

Kestirimci Bakım İçin Ara Katman Mimari Uygulaması

Osman Uslu

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı

Haziran 2023



Middleware Architecture Application for Predictive Maintenance

Osman Uslu

**MASTER OF SCIENCE THESIS**

Department of Computer Engineering

June 2023

Kestirimci Bakım İin Ara Katman Mimari Uygulaması

Osman USLU

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Eğitim ve Öğretim Yönetmeliğı Uyarınca  
Bilgisayar Mühendisliğı Anabilim Dalı  
Bilgisayar Donanımı Bilim Dalında  
YÜKSEK LİSANS TEZİ  
olarak hazırlanmıştır.

Danışman: Dr. Öğr. Üyesi Eyüp Çınar

Haziran 2023

## ONAY

Bilgisayar Mühendisliđi Anabilim Dalı Donanım Bilim Dalı Yüksek Lisans öğrencisi **Osman USLU**'nun YÜKSEK LİSANS tezi olarak hazırladığı “Kestirimci Bakım İçin Ara Katman Mimari Uygulaması” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Eyüp Çınar

İkinci Danışman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye: Dr. Öğr. Üyesi Eyüp Çınar

Üye: Prof. Dr. Ahmet Yazıcı

Üye: Doç Dr. Cemal Koçak

Üye:

Üye:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN

Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Dr. Öğr. Üyesi Eyüp Çınar danışmanlığında hazırlamış olduğum “Kestirimci Bakım İçin Ara Katman Mimari Uygulaması” başlıklı Yüksek Lisans tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallarına uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 19/06/2023

Osman USLU

İmza

## ÖZET

Teknolojinin gelişimine paralel olarak dijitalleşen endüstriyel sistemlerde, Endüstriyel Nesnelerin İnterneti (IIoT) ve Ara katman kavramları ön plana çıkmaktadır. IIoT'nin gelişimi sayesinde, endüstride verimin artırılması, zaman ve işgücünden tasarruf edilmesi, hata oranlarının düşürülmesi, kalitenin iyileştirilmesi ve kolay yönetim sağlanması gibi birçok fayda sağlanmaktadır. Endüstriyel ihtiyaçların giderilerek, yüksek verimliliğe sahip ortamların sağlanması için Ara Katman mimarilerine ihtiyaç duyulmaktadır. Ayrıca çalışma ortamlarında kullanılan sensörlerden toplanan verilerin anlamlandırılması ile makine ve ekipmanları üzerinde gerçekleştirilen anomali tespitleri sayesinde kestirimci bakım faaliyetleri yürütülebilmektedir. Anomali tespitlerinin önceden yapılabilmesi, bakım ihtiyaçlarının düşük maliyet ve yüksek güvenilirlikle karşılanabilmesini sağlamaktadır. Böylece kullanılan sistemler sürekli takip edilerek büyük oranda sürdürülebilir kılınmaktadır. Çalışmamızda, açık kaynak kodlu Fiware arakatmanı bileşenleri olan Orion Context Broker, OPC-UA IoT Agent ve Draco kullanılarak bir veri hattı sistemi tasarlanmaktadır. Tasarlanan bu sistemde, donanım bileşenleri olan V20 sürücü ile Plc arasında Modbus RTU iletişim protokolü, OPC-UA sunucu yayını yapan Plc ile arakatman arasında OPC-UA iletişim protokolü kullanılmaktadır. V20 motor sürücüsü, dokunmatik panel ekran ve Plc donanım bileşenlerinin tek tek programlanması ile oluşturulan test düzeneği üzerinden, Fiware bileşeni OPC-UA IoT Aracısı tarafından okunan, NGSİ veri modeline ait akım ve frekans verileri ara katmana aktarılmaktadır. Veriler, ara katman bileşeni olan Draco ile gerekli dönüşüm işlemlerine tabi tutularak PostgreSQL veri tabanına yönlendirilmekte ve burada saklanmaktadır. Son olarak veri tabanında depolanan akım verileri, öznitelik seçimi yöntemleri kullanılarak sadeleştirilmekte ve sekiz farklı makine öğrenme algoritması kullanılarak bir anomali tespiti uygulaması gerçekleştirilmektedir.

**Anahtar Kelimeler:** Kestirimci Bakım, Fiware, OPC-UA, ModBus, Yapay Öğrenme

## SUMMARY

In parallel with the development of technology, the concepts of Industrial Internet of Things (IIoT) and Intermediate layer come to the fore in industrial systems that digitize. Thanks to the development of IIoT, many benefits are provided in the industry, such as increasing efficiency, saving time and labor, reducing error rates, improving quality and providing easy management. Middleware architectures are needed to meet industrial needs and provide high-efficiency environments. In addition, predictive maintenance activities can be carried out thanks to the interpretation of the data collected from the sensors used in the working environments and the anomaly detection performed on the machinery and equipment. The ability to detect anomaly in advance ensures that maintenance needs can be met with low cost and high reliability. Thus, the systems used are continuously monitored and largely made sustainable. In our study, a data pipeline system is designed using open source Fiware middleware components, Orion Context Broker, OPC-UA IoT Agent and Draco. In this designed system, Modbus RTU communication protocol is used between V20 driver and Plc, which are hardware components, and OPC-UA communication protocol is used between Plc broadcasting OPC-UA server and middleware. The current and frequency data of the NGSII data model, which is read by the Fiware component OPC-UA IoT Agent, are transferred to the middle layer through the test setup created by programming the V20 motor driver, touch panel screen and Plc hardware components one by one. The data is forwarded to the PostgreSQL database and stored in the PostgreSQL database by undergoing the necessary transformation processes with the middleware component Draco. Finally, the flow data stored in the database is simplified by using feature selection methods and an anomaly detection application is carried out using eight different machine learning algorithms.

**Keywords:** Predictive Maintenance, Fiware, OPC-UA, ModBus, Machine Learning

## TEŐEKKÜR

Bu alıőmanın yürütölmesi sırasında desteęini esirgemeyen danıőmanım Dr. Öğr. Üyesi Eyüp ınar'a, yoğun alıőmalarım sırasında sabırla destek veren sevgili eőim Hatice'ye ve alıőmalarıma sürekli tatlı aralar verdirerek motivasyonumu en yüksek düzeyde tutan sevigili oęlum Ahmet Akif'e sonsuz teőekkürlerimi sunarım.



## İÇİNDEKİLER

### Sayfa

<b>ÖZET</b> .....	vi
<b>SUMMARY</b> .....	vii
<b>TEŞEKKÜR</b> .....	viii
<b>İÇİNDEKİLER</b> .....	ix
<b>ŞEKİLLER DİZİNİ</b> .....	xi
<b>ÇİZELGELER DİZİNİ</b> .....	xiv
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	xv
<b>1. GİRİŞ VE AMAÇ</b> .....	1
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	4
<b>3. MATERYAL VE YÖNTEM</b> .....	19
3.1. Materyal .....	20
3.1.1. Yazılım materyalleri .....	20
3.1.1.1. <u>Fiware ara katman yazılımı</u> .....	20
3.1.1.2. <u>OPC-UA IoT aracı</u> .....	22
3.1.1.3. <u>MongoDb</u> .....	23
3.1.1.4. <u>Postman</u> .....	23
3.1.1.5. <u>Draco</u> .....	23
3.1.1.6. <u>PostgreSQL</u> .....	24
3.1.1.7. <u>SoMachine</u> .....	24
3.1.1.8. <u>Vijeo Designer</u> .....	25
3.1.2. Haberleşme protokolleri .....	25
3.1.2.1. <u>Modbus RTU</u> .....	25
3.1.2.2. <u>OPC-UA</u> .....	27
3.1.3. Donanım materyalleri .....	27
3.1.3.1. <u>Elektrik motoru</u> .....	27
3.1.3.2. <u>Sürücü</u> .....	28
3.1.3.3. <u>HMI dokunmatik panel ekran</u> .....	30
3.1.3.4. <u>Plc</u> .....	30

## İÇİNDEKİLER

### Sayfa

3.1.4. Makine öğrenmesi algoritmaları .....	31
3.1.4.1. <u>Rasgele Orman (RF)</u> .....	31
3.1.4.2. <u>Karar Ağaçları (DT)</u> .....	32
3.1.4.3. <u>Gradyan Artırma Makinesi (GBM)</u> .....	32
3.1.4.4. <u>Hafif Gradyan Artırma Makinesi (LGBM)</u> .....	33
3.1.4.5. <u>K-En Yakın Komşuluk (K-nN)</u> .....	33
3.1.4.6. <u>Kategorik Artırma (CB)</u> .....	34
3.1.4.7. <u>Lojistik Regresyon (LR)</u> .....	34
3.1.4.8. <u>Çok Katmanlı Algılayıcılar (MLP)</u> .....	34
3.2. Yöntem.....	35
3.2.1. Test ortamının hazırlanması.....	35
3.2.2. Ara katman mimarisinin geliştirilmesi.....	40
3.2.3. Makine öğrenmesi yöntemleri.....	45
<b>4. BULGULAR VE TARTIŞMA</b> .....	48
4.1. Test Ortamı ve Ara Katman Gelişimi Sırasındaki Bulgular .....	48
4.2. Kestirimci Bakım İçin Gerçekleştirilen Anomali Tespiti Sırasındaki Bulgular .....	54
<b>5. SONUÇ VE ÖNERİLER</b> .....	60
<b>KAYNAKLAR DİZİNİ</b> .....	62

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. SWAMP katmanlı mimari (Kamienski vd., 2019) .....	12
2.2. In.IoT, Linksmart, Konker, Orion ve Sitewhere ara yazılımı dikkate alınarak 1, 15 ve 100 parametrenin gönderildiği başarılı tek bir isteğin paket boyutu analizi (Cruz vd., 2021) .....	13
2.3. In.IoT, Linksmart, Konker ve Orion ara katman yazılımı dikkate alınarak 1, 15 ve 100 parametrenin gönderildiği 5000 eşzamanlı kullanıcı için yanıt süresi analizi (Cruz vd., 2021) .....	14
2.4. Fiware, IoT Middleware ve OPC-UA'nın yıllara göre dağıtılmış akademik çalışmaların sayıları .....	16
3.1. Arakatman mimarisinin geliştirilmesine ait blok diyagram .....	19
3.2. NGSİ veri modeli ana öğeleri (Fonseca & Marquez, 2018) .....	21
3.3. OPC-UA IoT Aracısının sistem üzerindeki yeri ve görevi (Fiware-OPC-UA Agent, 2023) .....	22
3.4. Sunucu ve istemcilere ait tekli ve çoklu yayın modları (Urrea vd., 2016) .....	26
3.5. Modbus RTU protokol Veri Yapısı (Akkaya vd., 2015) .....	26
3.6. IFARLAB elektrik motoru test düzeneği (IFARLAB,2023) .....	28
3.7. Modbus makrosu sürücü bağlantı şeması (Siemens, 2023c) .....	30
3.8. Sürücü ve sonlandırma devresinin bağlantı şekli .....	36
3.9. Plc üzerindeki Modbus Slave yapılandırması ve değişkenler .....	38
3.10. Plc üzerindeki OPC-UA yayın ve sunucu konfigürasyonu .....	39
3.11. Vijeo Designer ile HMIGXU3500 dokunmatik panel ekran tasarımı .....	40
3.12. Portainer.io ara yüzü .....	41

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.13. Docker-compose.yml ve Config.json dosyalarına ait yapılandırma ayarları (a)Docker-compose.yml , (b)Config.json .....	42
3.14. DBCPConnectionPool yapılandırma ayarları .....	44
3.15. Draco'nun aktif hale getirilmesi için kullanılan REST API çağrı kodu .....	44
4.1. SoMachine uygulaması üzerinde Modbus RTU seri haberleşme kontrolü .....	49
4.2. OPC Test çalışmalarında kullanılan UaExpert uygulaması .....	49
4.3. Postman üzerinden IoT Aracı sürücüsüne yapılan REST çağrısı ve sonucu .....	50
4.4. Orion Context Broker'a ait test çalışmalarında kullanılan Postman ve MongoDB uygulamalarına ait ekran görüntüleri .....	51
4.5. Draco'ya ait sistem sağlığı kontrol REST çağrısı ve çağrı yanıtı .....	52
4.6. PgAdmin uygulamasından elde edilen ekran görüntüsü .....	53
4.7. Gradyan Artırma Makinesi (GBM) algoritması Confusion Matrixi ve Skor Tablosu .....	56
4.8. Rasgele Orman (RF) algoritması Confusion Matrixi ve Skor Tablosu .....	56
4.9. Hafif Gradyan Artırma Makinesi (LGBM) algoritması Confusion Matrixi ve Skor Tablosu .....	56
4.10. Kategorik Artırma (CatBoost) algoritması Confusion Matrixi ve Skor Tablosu .....	57
4.11. Karar Ağaçları (DT) algoritması Confusion Matrixi ve Skor Tablosu .....	57
4.12. K-En Yakın Komşuluk (K-nN) algoritması Confusion Matrixi ve Skor Tablosu ....	57
4.13. Lojistik regresyon (LR) algoritması Confusion Matrixi ve Skor Tablosu .....	58
4.14. Çok Katmanlı Algılayıcılar (MLP) algoritması Confusion Matrixi ve Skor Tablosu .....	58

## ŞEKİLLER DİZİNİ

### Sekil

### Sayfa

4.15. Makine öğrenmesi modellerinin doğruluk oranlarının grafiksel gösterimi ve sayısal değerleri .....	59
---	----



## ÇİZELGELER DİZİNİ

<b><u>Cizelge</u></b>	<b><u>Sayfa</u></b>
2.1. Literatür araştırması sırasında ara katman yazılımlarının incelenmesi ile ilgili ele alınan başlıca çalışmalar.....	5
2.2. Seçili ara katman yazılımlarının özet bilgileri (Medeiros vd., 2022).....	7
2.3. Fiware ara katman mimarisi kullanılan literatür örnekleri.....	17
3.1. V20 sürücü kontrol makroları (Siemens, 2023c).....	29
3.2. Sürücü Cn011 Modbus yapılandırma ayarları (Siemens, 2023b).....	37

## SİMGELER VE KISALTMALAR DİZİNİ

### Simgeler

$A_i$	Doğruluk
$p_i$	Duyarlılık
$\pi_i$	Kesinlik

### Kısaltmalar

### Açıklama

API	Uygulama Programlama Arabirimi (Application Programming Interface)
CB	Kategorik Artırma (Categorical Boosting)
CNC	Bilgisayar Destekli Nümerik Kontrol (Computer Numeric Control)
COAP	Kısıtlı Uygulama Protokolü (Constrained Application Protocol)
CRC	Döngüsel Artıklık Kontrolü (Cyclic Redundancy Check)
DDS	Veri Dağıtım Servisi (Data Distribution Service)
DT	Karar Ağacı (Decision Tree)
GBM	Gradyan Artırma Makinesi (Gradient Boosting Machine)
GE	Genel Etkinleştirici (Generic Enabler)
HMI	İnsan Makine Ara yüzü (Human Machine Interface)
HTTP	Üstün Metin Transfer Protokolü (Hyper Text Transfer Protocol)

**SİMGELER VE KISALTMALAR DİZİNİ**

<b><u>Simgeler</u></b>	<b><u>Açıklama</u></b>
HTTPS	Güvenli Üstün Metin Transfer Protokolü (Hyper Text Transfer Protocol Secure)
HVAC	Isıtma, Havalandırma, İklimlendirme Sistemi (Heating, Ventilating and Air Conditioning)
IFARLAB	Akıllı Fabrika ve Robotik Laboratuvarı (Intelligent Factory and Robotics Laboratory)
IIOT	Endüstriyel Nesnelerin İnterneti (Industrial Internet Of Things)
IOT	Nesnelerin İnterneti (Internet of Things)
IP	İnternet Protokolü (Internet Protocol)
IRT	Eşzamanlı verilerin gerçek zamanlı olarak aktarılması (Isochronous Real-Time)
KOBİ	Küçük ve Orta Büyüklükteki İşletmeler
KNN	K-En Yakın Komşuluk (k-Nearest Neighbors)
LGBM	Hafif Gradyan Artırma Makinesi (Light Gradient Boosting Machine)
LR	Lojistik Regresyon (Logistic Regression)
M2M	Makineden Makineye (Machine to Machine)
MLP	Çok katmanlı Algılayıcılar (Multi Layer Perceptron)
MQTT	Mesaj Kuyruğu Telemetri Aktarımı (Message Queuing Telemetry Transport)

**SİMGELER VE KISALTMALAR DİZİNİ**

<b><u>Simgeler</u></b>	<b><u>Açıklama</u></b>
NGSI	Yeni Nesil Servis Arayüzleri (Next Generation Service Interfaces)
NDN	Adlandırılmış Veri Ağı (Named Data Networking)
OPC-UA	Açık Platform İletişimi Birleşik Mimarisi (Open Platform Communications Unified Architecture)
PLC	Programlanabilir Mantıksal Denetleyici (Programmable Logic Controller)
REST	Temsili Durum Transferi (Representational State Transfer)
RF	Rastgele Orman (Random Forest)
ROS	Robot İşletim Sistemi (Robot Operating System)
RTU	Uzaktan Yönetim Birimi (Remote Terminal Unit)
TCP	İletim Kontrol Protokolü (Transmission Control Protocol)
USS	Evrensel seri arayüzler (Universelles Serielles Schnittstellen)
VF-OS	Sanal Fabrika Açık İşletim Sistemi (Virtual Factory Open Operating System)

## 1. GİRİŞ VE AMAÇ

Teknolojinin hızlı ilerleyişi ile endüstriyel alanda yüksek teknolojinin kullanımı yaygınlaşırken, insan gücüne ihtiyaç azalmış, makineleşme ve otomasyon sistemleri ön plana çıkmıştır. Endüstriyel alandaki ihtiyaçlar ise sanayide robotların, makinelerin, sensörlerin ve yazılım sistemlerinin gelişimini sağlamıştır. Dördüncü sanayi devriminin en büyük amacı, birbirleri ile iletişim kurabilen, sensörler sayesinde buldukları ortamı algılayabilen ve topladıkları verileri analiz ederek ihtiyaçları saptayabilen makineler üretilmesidir. Bu makinelerin üretimde aktif rol alarak hızlı, kaliteli, ucuz ve insan hatalarından arındırılmış bir üretim sistemi geliştirilmesi hedeflenir. Geliştirilen sistemlerin geleneksel bakım yöntemleri ile yönetilmesi bakım ihtiyaçlarının karşılanmasını zorlaştırırken, maliyetlerini son derece artırmaktadır. Kestirimci bakım teknikleri sayesinde düşük maliyet ve yüksek güvenilirlikle bakım ihtiyaçları karşılanmaktadır. Bakım ihtiyaçlarının karşılanması ve maliyetlerinin düşürülmesi için veri kaynağı olan sensörler sayesinde toplanan veriler, endüstriyel haberleşme kanalları ile veri tabanlarına aktarılmaktadır. Verilerin çeşitli algoritmalar ile eğitilmesi sayesinde yapılan analizler, sistem bileşenlerinin bozulmalarını tespit etmek ve arıza durumları oluşmadan önce bakım yapılması için sistem yöneticilerini bilgilendirmektedir. Sang vd. (2020) belirledikleri çalışma alanındaki CNC ve diğer operasyon makinelerinde bulunan sensörlerden toplanan verileri ve Fiware bileşenlerini kullanarak Fiware ara katman yazılımının kestirimci bakımı nasıl desteklediğini araştırmıştır.

Dördüncü sanayi devrimi ile ön plana çıkan Nesnelerin İnterneti gibi akıllı çözümlerin geliştirilmesini sağlamak amacıyla çeşitli ara katman yazılımları kullanılmaktadır. Vikash vd. (2021) kablosuz sensörlerin içinde yer alacağı, IoT sistemlerinin geliştirilmesi sırasında ortaya çıkan sorunları inceleyerek bir çalışma gerçekleştirmekte ve belirlenen bazı ara katman çözümlerinde karşılaştırmalı bir inceleme yaparak ara katman yazılımlarını değerlendirmektedir.

Ara katman platformlarından biri olan Fiware, diğer platformlar ve bunlara ait bileşenleri birlikte çalıştırabilen açık kaynak kod sistemine sahip uygulama odaklı bir ekosistemdir. Açık kaynak olması, herhangi bir telif hakkı ihtiyacının bulunmaması,

sürdürülebilir olması, geliştirilmesinin ve entegrasyonunun kolay olması Fiware platformunun en büyük avantajlarıdır. Bu avantajlar kullanılabilirlik ve yönetim gücünü artırırken, maliyetleri düşürmektedir. Sinanovic vd. (2022) Fiware platformu ile ilgili temel bilgiler vererek, platformun neden tercih sebebi olması gerektiğini avantajlarıyla birlikte değerlendirmiş ve Fiware platformuna ait bileşenleri kullanarak akıllı şehir kullanım senaryosu ile ilgili bir çalışma gerçekleştirmiştir. Fiware platformu geliştiricileri 2018 yılında servis edilen ve Fiware platformuna özgü olarak geliştirilen “OPC-UA IoT Aracısı” adlı bileşeni deneysel bir fabrika ortamında kullanarak test etmiştir. Fiware OPC-UA IoT Aracısı kullanımı, üretim tesislerinden gelen endüstriyel IoT verilerini diğer Fiware bileşenleri ile entegre edecek, bu sayede bağlama duyarlı hizmetlerin ve akıllı uygulamaların geliştirilmesi sağlanacaktır. Bu deneyde, istasyonların güç durumlarını verimli bir şekilde yöneterek makinelerin boştaki sürelerini ve bekleme durumlarını lehine azaltmayı hedeflemişlerdir (Marguglio ve Aiello, 2018).

