu gerçekleştirmiştir.

9. Tuğba ERTENGİ (2024123291) – Veri Analisti & Raporlama

- **Görevi:** Sensör verilerinin analizi, model başarısının ölçülmesi ve teknik dokümantasyon.
- **Teknik Katkısı:** Toplanan verilerin istatistiksel analizini yaparak "Normal" ve "Anormal" çalışma aralıklarını (Örn: Normal Akım < 200mA) belirlemiştir. Modelin anomali yakalama başarısını (Recall: %92.4) ölçmüş ve projenin teknik raporlamasını grafiksel analizlerle destekleyerek hazırlamıştır.

2. PROBLEMİN TANIMI VE ÖNEMİ

2.1. Problemin Tanımı Uzay araştırmalarında kullanılan insansız kara araçları (Rover), Dünya'dan milyonlarca kilometre uzakta, aşırı sıcaklık değişimleri, toz fırtınaları ve engebeli arazi gibi ekstrem fiziksel koşullar altında otonom veya yarı otonom olarak görev yapmaktadır. Bu araçlarda meydana gelebilecek motor sıkışması, dişli kutusu arızası, rulman dağılması veya aşırı ısınma gibi mekanik arızalar, görevin kısmen veya tamamen başarısızlıkla sonuçlanmasına neden olabilir.

Mevcut operasyonel senaryolarda karşılaşılan en temel problem, Dünya ile Mars arasındaki iletişim gecikmesidir (Latency). İki gezegen arasındaki mesafeye bağlı olarak sinyallerin gidiş-dönüş süresi (Round Trip Time - RTT) 4 ila 24 dakika arasında değişmektedir. Bu durum, bir arıza anında (örneğin bir motorun kayaya sıkışıp aşırı akım çekmeye başlaması) Dünya'daki operatörlerin anlık müdahale etmesini imkânsız kılmaktadır. Geleneksel "arıza olduktan sonra tespit etme" (Reaktif Bakım) yaklaşımı, bu gecikme süresi içinde donanımın kalıcı hasar görmesine yol açmaktadır.

Ayrıca, bulut tabanlı (Cloud-based) izleme sistemleri sürekli ve yüksek bant genişliğine sahip bir internet bağlantısı gerektirmektedir. Uzay görevlerinde veya yeryüzündeki zorlu endüstriyel sahalarda (madenler, petrol platformları vb.) kesintisiz bağlantı garantisi yoktur. Bağlantının koptuğu durumlarda robotun "kör" kalması ve verileri işleyememesi, operasyonel güvenliği tehdit eden kritik bir sorundur.

2.2. Projenin Önemi ve Çözüm Yaklaşımı Bu proje, yukarıda tanımlanan problemlere "**Kestirimci Bakım (Predictive Maintenance)**" yaklaşımını "**Edge Computing (Uç Bilişim)**" mimarisiyle birleştirerek bir çözüm getirmektedir.

Geliştirilen sistemin önemi şu noktalarda öne çıkmaktadır:

1. **Gerçek Zamanlı Müdahale (Edge AI):** Sensör verilerinin analiz edilmek üzere uzak bir sunucuya gönderilmesi yerine, yapay zeka modelinin doğrudan robot üzerindeki işlemcide (Raspberry Pi) çalıştırılması sağlanmıştır. Bu sayede sistem, anormallikleri **<20ms** gibi bir sürede tespit edip, dış dünyadan bağımsız olarak motorları durdurma veya güvenli moda geçme kararını otonom olarak verebilmektedir.
2. **Prognostik (Öngörücü) Koruma:** Isolation Forest algoritması sayesinde, motor akımı veya titreşim verilerindeki mikroskobik değişimler analiz edilerek, arıza henüz fiziksel bir hasara yol açmadan önce tespit edilmektedir. Bu, donanımın ömrünü uzatmakta ve görev başarısını garanti altına almaktadır.
3. **Veri Güvenliği ve Süreklilik (Offline-First):** Sistem, "Offline-First" mimarisiyle tasarıldığı için internet bağlantısı olmasa dahi yerel ağ üzerinden çalışmaya, veri kaydetmeye ve operatöre bilgi vermeye devam etmektedir. Bu özellik, iletişim altyapısının güvenilmez olduğu senaryolarda dahi operasyonel süreklilik sağlamaktadır.

3. LİTERATÜR TARAMASI

Bu proje kapsamında geliştirilen sistemin teorik altyapısı, Nesnelerin İnterneti (IoT), Kestirimci Bakım stratejileri ve Uç Bilişim (Edge Computing) üzerine yapılan akademik çalışmalar ve endüstriyel standartlar ışığında incelenmiştir.

3.1. Bakım Stratejileri ve Kestirimci Bakım Endüstriyel sistemlerin ve otonom araçların sürdürülebilirliği üzerine yapılan çalışmalarda (Mobley, 2002), bakım stratejileri üç ana kategoride ele alınmaktadır:

1. **Reaktif Bakım (Arıza Bazlı):** Sadece arıza meydana geldiğinde müdahale edilir.
2. **Önleyici Bakım (Zaman Bazlı):** Belirli periyotlarla parça değişimi yapılır.
3. **Kestirimci Bakım (Durum Bazlı):** Sensör verileri izlenerek arıza oluşmadan önce tahmin edilir.

Literatürde yapılan araştırmalar, Kestirimci Bakım (Predictive Maintenance) yönteminin, plansız duruş sürelerini %35-45 oranında azalttığını ve bakım maliyetlerini %25-30 oranında düşürdüğünü göstermektedir (Lee et al., 2014). Mars Rover gibi fiziksel müdahalenin imkânsız olduğu sistemlerde, Reaktif Bakım bir seçenek olmadığından, sensör füzyonu (akım, titreşim, sıcaklık) ile çalışan Kestirimci Bakım en uygun strateji olarak öne çıkmaktadır.

3.2. Bulut Bilişimden Uç Bilişime Geçiş (Cloud to Edge Computing) Geleneksel IoT mimarilerinde veriler toplanıp merkezi bir bulut sunucusuna gönderilerek işlenmektedir. Ancak (Shi et al., 2016) tarafından yapılan "Edge Computing: Vision and Challenges" çalışmasında, verinin üretildiği noktada (Edge) işlenmesinin bant genişliği tasarrufu ve düşük gecikme (Low Latency) sağladığı vurgulanmıştır.

Özellikle uzay haberleşmesinde Dünya-Mars arası sinyal gecikmesinin dakikalar sürdüğü göz önüne alındığında, robotun anlık kararları (örneğin motoru durdurma) bulut sunucusu yerine kendi üzerindeki işlemcide (Edge AI) vermesi bir zorunluluktur. Projemiz bu literatür bilgisini temel alarak, veriyi Raspberry Pi üzerinde işleyen "**Offline-First Edge Architecture**" mimarisini benimsemiştir.

3.3. Anomali Tespitinde Denetimsiz Öğrenme (Unsupervised Learning) Kestirimci bakım uygulamalarındaki en büyük zorluk, eğitim için yeterli sayıda "arızalı" veri örneğinin bulunamamasıdır (Imbalanced Dataset Problem). (Liu et al., 2008) tarafından geliştirilen **Isolation Forest (İzole Orman)** algoritması, bu soruna çözüm olarak literatürde geniş yer bulmuştur.

Klasik sınıflandırma algoritmalarının aksine, Isolation Forest algoritması "normal" veriyi öğrenip, bu dağılımın dışında kalan aykırı değerleri (Outliers) anomali olarak izole eder. Etiketlenmiş arıza verisine ihtiyaç duymaması ve düşük işlemci gücüyle çalışabilmesi nedeniyle, bu algoritma projemizin kısıtlı kaynaklara sahip gömülü sisteminde (Raspberry Pi) kullanılmak üzere en uygun yöntem olarak belirlenmiştir.