Endüstriyel alanlardaki gelişim, haberleşme protokollerinde incelenerek, sahada kullanılacak uygun protokol arayışlarını başlatmıştır. Endüstriyel IoT alanında kullanılan sistemlerde, tesis içi esnek ve sağlam iletişim için daha fazla önem kazanan OPC-UA haberleşme protokolü, dünya çapında en önemli protokollerden biri haline gelmektedir. Silva vd. (2021) protokol özellikleri ile hem son kullanıcı hem de IIoT ortamlarında kullanılmakta olan MQTT, COAP ve OPC-UA protokollerine ait test ve değerlendirmeler gerçekleştirmiştir. Büyük ölçekli FIT-IoT test ortamı kullanılarak gerçekleştirilen deneysel test sonuçlarında, CoAP' ın tüm senaryolarda en düşük tamamlanma süresi ile çalışan protokol olduğu, OPC-UA'nın ise CoAP ve MQTT'ye oranla daha yüksek tamamlanma süreleriyle çalıştığı belirtilmiştir.

Çalışmamızda, açık kaynak kodlu Fiware arakatmanı ve bileşenleri kullanılarak bir veri hattı sistemi tasarlanmaktadır. Tasarlanan bu sistemde, donanım bileşenleri arasında Modbus RTU iletişim protokolü, arakatman ile donanım bileşenleri arasında OPC-UA iletişim protokolü kullanılmaktadır. Laboratuvar ortamında; elektrik motoru, V20 motor sürücü, dokunmatik panel ekran ve Plc donanım bileşenleri ile oluşturulan test düzeneği üzerinden, Fiware bileşeni OPC-UA IoT Aracısı tarafından okunan akım verileri ara katman içerisinde dönüştürülerek PostgreSQL veri tabanında saklanmaktadır. Son olarak

veri tabanında depolanan akım verileri çeşitli makine öğrenme algoritmaları kullanılarak bir anomali tespiti uygulaması gerçekleştirilmektedir.

Çalışmaya ait düzenleme yapısı şu şekilde gerçekleştirilmektedir. Takip eden bir sonraki bölümde çalışmada kullanılan teknik ve yöntemler ile ilgili literatür çalışmalarına yer verilmektedir. Üçüncü bölümde, tasarımı gerçekleştirilen ara katman mimarisine ait materyal ve metot hakkındaki bilgilere yer verilmektedir. Dördüncü bölümde, ara katman mimarisinin geliştirilmesi sırasında sensör verilerinin okuma sonuçlarına, iletişim kanallarının nasıl kurulduğuna, çalışma esnasında karşılaşılan zorluklara ve makine öğrenmesi algoritma sonuçlarına ait bulguların detaylarına yer verilmektedir. Beşinci bölümde, oluşturulan arakatman mimarisinde okunan akım verilerine ve farklı algoritmalara ait performans ve test sonuçları ile sonraki çalışmalar için öneriler yer almaktadır. Çalışma, sonuçlar ve öneriler bölümü ile sonlandırılmaktadır.

## 2. LİTERATÜR ARAŞTIRMASI

Tez çalışmasının gerçekleştirilmesi sırasında yapılan literatür incelemelerine bu başlık altında yer verilmektedir. Joseph vd. (2017) akıllı şehircilik adına yaptıkları çalışmada, şehirdeki birbirinden farklı sistemler arasında bir iletişim katmanı olarak çalışan ve elde edilen veriler üzerinde kontrolü sağlayan IoT tabanlı bir ara katman mimarisi denemesi gerçekleştirmektedir. Mimarinin gerçekleştirilmesinde iletişim kanalı olarak MQTT, HTTPS, HTTP ve CoAP protokollerini, test verilerinde ise araç takip cihazları ve cep telefonlarından alınan verileri kullanmaktadır. Ngu vd. (2017) ilk olarak akıllı saat aracılığı ile elde edilen sensör verileri yardımıyla kandaki alkol miktarının tahmini için tasarlanmış bir IoT uygulaması ve IoT ara yazılımı denemesi gerçekleştirmiştir. Daha sonra ise mevcut IoT ara yazılımının yetenekleri üzerine yapılan değerlendirmeler ile IoT ara katman yazılımı geliştirilmesine ait zorlukların ve kolaylık sağlayan teknolojilerin kapsamlı bir şekilde analizini gerçekleştirmiştir. Zhang vd. (2021) çalışmalarında bir IoT ara katman yazılımı geliştirilmesi sırasında karşılaşılan gereksinimleri ve zorlukları analiz ederek bu alandaki araştırmaların mevcut durumunu değerlendirmekte ve IoT ara yazılımları için teknik bir sınıflandırma kullanarak karşılaştıkları sorunları özetlemektedir. Kılıç (2022) çalışmasında IoT ara yazılımları kapsamında sıklıkla kullanılan DDS adı verilen veri dağıtım servisi ara katman mimarisini, ağ topolojileri, ağ kesintileri, bant genişliği daralmaları ve paket kayıpları gibi belirlediği kriterler ile analiz ederek bir simülasyon ortamı tasarlamaktadır. Buna göre, veri iletiminin sağlandığı belirli noktalarda yapılan değişiklikler ile paket kayıpları, ağ gecikmeleri ve servis kesintileri gibi durumların incelenebilmesi sağlanmaktadır.

Literatür araştırması kapsamında, ara katman yazılımlarına ait çalışmalarda gerçekleştirilen sınıflandırma, karşılaştırma ve değerlendirmelerin yapıldığı ara katman yazılımları Çizelge 2.1.'de görüntülenmektedir. Çalışmalarda her bir ara katman yazılımı ayrı ayrı ele alınarak özellikleri incelenmekte ve diğer ara katman yazılımları ile karşılaştırmaları yapılmaktadır.

Çizelge 2.1. Literatür araştırması sırasında ara katman yazılımlarının incelenmesi ile ilgili ele alınan başlıca çalışmalar.

Referans	Çalışma	İncelenen Ara Katman Yazılımı
Razzaque vd., 2016	Middleware for Internet of Things: A Survey	Hermes, EMMA, GREEN, RUNES, PRISMA, SensorBus, Mires, Hydra, Sensewrap, MUSIC, TinySOA, SOCRADES, SENSEI, ubiSOAP, Servilla, KASOM, CHOReOS, MOSDEN, Xively, CarrIoT, Echelon, Mate, Melete, MagnetOS, Squawk, Sensorware, Extended Mate, DVM, DAViM, SwissQM, TinyVM, TinyReef, Impala, Smart messages, ActorNet, Agilla, Ubiware, UbiROAD, AFME, MAPS, MASPOT, TinyMAPS, LIME, TeenyLIME, TinyLIME, TS-Mid, A3-TAG, SINA, COUGAR, IrisNet, Sensation, TinyDB, GSN, KSpot, HyCache, AutoSec, Adaptive middleware, MiLAN, TinyCubus, MidFusion
Ngu vd., 2017	IoT Middleware: A Survey on Issues and Enabling Technologies	Hydra, GSN, Google Fit, Xively, Paraimpu, Calvin, Node-RED, Ptolemy Accessor Host
Cruz vd., 2018	A Reference Model for Internet of Things Middleware	Alljoyn, Amazon IoT platform, Artik cloud, Autodesk fusion connect, Carriots, Chorevolution, Cloudplugs, Devicehive, EVRYTHNG, Fiware, GroveStreams, InatelPlat, Iotivity, Kaa, Konker, Linksmart, Losant, M2Mlabs, Microsoft Azure IoT Suite, Nimbits, Nitrogen, OpenIoT, Sitewhere, Stack4Things, Tago, Telit IoT platform, Temboo, ThingSpeak, Thingworx IoT platform, Ubidots, WSO2 IoT server, Webinos, Xively
Farahzadi vd., 2017	Middleware technologies for cloud of things: a survey	Aura, ABC&S, Capnet, Carriots, CARISMA, CHOReOS, C-MOSDEN, COPAL, CoMiHoC, DropLock, Gaia, GSN, Link smart, OpenIoT, Rimware, SOCAM, ThingWorx, UPnP, VIRTUS, Xively

Çizelge 2.1. Literatür araştırması sırasında ara katman yazılımlarının incelenmesi ile ilgili ele alınan başlıca çalışmalar (devam).

Referans	Çalışma	İncelenen Ara Katman Yazılımı
Benayache vd., 2019	MsM: A microservice middleware for smart WSN-based IoT application	PRISMA, Mires, Linksmart, TinySOA, CHOReOS, VMSTAR, MagnetOS, LIME, TinyLIME, Agilla, Ubiware, TinyDB, GSN, Adaptive Middleware, MiLAN, Cloud-WBAN, Cloudlet-based WBANs
Agarwal ve Alam 2020	Investigating IoT Middleware Platforms for Smart Application Development	AWS IoTPlatform, Microsoft Azure IoT Hub, IBM Watson IoT Platform, Google IoT Platform, Oracle IoT Platform, Kaa, ThingSpeak, Carriots, Temboo, Samsara, Particle Cloud
Cavalcanti vd., 2021	Adaptive Middleware of Things	AMOT platform
Aslam ve Curry, 2021	A Survey on Object Detection for the Internet of Multimedia Things (IoMT) using Deep Learning and Event-based Middleware: Approaches, Challenges, and Future Directions	CORBA, SIENA, CEA, SECO, JEDI, Elvin, Hermes, STEAM, ToPSS, S-ToPSS, OPS, A-TOPSS, GREEN, EMMA, Mires, SensorBus, WebSphere MQ, MiSense, Rebeca, FOMatch, PSWare, TinyDDS, Prisma, Mosquitto, Approximate Event Matcher
Pradeep vd., 2021	A holistic approach to a context-aware IoT ecosystem with Adaptive Ubiquitous Middleware	Gaia, CoBrA, CAMUS, AmbieAgents, SOCAM, CAMPS, TeenyLIME, SAMOA, Agent based Middleware, Hydra, iConAwa, UbiRoad, KASOM, CAMPUS, CASF, CoCaMAAL, SeCoMan, CONASYS, Cooperative Middleware, MiSci
Vikash vd., 2021	Middleware Technologies for Smart Wireless Sensor Networks towards Internet of Things: A Comparative Review	MiLAN, Mires, EnviroTrack, SINA, DSWare, Mate, MagnetOS, Impala, Agilla, TinyLime

Çizelge 2.1. Literatür araştırması sırasında ara katman yazılımlarının incelenmesi ile ilgili ele alınan başlıca çalışmalar (devam).

Referans	Çalışma	İncelenen Ara Katman Yazılımı
Zhang vd., 2021	Middleware for the Internet of Things A survey on requirements, enabling technologies, and solutions	Hydra, CoCaMAAL, BDCaM, Atlas, IoT-MP, PRISMA, SeCoMan, SenSocial, Mate, SensorWare, Actinium, MODE, TinyDBi SINA, Cougar, DSWare, Agilla, Eagilla, ACOSO, Sensomax
Medeiros vd., 2022	Middleware for the Internet of Things: a systematic literature review	In.iot, SEnviro Connect, CUPUS, InterSCity, Middle-Brigde, CoIoT, M-Hub/CDDL, SWaMM, Thinger.io, Cilia, DeXMS, MSM, SmartCity Aware, MinT, eVATAR+, MSOAHIoT, RDS

Medeiros vd. (2022) IoT için geliştirilen ve güncelliğini koruyan ara katman platformlarını listelemiş, bu platformları uygulama alanları, mimarileri, gereksinimleri, kullanılan iletişim protokolleri ve hangi fiziksel katmanlarda kullanıldığı gibi başlıklardan yola çıkarak literatürün sistematik bir incelemesini gerçekleştirmiştir.

Çizelge 2.2. Seçili ara katman yazılımlarının özet bilgileri (Medeiros vd., 2022).

Ara Katman	Uygulama	Mimari	Gereksinimler	Protokoller	Fiziksel Katman
In.iot	Akıllı Enerji	Cloud	Birlikte çalışabilirlik, Ölçeklenebilirlik, Güvenilirlik	MQTT, CoAP, HTTP	-
SEnviro Connect	Akıllı Tarım	Edge	Birlikte çalışabilirlik, Ölçeklenebilirlik, Güvenilirlik	HTTP, MQTT	2G/3G
-	Cihaz yönetimi	Cloud	Birlikte çalışabilirlik, Güvenilirlik	HTTP/REST, MQTT, COAP	WiFi, BLE, ZigBee, Z-Wave, M-Bus, CANBus

Çizelge 2.2. Seçili ara katman yazılımlarının özet bilgileri (Medeiros vd., 2022) (devam).

Ara Katman	Uygulama	Mimari	Gereksinimler	Protokoller	Fiziksel Katman
CUPUS	Mobil Kalabalık Algılama	Cloud	Esneklik	HTTP/GCM	BLE
-	Kanalizasyon Arıtma İstasyonları	Edge	Birlikte çalışabilirlik,	MQTT, HTTP	Modbus, RS485, RS422, RS232
M-Hub/CDDL	Bağlam Kalitesi	Cloud	Bağlam Yönetimi	MQTT	Bluetooth
-	Akıllı Çevre	Cloud	Esneklik	HTTP/REST, AMQP	WiFi
InterSCity	Akıllı Şehir	Cloud	Birlikte çalışabilirlik, Ölçklenebilirlik, Güvenilirlik, Esneklik	HTTP/REST, MQTT	-
Middle-Brigde	Farklı Uygulama Protokolleri	Edge	Birlikte çalışabilirlik	MQTT, CoAP, DDS, WebSocket, HTTP/REST	-
-	e-Sağlık	Edge	Birlikte çalışabilirlik, Güvenilirlik	-	WiFi, Bluetooth
CoIoT	Akıllı Labaratuar	Edge	Bağlam Yönetimi, Birlikte çalışabilirlik	HTTP/REST	WiFi
SWaMM	Akıllı Sulama	Cloud	Birlikte çalışabilirlik	HTTP/REST, MQTT	WiFi, M-Bus, LoRa
Thingier.io	Veri Entegrasyonu	Cloud	Bağlantı, Birlikte çalışabilirlik	HTTP/REST, MQTT, CoAP	WiFi, Lora, SigFox

Çizelge 2.2. Seçili ara katman yazılımlarının özet bilgileri (Medeiros vd., 2022) (devam).

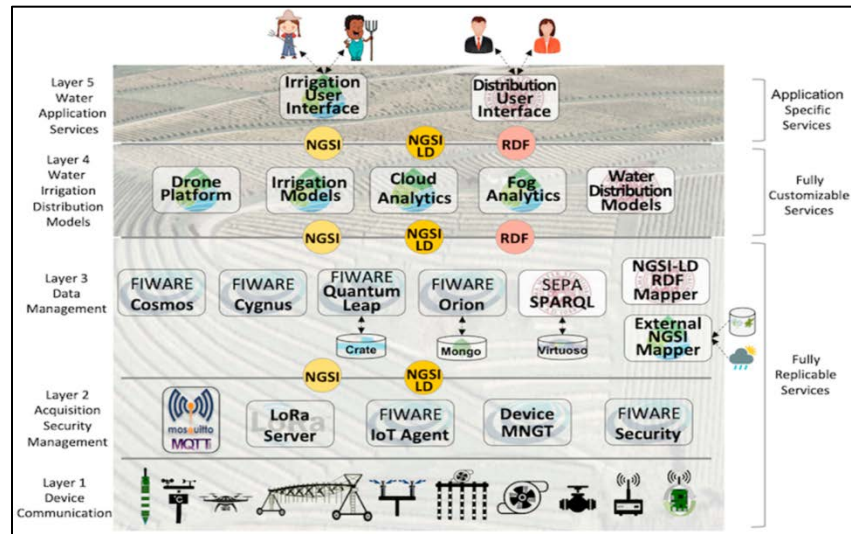
Ara Katman	Uygulama	Mimari	Gereksinimler	Protokoller	Fiziksel Katman
DeXMS	Farklı Uygulama Protokolleri	Edge	Birlikte çalışabilirlik	MQTT, CoAP, Web-Socket, HTTP/REST	-
MSM	Cihaz yönetimi	Cloud	Birlikte çalışabilirlik, Mahremiyet, Güvenlik	MQTT, HTTP, DDS, CoAP, WebSocket	-
SmartCity Aware	Akıllı Şehir	Cloud	Esneklik	HTTP/REST	WiFi
MinT	Cihaz yönetimi	Edge	Birlikte çalışabilirlik, Güvenilirlik	MQTT, CoAP, HTTP/REST	WiFi, Bluetooth, BLE, ZigBee
-	Akıllı Bina	Edge	Birlikte çalışabilirlik	Socket	Wi-Fi, Ethernet
MSOAH-IoT	Akıllı Ev	Edge	Birlikte çalışabilirlik	HTTP/REST	WiFi, Bluetooth, 6Low-PAN, ZigBee
RDS	Cihaz yönetimi	Edge	Birlikte çalışabilirlik, Ölçeklenebilirlik	DDS	-
eVATAR+	Akıllı Ev	Edge	Birlikte çalışabilirlik	HTTP/REST	-

Çizelge 2.2’de görüldüğü üzere gerçekleştirilen sistematik literatür incelemesindeki ara katman yazılımlarında, veri iletişiminin sağlanması için birçok veri protokolü kullanılmış ve bu protokollerin yetenekleri sayesinde farklı alanlarda çeşitli uygulamalar geliştirilmiştir.

Mourtzis vd. (2018) OPC-UA haberleşme protokolü standartlarına dayanan bir makine modeli ile freze ve CNC tezgâhlarını dijitalleştiren bir model tasarımı gerçekleştirmiştir. Laboratuvar ortamında eski takım tezgahlarını dijitalleştirmek için bir veri toplama cihazı geliştirerek OPC-UA haberleşme protokolü sayesinde bu sistemleri birbirine entegre etmişlerdir. Bouloukakis vd. (2019) IoT ara yazılım katmanlarına ait etkileşim protokollerinin çok fazla olması sebebi ile farklı yapıdaki nesnelerin birbirine bağlanmasını desteklemek amacıyla geliştirilen protokol araçlarının sistematik bir incelemesini gerçekleştirmiştir. Veri alışverişi bağlayıcı modeline dayanan sistematik inceleme sayesinde, nesnelerin hafif arabulucular aracılığı ile birbirine bağlanabildiğini kanıtlamaktadırlar. Liu vd. (2019) CNC takım tezgâhları ve çeşitli yazılım uygulamaları arasında OPC-UA ve MTConnect tabanlı bir Siber Fiziksel Takım Tezgâhları Platformu tasarımı gerçekleştirmiştir. Çalışmalarında, OPC-UA ve MTConnect tabanlı sistemin birlikte çalışabilirliğini inceleyerek, geliştirdikleri platformun sağladığı avantajları gösterebilmek için OPC-UA protkolü ile haberleşen, gelişmiş bir arayüzü bulunan ve bulut ortamında çalışan bir uygulama geliştirmişlerdir. Cavalieri vd. (2019) OPC-UA ve OPC-UA'ya ait sunuculara, web tarayıcılar aracılığıyla, kaynak odaklı bir şekilde erişim sağlayabilen, REST mimarisine dayalı OPC-UA Web Platformu tasarımı gerçekleştirmiştir. Gerçekleştirdikleri çalışmada, diğer OPC çalışmalarından farklı olarak, kullanıcı ve platform arasında değiş tokuş edilen mesajların sayısını sınırlamış ve OPC-UA bilgi modelinin görünümünü kullanıcının bakış açısından basitleştirerek, kullanıcının belirlenen kavramlar hakkında sınırlandırılmış bir bilgiye sahip olmasını sağlamışlardır. Gonzalez vd. (2019) çalışmalarında OPC-UA'nın uygulanabilirlik ve yetenekleri bakımından incelemesini yapmış, OPC iletişim kanalının gelişimi ile ilgili ayrıntılardan bahsederek literatürdeki yakın tarihli OPC tabanlı sistemlerin genel bir araştırmasını gerçekleştirmiştir. Coito vd. (2020) akıllı ilaç otomasyon sistemi geliştirmek için bir ara katman yazılımı kullanarak endüstriyel bir örnek uygulama gerçekleştirmiştir. Bu uygulama da OPC-UA, TCP/IP ve PROFINET IRT gibi haberleşme kanallarını değerlendirmişlerdir. Muniraj ve Xu (2021) çalışmalarında OPC serisine ait standartları incelemiş ve akıllı bir fabrikada makineler arası iletişim için OPC-UA standartları ile geliştirilmiş bir uygulama gerçekleştirmiştir. Vaclavova vd. (2022) IIoT cihazların sağladığı özelliklerin sınırlı olması nedeniyle, OPC-UA'nın avantajlarından faydalanarak "Yüksek Seviyeli Sonlu Durum Makinesi" modeline dönüştürülmüş bir IIoT cihazı tasarımı gerçekleştirmiştir. Gelişmiş izleme ve veri toplama sağlayabilen bu tasarım

sayesinde üretim sürecinin kontrolünü daha iyi yapabileceklerdir. Fernandez vd. (2016) Las Palmas de Gran Canaria limanında bulunan gemi, şamandıra ve meteoroloji istasyonlarından sensör verilerini toplama, depolama ve analiz etme işlemlerini gerçekleştirmek için Fiware tabanlı bir akıllı liman uygulaması gerçekleştirmiştir. Çalışma ile Fiware ara katman yazılımına ait ayrıntılı bir inceleme sağlanmıştır. Martinez vd. (2016) hazırladıkları test ortamında, Fiware'in tarımsal uygulamaların geliştirilmesi için uygun bir ara katman olup olmadığını, hassas tarımda nasıl kullanılacağını, bu tür uygulamalar için uygunluğunu, ölçeklenebilirliğini ve etkinliğini inceleyen bir çalışma gerçekleştirmiştir. Hassas tarım alanında nesnelerin interneti kullanım durumu senaryosunun analizine yer vermişlerdir. Preventis vd. (2016) Fiware platformuna dayanan IoT-A tabanlı bir mimari denemesi ile bu iki platformun birlikte çalışabilirliğini ve platformların yeteneklerini incelemiştir. Gün koşullarında, Fiware platformunun yeterli araçlara sahip olmaması sebebi ile IoT-A standartlarına sahip mimarilerin tam anlamıyla desteklenemeyeceğini değerlendirmektedirler. Cardoso vd. (2017) belirledikleri bir dizi nitel ve nicel ölçüm özellikleri ile ETSI M2M ve Fiware ara katman mimarilerini bir akıllı şehir senaryosunda kıyaslayarak ara katman yazılımlarının güçlü ve zayıf yönlerini ortaya koymuşlardır. Ferreira vd. (2017) özellikle tarım alanında bulunan endüstri çalışmalarını hedef alarak, VF-OS projesi ara yüz uygulamalarına veri sağlamak için "Sanal Fabrika Sistemleri" kapsamında Fiware genel etkinleştiricilerini kullanan bir çalışma gerçekleştirmişlerdir. Çalışmada VF-OS'nin temel işlevlerini desteklemek için Fiware bileşenlerinden MRCoAP, Orion Context Broker, STH - Comet ve IDAS kullanılarak IoT hizmeti etkinleştirme, karmaşık veri işleme ve içerik yönetimi gibi çözümler sağlanmıştır. Alonso vd. (2018) Fiware ara katman bileşenlerini kullanarak "Endüstriyel Veri Alanı Referans Mimarisi"nin bir uygulaması ile makinelerin kestirimci bakımını gerçekleştirmek ve üretim süreçlerindeki kusurları tespit etmek istemiştir. Bunun için bir Freze Makinesi ve bir Koordinat Ölçüm Makinesi tarafından üretilen verileri analiz eden gerçek bir sanayi senaryosunu devreye alarak gerçekleştirmişlerdir. Araujo vd. (2019) IoT cihazlarının büyük ölçekli veri değişimlerini içeren kapsamlı bir test ortamı oluşturarak, MQTT ve CoAP tabanlı LWM2M protokolünü kullanan bulut tabanlı Fiware platformunun kapsamlı bir analizini sunmuştur. Celesti vd. (2019) sensörler ve düşük maliyetli IoT tıbbi cihazlarından sağlanan veriler ile hastaların sağlık durumuna ait parametreleri doğrudan evlerinde toplayabilen, Fiware tabanlı Uzaktan Hasta İzleme Sistemi denemesi gerçekleştirmiştir. Cirillo vd. (2019) IoT platformlarının içinden, bir araştırma olarak

başlayıp ticari bir düzeye geçiş yapan Fiware platformunun yeteneklerini, küresel IoT pazarı, akıllı şehircilik uygulamaları ve IoT artırılmış otonom sürüş senaryoları üzerinden gerçekleştirilen çalışmalar ile değerlendirmiştir. Her bir kullanım durumunu ve bunlara ait kritik sorunları, tasarım, uygulama ve dağıtım aşamalarında elde edilen sonuçlar ile analiz etmişlerdir. Baca Gomez vd. (2019) kullanıcıların hedeflerine ulaşmak için hava kirliliği ve çevre koşulları gibi etkenleri hesaba katarak en iyi rotayı belirlemesine olanak tanıyan Yeşil Rota adlı bir uygulamayı, Cloudino ve Fiware teknolojilerini kullanarak geliştirmiştir. Geliştirdikleri “Hava Kalitesi İzleme Sistemi” sayesinde çeşitli sensörlerden veriler toplamış ve bu verileri Fiware bileşenlerinde işleyerek en uygun rota hesaplamalarını yapmışlardır. Zamora-Izquierdo vd. (2019) test ortamı olarak belirledikleri tesiste, Fiware platformuna ait bileşenleri kullanarak akıllı tarım uygulaması gerçekleştirmiştir. Tesis içindeki verilerin toplanması için, sensörler ile güneş radyasyonu, nem, sıcaklık, CO<sub>2</sub>, pH ölçer, elektriksel iletkenlik, sıvı tüketimi, basınç ölçümü yapılırken, diğer aktüatörler ile toprak ve su besleme pompaları, vanalar ve cihazların kontrolü yapılmaktadır. Sensörler ve aktüatörlerden toplanan veriler, MQTT ve COAP gibi Iot protokolleri kullanılarak oluşturulan “Siber Fiziksel Sistem”e aktarılırken, dikey yönlü bulut erişimi için yeni nesil hizmet arayüzü olarak adlandırılan NGSI kullanılmaktadır. Kamienski vd. (2019) tarımda hassas sulama sağlamak için geliştirilen IoT tabanlı akıllı su yönetimi platformu SWAMP ile Fiware platform entegrasyonunu sağlayarak Fiware bileşenlerinin performans analizini değerlendirmiştir. Şekil 2.1’de SWAMP katmanlı mimari bileşenleri görüntülenmektedir.

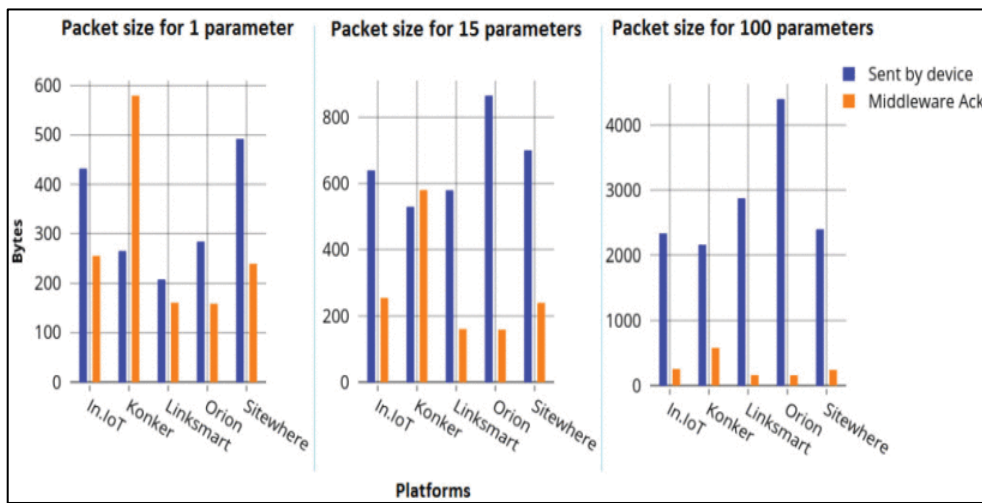


Şekil 2.1. SWAMP katmanlı mimari (Kamienski vd., 2019).

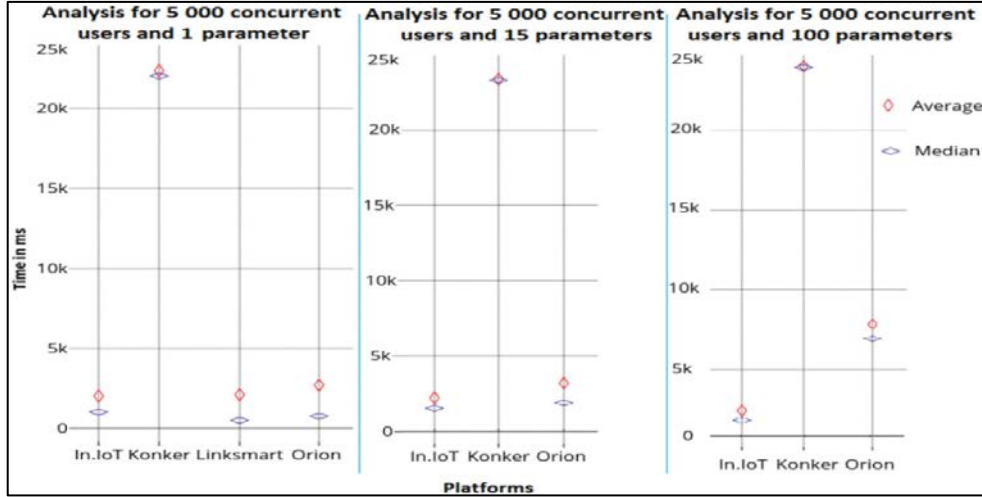
Beş katmanlı olarak tasarlanan mimarinin ikinci katmanında güvenlik yönetimi için kullanılan Fiware bileşenlerine, üçüncü katmanda ise Fiware ve SEPA SPARQL motoruna dayalı veri yönetiminin gerçekleştirildiği, veri depolama, işleme ve dağıtımından sorumlu yazılım bileşenlerine yer verilmiştir.

Pozo vd. (2020) Fiware bileşenleri ile akıllı binalardaki kablosuz iletişim senaryolarının güvenlik durumlarını ele alarak IoT cihazlarının korunması amaçlı erişim kontrol mimarisini değerlendirmiştir. Bauer vd. (2021) çalışmalarında akıllı şehirler için dijital ikiz kavramı ve kullanımını açıklamaktadır. Dijital ikizlerin oluşturulmasında NGSI-LD tabanlı Fiware bileşenleri ile tahmine dayanan dijital ikiz verilerini keşfetmek, yönetmek ve uygulamak için teknolojilerin nasıl kullanılabileceğini açıklamışlardır.

Cruz vd. (2021) çalışmalarında tanımlanan ara yazılım endişelerini ele alarak, In.IoT adlı ölçeklenebilir, güvenli ve yenilikçi yeni bir ara katman yazılımı gerçekleştirmiştir. Gerçekleştirdikleri ara katman yazılımını literatürde mevcut olan ara katman yazılımlarından bazıları ile karşılaştırarak değerlendirmişlerdir. Şekil 2.2’de değerlendirmeye tabi tutulan In.IoT, Linksmart, Konker, Orion ve Sitewhere ara yazılımlara ait 1, 15 ve 100 parametrenin gönderildiği, başarılı tek bir isteğin paket boyutu analizleri görülürken Şekil 2.3’de 1, 15 ve 100 parametrenin gönderildiği, 5000 eşzamanlı kullanıcı için yanıt süresi analizi yer almaktadır.



Şekil 2.2. In.IoT, Linksmart, Konker, Orion ve Sitewhere ara yazılımı dikkate alınarak 1, 15 ve 100 parametrenin gönderildiği başarılı tek bir isteğin paket boyutu analizi (Cruz vd., 2021).



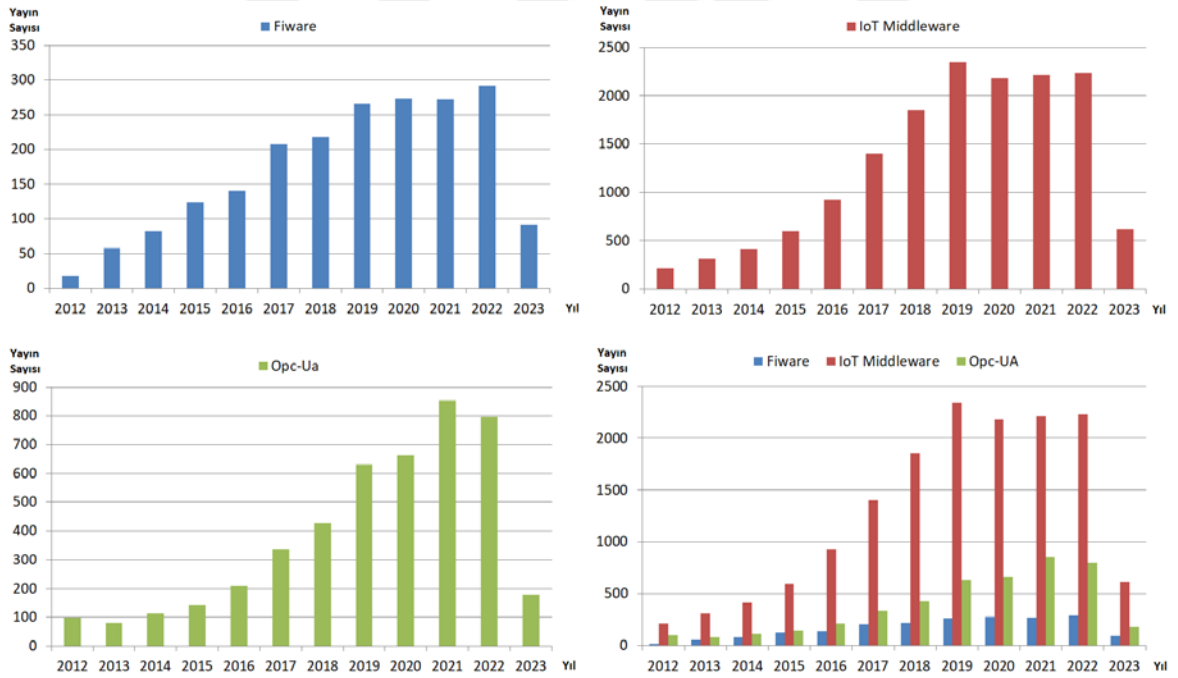
Şekil 2.3. In.IoT, Linksmart, Konker ve Orion ara katman yazılımı dikkate alınarak 1, 15 ve 100 parametrenin gönderildiği 5000 eşzamanlı kullanıcı için yanıt süresi analizi (Cruz vd., 2021).

Llopis vd. (2021) akıllı şehir uygulamalarının gelişimini desteklemek için Fiware platformunu kullanarak mikro hizmetlere dayalı bir mimari olan MI-FIWARE tasarımı gerçekleştirmiştir. Mimari, bağlam aracı ile geliştirilen kullanıcı ara yüzü arasında bir mikro hizmet kullanmakta ve bağlam aracı tarafından gönderilen verileri önceden işleyerek kullanıcı ara yüzü tarafından yapılan istek ve abonelikleri yönetmektedir. Sousa vd. (2021) çalışmalarında IoT sistemlerinin güvenlik yeteneklerine ait provizyon, kimlik doğrulama ve erişim kontrolü gibi durumları analiz ederek, YubiAuthIoT ve Fiware bileşenlerini bir arada çalıştıran bir tasarım gerçekleştirmişlerdir. Loss vd. (2021) Covid19 pandemisi sonrası bir dijital aşılama kanıtı sunmak ve dünya çapında çoğu aşılama sistemini birbiri ile entegre etmek için bir çalışma gerçekleştirmiştir. Bu çalışmada, NGSİ standartlarını kullanarak bileşenlerin entegrasyonunu sağlayan ve Fiware platformu bileşenlerinden biri olan Canis Major adlı yeni bir genel etkinleştiriciden faydalanmışlardır. Entegre edilen sistemdeki verilerin güvenilir olması için merkezi olmayan veri kayıt defteri Blockchain'i, NGSİ ile birlikte kullanmışlardır. Bu sayede entegre aşılama sistemlerinin, pandeminin sona ermesine katkıda bulunmak için bir kişinin ne zaman ve ne tür bir aşı yaptırdığının güvenilir bir şekilde doğrulayabileceğini savunmuşlardır. Armando vd. (2022) çok sayıda cihaz içeren IoT sistemlerine ait kaynak yönetiminin verimli bir şekilde yapılması konusunu ele alarak Ultralight ve LwM2M IoT araçları ile farklı senaryolar üzerinde karşılaştırmalı bir değerlendirme gerçekleştirmiştir. Değerlendirme kriteri olan, ağ üzerindeki paket kaybı, teslimat gecikmesi ve bant genişliği

doluluk ölçümlerini analiz ederek Ultralight ve LwM2M IoT araçlarının avantaj ve dezavantajlarına yer vermişlerdir. Barriga vd. (2022) IoT sistemlerinin tasarımı, kod üretimi ve simülasyonu için Fiware ekosistemine ait altyapı üzerinde oluşturulan, SimulateIoT tabanlı akıllı bina ve tarımsal IoT ortamına ait senaryolardan oluşan bir çalışma gerçekleştirmiştir. Oluşturdukları Fiware IoT ortamı, docker konteyner kapsayıcılarında mikro hizmetler olarak dağıtılan ve hizmet odaklı bir mimari elde etmeye uygunluk oluşturan sensörler, aktüatörler, sis düğümleri, bulut düğümleri ve bu bileşenlere ait analitik özellikleri içermektedir. Conde vd. (2022) fiziksel ve sanal dünyalar arasındaki ilişkiden yararlanarak, endüstride kullanımı gün geçtikçe artan bir “Dijital İkiz” tasarımı gerçekleştirmek için Fiware ekosistemini kullanmıştır. Tasarımın gerçekleştirilmesinde tüm bileşenleriyle eksiksiz bir otopark kullanım örneği sunulmaktadır. Çalışma sonunda, Fiware ekosisteminin “Dijital İkiz” geliştirmek için gerçek bir referans seçeneği oluşturduğu, Fiware bileşenlerinin ve akıllı veri modellerinin, gerçek zamanlı toplu veri kullanımı ve işlenmesi, ölçeklenebilirlik, bulut bilişim, veri modelleme ve güvenlik gibi alanlarda, ilgili literatürde formüle edilen zorlukların üstesinden geldiği sonucuna ulaşılmaktadır. Hail vd. (2022) veri iletişimini sağlamak için IP adresleri yerine hiyerarşik adlar kullanan “Adlandırılmış Veri Ağı” iletişim modeli tabanlı IoT-NDN ile Fiware yazılım entegrasyonunu sağlayan bir mimari geliştirmiştir. Geliştirilen mimaride bağlam bilgilerini toplamak ve bu bağlamları bulut tabanlı IoT uygulamalarında yönetmek için NGSİ-v2, Context Broker, Cygnus, Keyrock, QuantumLeap, Draco, Cosmos, STH Comet gibi Fiware bileşenleri kullanılmıştır. Loss vd. (2022) Fiware ve bileşenlerinin, Blockchain teknolojisi ile tam anlamıyla bir entegrasyona sahip olmamasından, bu entegrasyonu sağlamak için bir dizi yeni bileşen üzerinde çalışmıştır. Tasarladıkları “Fiware Blockchain Adaptörü” yardımıyla Blockchain teknolojisini, Fiware ekosistemi üzerinden akıllı şehir uygulamaları senaryolarında kullanılabilir hale getirmişlerdir. Örnek senaryo olarak, küresel bir salgın sonrası aşılama ile ilgili Akıllı Şehir uygulaması gerçekleştirmişlerdir. Namba ve Nishi (2022) Fiware mimarisi içerisinde, veri paketlerindeki kişisel bilgilerle ilişkilendirilen veri akışlarının etiketlerini kullanarak uç noktada veri yollarını yöneten bir mekanizma oluşturmuştur. Sonrasında, oluşturdukları simülasyon ortamında kişisel bilgilerin toplanması için işlem süresi gecikmelerini ölçmüş ve bir karşılaştırma ile sonuçları değerlendirmişlerdir. Kalay vd. (2022) laboratuvar ortamında MQTT ve FIROS haberleşme protokollerini kullanan, Fiware ekosistemine dayalı olarak çalışan bir sistem tasarlamıştır. Bu sistem üzerinden toplanan zaman serisi sensör verilerini çalışmalarına

veri kaynağı olarak kullanmış, farklı konum ve oranlarda eksik veriler oluşturmuşlardır. Oluşturulan veriler üzerinde çeşitli makine öğrenmesi algoritmaları kullanarak eksik veri atama yöntemlerini incelemiş ve performansları üzerinde karşılaştırma gerçekleştirmişlerdir.

Fiware, IoT Ara katman ve Opc-UA ile ilgili olarak, 2012 yılından Haziran 2023'e kadar, her biri için ayrı ayrı yapılan akademik çalışmalar incelenerek gelişimleri ve kullanımları hakkında bir analiz gerçekleştirilmiştir. Yapılan bu analiz, "Web Of Science" ve "Scopus" akademik veri tabanlarında yer alan, "fiware" ve "fi-ware" anahtar kelimeleriyle ilişkili 2.402 yayını, "iot middleware" ve "iot-middleware" anahtar kelimeleriyle ilişkili 15.306 yayını, "opcua", "opc ua" ve "opc-ua" anahtar kelimeleriyle ilişkili 4.535 yayını olmak üzere toplamda 21.883 yayını kapsamaktadır. Şekil 2.4'de Fiware, IoT Ara katman ve Opc-UA ile ilgili olarak 2012 ve 2023 yılları arasında ortaya konan akademik çalışmaların yıllara göre dağılımı yer almaktadır.



Şekil 2.4. Fiware, IoT Middleware ve OPC-UA'nın yıllara göre dağıtılmış akademik çalışmaların sayıları.

Fiware platformu, IoT ara katman ve OPC-UA haberleşme protokolü ile ilgili olarak Şekil 2.4'de akademik çalışmaların yıllara göre dağılımında görüntülediği gibi, bu kavramları temel alan akademik çalışmalar gün geçtikçe artmaktadır. Bu

grafiklerden yapılacak bir çıkarım ise, teknolojinin gelişimi ve makineleşme ile birlikte nesnelerin interneti konusu üzerinde artan bir talep olduğudur. Tez çalışmamızda, Fiware, IoT Middleware ve OPC-UA kavramları ayrı ayrı incelenmekte ve bu kavramlar tek bir çatı altında toplanarak çalışma gerçekleştirilmektedir.

Fiware ara katman mimarisi ile tasarlanan ve literatürde yer verilen örnek yayınlara ait çalışmaların, yayın bilgileri, kullanılan Fiware ara katman mimarisi bileşenleri, uygulama alanları ve kullandıkları protokollere ait bilgiler Çizelge 2.3.' de görüntülenmektedir.

Çizelge 2.3. Fiware ara katman mimarisi kullanılan literatür örnekleri.