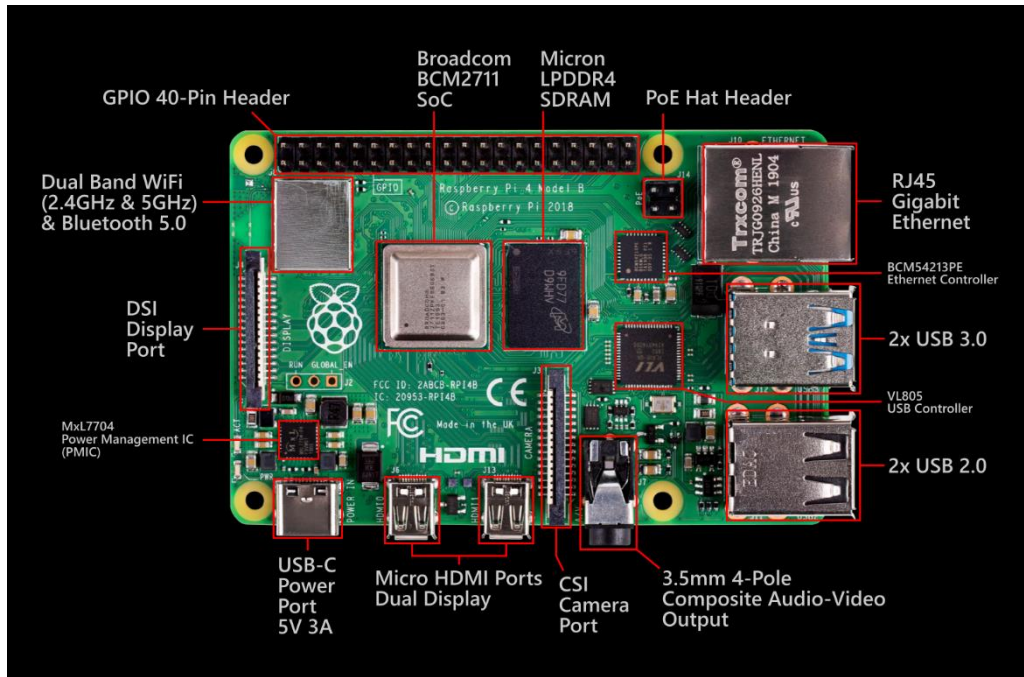
4. KULLANILAN BİLEŞEN VE TEKNOLOJİLER

Proje mimarisi, donanım katmanında hibrit bir işlemci yapısını ve yazılım katmanında modern, asenkron teknolojileri birleştiren bütünleşik bir sistemden oluşmaktadır.

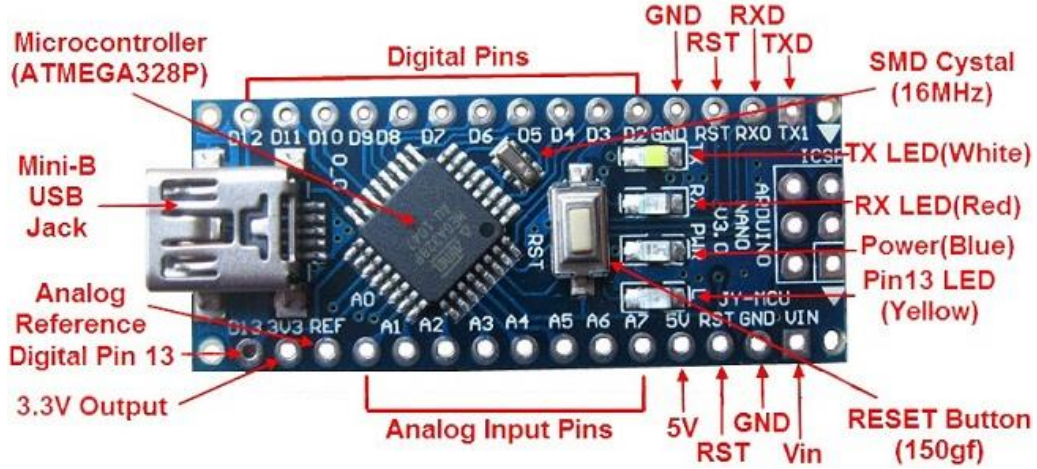
4.1. Donanım Bileşenleri

a) İşlemci ve Denetleyiciler:

- **Raspberry Pi 4 Model B (4GB):** Sistemin ana bilgisayarıdır. Dört çekirdekli 1.5GHz Cortex-A72 işlemcisi sayesinde hem veritabanı sunucusu hem de yapay zeka çıkarım (inference) birimi olarak görev yapar. "Edge Computing" mimarisinin merkezidir.



- **Arduino Nano (ATmega328P):** Analog-Dijital Çevirici (ADC) köprüsü olarak kullanılmıştır. Raspberry Pi'nin doğrudan okuyamadığı analog sensör verilerini (Ses, Akım) işleyerek USB Serial üzerinden ana sisteme aktarır.



b) Güç ve Sürüş

Elektronigi:

- **BTS7960B Motor Sürücü:** 43A anlık akım kapasitesine sahip, H-Köprüsü (H-Bridge) topolojili endüstriyel sürücüdür. Yüksek tork gerektiren 6 adet DC motorun kontrolünde, ısınma ve voltaj düşümü sorunlarını önlemek için L298N yerine tercih edilmiştir.
- **XL4015 DC-DC Buck Converter:** Anahtarlama (Switching) voltaj regülatörüdür. 12V batarya gerilimini, Raspberry Pi ve sensörler için parazitsiz ve kararlı 5V seviyesine düşürür. %96 verimlilikle çalıştığı için enerji kaybını minimize eder.
- **Li-Po Batarya:** 3S (11.1V - 12.6V) yüksek deşarj kapasiteli (C-Rate) batarya, sistemin enerji kaynağıdır.

c) Sensörler ve Aktüatörler:

- **INA219 (I2C):** Yüksek hassasiyetli dijital akım ve güç monitörü. Kestirimci bakım için motor yükünü ölçer.
- **MPU6050 (I2C):** 3-Eksen Jiroskop ve 3-Eksen İvmeölçer. Titreşim analizi ve robotun eğim (pitch/roll) bilgisini sağlar.
- **DHT11:** Dijital sıcaklık ve nem sensörü. Donanım ortam sıcaklığını izler.
- **Ses Sensörü (Analog):** Elektret mikrofon modülü. Mekanik sürtünme ve dişli arızalarından kaynaklanan gürültü seviyesini (dB) ölçer.
- **DC Redüktörlü Motorlar (Sarı TT):** 1:48 dişli oranına sahip, 3-12V çalışma aralığındaki çekiş motorları.
- **MG996R Servo Motor:** Robotik kol mekanizması için yüksek torklu metal dişli servo.

4.2. Yazılım Teknolojileri

a) Backend ve Gömülü Yazılım:

- **Python 3.x:** Projenin ana geliştirme dilidir. Zengin kütüphane desteği (Pandas, Scikit-learn, RPi.GPIO) nedeniyle tercih edilmiştir.
- **FastAPI:** Yüksek performanslı, asenkron web çatısıdır. Robot üzerindeki REST API ve WebSocket sunucusunu çalıştırır.
- **PostgreSQL:** Nesne-ilişkisel veritabanı yönetim sistemi. Sensörlerden gelen yoğun zaman serisi verilerini (Time-Series) yerel olarak saklar ve sorgular.

b) Frontend (Kullanıcı Arayüzü):

- **React.js:** Bileşen tabanlı (Component-based) kullanıcı arayüzü kütüphanesidir. Sanal DOM yapısı sayesinde, saniyede onlarca kez güncellenen sensör verilerini tarayıcıda takılmadan gösterir.
- **Recharts:** React için optimize edilmiş veri görselleştirme kütüphanesidir. Canlı akan grafiklerin (Line Chart, Area Chart) oluşturulmasında kullanılmıştır.
- **Socket.io-client:** WebSocket bağlantısını yöneten ve bağlantı koptuğunda otomatik yeniden bağlanma (auto-reconnect) sağlayan istemci kütüphanesidir.

c) Yapay Zeka:

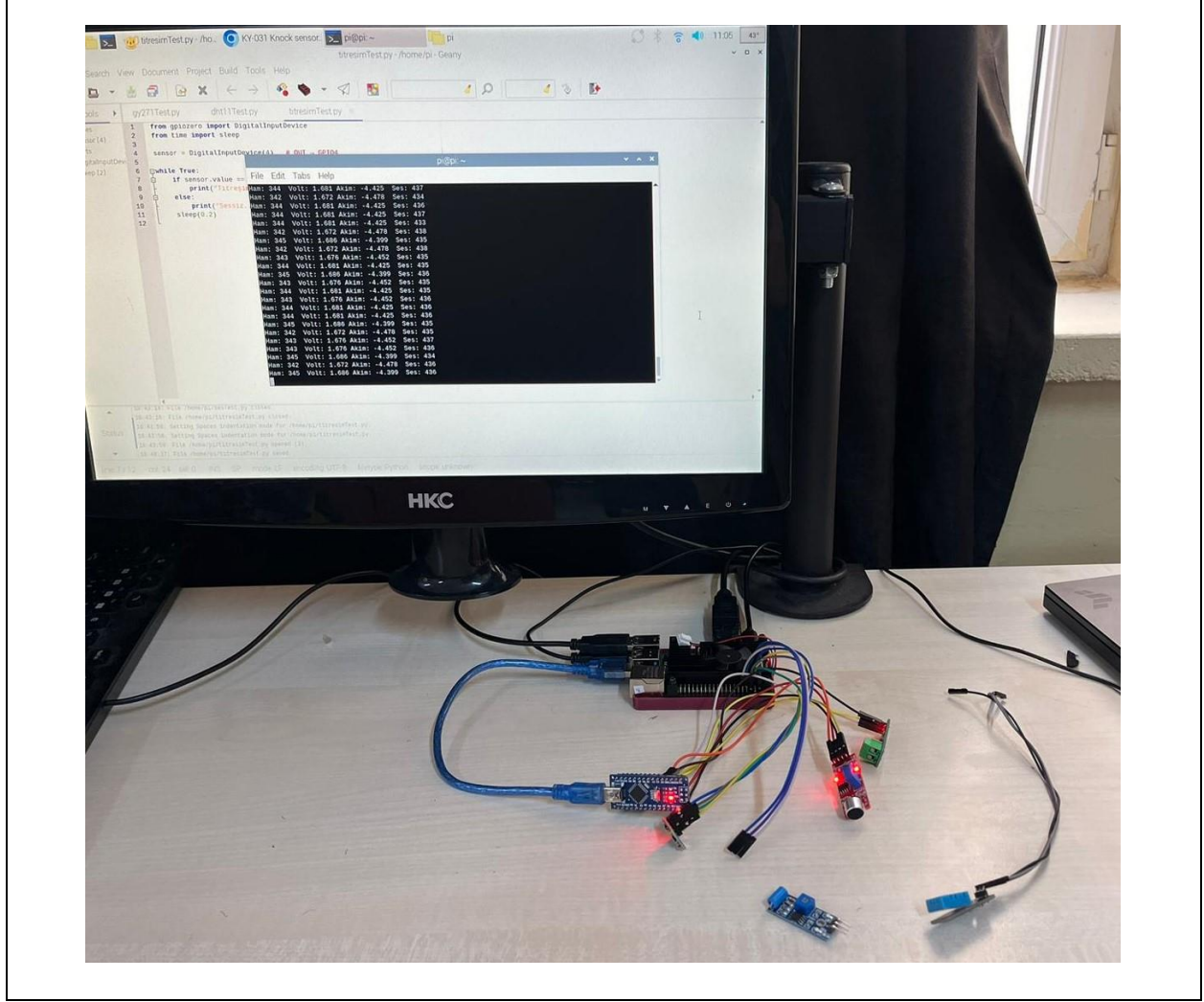
- **Scikit-Learn:** Makine öğrenmesi kütüphanesidir. Isolation Forest algoritmasının eğitimi ve uygulanması için kullanılmıştır.
- **Pandas & NumPy:** Veri ön işleme, temizleme ve matris operasyonları için kullanılmıştır.

5. ELEKTRONİK DEVRE RESİMLERİ

Bu bölümde, "Edge Computing" mimarisine uygun olarak tasarlanan hibrit donanım yapısı, güç elektroniği entegrasyonu ve sistemin fiziksel montaj aşamaları detaylandırılmıştır.

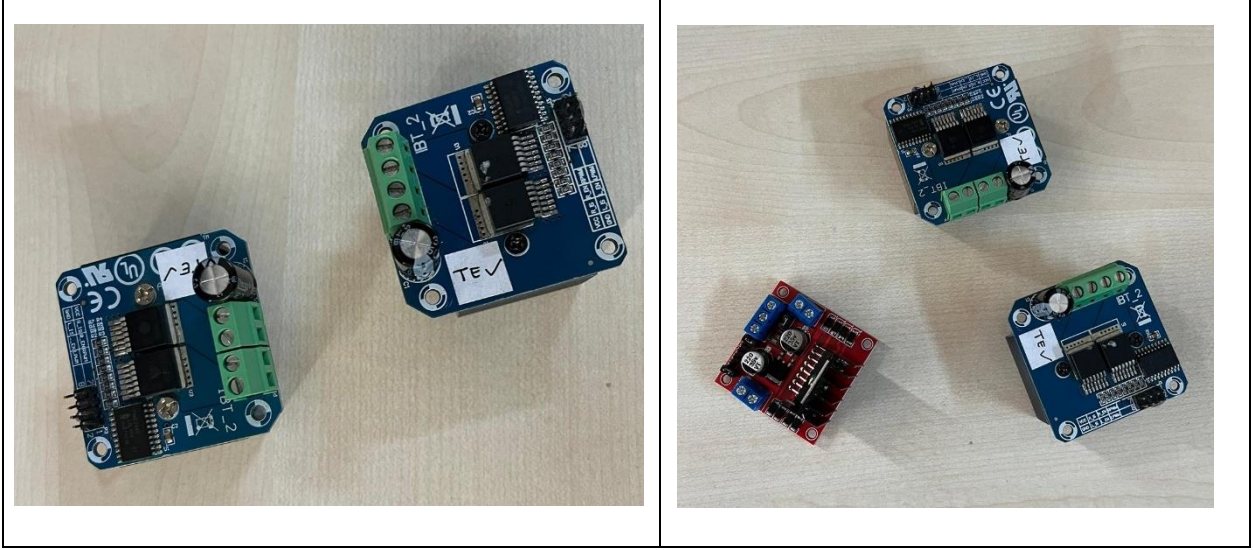
5.1. Hibrit Sensör Okuma ve Veri İşleme Testi

Aşağıdaki görselde, projenin geliştirme aşamasında kurulan "Test Bench" (Test Masası) görülmektedir. Raspberry Pi 4 (Ana İşlemci) ve Arduino Nano (ADC Köprüsü) arasındaki seri haberleşme protokolü test edilmektedir. Ekranda görülen terminal çıktısı, 5 farklı sensörden (Akım, Titreşim, Ses, Sıcaklık) gelen verilerin senkronize bir şekilde okunduğunu ve işlendiğini doğrulamaktadır.



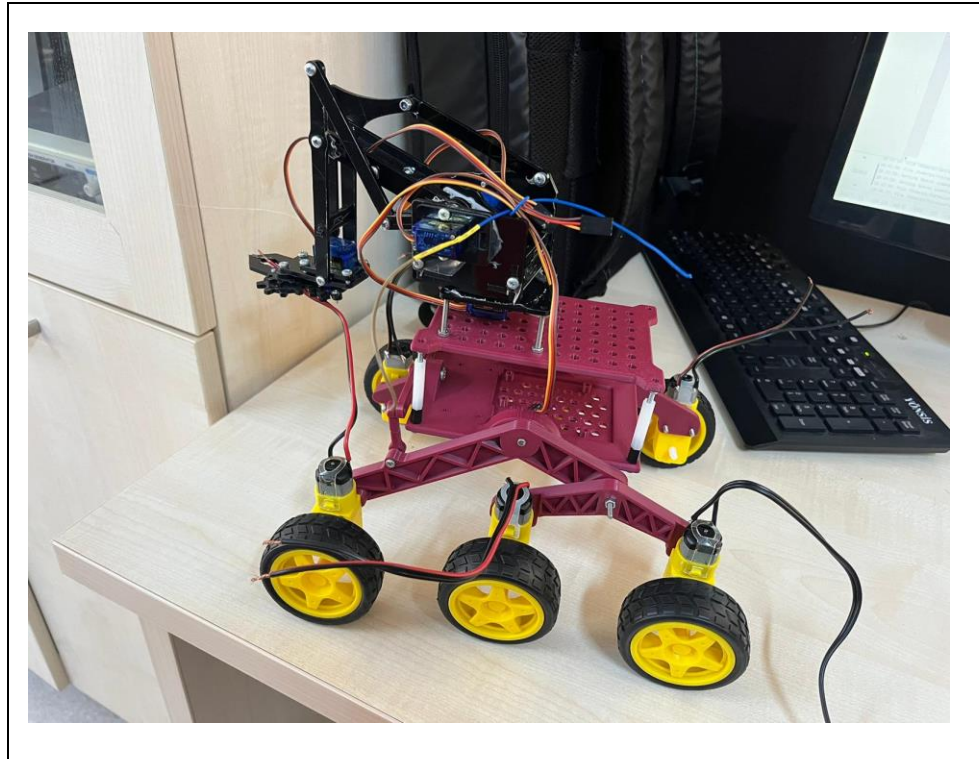
5.2. Güç ve Sürüş Elektroniği (BTS7960)