Referans	Çalışma	Fiware GE	Uygulama	Protokol
Martinez vd., 2016	A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture	Context Broker, Protocol Adapter, Data Handling, IoT Discovery, IoT Broker, Backend Device Management, Cosmos, Cygnus-ngsi	Akıllı Tarım	UltraLight 2.0 / HTTP, MQTT, LWM2M / COAP
Lopez-Riquelme vd., 2017	A software architecture based on FIWARE cloud for Precision Agriculture	Context Broker, Cosmos, Cygnus Injector, Mashup, Wirecloud	Akıllı Tarım	HTTP / REST
Rodriguez vd., 2018	FIWARE Open Source Standard Platform in Smart Farming - A Review	Context Broker, Cosmos, Cygnus, IDAS, KeyRock	Akıllı Tarım	UltraLight 2.0
Baca Gomez vd., 2019	A Novel Air Quality Monitoring Unit Using Cloudino and FIWARE Technologies	AirQuality-Observed, Orion Context Broker, QuantumLeap	Akıllı Şehir	HTTP / REST
Celesti vd., 2019	How to Develop IoT Cloud e-Health Systems Based on FIWARE: A Lesson Learnt	KeyRock, Wilma, TheObjectStorage, IaaS, Wirecloud, AuthZForce	e-sağlık	HTTP / REST, TCP, WebSocket

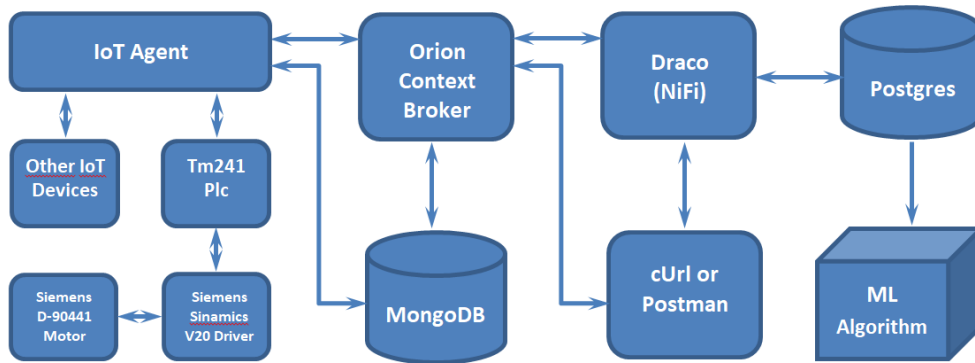
Çizelge 2.3. Fiware ara katman mimarisi kullanılan literatür örnekleri (devam).

Referans	Çalışma	Fiware GE	Uygulama	Protokol
Pozo vd., 2020	Evaluation of an IoT Application-Scoped Access Control Model over a Publish / Subscribe Architecture Based on FIWARE	Context Broker, Keyrock, Wilma PEP	Bağlam Kalitesi	HTTP, COAP, OAuth 2.0
Kalay vd., 2022	A Predictive Maintenance System Design and Implementation for Intelligent Manufacturing	Context Broker, MQTT IoT Agent, Grafana	-	HTTP, MQTT, FIROS
Villalon vd., 2022	An IoT architecture for smart cities based on the Fiware platform	Context Broker, Keyrock, Wilma, AuthZForce, Wirecloud, AeroAllergen-Observed, AirQuality-Observed, WaterQuality-Observed, NoiseLevel-Observed, QuantumLeap, CrateDb	Akıllı Şehir	HTTP, MQTT, TCP
Aizaga-Villon vd. 2022	FIWARE-Based Telemedicine Apps Modeling for Patients Data Management	Context Broker, Draco, Cygnuse	e-sağlık	HTTP / REST

Bu tez çalışmasında, literatürde yapılan diğer kestirimci bakım çalışmalarından farklı olarak; elektrik motoru test düzeneği üzerinden toplanan akım sensör verileri, önce OPC-UA ara yüzü ile açık kaynak kod olan Fiware ara katman mimarisine, sonra ara katmanın mevcut hat ile bağlı olduğu veri tabanına aktarılmış ve aktarılan veriler ile makine öğrenmesi algoritmaları kullanılarak bir anomali tespiti uygulaması gerçekleştirilmiştir.

### 3. MATERYAL VE YÖNTEM

Ara katman mimarisinin tasarımının gerçekleştirileceği test ortamının oluşturulması için öncelikle işlem basamakları sıralanmaktadır. Geliştirilen ara katman mimarisinde, veri kaynağı olarak Siemens Sinamics V20 sürücüyeye bağlı üç fazlı Siemens D-90441 elektrik motoru kullanılmaktadır. Sürücü sayesinde okunan elektrik motorunun akım ve frekans verileri, Schneider TM241CE40T Plc'ye aktarılmaktadır. Sürücü ve Plc arasında gerçekleştirilen veri aktarımı için seri haberleşme protokolü olan Modbus RTU kullanılmaktadır. Çalışmada, TM241CE40T Plc'nin kullanılma sebebi olan en önemli özelliği, içerisinde OPC-UA'ya ait bir sunucu barındırmasıdır. Bu sunucu sayesinde, OPC-UA haberleşme protokolü kullanılarak elektrik motorundan toplanan akım ve frekans verileri Fiware ara katman yazılımına aktarılmaktadır. Fiware bileşenlerinden biri olarak çalışan OPC-UA IoT aracısı, sunucunun yaptığı yayına abone olarak verileri almakta ve yönetilmesi için Fiware ana bileşenlerinden Context Broker'a göndermektedir. Aynı zamanda MongoDB üzerinde abonelik bilgileri ve değişkenlerin son değerleri saklanmaktadır. Elde edilen veriler, Orion Context Broker tarafından, PostgreSQL veritabanında saklanacak SQL veri tabanı formuna dönüştürülmek üzere Draco'ya gönderilmektedir. Draco gerekli veri dönüşümü ve yönlendirmelerini gerçekleştirerek, verilerin PostgreSQL veri tabanına aktarılmasını sağlamaktadır. Böylece veri tabanına aktarılan veriler, anomali tespiti için gerekli veri seti kaynağını oluşturmaktadır. Sıralanan işlem basamakları, Şekil 3.1'de görülen Ara katman mimarisinin geliştirilmesine ait blok diyagram üzerinde görüntülenmektedir.



Şekil 3.1. Arakatman mimarisinin geliştirilmesine ait blok diyagram.

Mimarinin geliştirilmesi sırasında kullanılan her bir bileşenine ait teknik yapı ve açıklamalarına materyal bölümünde, mimarinin gerçekleştirilmesinin detaylarına ise yöntem bölümünde yer verilmektedir.

### **3.1. Materyal**

Ara katman mimarisinin geliştirilmesinde kullanılan materyaller Yazılım materyalleri, Haberleşme protokolleri, Donanım materyalleri ve Makine öğrenmesi algoritmaları olmak üzere dört başlık altında incelenmektedir.

#### **3.1.1. Yazılım materyalleri**

Ara katman mimarisinin geliştirilmesi sürecinde Fiware ara katman bileşenlerine ait bütün programlar Linux tabanlı Ubuntu 20.04 işletim sisteminde çalıştırılan Docker konteyner mimarisi üzerine inşa edilmiştir. Orinon Context Broker, OPC-UA IoT Aracısı, MongoDB, Postman, Draco (Nifi), PostgreSQL uygulamalarının tümü konteyner olarak entegre edilmiştir. Konteyner yönetiminde ise Portainer.io kullanılarak Docker konteyner takibi, analizi ve yönetimi kolaylaştırılmıştır.

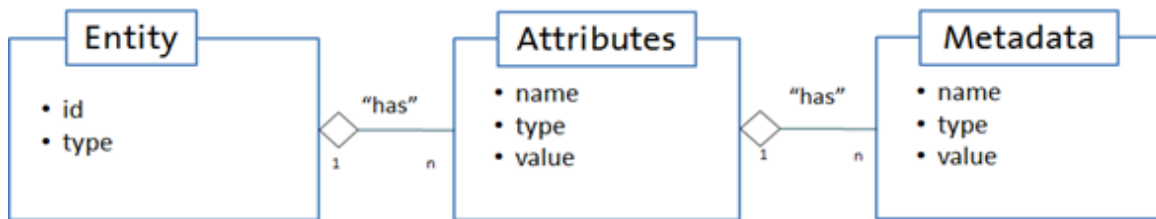
##### **3.1.1.1. Fiware ara katman yazılımı**

Ara katmanlar, birbirleri ile iletişim halinde bulunan platformlar arasında iletişimi gerçekleştirmek amacıyla çalıştıkları için kendi başlarına her hangi bir işlem gerçekleştiremezler. Nesnelerin interneti için geliştirilen sistemlerde veri iletişiminin sağlanması için birçok veri protokolü kullanılmış ve bu protokollerin yeteneklerine uygun birçok ara katman yazılımı geliştirilmiştir (Özdoğan ve Erdem 2020). Ara katman yazılımları genel olarak dağıtılmış sistemlerde işletim sistemleri, donanım bileşenleri ve haberleşme protokolleri arasında bir köprü işlevi gören ara yazılımlardır (Karasoy ve Çınar 2014).

Fiware ara katman yazılımı, farklı sektörlere yönelik akıllı uygulamaların geliştirilmesini kolaylaştırmak için telif hakkı olmayan ve uygulama odaklı yazılım platformu standartları ile şekillenmiş açık ve sürdürülebilir bir ekosistemdir. Fiware

ekosistemi; özellikle IoT alanlarında çalışan çok sayıda üretici ve servis sağlayıcı firmalar, girişimci bireyler, KOBİ'ler, yeni girişimler ve yeni yatırımcılar gibi içerisinde birçok paydaşı barındıran Fiware Topluluğu'nun ortak çalışmasının bir ürünüdür. Fiware, uygulamalar arasında veri alışverişini kolaylaştırmak için referans bağlam modeliyle ilişkili olarak tanımlanan veri modelleri sağlar. Fiware'in standart NGSI v2'si, güncellemeler, sorgular, kayıtlar ve abonelikler dahil olmak üzere bağlam bilgilerinin tüm yaşam döngüsünü yönetmeyi amaçlar (Gomez vd., 2019).

Orion Context Broker, ortamdan gelen verileri kullanan uygulamalar oluşturmak için yaygın olarak kullanılan bir Fiware bileşenidir. Context Broker'ın ana konsepti, veri bağlamı üreticisinin bilgi üretebilmesi ve üretilen verileri kullanacak kullanıcı veya uygulama hakkında önceden bilgi sahibi olmadan buluta yerleştirebilmesidir. Standart REST API işlevlerini uygulayarak bağlam bilgilerini büyük ölçekte işleyebilmektedir. Böylece geliştiricilerin güncellemeler, sorgular, kayıtlar ve abonelikler dâhil olmak üzere bağlam bilgilerinin tüm yaşam döngüsünü yönetmesini sağlamaktadır (Fiware - Orion, 2023). Context Broker, NGSI REST API'nin bir uygulamasıdır. Bulutta paylaşımına izin vermek ve farklı uygulamalar arasında birlikte çalışabilirliği sağlamak için bu bileşene gönderilen tüm bilgilerin NGSI formatında temsil edilmesi gerekir. NGSI veri modelindeki ana öğeler, Şekil 3.2'de gösterildiği gibi bağlam varlıkları, nitelikler ve meta verilerdir.



Şekil 3.2. NGSI veri modeli ana öğeleri (Fonseca & Marquez, 2018).

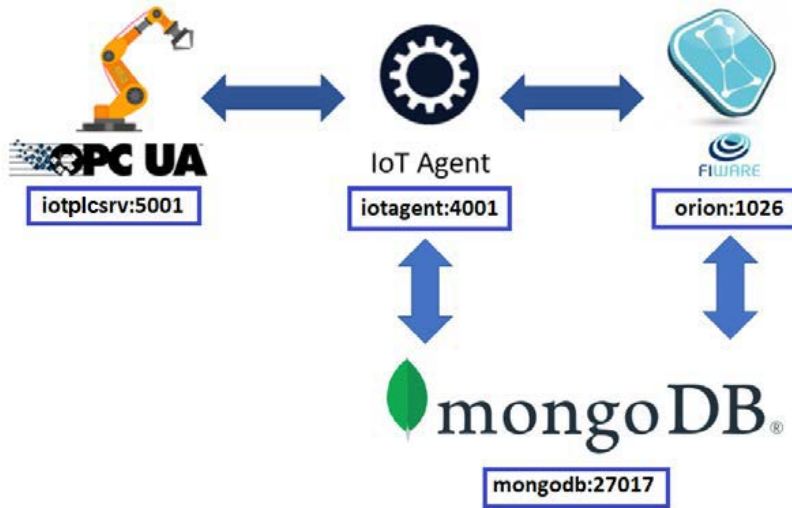
Bağlam varlıkları fiziksel yada mantıksal bir nesneyi temsil etmektedir. Her varlığa ait bir varlık kimliği ve varlık tipi bulunur. Bağlam nitelikleri, bağlam varlıklarının özellikleridir. NGSI veri modelinde, bağlam niteliklerinin bir nitelik adı, nitelik türü ve nitelik değeri vardır. Meta veriler, niteliklerin içerisinde yer alabileceği gibi ayrı bir veri parçası olarak da kullanılabilir. Oluşturulduğu yerde meta verinin rolünü, meta veri adını ve meta verinin tipini açıklamak için kullanılır (Fonseca & Marquez, 2018).

Fiware ekosistemi, oluşturulacak projelerin ihtiyaçlarını karşılayabilmek, özel hibrit platformlar oluşturmak ve projelerin daha hızlı çalışmasını sağlamak için Genel Etkinleştiriciler (GE) olarak da bilinen açık kaynaklı platform bileşenlerinin derlenmiş bir çerçevesidir. (Kalay vd, 2022) Herhangi bir Fiware tabanlı sistemin veya çalışmanın olmazsa olmaz tek zorunlu bileşeni, herhangi bir akıllı çözümde ana unsur işlevi gören Fiware Context Broker GE'dir.

### 3.1.1.2. OPC-UA IoT aracı

OPC-UA Aracı, üretim alanında bulunan OPC-UA cihazlarından veri yakalamayı ve bu verileri Fiware tabanlı bir sistemin üst seviyelerine göndermeyi amaçlayan, açık kaynak olarak dağıtılan bir Fiware bileşenidir. OPC-UA IoT aracısının hedeflenen amacı, OPC-UA sunucu yayını yapan saha cihazlarından, Fiware ekosistemine doğru veri akışını sağlayarak Orion Context Broker ile iletişimi kurmaktır (Fiware - OPC UA Agent, 2023).

Şekil 3.3'de OPC-UA IoT aracısının sistem üzerindeki yerini gösteren görsele yer verilmiştir.



Şekil 3.3. OPC-UA IoT Aracısının sistem üzerindeki yeri ve görevi (Fiware – OPC-UA Agent, 2023).

Çalışmamızda bu IoT Aracı kullanılarak, Plc üzerinde akım ve frekans değişkenlerini yayınlayan OPC-UA sunucusunu, Orion Context Broker'a bağlayarak bu verilerin Fiware ortamına aktarımı sağlanmaktadır. OPC-UA aracı verilerinde de, NGSI veri modeli kullanıldığı için Orion Context Broker'da otomatik olarak yayınlanmaktadır.

### 3.1.1.3. MongoDb

MongoDB, C++ programlama dili ile geliştirilmiş, ölçeklenebilen ve doküman tabanına dayanan açık kaynak kodlu bir NoSQL veritabanı uygulaması olarak karşımıza çıkmaktadır. Özellikle hızlı sonuçlar alınması gereken ve geleneksel veritabanı yönetim sistemlerinin yavaş kaldığı durumlarda tercih edilmektedir. MongoDB'ye ait kullanım alanları arasında, yüksek hacimli problemler, analiz için veri depolama, veri önbellekleme, web tabanlı içerik yönetim sistemleri gibi alanlar bulunmaktadır (Eken vd., 2014).

### 3.1.1.4. Postman

Postman, API testlerinde kullanılan en popüler test araçlardan biridir. Uygulama geliştiricilerinin kolayca API oluşturmasına, paylaşmasına, test etmesine ve belgelemesine olanak tanıyan çok kullanışlı bir uygulamadır. Geliştiriciler uygulama içinde geliştirdikleri kaynak kodların kapsamını artırmasına olanak sağlayan test senaryoları oluşturabilir ve HTTP isteklerini oluşturarak kaydedebilir veya yanıtlarını okuyabilirler (Postman, 2023). Postman uygulaması içinde oturum açılabilmesi ve çalışmaların bulut ortamında saklanabilmesi sayesinde, çalışmalara her yerden erişim sağlanabilmektedir (Shershneu ve Oskin, 2020). Uygulama, hızlı ve kolay bir şekilde kullanılabilmesi sebebi ile tercih edilmiştir.

### 3.1.1.5. Draco

Draco, verileri işlemek ve dağıtmak için kullanılan, Apache NiFi'ye dayalı olarak geliştirilen, akış tabanlı programlama kavramlarını temel alan güvenilir bir veri akışı sistemidir. Verilerin, yönlendirilme, dönüştürülme ve sistemsel ara buluculuk mantığını, güçlü ve ölçeklenebilir yönlendirilmiş grafikleri ile destekler (Velasquez vd., 2021). Sistemler arasındaki veri akışını otomatikleştirmek için oluşturulmuştur. Draco bildirim verilerini Carto, Cassandra, HDFS, Mongo, MySQL, PostgreSQL gibi farklı veri tabanlarına kaydedebilme yeteneğine sahiptir.

Draco, kendisi için belirlenmiş bağlantı noktası aracılığı ile Context Broker'dan gelen bildirim verilerini sürekli olarak takip etmektedir. Bir veri bildirimi alındığında,

NGSI verilerini PostgreSQL'e dönüştüren bir sonraki işlemciye ileterek kaydetmektedir. PostgreSQL'e kayıt işlemi, ilk olarak verilerin hangi şemaya kaydedileceğine karar verilmesi ile gerçekleşmektedir. Bu işlem, bildirimlerin 'fiware-service' başlığı tarafından yapılmakta, şemaya karar verme işlemi tamamlandıktan sonra 'fiware-servicepath' başlığına göre hangi tabloya kaydedileceğine karar verilmektedir (Fiware - Draco, 2023). Draco, kullanıcıların veri akışlarını kolayca yönetebilmesi için varsayılan olarak 9090 numaralı bağlantı noktasında çalışan bir kullanıcı ara yüzü sağlamaktadır.

### **3.1.1.6. PostgreSQL**

PostgreSQL, SQL programlama dilini kullanan bir veri tabanı yönetim sistemidir. Bu sistem, karmaşıklık seviyesi yüksek verilere ait çalışmalarını güvenli bir şekilde depolayabilen ve aynı oranda ölçeklendirebilen açık kaynak kodlu bir yapıya sahip, nesne ilişkili bir veritabanı sistemidir (Postgresql, 2023). PostgreSQL veritabanı yönetim sistemi, MySQL ve diğer açık kaynak kod sistemine sahip veritabanı yönetim sistemleri ile karşılaştırıldığında, mimari yapısının getirdiği özellikler sayesinde daha iyi bir performans göstermektedir.

PostgreSql veri tabanının görüntülenmesi ve yönetilmesi için PgAdmin uygulamasından faydalanılmaktadır. PgAdmin, PostgreSql veri tabanı yönetimi için kullanılan açık kaynak yönetim ve geliştirme platformu olarak bilinmektedir.

### **3.1.1.7. SoMachine**

Schneider Electric'in geliştirdiği SoMachine programlama yazılımı, mantık denetleyicilerinden yürütülen projeleri geliştirmek için tasarlanmış bir yazılımdır. SoMachine, lojik, hareket kontrolü, HMI ve ilgili ağ otomasyon fonksiyonları da dâhil olmak üzere tek bir yazılım ortamında makinenin tamamını geliştirmeye, yapılandırmaya ve devreye almaya yönelik makine üreticisi için tasarlanmış çözüm yazılımıdır (Schneider Electric, 2023a). Schneider Electric firması yazılımın EcoStruxure Machine Expert adı ile yeni bir sürümünü çıkarmasına rağmen, kullanım kolaylığı ve lisans işlemleri sebebi ile eski sürüm olan SoMachine V4.3 yazılımı kullanılmaktadır.

### 3.1.1.8. Vijeo Designer

SoMachine yazılımı ile birlikte de gelen bu yazılım, operatörlerin kullanacağı paneller oluşturmak ve insan/makine etkileşimi için geliştirilecek ara yüz aygıtlarında çalışma parametrelerini konfigüre etmek için kullanılabilen bir uygulamadır. Verilerin kaynaktan okunması, tasarım görsellerinin oluşturulması ve bu görsellerin görüntülenmesi gibi çalışmaların gerçekleştirilmesinde kullanılacak olan tüm araçları sağlamaktadır (Schneider Electric, 2023b). Schneider Electric firmasının geliştirdiği HMIGXU3500 dokunmatik panel ekranın görsel tasarımı, programlanması ve iletişim parametrelerinin yapılandırılması sırasında, Vijeo Designer V1.1 yazılımı kullanılmaktadır.