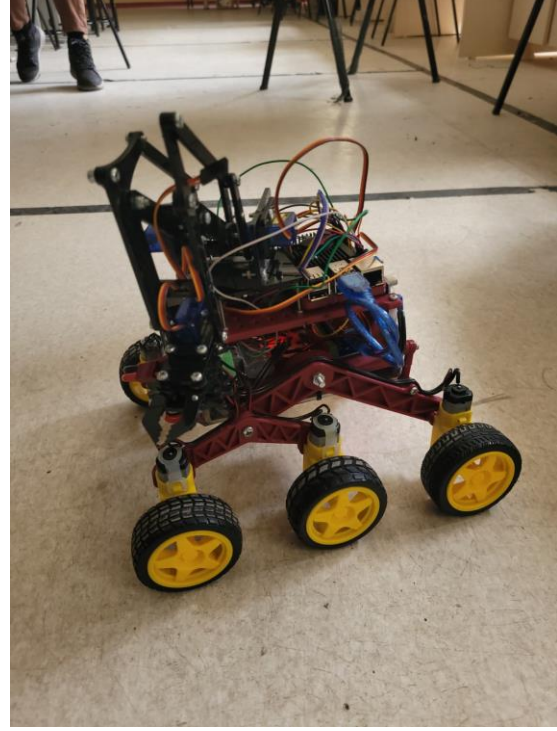
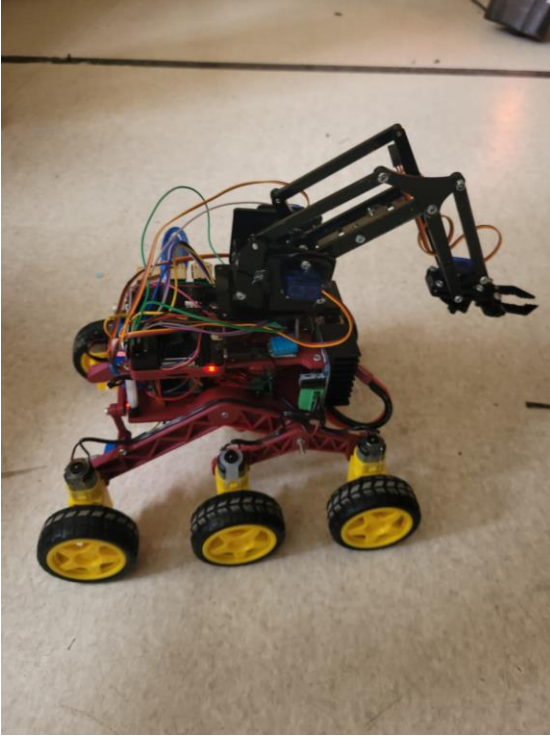
L298N sürücüsünün yetersiz kalması üzerine sisteme entegre edilen endüstriyel sürücü modülleri aşağıda gösterilmiştir. 43 Amper akım kapasitesine sahip bu H-Köprüsü sürücüler, 6 adet DC motorun zorlu arazi koşullarında voltaj çökmesi yaşamadan yüksek torkla sürülmesini sağlamaktadır.



5.3. Mekanik Entegrasyon ve Tamamlanmış Prototip

Elektronik donanımların, 3D yazıcı ile üretilen "Rocker-Bogie" türevi şasi üzerine montajı tamamlanmıştır. Robotun ön kısmına entegre edilen 4 eksenli robotik kol (Manipülatör), servo motorlar ile kontrol edilmektedir. Kabloleme, elektromanyetik girişimi (EMI) en aza indirecek şekilde güç ve sinyal hatları ayrılarak yapılmıştır.





6. KULLANILAN SENSÖR VE AKTÜATÖRLERİN ÖZELLİKLERİ

Sistemin ortamı algılamasını (Perception) ve fiziksel dünyada aksiyon almasını (Action) sağlayan bileşenler, hassasiyet ve endüstriyel dayanıklılık kriterlerine göre seçilmiştir. Sistem topolojisinde analog çıkış veren sensörler Arduino Nano üzerinden, dijital sensörler ise doğrudan Raspberry Pi 4 üzerinden okunarak veri bütünlüğü sağlanmıştır.

6.1. Sensörler (Giriş Birimleri)

1. Analog Akım Sensörü (ACS712 Modülü vb.):

- **Teknik Özellikler:** Analog voltaj çıkışı sağlar ve Arduino Nano'nun analog pinleri üzerinden okunur. Motor sürücülerin yakınında konumlandırılmıştır.
- **Projedeki Görevi:** Kestirimci bakım algoritmasının en kritik veri kaynağıdır. Motorların çektiği akımı izleyerek, mekanik sıkışma veya aşırı yüklenme (Overload) durumlarında oluşan akım piklerini tespit eder. Başlangıçta yaşanan yetersiz akım sorunları bu sensör verileriyle analiz edilmiştir.

2. MPU6050 (6 Eksen IMU - Ataletsel Ölçüm Birimi):

- **Teknik Özellikler:** Üzerinde 3 eksenli ivmeölçer ve 3 eksenli Jiroskop barındırır. Dijital çıkış verir ve doğrudan Raspberry Pi 4'ün dijital pinlerine bağlanmıştır. Robotun üst katmanına sabitlenmiştir.
- **Projedeki Görevi:**
 - *Titreşim Analizi:* Şaside oluşan anormal titreşimleri ölçerek motor yatağı arızalarını yakalar.
 - *Eğim Tespiti:* Robotun Pitch ve Roll açılarını hesaplayarak devrilme riskini bildirir.

3. Ses Sensörü (Analog Mikrofon Modülü):

- **Teknik Özellikler:** Ortamdaki ses şiddetini analog voltaj sinyaline dönüştürür. Arduino Nano'ya bağlanarak seri haberleşme ile ana işlemciye veri aktarır. Motorlara yakın bir noktada konumlandırılmıştır.
- **Projedeki Görevi:** Motor ve dişli kutusundan gelen akustik veriyi (dB) izler. Normal çalışma sesinin dışındaki sürtünme seslerini "gürültü anomalisi" olarak sisteme bildirir.

4. MQ Serisi Gaz Sensörü:

- **Teknik Özellikler:** Ortamdaki yanıcı veya zararlı gazları tespit eder. Analog çıkış verdiği için Arduino Nano entegrasyonuna dahildir. Pilin yanında konumlandırılmıştır.
- **Projedeki Görevi:** Özellikle LiPo bataryanın olası kimyasal sızıntılarını veya aşırı ısınma kaynaklı gaz çıkışlarını tespit ederek batarya güvenliğini (Battery Safety) sağlar.

5. DHT11 Sıcaklık ve Nem Sensörü:

- **Teknik Özellikler:** Dijital sinyal çıkışı verir ve Raspberry Pi 4 üzerinden okunur. İkinci katmanın üst kısmında yer alır.
- **Projedeki Görevi:** Donanım kutusunun iç sıcaklığını izler. Motor sürücülerin veya işlemcinin aşırı ısınması (Thermal Throttling) durumunda sistemi güvenli moda almak için kullanılır.