### 3.1.2. Haberleşme protokolleri

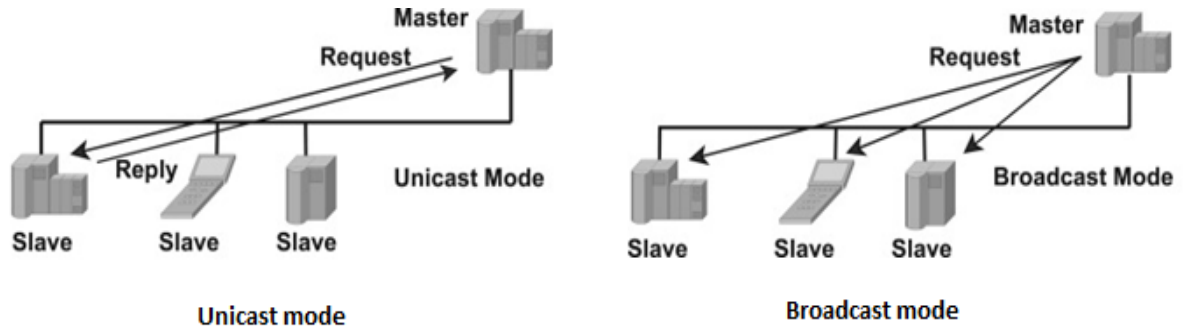
Ara katman mimarisinin geliştirilmesinde iki farklı haberleşme protokolünden faydalanılmıştır. Modbus RTU seri haberleşme protokolü ve endüstri alanında sıkça kullanılmaya başlanılan OPC-UA haberleşme protokolüne ait teorik bilgiler bu başlık altında incelenmektedir.

#### 3.1.2.1. Modbus RTU

Sürücü ve Plc arasında iletişim kurmak için, iki donanımında sahip olduğu Modbus RTU haberleşme protokolü kullanılmıştır. Modicon firması tarafından geliştirilen ve açık erişim olan bu protokol endüstride en çok kullanılan haberleşme protokollerinden birisidir (Çetin ve Yörükeren, 2020). Modbus RTU protokolü, master/slave ya da sunucu/istemci ilişkisi bulunan sistemlerin birbiri ile haberleşmesinde kullanılan seri haberleşme protokolüdür. Modbus RTU iletişim kanalına sahip bir ağ üzerinde bir adet sunucu ve 247 adet istemci aygıt çalıştırılabilmektedir (Erdoğan vd., 2020).

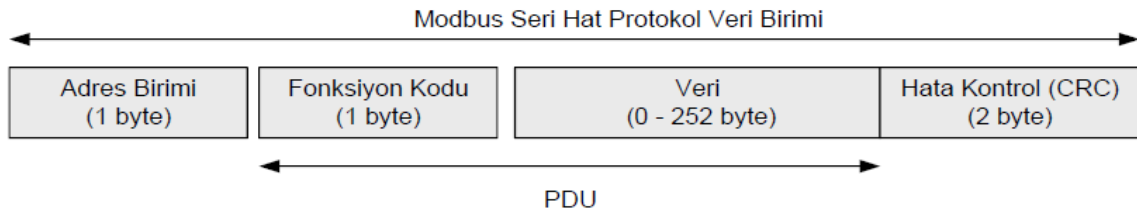
Tek bir noktaya yayın veya çoklu yayın modları aracılığıyla bir veya daha fazla alıcıya istek mesajı göndererek yalnızca birim iletişim başlatabilir. Tek bir noktaya yayın modunda Master, tek bir slave'e bir istek gönderir ve istek işlendikten sonra, alıcı yöneticiye bir yanıt mesajı gönderir. Çoklu yayın modunda ise Master, yayın adresini kullanarak tüm alıcılara aynı anda bir istek gönderir ve her alıcı isteği işler, ancak

yöneticiye hiçbir yanıt mesajı göndermez (Urrea vd., 2016). Şekil 3.4’ de Sunucu ve istemcilere ait tekli ve çoklu yayın modlarındaki iletişime ait görsele yer verilmektedir.



Şekil 3.4. Sunucu ve istemcilere ait tekli ve çoklu yayın modları (Urrea vd., 2016).

Veriler ikişer bitlik dizeler halinde yollanmaktadır ve iletilen mesajların güvenli bir şekilde gönderilmesi için 16 bitlik CRC hata kontrol mekanizması kullanılmaktadır. Bu sayede hem basitlik hem de güvenilirlik sağlanmaktadır (Akkaya vd., 2015). Şekil 3.5’de Modbus RTU’ya ait protokol veri yapısı görüntülenmektedir.



Şekil 3.5. Modbus RTU protokol Veri Yapısı (Akkaya vd., 2015).

Modbus RTU'da RS-232, RS-422 veya RS-485 fiziksel katmanları kullanılabilir. RS-232, aynı anda veri iletişimi sağlayan dupleks topoloji iken, RS-485, iletişim sırasında birbirini ardına yapılan yarı çift yönlü iletişim sağlayan bir topolojidir. Modbus RTU fonksiyon kodu 32 bit kayan noktalı olup tamsayı formatı kullanılmaktadır. Modbus RTU'da master cihazlar, slave cihazlara sorgu gönderir ve slave cihazlar, belirtilen koda göre master sorgusuna yanıt gönderir (Tamboli vd., 2015).

### **3.1.2.2. OPC-UA**

OPC-UA (Open Platform Communications Unified Architecture - Açık Platform İletişim Birleşik Mimarisi), temel olarak endüstriyel otomasyonda kullanılan, kontrol donanımı ve saha seviyesi cihazlarla etkileşim için bir standarda sahip olan, hizmet odaklı bir makineden makineye iletişim protokolüdür (Profanter vd., 2019).

Microsoft Windows platformuna dayalı OLE, COM ve DCOM teknoloji bağımlılığına tepki olarak oluşmuştur. OPC-UA, çeşitli ağlar üzerinden istemciler ve sunucular arasında mesajlar göndererek çeşitli sistem ve cihazların iletişim kurabildiği platformdan bağımsız bir standarttır (Lee vd., 2017). OPC-UA, aktarılan verileri açıklamak için bir bilgi modeli kullanırken, platformlar arası bir iletişim kanalı oluşturmaktadır. İstemcilerin ve sunucuların kimliğini garanti eden, sağlam ve güvenli iletişimi destekleyen, çeşitli saldırı türlerine karşı dirençli olduğu düşünülen bir protokoldür. Bu nitelikler ile tesis içi esnek ve sağlam iletişim sağladığından, Endüstriyel IoT alanında kullanılan sistemlere ait iletişim için daha fazla önem kazanarak dünya çapında en önemli protokollerden biri haline gelmiştir. OPC UA'nın başlıca gücü, adres alanı modelinin anlamsal açıklamasıdır. PLCopen, robotik veya bilgisayar görüşü gibi çeşitli alanlar için temel anlamsal açıklamaları genişleten çeşitli spesifikasyonları içermektedir. OPC-UA tarafından hizmete sunulan Yayınla/Abone Ol özelliği ile sunucular verileri yayınlatabilir ve istemciler veri kaynağından bağımsız olarak bu verilere abone olabilir (Profanter vd., 2019).

### **3.1.3. Donanım materyalleri**

Ara katman mimarisinin geliştirilmesinde, donanım bileşenleri olarak kullanılan Elektrik motoru, Sürücü, Plc ve HMI dokunmatik panel ekrana ait teorik bilgiler bu başlık altında incelenmektedir.

#### **3.1.3.1. Elektrik motoru**

Mimarinin veri kaynağı olarak Siemens D-90441 1LE10031AC422AA4 elektrik motoru kullanılmaktadır. Motor 3 faz olarak 1,5 Kw gücünde 50 Hz frekansta

çalışmaktadır (Siemens, 2023a). Test ortamında kullanılan elektrik motorunun üzerinde sağlam rulmanlar dışında laboratuvar ortamında bilinçli olarak oluşturulmuş farklı hatalara sahip rulmanlarda kullanılmaktadır. Şekil 3.6'da Eskişehir Osmangazi Üniversitesi Akıllı Fabrika ve Robotik Laboratuvarında (IFARLAB, 2023) laboratuvar ortamında oluşturulan elektrik motoru test düzeneği görülmektedir.



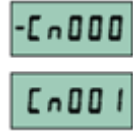
Şekil 3.6. IFARLAB elektrik motoru test düzeneği (IFARLAB,2023).

### 3.1.3.2. Sürücü

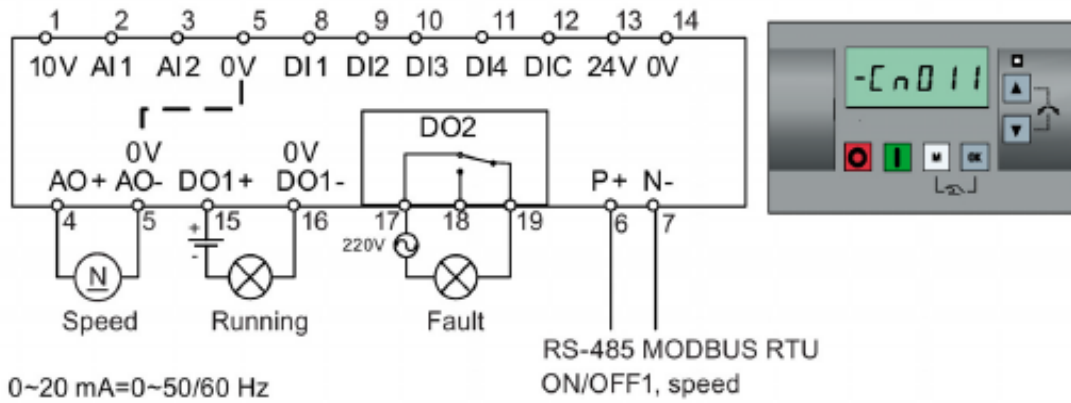
Elektrik motorunun çalışma kontrollerinin yapılması ve motora ait her türlü verinin okunması için Siemens 6SL3210-5BE27-5UV0 Sinamics V20 sürücü kullanılmaktadır. Endsütri ve basit kullanım senaryolarında kullanılan ve 3 faz çalışan bu sürücü 7.5 Kw nominal güce sahiptir. Siemens markasının V20 serisine ait olan bu sürücü, fan-pompa uygulamaları, kompresörler, genel makina uygulamaları, taşıma bandı sistemleri, makaralı ve zincirli konveyörler, değirmen, mikser, yoğurma makinası, vibrasyon gibi bir çok makine ve endüstri alanında hız kontrol birimi olarak kullanılmaktadır. Kullanılacak parametreler yardımıyla standart, genişletilmiş, uzman ve bakım olmak üzere, kullanıcılar tarafından erişim sağlanacak dört seviyeye sahiptir. Sürücüye ilk enerji verildiğinde sırasıyla; motor bilgileri, kontrol makrosu, uygulama makrosu ve temel ayarlar ile ilgili

yapılandırmaların yapılarak sürücünün devreye alınması gerekmektedir. Dâhili olarak USS ve Modbus RTU haberleşme kaynaklarını içerisinde barındırmaktadır. Kullanım alanına ve çalışmaya göre Çizelge 3.1’de görüntülenen kontrol makrolarından seçim yapılarak sürücü o makro ile çalıştırılabilmektedir. Çalışmamız kapsamında, sürücünün dâhili olarak sahip olduğu seri haberleşme protokolü olan Modbus RTU sayesinde Plc ile iletişim kurulmaktadır (Siemens, 2023b).

Çizelge 3.1.3.2 V20 sürücü kontrol makroları (Siemens, 2023c).

Makro	Açıklama	Panel Örnek Gösterim
Cn000	Fabrika Ayarı	 <p>Kontrol makrosunun yanındaki “-“ işareti seçilen makro oluşunu gösterir.</p>
Cn001	Temel Operatör Paneli ile Kontrol (BOP Kontrol)	
Cn002	Terminal Kontrol	
Cn003	Sabit Frekans Çalışma	
Cn004	Sabit Frekans Binary Çalışma	
Cn005	Sabit ve Analog Giriş ile Çalışma	
Cn006	Harici Push Butonlar ile Çalışma	
Cn007	Harici Push Butonlar ve Analog Giriş ile Çalışma	
Cn008	PID Kontrol (Setpoint Analog Giriş ile)	
Cn009	PID Kontrol (Sabit Değer ile)	
Cn010	USS Kontrol	
Cn011	Modbus Kontrol	

Sürücüye ait bağlantı makrolarının her birinde kullanılan bağlantı şekilleri ve şemaları farklılık göstermektedir. Kontrol makrosunu seçimi, sürücünün nasıl kontrol edileceği bilgisini sağlamaktadır. Kullanılacak olan Modbus RTU haberleşme için seçilmesi ve kullanılması gereken kontrol makrosu “Cn011 – Modbus Kontrol” olarak ayarlanmakta ve sürücünün üzerinde bulunan soket bağlantıları Şekil 3.7’deki örnek görselde yer verilen bağlantı şemasına göre tamamlanmaktadır. (Siemens, 2023c).



Şekil 3.7. Modbus makrosu sürücü bağlantı şeması (Siemens, 2023c).

### 3.1.3.3. HMI dokunmatik panel ekran

Test düzeneğimiz üzerindeki anlık değişkenleri okumak ve sürücü aracılığı ile elektrik motoruna gerekli parametrelerin gönderilmesi için Schneider HMIGXU3500 7" Dokunmatik Panel bir ekran kullanılmaktadır. Dokunmatik panel ekran, Magelis Easy GXU serisinin bir üyesidir. 7 inç LCD ekrana ve 800x480 piksel WVGA çözünürlüğe sahiptir. Uygulama için 32MB dahili bellek, 128MB dahili DDR ve 128kB dahili yedekleme belleği ile kullanıma sunulmaktadır. 20000 saatlik arka ışık ömrü ile 65536 rengi destekler. Nominal besleme voltajı 24V DC olan bir dokunmatik paneldir. RS422/RS485 COM1 bağlantısı kurulabilmesi için 9 pin erkek D-Sub konnektör ve USB tip mini bağlantıya sahiptir. Tekstil, paketleme, HVAC ve pompalama uygulamalarında kullanım için uygun görülmektedir (Schneider Electric, 2023d).

Ekranın, Plc'ye bağlantıları seri bağlantı olan RS485 bağlantı ile yapılmaktadır. Ekranın ait USB arayüzü üzerinden çalışan tasarım programı Vijeo Designer aracılığıyla, ekranın görselleri ve değişkenleri ihtiyaç duyulduğu şekilde tasarlanabilmektedir.

### 3.1.3.4. Plc

Donanım bileşenlerinin yönetimini üstlenmesi için test düzeneğinde Schneider TM241CE40T Plc kullanılmaktadır. Dâhili dijital giriş ve çıkışlara sahip olan bu plc, üzerinde 24 giriş ve 16 çıkış barındırmaktadır. 24 Volt DC gerilim ile çalışmaktadır. Modbus, Web, FTP, CANopen, Ethernet/IP bağlantılarını destekleyen Plc üzerinde iki adet Seri Port girişi, bir adet Usb ve bir adet Ethernet girişi bulunmaktadır (Schneider Electric,

2023c). İşlem gücü yüksek bir CPU ile kullanıma sunulan Plc, kartuş yuvaları sayesinde istenilen genişleme kartuşlarının takılabilesine olanak sağlamakta ve böylece istenilen modüllerin entegre edilmesi kolaylaştırılmaktadır. Üzerinde bulunan dâhili SD kart yuvası sayesinde, harici bir depolama birimi bulunmaktadır. Teknik altyapısının uygunluğu ile fieldbus mimarileri için 63 cihaza kadar dâhili CANopen özelliği sunabilmektedir. Yapılandırılabilen Ethernet portu sayesinde uzaktan erişime açılabilmekte, internet üzerinden hem konfigürasyon hem yönetim gerçekleştirilebilmektedir. 8 MB program ve 64 MB sistem belleğine sahiptir. Bağlantıların maksimum sayısı ise, 8 Modbus sunucusu, 8 SoMachine protokolü, 10 web sunucusu, 4 FTP sunucusu, 16 Ethernet/IP hedefi ve 8 Modbus istemcisi olarak sıralanabilmektedir (Schneider Electric, 2023c).

### 3.1.4. Makine öğrenmesi algoritmaları

Teknolojinin hızlı gelişimi ve takibinin zor olması belirli nesne ve kavramların kabul edilebilir doğruluklarla tanımlanmasında zorluklar yaratmaktadır. Bu zorlukların giderilmesi için, nesne ve kavramlar örneklerle birlikte makinelere aktarılmaktadır. Makinere aktarılan bu örneklerin bilgisayarlar aracılığı ile bilgiye dönüştürülebilmesi gerekmektedir. İşte bu dönüşümlerin yapılması ve örneklerin bilgisayarlar tarafından yorumlanması işlemlerinde makine öğrenmesi algoritmaları devreye girmektedir (Dündar vd., 2021).

Bu çalışma özelinde test düzeneği üzerinden toplanan akım verileri üzerinde yapılacak olan anomali tespitinde Gradyan Artırma Makinesi (GBM), Rasgele Orman (RF), Karar Ağaçları (DT), Çok Katmanlı Algılayıcılar (MLP), K-En Yakın Komşuluk (K-nN), Hafif Gradyan Artırma Makinesi (LGBM), Lojistik regresyon (LR) ve Kategorik Artırma (CatBoost) makine öğrenmesi algoritmaları kullanılarak bir değerlendirme yapılmaktave her makine öğrenmesi algoritmasına ait sonuca yer verilmektedir.

#### 3.1.4.1. Rasgele Orman (RF)

Rastgele Orman algoritmaları, karar ağacı temelli topluluk öğrenme algoritmalarıdır ve torbalama tekniğine dayanır. Bu algoritmada, sınıflandırıcıların bir orman oluşturduğu ve her sınıflandırıcının bir karar ağacı olduğu kabul edilir (Duman ve Başaraner, 2022). Her bir düğümü, özniteliklerin tamamıyla bölerek en iyi dalı belirlemek

yerine, her düğümde rastgele bir öznelik seçimi yaparak ayrı karar ağaçları oluşturulur. Sınıflandırma esnasında her karar ağacı, eğitimde kullanılmayan örneklere ait sınıflandırmayı değerlendirir. Böylece en yüksek değere sahip olan sınıf seçimi yapılarak sonuç döndürülür (Alan ve Karabatak, 2020).

### **3.1.4.2. Karar Ağaçları (DT)**

Karar ağaçları, sınıflandırma çalışmalarında yüksek kullanım oranına sahip bir algoritmadır. Genellikle sınıflandırma ve regresyon ağaçlarını temsil etmek için kullanılan karar ağaçları, bağımsız bir öğrenme yöntemi türüdür. Karar ağaçlarının kullanılmasındaki en büyük avantaj, oluşturulmasının ve yorumlanmasının kolay olmasıdır. Mimarisi incelendiğinde gerçek bir ağaçta bulunduğu gibi kök düğüm, dal ve yapraklardan oluştuğu görülmektedir. Karar ağaçları, sınıflandırma yöntemleri, özellikler ve hedefler doğrultusunda, karar düğümleri ile yaprak düğümlerinden meydana gelen, ağaç yapısı formunda bir model oluşturur. Özellik veya karar düğümü seçimi, her özelliğin sıralamasını belirleyerek, eğitim verilerinden elde edilen başlıkları tanımlar ve hangi özelliğin seçileceğini belirler. Veri kümesi içerisinde birden fazla ağaç elde edilebilmesine rağmen, boyutu en küçük olan ağaç tercih edilir. Yapraklardaki tüm elemanlar aynı sınıfta yer aldığı ve sınıflandırılacak değer kalmadığında, karar ağacı modeli tamamlanır ve döngü sona erer (Özlüer Başer vd., 2021).

### **3.1.4.3. Gradyan Artırma Makinesi (GBM)**

Gradyan Artırma, regresyon ve sınıflandırma problemlerinde tahmin edilen değerler ile gözlenen değerler arasındaki uyumu iyileştirmek için kullanılan, artırma tekniğine dayalı, karar ağacı tabanına sahip topluluk öğrenme algoritmalarından biridir. (Tütüncü ve Gürsakal, 2023). Bu algoritma, tahminlerini rastgele orman algoritmasından farklı bir teknik kullanarak gerçekleştirir. Bu işlemlerdeki tahminlere ulaşırken, eş zamanlı bir şekilde ağaç oluşturmak yerine birbiri ardına oluşturulan sıralı bir yaklaşım benimseyerek gerçekleştirmektedir. Oluşturulan model aracılığıyla elde edilen hata veya gerçek değer ile tahmin değeri arasındaki fark “Gradyan” olarak tanımlanmaktadır. Gradyan Artırma yönteminde, oluşturulan her bir karar ağacı kendinden bir önce bulunan

karar ağacının hatasını tahmin etmektedir. Bu yaklaşım sayesinde, hatalar aşamalı olarak azaltılarak model tamamlanmaktadır (Saygın ve Baykara, 2021).

#### **3.1.4.4. Hafif Gradyan Artırma Makinesi (LGBM)**

XGBoost'un eğitim süresi performansını iyileştirmek için geliştirilen Gradyan Artırma Makinesi türlerinden biridir (Saygın ve Baykara, 2021). Hafif Gradyan Artırma Makinesi algoritmaları büyük boyutlu verileri rahatlıkla işleyebilen ve yüksek seviyeli bir şekilde uygun hale getirebilen karar ağacı öğrenme algoritmalarıdır. Bu algoritmadaki karar ağaçlarına ait eğitim süreleri, yapılan hesaplama ve bölünme sayısı ile doğru orantılıdır. Bu sayede eğitim süresi ve kaynak kullanımı azaltılmaktadır. Algoritma, karar ağacını en uygun şekilde yaprak bilgisine bölerken, diğer artırma algoritmaları karar ağacını derinlik veya seviye bilgisine bölmektedir. Hafif Gradyan Artırma Makinesi aynı yaprak üzerinde büyüdüğünden, oluşacak kaybı, yaprak bilgisini temel alan algoritmalarından ve seviye bilgisini temel alan algoritmalarından daha fazla azaltmaktadır. Bu sayede diğer artırma algoritmalarının herhangi birinden elde edilebilecek değerlere kıyasla daha iyi doğruluk değerlerine ulaşabilmektedir (Khandelval, 2017).