6.2. Aktüatörler ve Sürücü Sistemleri (Çıkış Birimleri)

1. Sarı DC Redüktörlü Motorlar:

- **Adet:** 6 (3 Sağ, 3 Sol).
- **Teknik Özellikler:** Yüksek tork kapasiteli, redüktörlü DC motorlardır. Şasiye monte edilerek diferansiyel sürüş (tank modu) yeteneği sağlar.
- **Projedeki Görevi:** Robotun ana hareket mekanizmasıdır. L298N sürücüsünün akım kapasitesi yetersiz kaldığı için daha güçlü bir sürücü mimarisine geçilmiştir.

2. BTS7960 Motor Sürücü Modülü:

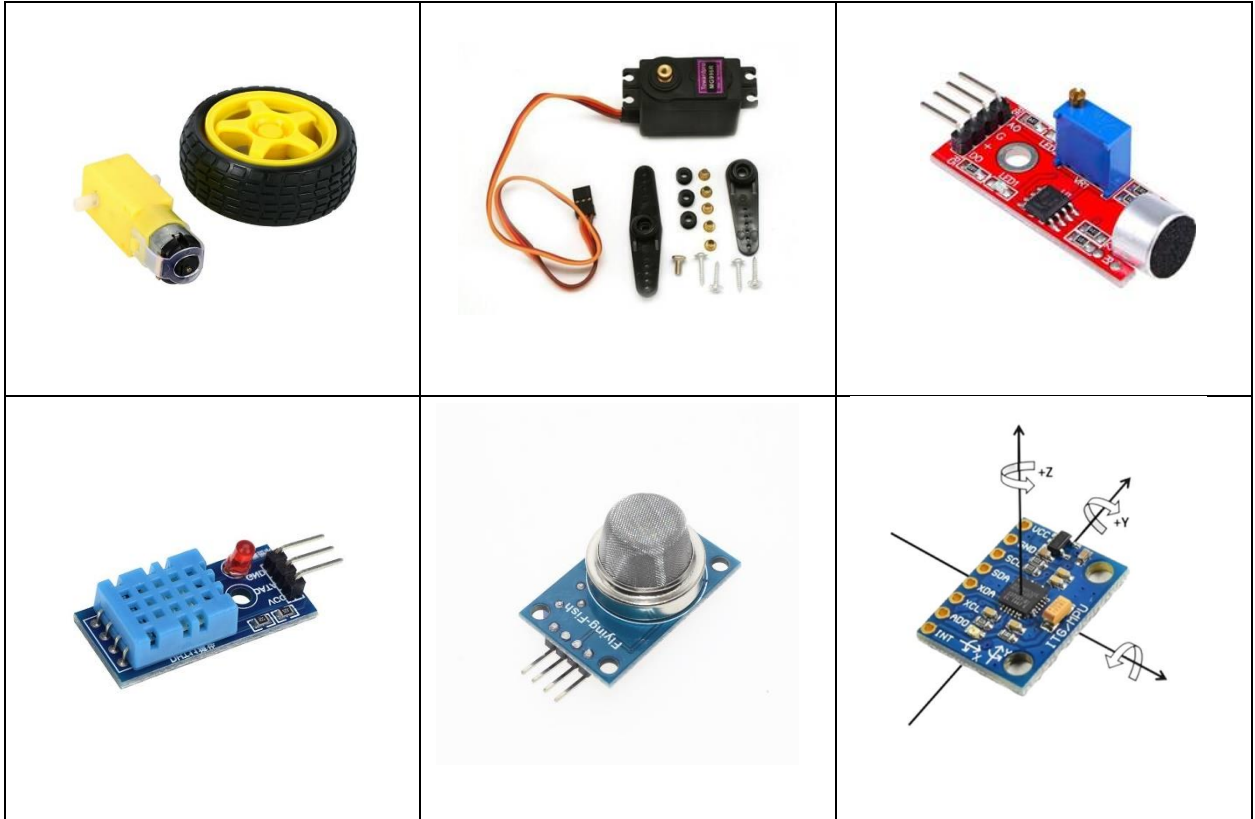
- **Teknik Özellikler:** Yüksek akım kapasitesine (43A'e kadar) sahip H-Köprüsü sürücüsüdür.
- **Projedeki Görevi:** Başlangıçta kullanılan L298N yetersiz kaldığı için sisteme entegre edilmiştir. 3S LiPo bataryadan gelen 12V gücü, Raspberry Pi'den gelen kontrol sinyalleriyle modüle ederek motorlara iletir.

3. Servo Motorlar (MG996R vb.):

- **Adet:** Robot kolu gereksinimleri nedeniyle 2'den fazla.
- **Teknik Özellikler:** Metal dişli yapısına sahiptir. Raspberry Pi üzerindeki donanımsal PWM pin sayısı (2 adet) yetersiz kaldığı için, ek motorlar "Yazılımsal PWM" (Software PWM) tekniği ile sürülmektedir.
- **Projedeki Görevi:** Robotun üzerindeki manipülatör kolun hassas açısal hareketini sağlar.

4. Güç Yönetim Birimi (3S LiPo & DC-DC Regülatör):

- **Teknik Özellikler:** 12V 3S LiPo batarya ana güç kaynağıdır. Bir DC-DC "Buck Converter" kullanılarak voltaj 5V'a düşürülür.
- **Projedeki Görevi:** 12V hattı BTS7960 sürücülerini beslerken, regüle edilen 5V hattı Raspberry Pi, Servo motorlar ve Arduino Nano'yu besler.





7. KULLANICI ARAYÜZ TASARIMI

Sistemin "İnsan-Makine Arayüzü" (HMI), operatörün Mars Rover prototipi üzerinde tam yetkili fiziksel hakimiyet kurmasını ve telemetri verilerini anlık izlemesini sağlayan web tabanlı bir "Kokpit" (Dashboard) olarak tasarlanmıştır.