#### **3.1.4.5. K-En Yakın Komşuluk (K-nN)**

K-En Yakın Komşuluk algoritması çalışma başlangıcında veri kümesini ilk önce eğitim ve test verisi olarak ikiye ayırmaktadır. Test verisi kümesine ait bir bulgunun sınıflandırılmasında tanımlama boşluğunda incelenen bulgunun, eğitim verisi kümesine ait bulguların her birisine ait uzaklığı ayrı ayrı hesaplanmaktadır. Bu işlem yapılırken sınıflandırılması planlanan yeni bulgunun daha önceki bulgulardan K tanesine olan yakınlığına bakılmaktadır. En yakın komşularının hangi sınıflara dâhil oldukları, daha önceden belirlenen bir K değeri kadar incelenerek, bulgu sayısının en fazla olduğu sınıf hangisi ise, incelenen örnek o veri sınıfına dâhil edilmektedir. K değerinin 1 olması halinde en yakın komşusuna ait veri sınıfına eklenmektedir. K değerinin belirlenmesi, sonuca ulaşılmasında önemli bir rol oynamaktadır. K değerinin çift olması, bulgunun tüm sınıflara aynı yakınlıkta olması durumunu veya eşitlik durumunu oluşturabilmektedir. Bu sebeple, K değeri genelde tek sayı olacak şekilde belirlenmektedir. Daha önce gerçekleştirilen

çalışmalarda, K değeri olarak 1, 3 ve 5 gibi değerler atandığında, başarı düzeyi yüksek sonuçlara ulaşıldığı görüntülenmiştir. (Dudoit vd., 2002).

#### **3.1.4.6. Kategorik Artırma (CB)**

Kategorik Artırma Algoritması, Yandex firmasında çalışan mühendislerin geliştirdikleri açık kaynak kod sistemine sahip bir algoritmadır (Sevli, 2022). Algoritma, veri kümesi içinde yer alan kategorik öznitelikleri işleyen, az veri ile de başarılı sonuçlar elde eden Gradyan Arttırma Karar Ağacı algoritmasıdır. Kategorik verileri otomatik olarak işleyebilen, kullanımı kolay ve yüksek performansa sahip bir algoritmadır. Gradyan Arttırma Karar Ağacı algoritmalarında genel olarak, kategorik özniteliklerine ait ön işleme süreleri değerlendirilirken bu algoritmada farklı olarak eğitim süresi değerlendirilmektedir (Kuş vd, 2021).

#### **3.1.4.7. Lojistik Regresyon (LR)**

Lojistik regresyon algoritması, üzerinde çalışılan veri setindeki gözlemlerin gruplara atanması için kullanılan, analize dayalı bir yöntemdir. Kaç sınıfa sahip olduğu bilinen algoritmada, mevcut veriler kullanılarak sınıflandırmaya ait bir model oluşturulmaktadır. Bu sayede verilere eklenmesi planlanan yeni gözlemlerin, belirlenen sınıflara atanması gerçekleştirilebilmektedir. Ayrıca, bu algoritma, bağımsız değişkenler ile iki veya daha fazla kategoriye sahip sınıf değişkeni arasındaki neden-sonuç ilişkisine ait derecelendirmenin gerçekleştirildiği bir yöntemdir. Burada bağımlı ve bağımsız değişkenler arasındaki ilişkinin önem derecesi analiz edilerek, üzerinde çalışılacak değişkene ait tahmin değerleri ile ulaşılan değerler arasında karşılaştırılma yapılmaktadır. Üzerinde çalışılan değişkenin öğrenme modeli içinde bulunduğu durumlarda daha iyi ve doğru tahminlere ulaşması, değişkenin model için önemli bir değişken olduğunu göstermektedir (Hosmer Jr. vd., 2013).

#### **3.1.4.8. Cok Katmanlı Algılayıcılar (MLP)**

Cok Katmanlı Algılayıcılar girdi katmanı, ara katman ve çıktı katmanından oluşmaktadır. Tek bir katmana sahip algılayıcıdan farkı ara katmanın, girdi ve çıktı

katmanları arasında bir geçiş sağlayan köprü görevi görmesidir. Ara katman, mevcut probleme göre daha iyi bir karar verilebilmesi için girdi katmanından ara katmana gönderilen girdileri, çıktı katmanına göndermeden önce bir değerlendirmeye tabi tutmaktadır. Model içinde kaç tane ara katman bulunacağını problemin durumu belirlediği için ara katman sayısı artırılabilir (Özdemir ve Ballı, 2020).

## **3.2. Yöntem**

Ara katman mimarisinin geliştirilmesinde kullanılan yöntemler test ortamının hazırlanması, ara katman mimarisinin geliştirilmesi ve makine öğrenmesi yöntemleri olmak üzere üç başlık altında incelenmektedir.

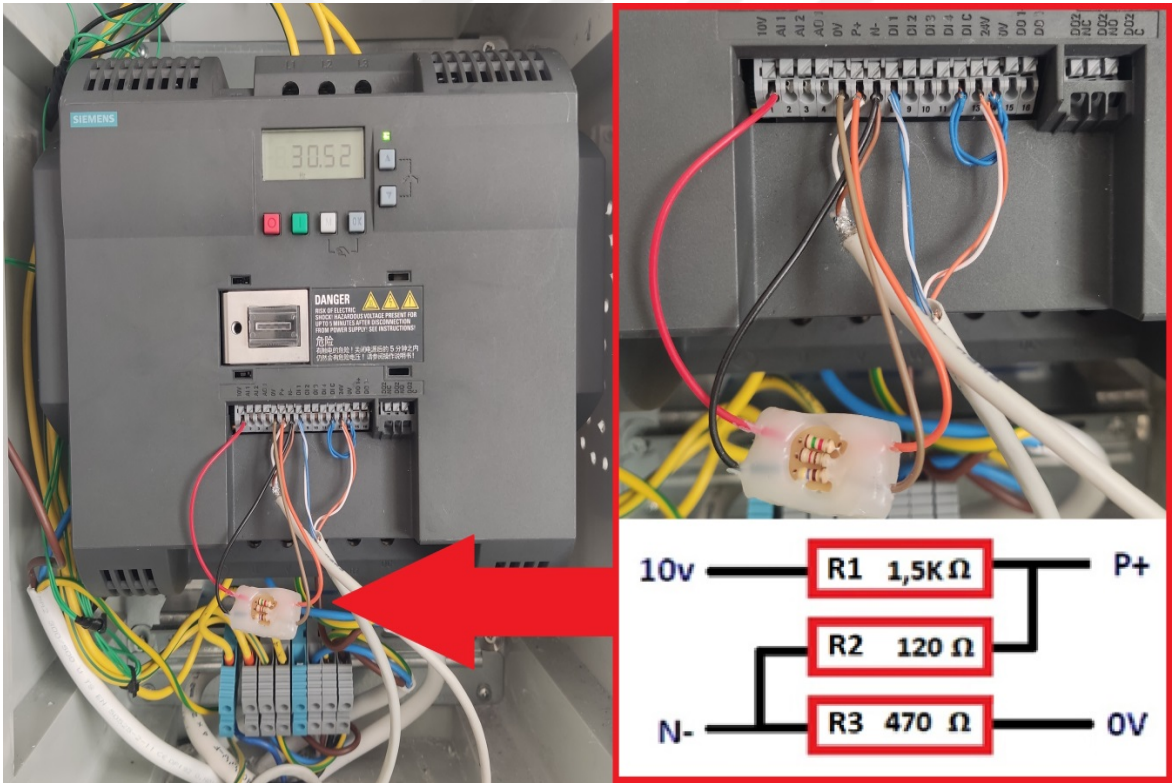
### **3.2.1. Test ortamının hazırlanması**

Çalışmaya donanım bileşenlerinin hazırlanarak aktif hale getirilmesi ile başlanmaktadır. Plc ve sürücünün farklı marka olması, birlikte çalışabilirliği için ayrıntılı bir inceleme ve tarama gerektirmiştir. İncelemeler sonucunda motordaki akım verilerinin okunabilmesi için herhangi bir sensör kullanımına gerek duymaksızın, direk sürücü üzerinden geçen akım verilerinin okunmasına, Plc ve sürücünün sahip olduğu ortak haberleşme kanalı olan Modbus RTU ile sürücüden Plc'ye aktarılmasına karar verilmektedir. Plc ve sürücü arasında Modbus RTU haberleşme hattının kurulması için RS 485-a ucu olarak sürücünün 7. ucu olan N- ucundan alınarak Plc TB SerialPort 2 üzerinde D0'a bağlanmaktadır. 485-b ucu olarak ise sürücünün 6. ucu olan P+ ucundan alınarak Plc TB SerialPort 2 üzerinde D1'a bağlanmaktadır. Bu sayede Modbus RTU seri haberleşme hattına ait fiziki bağlantı kurulumu gerçekleştirilmektedir.

Modbus haberleşme protokolünün dezavantajlarından biri çevre koşullarından ve manyetik alan etkilerinden kolay etkilenebilmesidir. Sürücü ve Plc arasındaki iletişim hattının dış ortam koşullarından izole edilebilmesi için elektromanyetik parazit engelleme özelliği olan koruma katmanlı kablo kullanılmakta ve kablonun şasesi, Plc şase ile topraklanmaktadır. Kurulan iletişim hattı, besleme kablolarından bağımsız bir şekilde farklı kablo kanalı üzerinden geçirilerek dış ortam etkileri azaltılmaya çalışılmaktadır. Bu

sayede haberleşmenin sağlanması sırasında oluşacak aksaklık ve kesintilerin önüne geçilmektedir.

Sürücüye ait teknik dokümanda belirtildiği üzere, sanayi ortamlarında birçok sürücü sıralı olarak birbirinin arkasına eklenerek kullanılmaktadır. Bu sebepten sistemin nerede sonlandığının anlaşılması adına en son kullanılan sürücünün sisteme tanımlanması gerekmektedir. Bu tanımlama, son sürücüye eklenecek bir direnç devresi ile sonlandırılarak gerçekleştirilmektedir. Çalışmamız içinde tek sürücü kullanıldığı için sürücü soket bağlantıları, kullanılan makroya göre yapıldıktan sonra hazırlanan küçük bir sonlandırma devresi ile sürücünün son sürücü olması sağlanmaktadır. Hazırlanan sonlandırma direnç devresi ve sürücüye bağlantıları Şekil 3.8’de görüntülenmektedir.



Şekil 3.8. Sürücü ve sonlandırma devresinin bağlantı şekli.

Sürücü üzerinde yapılandırma ayarlarının bulunduğu yönetim parametrelerine “uzman” modunda giriş ile Cn011 Modbus RTU kontrol makrosu seçimi yapılmaktadır. Sürücü, içerisinde hazır bulunan makrolar ile kullanılacak olan haberleşme kanalına ait hazır şablonlar sunulmaktadır. Makro seçimi sonrasında yapılandırma ayarları, Çizelge 3.2’de görüldüğü parametreler doğrultusunda Cn011 Modbus RTU kontrol makrosuna

uygun şekilde yapılandırılarak tamamlanmaktadır. Her parametrenin ve varsayılan değerinin çalışmaya uygun şekilde seçilmesi gerekmektedir.

Çizelge 3.2.1 Sürücü Cn011 Modbus yapılandırma ayarları (Siemens, 2023b).

Parametre	Tanım	Varsayılan Değer	Cn011 Varsayılan Değerleri	Açıklama
P0700[0]	Komut kaynağı seçimi	1	5	RS485 komut kaynağı
P1000[0]	Frekans seçimi	1	5	RS485 hız ayar noktası
P2023[0]	RS485 protokol seçimi	1	2	MODBUS RTU protocolü
P2010[0]	USS-MODBUS baudrate	6	6	Baudrate 9600 bps
P2021[0]	MODBUS adresi	1	1	İnvertör için MODBUS adresi
P2022[0]	MODBUS cevap zaman aşımı	1000	1000	Yanıtı yöneticiye geri göndermek için maksimum süre
P2014[0]	USS-MODBUS telegram kapatma süresi	2000	100	Veri alma zamanı
P2034	RS485 MODBUS paritesi	2	2	RS485, MODBUS telegrams Paritesi
P2035	RS485'te MODBUS durdurma bitleri	1	1	RS485 üzerindeki MODBUS telegramlarındaki durdurma bitlerinin sayısı

Plc üzerinde kullanılan iki seri porttan biri sürücü ile haberleşmeyi sağlamak diğeri ise HMI dokunmatik panel ekran ile iletişimi kurmak için kullanılmaktadır. İki seri porta

ait yapılandırmalar ayrı ayrı yapılmış ve iletişim kanalları kurularak bağlantılar aktif hale getirilmiştir. Dahili bulunan seri portlar RS485 standardını desteklemektedir.

Donanım bileşenlerinin hazırlanarak fiziki bağlantıların kurulmasını takiben gerekli yazılım yüklemeleri yapılmakta ve Plc'nin sürücü aracılığı ile elektrik motorundan verileri okuması için gerekli programın yazılması işlemi başlatılmaktadır. M241 sınıfı Plc'ye ait programlama ve her türlü yapılandırma, Schneider Elektrik firmasına ait SoMachine programı ile gerçekleştirilmektedir.

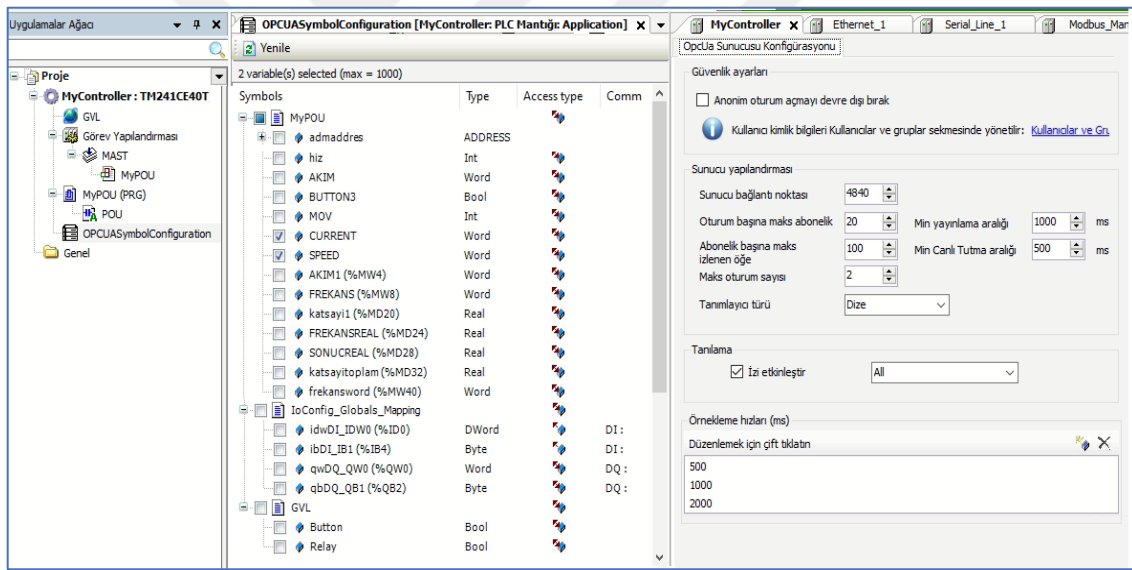
Plc üzerinde geliştirilecek olan uygulamada hem veri okuma hem de veri yazma işlemi gerçekleştirileceği için Read Holding Register fonksiyonu kullanılmaktadır. Akım ve frekans verileri Plc üzerinden okunabilirken, frekans değişiklikleri ve motorun çalışma durumuna dair durumlar için ise yazma işlemi gerçekleştirilmektedir. Read Holding Register ile Akım, Frekans (Hız) ve Motorun çalışma durumu verisi olmak üzere oluşturulan değişkenler ile Şekil 3.9'da görüldüğü üzere Plc Modbus Slave olarak tanımlanmaktadır.

Değişken	Eşleme	Kanal	Adres	Tür	Varsayılan Değer	Birim	Açıklama
		Channel 1	%IW3	ARRAY [0..0] OF WORD			Read Holding Registers
		Channel 1[0]	%IW3	WORD			READ 16#0019 (=00025)
		Channel 2	%QW2	ARRAY [0..0] OF WORD			Write Multiple Registers
		Channel 2[0]	%QW2	WORD			WRITE 16#0064 (=00100)
		Channel 3	%QW3	ARRAY [0..0] OF WORD			Write Single Register

Şekil 3.9. Plc üzerindeki Modbus Slave yapılandırması ve değişkenler.

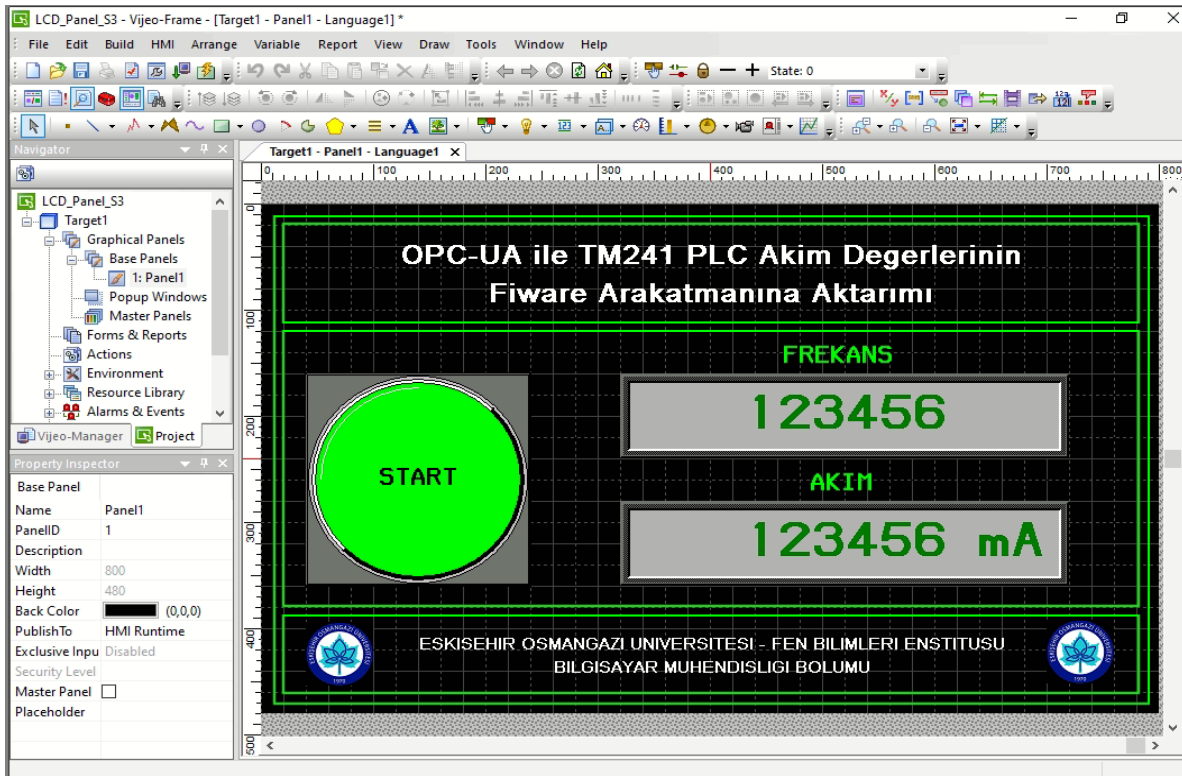
Oluşturulan yapılandırma ve tanımlamalar sayesinde Plc, sahip olduğu seri port üzerinden Modbus RTU haberleşme kanalını kullanarak veri okuyacak yeterliliğe ulaşmaktadır. Akabinde sürücüden gelen verilerin okunması ve yönetilebilmesi için gerekli giriş ve çıkış değişkenleri tanımlanarak, Plc'nin planlanan tüm işlemleri yapacak şekilde programlanması gerçekleştirilmektedir. Motor, sürücü ve Plc donanımlarının fiziki bağlantıları sağlandıktan sonra yapılan testlerde, motordan gelen akım verilerinin Plc'ye aktarımının başladığı görülerek MODBUS RTU veri hattının kurulumu tamamlanmaktadır.

Test düzeneğimizde OPC-UA sunucusu, Plc donanımımız üzerinde dâhili olarak bulunmaktadır. OPC-UA sunucu konfigürasyonu penceresi içerisinde bağlantı noktası, yayın ve bağlantı süreleri, abonelik sayıları gibi ayarlar yapılandırılmaktadır. OPCUA sembol konfigürasyonu penceresinden istemciler tarafından erişilebilecek olan sunucunun yayınlayacağı değişkenler tanımlanmaktadır. Sunucu üzerindeki OPC-UA Sembol konfigürasyonu, sadece akım ve frekans değişkenlerini yayınlanacak şekilde sınırlandırılmaktadır. Şekil 3.10'da Plc üzerinde hazırlanan programlama ile yapılandırılan OPC-UA sunucu üzerinden yayınlanacak olan değişkenlerin tanımlamaları, yapılandırılmaları, hangi değişkenlerin yayınlanacağına dair tercihler ve sunucunun hangi bağlantı noktasından ne sıklıkla yayın yapacağı bilgilerinin yer aldığı yapılandırma ayarları görüntülenmektedir. Plc üzerindeki Ethernet ve OPCUA sunucu yapılandırılmaları yapılarak sunucu yayın için hazır hale getirilmektedir.



Şekil 3.10. Plc üzerindeki OPC-UA yayın ve sunucu konfigürasyonu.

Schneider Elektrik firmasının diğer bir uygulaması olan Vijeo Designer programı ile HMI dokunmatik panel ekrana ait görsel tasarımları ve bu ekranın Plc ile haberleşmesi için gerekli programlamalar tamamlanmıştır (Şekil 3.11). Ekran üzerinde bulunan Start-Stop butonu ile motor çalıştırılabilmekte, frekans değeri dokunmatik panel ekran üzerinden değiştirilerek motorun hız kontrolü yapılabilmektedir. Akım bölümünde ise motorun anlık olarak harcadığı akım değerleri gözlenebilmektedir.



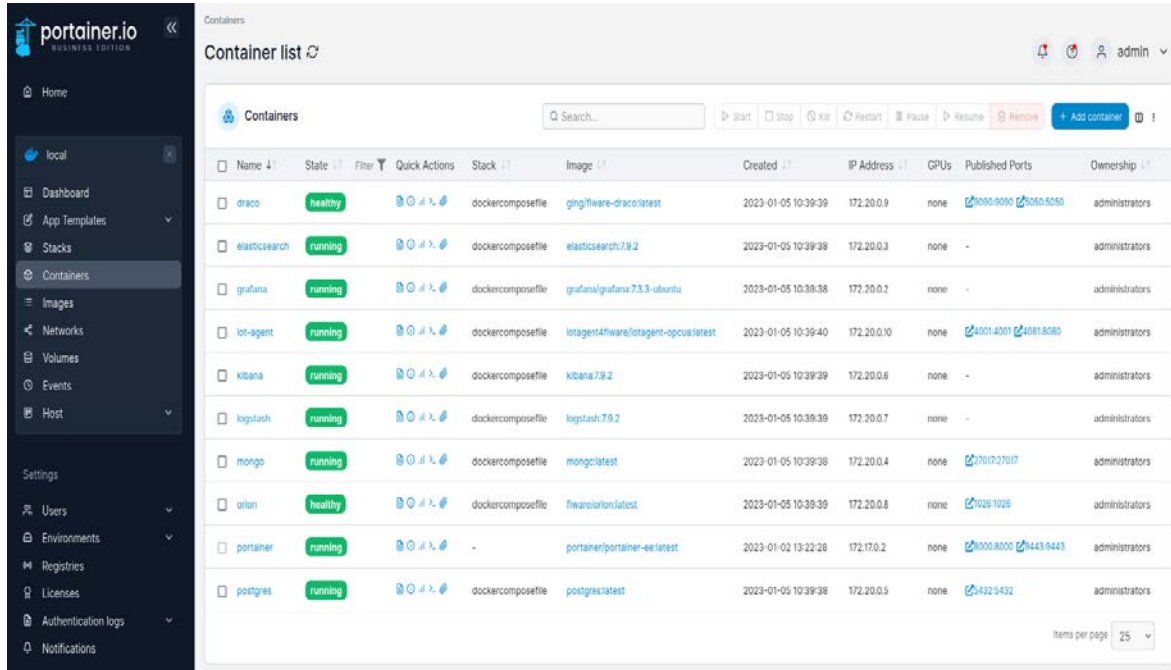
Şekil 3.11. Vijeo Designer ile HMIGXU3500 dokunmatik panel ekran tasarımı.

### 3.2.2. Ara katman mimarisinin geliştirilmesi

Fiware arakatman yazılımı Linux tabanlı bir yazılım olup Ubuntu işletim sistemi üzerinde çalışmaktadır. Kurulan Ubuntu sunucuda çalışacak olan Fiware ve bileşenleri için Docker Konteyner teknolojisi kullanılmaktadır. Docker ve Docker Compose kurulumları ile konteyner yapısı oluşturularak yapılandırılmaları gerçekleştirilmektedir. Oluşturulan konteyner yapılarının kontrol ve yönetimi Şekil 3.12'de ara yüz görseline yer verilen Portainer.io yazılımı ile takip edilmektedir.

Çalışmada kullanılacak olan her bir konteyner için tek tek image yüklemesi yapılmaktadır. Fiware ekosistemi bileşenleri Orion Context Broker ve OPC-UA IoT Aracı arası test çalışmaları, Fiware platformunun yayınladığı OPC-UA IoT Aracısı, araba örnek çalışması ile gerçekleştirilmektedir (Fiware – OPC UA Agent, 2023). Test çalışmaları sonrası IoT Aracısı bileşeni, Plc üzerinde çalışan OPC-UA sunucu yapılandırmasına göre yeniden yapılandırılmakta ve bağlantı sağlanmaktadır.

Çalışma sırasında PLC, veritabanı ve Fiware bileşenleri ile iletişim kurulması yönünde kullanılan API testlerinin tamamı Postman uygulaması ile gerçekleştirilmektedir. Postman, hızlı ve kolay bir şekilde kullanılabilmesi yönünden tercih sebebi olmaktadır.



Şekil 3.12. Portainer.io ara yüzü.

Fiware platformunda çalışmaların sorunsuz bir şekilde tamamlanabilmesi için en önemli konulardan biri, konteyner yapılarının sağlıklı çalışabilmesi için yapılandırma ayarlarının bulunduğu “Docker-compose.yml” dosyasının yapılandırılmasıdır. Bu dosya bütün Fiware bileşenlerine ait yapılandırma ayarlarını içerisinde barındırmaktadır. Bu sayede bileşenler nerede ve nasıl çalışacakları bilgisine ulaşmaktadır.

Şekil 3.13 (a)’da “Docker-compose.yml” dosyasına ait yapılandırılma ayarları görüntülenmektedir. Ayrıca Fiware platformunun, verileri alacağı bir sürücü tanımlaması gerekmektedir. Sürücü tanımlamaları “Config.json” dosyası ile yapılmaktadır. Config.json dosyası eklenen sürücünün OPC-UA sunucusu aracılığı ile kullanacağı servis, bağlantı noktası, bellek tipi ve aktarımı gerçekleştirilecek verilere ait değişkenler gibi yapılandırma ayarlarını içermektedir. Şekil 3.13 (b)’da “Config.json” dosyasına ait yapılandırılma ayarları görüntülenmektedir.

```

version: "3"
services:
  ##### ORION #####
  orion:
    image: fiware/orion:latest
    hostname: orion
    container_name: orion
    ports:
      - "1026:1026"
    depends_on:
      - mongo
    command: -statCounters -dbhost mongo -logLevel INFO
  ##### ORION MONGO #####
  mongo:
    image: mongo:latest
    hostname: mongo
    container_name: mongo
    ports:
      - "27017:27017"
    command: --nojournal
  ##### IOT AGENT #####
  iotage:
    image: iotagent4fiware/iotagent-opcua:latest
    hostname: iotage
    container_name: iotage
    ports:
      - "4001:4001"
      - "4081:8080"
    extra_hosts:
      - "iotcarsrv:192.168.8.55"
    depends_on:
      - mongo
      - orion
    volumes:
      - ./AGECONF:/opt/iotagent-opcua/conf
      - ./certificates:/opt/iotagent-opcua/certificates
    environment:
      - IOTA_REGISTRY_TYPE=memory
      - IOTA_LOG_LEVEL=DEBUG
      - IOTA_MONGO_HOST=mongo
      - IOTA_MONGO_DB=mongo
      - IOTA_FALLBACK_TENANT=opcua_car
      - IOTA_RELAX_TEMPLATE_VALIDATION=true
  ##### DRACOOO #####
  draco:
    image: ging/fiware-draco:latest
    hostname: draco
    container_name: draco
    depends_on:
      - postgres
    environment:
      - NIFI_WEB_HTTP_PORT=9090
    ports:
      - "9090:9090"
      - "5050:5050"
  ##### POSTGREEE #####
  postgres:
    image: postgres:latest
    hostname: postgres
    container_name: postgres
    expose:
      - "5432"
    ports:
      - "5432:5432"
    environment:
      - "POSTGRES_PASSWORD=password"
      - "POSTGRES_USER=postgres"
      - "POSTGRES_DB=postgres"
  {
    "logLevel": "INFO",
    "multiCore": false,
    "relaxTemplateValidation": true,
    "contextBroker": {
      "host": "orion",
      "port": 1026,
      "service": "opcua_plc",
      "subservice": "/plcl"
    },
    "server": {
      "port": 4001,
      "baseRoot": "/"
    },
    "deviceRegistry": {
      "type": "memory"
    },
    "mongodb": {
      "host": "mongo",
      "port": "27017",
      "db": "iotagent",
      "retries": 5,
      "retryTime": 5
    },
    "types": {
      "Device": {
        "service": "opcua_plc",
        "subservice": "/plcl",
        "active": [
          { "name": "AKIM", "type": "Number" },
          { "name": "FREKANS", "type": "Number" }
        ],
        "lazy": [],
        "commands": []
      }
    },
    "browseServerOptions": null,
    "service": "opcua_plc",
    "subservice": "/plcl",
    "providerUrl": "http://iotage:4001",
    "pollingExpiration": "200000",
    "pollingDaemonFrequency": "20000",
    "deviceRegistrationDuration": "P1M",
    "defaultType": null,
    "contexts": [
      {
        "id": "plcl_plc",
        "type": "Device",
        "service": "opcua_plc",
        "subservice": "/plcl",
        "polling": false,
        "mappings": [
          {
            "ocb_id": "AKIM",
            "opcua_id": "ns=2;s=MyPOU.CURRENT",
            "object_id": "ns=2;s=MyPOU.CURRENT",
            "inputArguments": []
          },
          {
            "ocb_id": "FREKANS",
            "opcua_id": "ns=2;s=MyPOU.SPEED",
            "object_id": "ns=2;s=MyPOU.SPEED",
            "inputArguments": []
          }
        ]
      }
    ],
    "contextSubscriptions": []
  }

```

(a) Docker-compose.yml

(b) Config.json

Şekil 3.13. Docker-compose.yml ve Config.json dosyalarına ait yapılandırma ayarları.

Docker-compose.yml ve Config.json dosyalarının ayarları yapılarak OPC-UA IoT Aracısı, OPC-UA Client olarak çalıştırılmakta, Plc akım ve frekans verilerine ait yayının okunması sağlanmaktadır. OPC-UA IoT Aracısı, OPC-UA verilerini NGSI veri modeli kullanarak Fiware bileşeni olan Orion Context Broker'da yayınlanmasını sağlayan bir araçtır. Test düzeneğimizde Plc üzerinden aktarılan veriler OPC-UA sunucusu yayınları ile OPC-UA IoT Aracısına iletilmektedir. Sürüciden Plc aracılığı ile gelen veriler, istemci tarafından alınmadan önce OPC-UA sunucu adres alanında nitelikler olarak tutulmaktadır. Böylece, sunucu tarafından kullanılan, Plc üzerindeki verileri kontrol etmekte mümkün olmaktadır. Plc üzerindeki sürücü motor verilerine erişim, bir abonelik mekanizması aracılığıyla sağlanmaktadır. OPC-UA istemcisi erişmek istediği akım ve frekans verilerini

içeren bir abonelik oluşturmaktadır. İstemci, bu parametreleri kullanarak sunucudan veri göndermesini istemekte, sunucu ise isteklerin devam durumunu belirlemektedir. Yani OPC-UA IoT Aracısı, Orion Context Broker ile OPC-UA sunucusu arasında bir OPC-UA istemcisi gibi davranarak köprü görevi görmektedir.

OPC-UA IoT Aracısı, sunucuya yaptığı abonelik ile verilere ulaşmakta, bu verileri Orion ve MongoDB'ye göndermektedir. MongoDB kendisine iletilen verileri anlık olarak saklamakla ve en son veriyi döndürmekle görevlidir. Yani ilişkisel olmayan verilerin saklandığı bu veri tabanı, kalıcı verilerin saklandığı bir veri tabanı değildir. Veriler üzerinde analiz yapılabilmesi için kalıcı bir veri tabanına aktarılması ve orada saklanması gerekmektedir. Bu verilerin saklanması için geliştirilen sistem üzerinde PostgreSQL veritabanı kullanılmaktadır. Ancak MongoDB üzerindeki bulunan yapılandırılmamış verilerin, ilişkisel olmayan veri türlerini ifade etmesi ve bu verileri ilişkisel tablolardan farklı bir formatta depolaması sebebi ile PostgreSQL veritabanına doğrudan aktarımı sağlanamaz. Verilerin Sql veritabanı formatına uygun bir şekilde anlamlandırılması ve dönüştürülmesi gerekmektedir. Bu noktada devreye Apache Nifi tabanlı olan Draco uygulaması girmektedir. Draco, Orion Context Broker'dan gelen verileri işleyerek dönüştürecek ve PostgreSQL veritabanına yönlendirecektir.

Draco ara yüzüne ulaşılması için konteyner üzerinde yapılan düzenlemede ara yüzün çalışma bağlantı noktası olarak 9090 bağlantı noktası seçilmektedir. Ara yüze ulaşıldıktan sonra şablonlar içerisinde çalışmamızda kullanılacak olan “NGSIToPostgreSQL Processor” şablonu eklenmekte ve işlemci bağlantıları yapılandırılarak mevcut yapılandırmaya ulaşılmaktadır. SQL işlemcileri bir bağlantı havuzu gerektirdiğinden verilerin nasıl kaydedileceğinin ayarlanması ve işlemci ile birlikte gelen DBCPConnectionPool'un kurulması gerekmektedir. Kullanılacak olan PostgreSQL sunucusuna bağlanmak için Database Connection URL kullanılmaktadır. URL olarak, çalışan postgresql docker konteynera ait ip yazılabileceği gibi, docker konteynerın alan adı ile de ulaşılabilir. PostgreSQL'in varsayılan bağlantı noktası “5432” olan bağlantı noktası kullanılmaktadır. Yapılandırmada Şekil 3.14'deki görselde görüntülediği gibi kapsayıcı adı, veri tabanı kullanıcı adı ve kullanıcı şifresi girilerek gerçekleştirilmektedir.

Property	Value
Database Connection URL	jdbc:postgresql://postgres:5432/
Database Driver Class Name	org.postgresql.Driver
Database Driver Location(s)	./drivers/postgresql-42.2.4.jar
Database User	veri tabanı kullanıcı adı
Password	veri tabanı kullanıcı şifresi
Max Wait Time	500 millis
Max Total Connections	8
Validation query	No value set

Şekil 3.14. DBCPConnectionPool yapılandırma ayarları.

Yapılandırma ayarları yapıldıktan sonra DBCPConnectionPool, ilgili menüden aktif edilmekte ve daha sonra NGSIToPostgreSQL işlemcisi çalıştırılmaktadır. Veri almak için hazır hale gelen Draco, Şekil 3.15'de görülen REST API çağrı kodu ile aktifleştirilmektedir. Orion Context Broker tarafından gönderilen veriler, bu kod aracılığı ile takip edilmeye başlanmaktadır. Takibi başlatılan veriler içindeki ilk değişim ile aktif verilerin geldiği yorumlanmakta ve veri aktarımı başlatılmaktadır. Veri yığımlarının önüne geçilmesi için arka arkaya gelen aynı veriler tekrarlanmayacak ve her gelen yeni veri ile değişim sağlanarak yeni veri Draco'ya aktarılacaktır.

```
curl -iX POST \
  'http://localhost:1026/v2/subscriptions' \
  -H 'Content-Type: application/json' \
  -H 'fiware-service: opcua_plc' \
  -H 'fiware-servicepath: /plc1' \
  -d '{
    "description": "Notify Draco of all context changes",
    "subject": {
      "entities": [
        {
          "idPattern": ".*"
        }
      ]
    },
    "notification": {
      "http": {
        "url": "http://draco:5050/v2/notify"
      }
    },
    "throttling": 5
  }'
```

Şekil 3.15. Draco'nun aktif hale getirilmesi için kullanılan REST API çağrı kodu.

### 3.2.3. Makine öğrenmesi yöntemleri

Makine öğrenmesi çalışmalarında, Python programlama dilini destekleyen ücretsiz ve açık kaynaklı Anaconda ile birlikte gelen Jupyter Notebook ile gerçekleştirilmiştir (Jupyter, 2023). Uygulamada makine öğrenmesi algoritmalarından Rasgele Orman (RF), Karar Ağaçları (DT), Gradyan Artırma Makinesi (GBM), Hafif Gradyan Artırma Makinesi (LGBM), K-En Yakın Komşuluk (K-nN), Kategorik Artırma (CB), Lojistik regresyon (LR) ve Çok Katmanlı Algılayıcılar (MLP) kullanılırken, Numpy, Pandas, Matplotlib, Searborn, Statsmodel, Scipy, Scikit-learn Python kütüphanelerinden faydalanılmıştır.

Numpy, neredeyse bütün Python kitaplıklarının temelini oluşturan, bilimsel veya sayısal hesaplama yapabilen, matematiksel işlemlerin hızlı bir şekilde gerçekleştirilmesinde kullanılan kütüphanedir. Pandas, etiketlenmiş ilişkisel veriler ile sezgisel ve basit bir çalışma gerçekleştirmek üzere tasarlanmış Python kütüphanesidir. Kütüphanenin yetenekleri arasında ön işleme ve analiz gerçekleştirebilmesi yer almaktadır. Ön işleme ve analiz, kütüphane içerisinde verilerin işlenebilmesi, yönetilebilmesi ve görselleştirilebilmesi sayesinde gerçekleştirilmektedir. Matplotlib, iki boyuta sahip olan grafik çizimlerini oluşturmak ve belirlenen biçimlerde çıktılara ulaşmak için kullanılan bir kütüphanedir. Searborn, daha çok istatistiksel modellerin görselleştirilerek sunulması için kullanılan bir kütüphanedir. Bu kütüphane verileri özetleyerek, verilerin genel dağılımlarını ifade eden görseller sunmaktadır. Mimari özelliği bakımından Matplotlib kütüphanesine dayanmaktadır (Ikonomakis vd., 2005). Statsmodel, özellikle istatistikler için oluşturulmuş bir Python kütüphanesidir. İstatistik modelleri NumPy, SciPy modellerine dayanmaktadır (Akarsu ve Dilbaz Alacahan, 2021). Scipy, içerisinde Fourier transform, interpolasyon, optimizasyon ve istatistiksel algoritmaları barındıran, mühendislerin, matematikçilerin ve bilim insanlarının kullanımına sunulan bir kütüphanedir (Scipy, 2023). Scikit-learn makine öğrenmesine ait modeller oluşturmak için kullanılan bir kütüphanedir. Bu kütüphane, sınıflandırma, kümeleme ve regresyon gibi işlemleri gerçekleştirmek için kullanılan bir öğrenme algoritmasına sahiptir (Bitdegree, 2023).

Makine öğrenmesi modelleri oluşturulurken öznitelik seçimi (feature selection) yapılarak veri kümesinde bulunan tüm değişkenlerden bir alt küme oluşturulmaktadır. Öznitelik seçimi; basit modeller oluşturmak ve yorumlamayı kolaylaştırmak, eğitim süresini kısaltmak, aşırı öğrenmeyi azaltmak, modelin uygulanmasını daha kolay hale getirmek, gereğinden fazla değişkenlerden kurtulmak ve yüksek boyutlu veri setlerinde kötü öğrenmeyi engellemek için uygulanmaktadır (Brownlee, 2019). Çalışmamızda Min, Max, Mean, Rms, Var, Std, Kurtosis, P2p öznitelik seçim yöntemleri kullanılarak makine öğrenmesi modelleri geliştirilmiştir.

Veri analizi gerçekleştirilirken en doğru öğrenme modelinin hangisi olduğuna karar vermek için öğrenme modelinin çıktılarının özenle değerlendirilmesi gerekmektedir. Gerçekleştirilen akademik çalışmalar incelendiğinde model performansını ve etkinliğini belirlemek için en çok kullanılan metriklerin Kesinlik (Precision), Duyarlılık (Recall) ve Doğruluk (Accuracy) metrikleri olduğu görülmektedir. Bu metriklerin hesaplanmasında ilk olarak modele ait Doğru Pozitif, Doğru Negatif, Yanlış Pozitif ve Yanlış Negatif gibi sınıflandırmaları açıklamak gerekmektedir. Doğru Pozitif (TP), bir koşulun veya özelliğin var olduğunu doğru bir şekilde gösteren, Doğru Negatif (TN), bir koşulun veya özelliğin var olmadığını doğru bir şekilde gösteren, Yanlış Pozitif (FP), belirli bir koşulun veya özelliğin var olduğunu yanlış bir şekilde gösteren, Yanlış Negatif (FN), belirli bir koşulun veya özelliğin var olmadığını yanlış bir şekilde gösteren test sonuçlarıdır (Ikonomakis vd., 2005).

Kesinlik " $\pi_i$ " ile temsil edilmekte ve herhangi bir  $d$  belgesinin,  $c_i$  sınıfına eklenmesi halinde, sınıflandırmaya ait doğru olma olasılığını ifade etmektedir (Denklem 3.1). Duyarlılık " $p_i$ " ile temsil edilmekte ve belgelerin asıl olarak  $c_i$  sınıfında olması gerekirken, kaç tanesinin belirtilen sınıfta olduğu olarak tanımlanmaktadır (Denklem 3.2). Doğruluk " $A_i$ " ile temsil edilmekte ve algoritmanın doğru sonuçlar elde etmesi konusundaki yeteneğini göstermektedir (Denklem 3.3). Burada belirtilecek hata oranına, doğruluk değerinin 1'e tamlayanı hesaplanarak ulaşılmaktadır. F1-skoru ise, uç durumlarda oluşan değerlerin dikkate alınmaması gibi durumları engellemek için Kesinlik ile Duyarlılık değerlerine ait harmonik ortalama alınarak hesaplanmaktadır (Denklem 3.4). Metriklere ait hesaplama formülleri aşağıda sırasıyla görüntülenmektedir (Ikonomakis vd., 2005).

$$Kesinlik (\pi_i) = \frac{TP_i}{TP_i+FP_i} \quad (3.1)$$

$$Duyarluluk (p_i) = \frac{TP_i}{TP_i+FN_i} \quad (3.2)$$

$$Doğruluk (A_i) = \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i} \quad (3.3)$$

$$Skor (F_1) = 2 * \frac{Kesinlik * Duyarluluk}{Kesinlik + Duyarluluk} \quad (3.4)$$

Bu bölüm başlığı altında yer alan materyal ve yöntemler kullanılarak, Plc üzerinde bulunan OPC-UA sunucusundan, Fiware ara katman bileşeni olan OPC-UA IoT Aracısı ile çekilen veriler, Draco ile Sql veri formuna dönüştürülerek PostgreSQL veri tabanına aktarılmaktadır. Aktarılan veriler ise çeşitli makine öğrenmesi algoritmaları kullanılarak anomali tespiti gerçekleştirilmektedir.