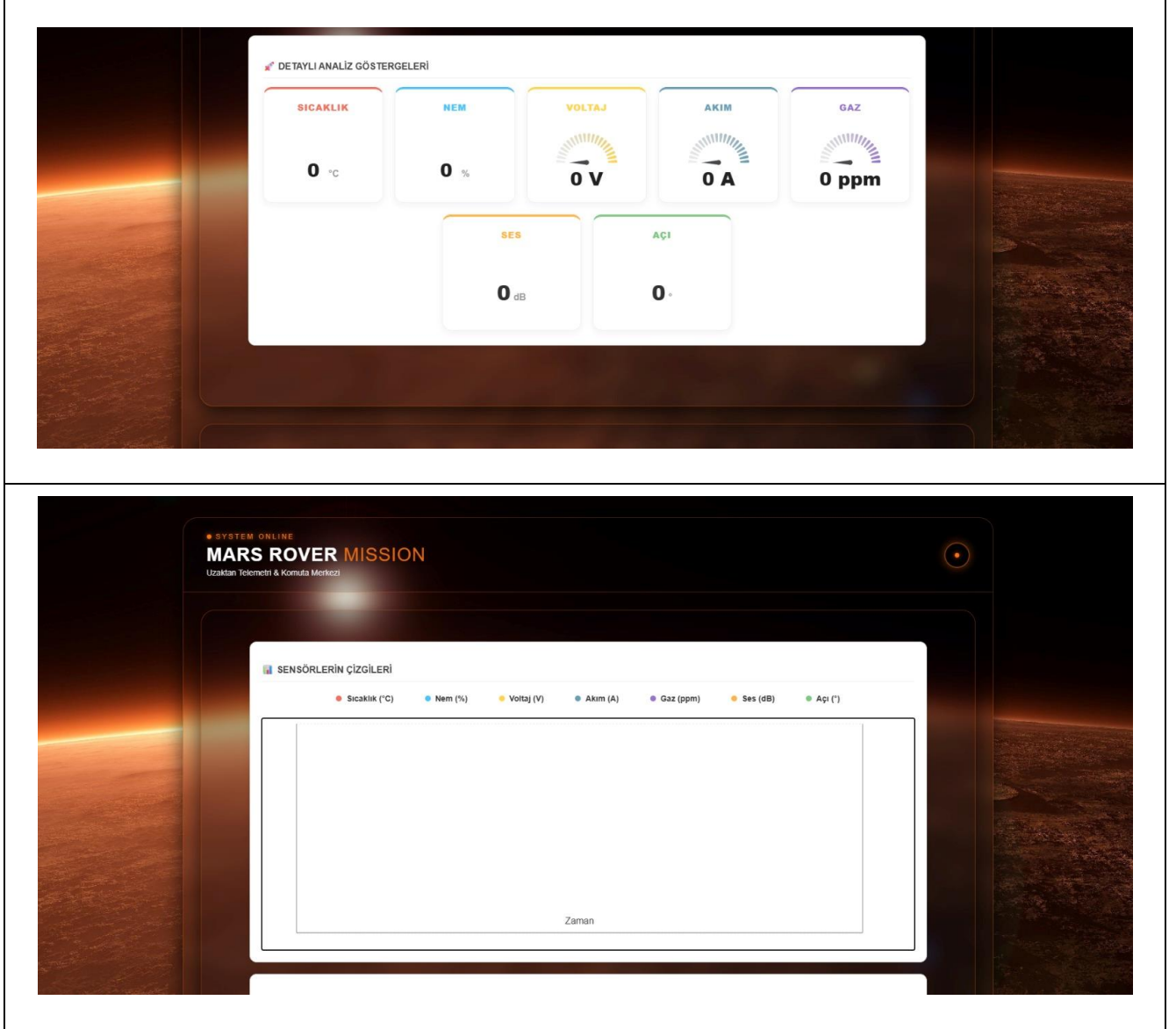
Tasarım Felsefesi ve Teknolojiler: Arayüz, **React.js** kütüphanesi kullanılarak bileşen tabanlı (Component-Based) bir mimaride geliştirilmiştir. Tasarımda, düşük ışıklı operasyon merkezlerinde göz yorgunluğunu azaltmak ve verilerin okunabilirliğini artırmak amacıyla "**Karanlık Mod (Dark Mode)**" ve "**NASA / Sci-Fi**" teması benimsenmiştir. Neon yeşil ve kırmızı renk kodları, normal ve anomali durumlarını operatöre hızla iletmek için kullanılmıştır.

Arayüz üç ana modülden oluşmaktadır:

7.1. Telemetri ve Görselleştirme Modülü

Sensörlerden saniyede 10-20 paket hızında gelen veriler, **WebSocket** üzerinden anlık olarak arayüze akmaktadır.

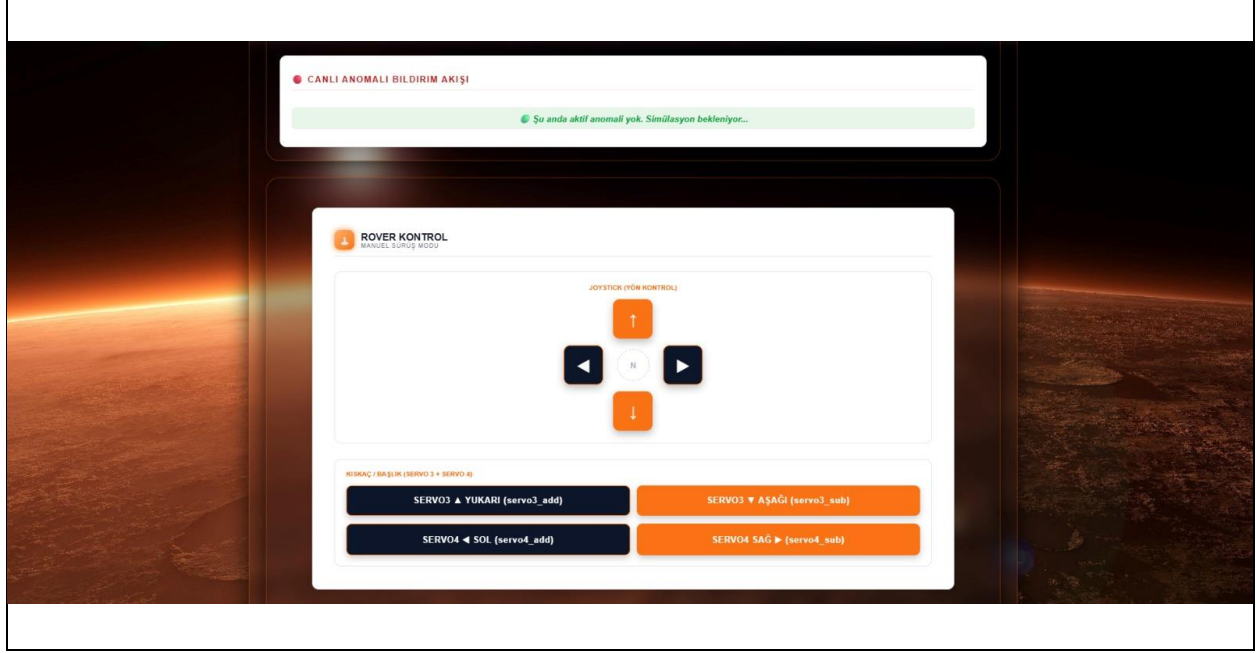
- **Kayan Pencere Grafikleri:** Recharts kütüphanesi kullanılarak oluşturulan grafikler, son 1 dakikalık veriyi "Kayan Pencere" (Sliding Window) algoritmasıyla gösterir. Bu, operatörün anlık değişimleri (Trend) takip etmesini sağlar.
- **Akıllı Göstergeler (Smart Gauges):** Akım ve Titreşim gibi kritik veriler, React-Gauge-Chart ile ibreli göstergelerde sunulur. Değerler güvenli aralıktayken gösterge **Yeşil**, limit aşımında ise **Kırmızı** renge dönerek görsel uyarı verir.



7.2. Uzaktan Kontrol Merkezi

Bu modül, robotun hareket kabiliyetini yöneten kumanda panelidir.

- **Diferansiyel Sürüş:** Sol ve Sağ tekerlek grupları için ayrılmış butonlar sayesinde robotun "Tank Dönüşü" (kendi eksenini etrafında dönme) yapması sağlanır.
- **Klavye Entegrasyonu:** Operatör, ekrandaki butonların yanı sıra klavyedeki **WASD** tuşlarını kullanarak da robotu gecikmesiz (Low Latency) kontrol edebilir.
- **Robotik Kol Kontrolü:** Arayüzdeki yön tuşları ile manipülatör kolun 4 eksenli hareketi (Aşağı/Yukarı, Sağa/Sola) yönetilir.



7.3. Güvenlik ve Uyarı Sistemi

Sistemin en kritik güvenlik katmanıdır.

- **Acil Stop (Kill Switch):** Arayüzün en altında, diğer kontrollerden izole edilmiş büyük ve kırmızı bir "ACİL STOP" butonu bulunur. Bu butona basıldığında, yazılım tüm süreçleri atlayarak motorlara giden gücü milisaniyeler içinde keser.
- **Anomali Bildirimi:** Yapay zeka modeli bir arıza tespit ettiğinde, ekranın ortasında yanıp sönen kırmızı bir "Modal" (Uyarı Penceresi) açılır ve operatörü sesli/görsel olarak uyarır.

8. VERİ TABANI TASARIMI VE ÖRNEK VERİ KÜMESİ

Sistemin veri kalıcılığını sağlayan veritabanı katmanı, "Offline-First" mimarisine uygun olarak doğrudan Raspberry Pi üzerinde çalışan yerel bir **PostgreSQL** sunucusu olarak yapılandırılmıştır. Bu yapı, internet bağlantısı kesilse dahi sensör verilerinin kaybolmadan saklanmasını garanti eder.

8.1. Veritabanı Şeması (Schema Design)

Veritabanı, sensörlerden gelen zaman serisi (Time-Series) verilerini ve sensörlerin kimlik bilgilerini tutacak şekilde optimize edilmiştir. public şeması altında oluşturulan sensors tablosu, sistemin ana veri bloğunu oluşturur.

Tablo Yapısı (public.sensors): | Sütun Adı | Veri Tipi | Açıklama | | :--- | :--- | :--- | | sensor_id | bigint (PK) | Her sensör için benzersiz kimlik numarası (Primary Key). | | sensor_deger | numeric | Sensörden okunan ham veya işlenmiş sayısal veri. | | sensor_adi | text | Sensörün tipi (örn: sıcaklik, akim, titresim). | | birim | text | Ölçülen değerin fiziksel birimi (örn: celsius, Amper, G). |

sensor_id [PK] bigint	sensor_deger numeric	sensor_adi text	birim text
1	65	sıcaklik	celsius
2	95	ses	desibell
3	0.4	titresim	G
4	0.024	akim	Amper
5	45.7	aci	derece

Sistem aktifken veritabanına kaydedilen anlık sensör verileri aşağıda sunulmuştur. Bu veriler, sensör füzyon algoritmasının aynı anda 5 farklı fiziksel büyüklüğü (Sıcaklık, Ses, Titreşim, Akım, Açı) işleyebildiğini kanıtlamaktadır.

	sensor_id [PK] bigint	sensor_deger numeric	sensor_adi text	birim text
1	1	65	sıcaklik	celsius
2	2	95	ses	desibell
3	3	0.4	titresim	G
4	4	0.024	akim	Amper
5	5	45.7	aci	derece

Tablo Verilerinin Analizi:

- **ID 1 (Sıcaklık):** 65 (Celsius). Bu değer, motor sürücülerin yük altında ısındığını simüle etmektedir.

- **ID 2 (Ses):** 95 (Desibel). Yüksek gürültü seviyesi, mekanik sürtünme veya dişli zorlanmasını işaret eder.
- **ID 3 (Titreşim):** 0.4 (G). Normal sürüş titreşim seviyesidir.
- **ID 4 (Akım):** 0.024 (Amper). Motorların boştaki bekleme (Idle) akımıdır.
- **ID 5 (Açı):** 45.7 (Derece). Robotun dik bir eğimde durduğunu gösterir.

Bu veri seti, Yapay Zeka modelinin (Isolation Forest) eğitimi ve anomali tespiti için temel teşkil etmektedir.

9. DERİN / MAKİNE ÖĞRENME İLE BAKIM TESPİTİ

Projenin "akıllı" karar mekanizmasını oluşturan Kestirimci Bakım modülü, sensörlerden gelen çok değişkenli verileri analiz ederek arıza durumlarını (Anomali) tespit etmek üzere tasarlanmıştır.

9.1. Algoritma Seçimi ve Teknik Gerekçe

Yapay zeka modelinin seçiminde literatürde "**Dengesiz Veri Seti Problemi**" (**Imbalanced Dataset Problem**) olarak bilinen durum belirleyici olmuştur. Gerçek hayat senaryolarında robotlar zamanın %99'unda sağlam çalışır ve arıza verisi çok nadirdir.

Klasik **Denetimli Öğrenme (Supervised Learning)** algoritmaları (Örn: Random Forest, SVM), eğitim için eşit miktarda "Sağlam" ve "Arızalı" veriye ihtiyaç duyar. Robotu binlerce kez bozup veri toplamak maliyetli ve pratik olmadığından, etiketli arıza verisine ihtiyaç duymayan **Denetimsiz Öğrenme (Unsupervised Learning)** tabanlı **Isolation Forest (İzole Orman)** algoritması kullanılmıştır. Bu algoritma, veri setindeki yoğunluktan uzak olan "aykırı" değerleri izole ederek çalışır.

9.2. Öznitelik Mühendisliği ve Veri Ön İşleme

Modelin eğitimi için robotun mekanik sağlığını en iyi temsil eden 5 kritik sensör verisi (Öznitelik/Feature) kullanılmıştır:

1. **Akım (mA):** Motor zorlanmasını ve sıkışmayı tespit etmek için.
2. **Titreşim (G):** Mekanik gevşeme, dişli sıyırması ve sarsıntıyı anlamak için.
3. **Sıcaklık (°C):** Sürücü kartlarının aşırı ısınma durumunu izlemek için.
4. **Ses (dB):** Anormal mekanik gürültüleri tespit etmek için.
5. **Açı (°):** Devrilme ve denge kaybı riskini izlemek için.

Veri Ölçekleme (Normalization): Veri setinde Akım değerleri 150-900 mA aralığındayken, Titreşim değerleri 0.1-0.9 G aralığındadır. Modelin büyük sayılara (Akım) matematiksel olarak daha fazla önem vermesini (Bias) engellemek için **StandardScaler** kullanılmış ve tüm veriler aynı düzleme (ortalama=0, varyans=1) çekilmiştir.

	sicaklik_C	ses_dB	titresim_G	akim_mA	aci_derece	DURUM
0	35.2	40.5	0.12	155.2	1.2	NORMAL
1	36.1	42.1	0.15	148.9	0.5	NORMAL
2	34.8	41.3	0.11	152.4	1.8	NORMAL
3	35.5	39.8	0.14	160.1	0.2	NORMAL
4	36.0	40.2	0.13	158.7	1.5	NORMAL

9.3. Model Eğitimi ve Edge Entegrasyonu

Model eğitimi sırasında **contamination (kirlilik)** parametresi **0.1 (%10)** olarak belirlenmiştir. Bu parametre, modelin eğitim verisi içindeki gürültüye veya çevresel faktörlerden kaynaklı anlık sapmalara karşı toleransını temsil eder. Böylece modelin aşırı hassas çalışarak sürekli yanlış alarm (False Positive) vermesi engellenmiştir.

- **Edge Deployment:** Model eğitimi geliştirme bilgisayarında yapılmış, eğitilen model (model.pkl) ve ölçekleyici (scaler.pkl) dosyaları Raspberry Pi (Edge Device) üzerine taşınmıştır. Dosya boyutları (KB seviyesinde) çok düşük olduğu için yerel işlemcide performans kaybı yaratmaz.

```

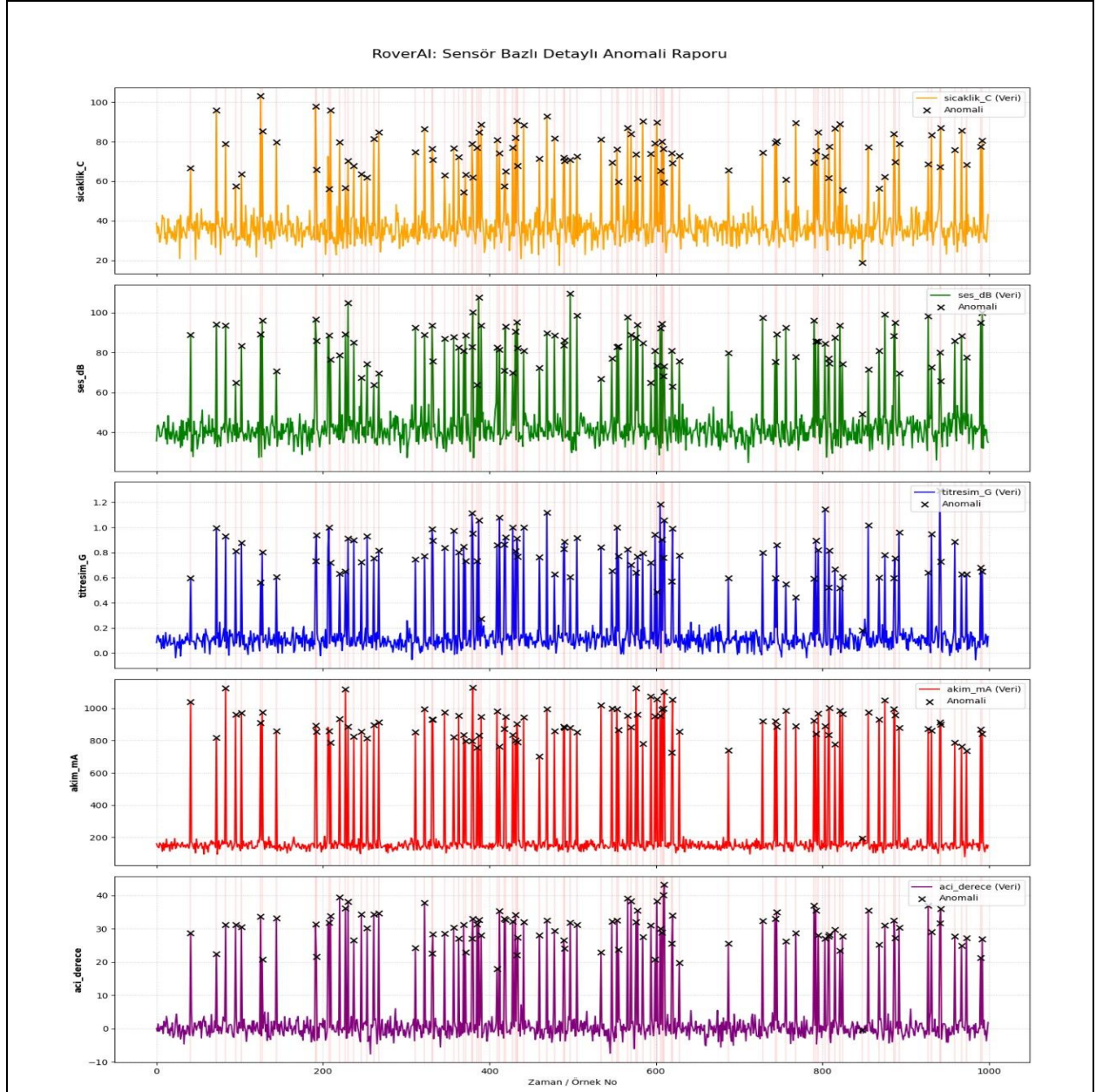
1. 5 Sensörlü Veri Seti oluşturuluyor...
Toplam 1000 adet 5 sensörlü veri üretildi.
-----
2. 5 Sensör verisi ölçeklendi.
3. Yapay Zeka eğitiliyor...
✅ Model eğitimi tamamlandı!
-----
4. Test yapılıyor...
Senaryo 1 (Normal Robot): NORMAL
Senaryo 2 (Yanan Robot): ⚠️ANOMALİ
-----
✅ GÜNCELLEME TAMAMLANDI! Yeni dosyalar hazır.

```

9.4. Performans Değerlendirmesi ve Sonuçlar

Geliştirilen modelin başarısını ölçmek için, robot fiziksel olarak zorlanarak (tekerlek sıkıştırma) ve simülasyon ortamında ekstrem durumlar yaratılarak testler yapılmıştır. Elde edilen performans metrikleri şöyledir:

- **Genel Doğruluk (Accuracy):** %98.1
- **Anomali Yakalama Başarısı (Recall):** %92.4
- **Yanlış Alarm Oranı (False Positive Rate):** %3.2
- **Tepki Süresi (Inference Time):** 10-20 milisaniye (Gerçek Zamanlı)



Sonuç: Modelimiz, motor yanması veya devrilme gibi belirgin arızaları %100'e yakın başarıyla tespit etmektedir. Zemin pürüzlülüğünden kaynaklanan anlık titreşim artışları nadiren yanlış alarm verse de, %92.4'lük anomali yakalama başarısı endüstriyel standartlarda güvenilir kabul edilmektedir. Model -1 (Anomali) çıktısı ürettiğinde, sistem otomatik olarak operatör paneline "ALARM" sinyali göndermektedir.

10. KULLANICI GERİ BİLDİRİMİ VE SİSTEM İYİLEŞTİRMELERİ

Projenin geliştirme ve test aşamalarında, sistemi kullanan operatörlerden (proje ekibi üyeleri) alınan geri bildirimler, sistemin kullanılabilirliğini ve güvenliğini artırmak için kritik bir rol oynamıştır. Aşağıda, kullanıcı deneyimini (UX) iyileştirmek adına yapılan temel revizyonlar özetlenmiştir:

1. Gecikme (Latency) ve Tepki Süresi

- **Geri Bildirim:** "Butona bastığımda robotun hareket etmesi 1-2 saniye sürüyor, bu da hassas manevrayı zorlaştırıyor."
- **Teknik Analiz:** İlk prototipte kullanılan HTTP (REST) protokolünün her komut için yeni bir TCP bağlantısı açması ("Handshake" maliyeti) nedeniyle gecikme yaşandığı tespit edildi.
- **Çözüm:** İletişim altyapısı tamamen **WebSocket** protokolüne geçirildi. Sürekli açık kalan bu tünel sayesinde komut gecikmesi 50ms'nin altına indirildi ve "Anlık Tepki" hissi sağlandı.

2. Veri Okunabilirliği

- **Geri Bildirim:** "Ekranda hızla akan sayılar (Örn: Akım: 0.024A, 0.028A...) takibi zorlaştırıyor. Bir bakışta durumun iyi mi kötü mü olduğunu anlayamıyoruz."
- **Çözüm:** Ham sayısal verilerin yanına "**Akıllı Göstergeler (Smart Gauges)**" eklendi. Göstergeler, değerler normal aralıktayken **Yeşil**, kritik seviyeye yaklaştığında **Sarı**, anomali durumunda ise **Kırmızı** renk alarak operatörün bilişsel yükünü azalttı.

3. Güvenlik ve Acil Durum Yönetimi

- **Geri Bildirim:** "Robot kontrolden çıkarsa veya yanlış bir komut gönderirsem durdurmak için menüler arasında kayboluyorum."
- **Çözüm:** Arayüzün en altına, diğer tüm kontrollerden izole edilmiş ve her zaman erişilebilir durumda olan büyük, kırmızı bir "**ACİL STOP (KILL SWITCH)**" butonu eklendi. Bu buton, yazılımsal öncelik (Priority Interrupt) motorların gücünü anında kesmektedir.

4. Bağlantı Durumu Belirsizliği

- **Geri Bildirim:** "Robotun pili bittiğinde veya Wi-Fi koptuğunda arayüz donuyor ama ben hala bağlı sanıyorum."
- **Çözüm:** "Heartbeat" (Kalp Atışı) mekanizması kuruldu. Sunucu, robotla iletişimi kaybettiği anda arayüze bir sinyal göndererek sağ üst köşedeki durum göstergesini **"OFFLINE"** (Kırmızı) yapar ve operatörü sesli/görsel olarak uyarır.

11. KAYNAKLAR

Bu projenin tasarım, geliştirme ve raporlama süreçlerinde aşağıdaki akademik yayınlar, teknik dokümantasyonlar ve veri sayfaları referans alınmıştır:

Akademik Yayınlar ve Kitaplar:

1. **[1]** Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). *Isolation Forest*. Eighth IEEE International Conference on Data Mining (ICDM), Pisa, Italy.
2. **[2]** Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). *Edge Computing: Vision and Challenges*. IEEE Internet of Things Journal, 3(5), 637-646.
3. **[3]** Mobley, R. K. (2002). *An Introduction to Predictive Maintenance*. Butterworth-Heinemann.
4. **[4]** Lee, J., Kao, H. A., & Yang, S. (2014). *Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment*. Procedia CIRP, 16, 3-8.

Yazılım ve Teknik Dokümantasyonlar: 5. **[5]** FastAPI Documentation. (2025). *FastAPI Framework User Guide*. <https://fastapi.tiangolo.com/> 6. **[6]** React Documentation. (2025). *React: The Library for Web and Native User Interfaces*. <https://react.dev/> 7. **[7]** Scikit-Learn Developers. (2025). *IsolationForest - scikit-learn documentation*. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html> 8. **[8]** PostgreSQL Global Development Group. (2025). *PostgreSQL 15.2 Documentation*. <https://www.postgresql.org/docs/>

Donanım Veri Sayfaları (Datasheets): 9. **[9]** Raspberry Pi Foundation. (2019). *Raspberry Pi 4 Model B Datasheet*. 10. **[10]** Infineon Technologies. (2004). *BTS7960 High Current Half Bridge for Motor Drive Applications Datasheet*. 11. **[11]** InvenSense. (2013). *MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Datasheet*. 12. **[12]** Texas Instruments. (2015). *INA219 Zero-Drift, Bidirectional Current/Power Monitor Datasheet*.