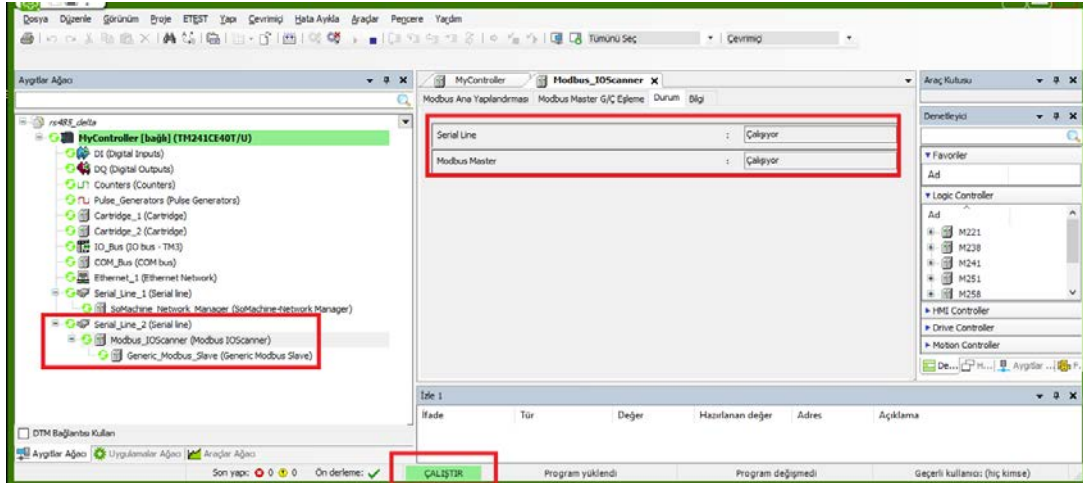
## 4. BULGULAR VE TARTIŞMA

Kestirimci bakım için ara katman mimari uygulamasına yönelik test ortamı ve arakatman mimarisinin geliştirilmesi ile kestirimci bakım için gerçekleştirilen anamoli tespiti sırasında karşılaşılan bulgular bu bölümde detaylı olarak incelenmekte ve tartışılmaktadır.

### 4.1. Test Ortamı ve Ara Katman Gelişimi Sırasındaki Bulgular

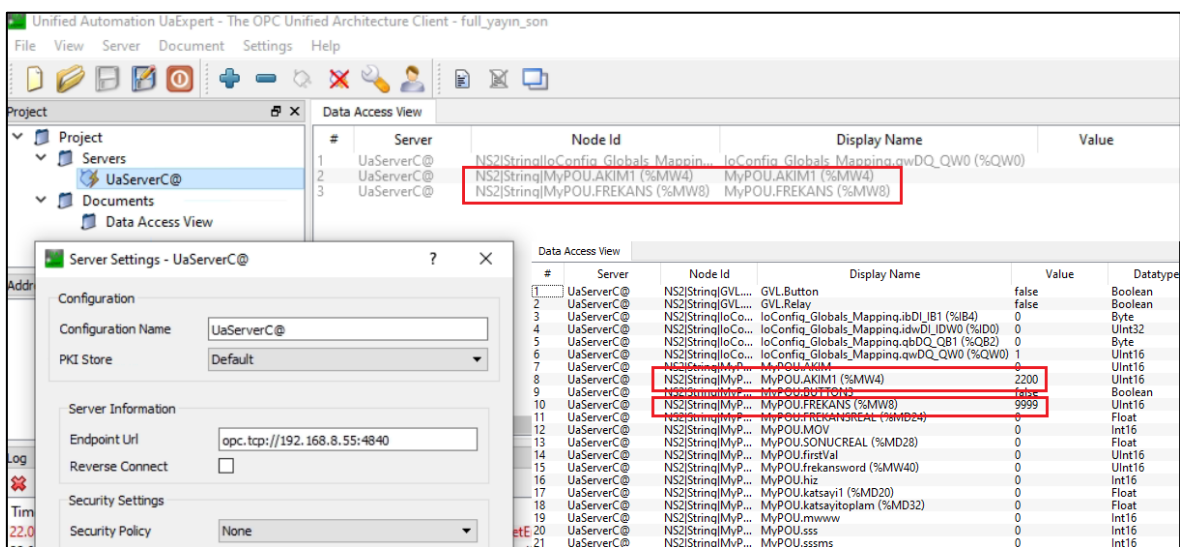
Donanım bileşenlerinin hazırlanması ile başlatılan çalışmalarda elektrik motoru ve sürücü şebekeden beslenecek şekilde 3 faz olarak birbirine bağlanmaktadır. Modbus RTU haberleşme protokolü ile sürücü ve Plc'nin haberleşebilmesi için iki kablolu hat ile seri bağlantı kurulmaktadır. Bu haberleşme protokolünün dezavantajlarından biri çevre koşullarından ve manyetik alan etkilerinden kolay etkilenebilmesidir. Bu sebeple yalıtımlı koaksiyel kablo kullanılarak şase bağlantısı yapılmakta ve besleme kablolarından bağımsız bir şekilde farklı kanal üzerinden bağlantı sağlanmaktadır. Buna rağmen gerçekleştirilen ilk testlerde bekleme modunda hiçbir hata alınmamasına ve iki bileşenin sağlıklı olarak çalışıyor görünmesine rağmen, elektrik motoru çalıştırılarak sürücü devreye girdiğinde sürücü üzerinde F72 hatası alınmaktadır. Hatanın sebebi sürücü kullanım kılavuzunda araştırıldığında Modbus seri haberleşme hatası olduğu tespit edilmektedir. Hatanın giderilmesi adına gerçekleştirilen incelemelerde, sürücüde P2014[0] nolu parametre olan ve varsayılan olarak 100 ms değerine sahip veri alma zamanının 2000ms civarına çekilerek hatanın giderilebileceği bilgisine ulaşılmaktadır. Kademeli olarak yükseltile P2014[0] parametre değerinin ara değerlerinde, elektrik motoruna ait çalışma süresi giderek artmasına rağmen sağlıklı bir çalışma durumuna 2000ms sürelerinde erişilmektedir.

Sürücü üzerindeki akım verilerinin değişkenliği 30 Hz çalışma frekansında değerlendirilmektedir. Modbus RTU seri haberleşme protokolü ile Plc ve Sürücü arasında kurulan iletişim hattının kontrolleri Şekil 4.1'de görülen SoMachine uygulamasında sağlanmaktadır.



Şekil 4.1. SoMachine uygulaması üzerinde Modbus RTU seri haberleşme kontrolü.

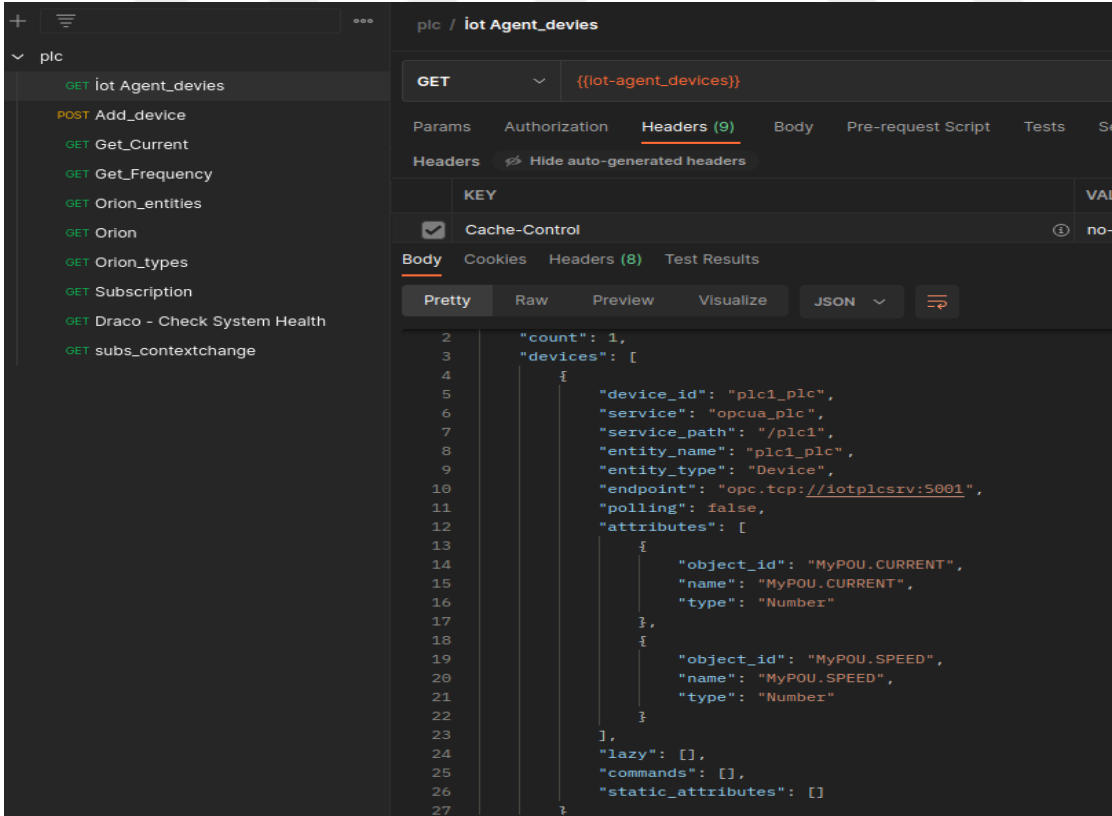
Sürücü ve Plc arasındaki seri bağlantı sağlandıktan sonra sıra Plc ile Fiware platformunu üzerinde barındıracak olan sunucunun OPC-UA haberleşme protokolü ile bağlantısının kurulması işlemine gelmektedir. Plc üzerinde bulunan OPC-UA sunucu yayın ayarları gerçekleştirildikten sonra yayına ait testler, ağ içinde kullanılan farklı bir bilgisayar üzerinde kurulu olan UaExpert programı aracılığı ile yapılmıştır. Yapılan testlerde istemci olarak kullanılan bilgisayar sunucu yayın bilgilerine erişimi sağlamış ve OPC-UA haberleşme protokolü aracılığıyla sunucu/ istemci haberleşme kanalı kurulmuştur. OPC-UA Bağlantı testleri Şekil 4.2’de görüntülenen ve bir OPC uygulaması olan UaExpert ile gerçekleştirilmiştir.



Şekil 4.2. OPC Test çalışmalarında kullanılan UaExpert uygulaması.

Kurulan Ubuntu sunucuda çalışacak olan Fiware ara katman mimarisinde, Orion Context Broker, OPC-UA IoT Aracısı ve Draco GE'leri çalıştırılmaktadır. Her bir Fiware bileşeni Docker Konteyner olarak sisteme entegre edilmiş ve çalışmaları anlık olarak incelenmiştir. IoT Aracısı için oluşturulan sürücü sağlıklı bir şekilde çalıştırılmış, sürücü üzerinde tanımlanan değişkenler sayesinde sunucunun yaptığı yayın bilgilerine abone olunarak verilere erişim gerçekleştirmiştir. Fiware bileşenlerine ait ilk test aşamalarında Fiware platformunda verilen Araba örneği değerlendirilmiştir. Örnek üzerinde REST API kodları kullanılarak örnekteki arabaya ait, Araba (nesne), Hız (öznitelik), Hızlandırmak (metot), Durdur (metot), Motor (nesne), Sıcaklık (öznitelik), Oksijen (öznitelik) ve kullanıcı tanımlı doğrulama sensörlerindeki değişkenler üzerinde değişiklikler yapılabilmiş ve yeni değişkenler tekrar okunabilmiştir.

Fiware bileşenlerimizden olan OPC-UA Aracısına ait yapılandırma dosyalarının düzenlenmesini takiben test çalışmaları başlatılmıştır. Postman uygulamasında kullanılan REST çağruları ile IoT Aracı sürücüsüne bağlantı kontrolleri sağlanmıştır. Şekil 4.3'de Postman üzerinden sürücü bilgilerine yapılan REST çağrısı ve sonucu görüntülenmektedir.



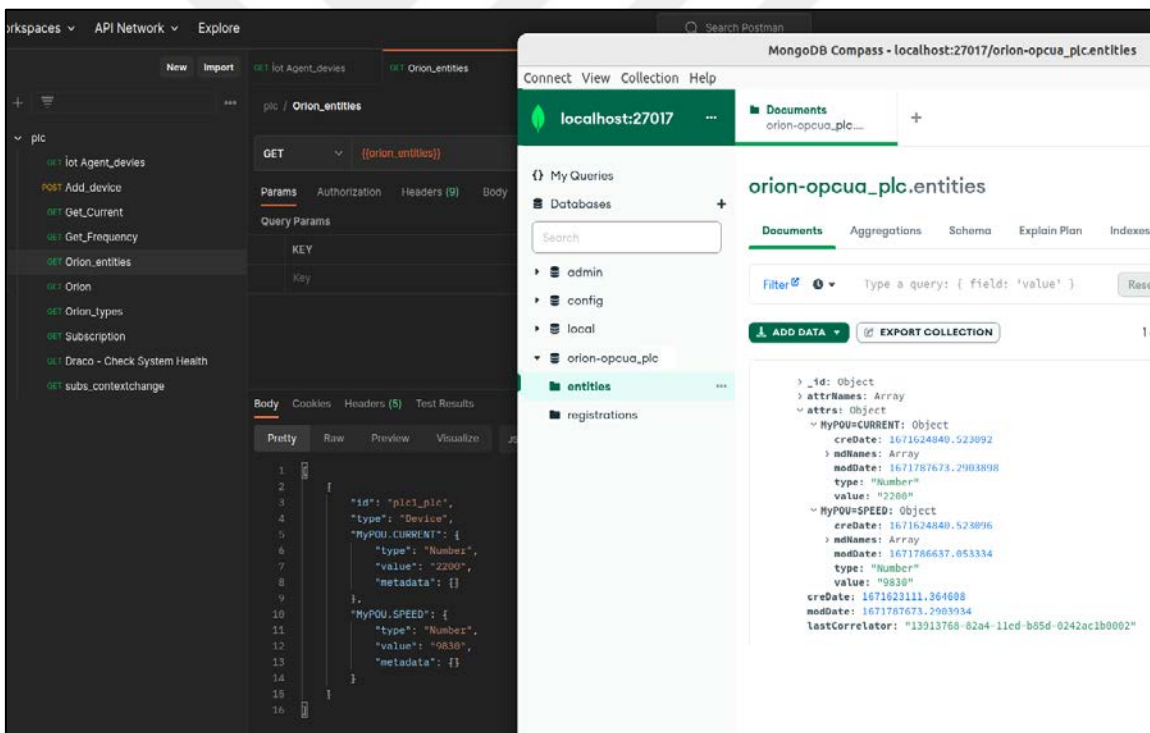
```

2  "count": 1,
3  "devices": [
4    {
5      "device_id": "plc1_plc",
6      "service": "opcua_plc",
7      "service_path": "/plc1",
8      "entity_name": "plc1_plc",
9      "entity_type": "Device",
10     "endpoint": "opc.tcp://iotplcsrv:5001",
11     "polling": false,
12     "attributes": [
13       {
14         "object_id": "MyPOU.CURRENT",
15         "name": "MyPOU.CURRENT",
16         "type": "Number"
17       },
18       {
19         "object_id": "MyPOU.SPEED",
20         "name": "MyPOU.SPEED",
21         "type": "Number"
22       }
23     ],
24     "lazy": [],
25     "commands": [],
26     "static_attributes": []
27   }

```

Şekil 4.3. Postman üzerinden IoT Aracı sürücüsüne yapılan REST çağrısı ve sonucu.

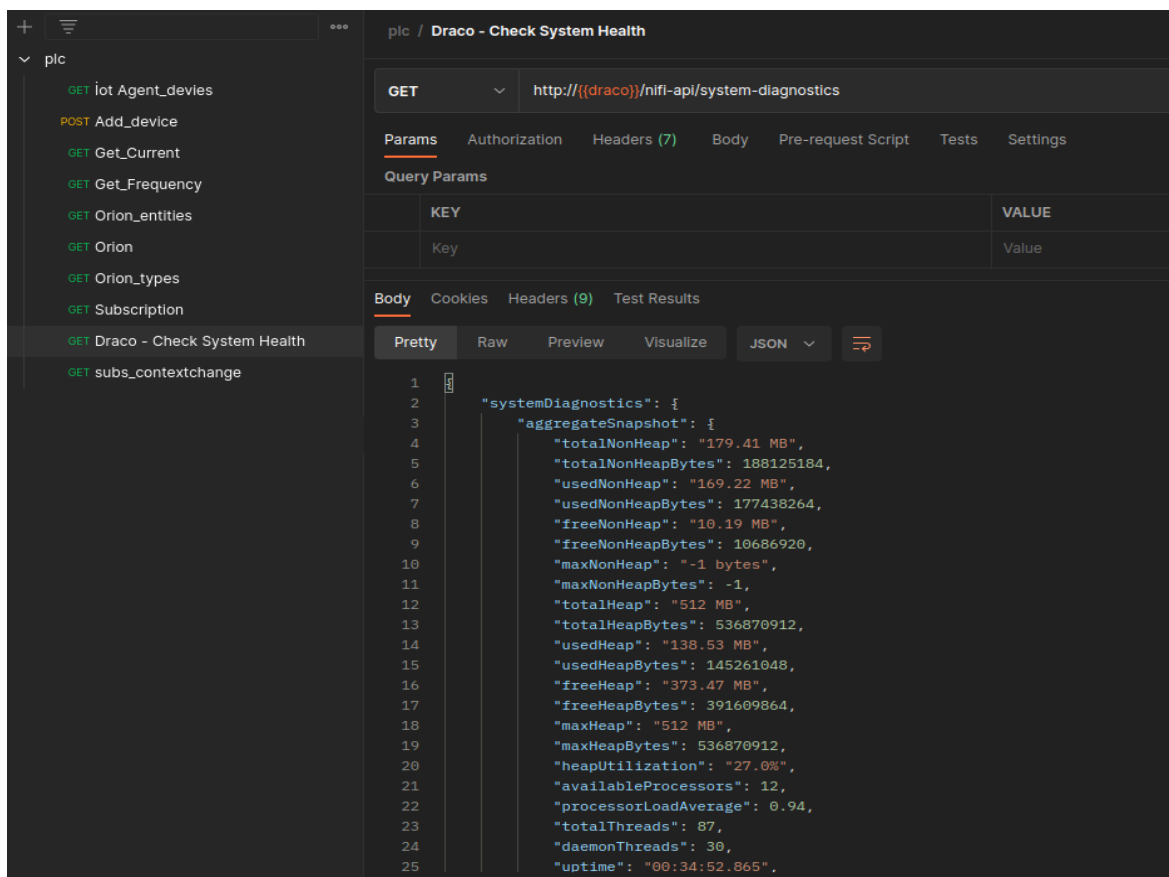
IoT Aracısının REST çağrılarında verdiği olumlu cevaplar sonrasında sıra Orion Context Broker ile bu verilerin alınıp alınmadığının test edilmesine gelmektedir. IoT Aracısı yapılandırma gereği verileri hem Orion Context Broker'a hemde MongoDB'ye gönderecek şekilde ayarlandığından, yapılandırılmalarının eksiksiz olarak tamamlanmasının kontrol edilmesi için yapılacak test çalışmaları sırasında Postman ve MongoDB uygulamalarının ikisi birden kullanılmıştır. Orion ve MongoDB'de gerçekleştirilen REST API çağrılarında başarılı sonuçlar alınmıştır. OPC-UA IoT Aracısı ile sunucuya abonelik işlemi gerçekleştirilmiş ve Plc üzerindeki motor akım ve frekans verilerinin Orion Context Broker'a anlık aktarımları gerçekleştirilmiştir. Böylece elektrik motoru ve sürücü ile başlayan veri hattı Orion'a kadar başarılı bir şekilde kurulmuştur. Şekil 4.4'de Orion Context Broker'a ait test çalışmalarında kullanılan Postman ve MongoDB uygulamalarına ait ekran görüntüleri yer almaktadır.



Şekil 4.4. Orion Context Broker'a ait test çalışmalarında kullanılan Postman ve MongoDB uygulamalarına ait ekran görüntüleri.

Orion Context Broker ile toplanan veriler PostgreSQL veri tabanına aktarılmak üzere Draco'nun 9090 bağlantı noktasına gönderilmektedir. Bu bağlantı noktasını sürekli olarak dinleyen Draco, gelen verileri işleyerek PostgreSQL veri tabanına aktarmaktadır. Çalışmamızda NGSI veri modeline dayanan Fiware bileşenleri kullanıldığından,

çalışmanın iletilebilmesi için Draco ara yüzünde NGSI tabanlı şablon kullanılmaktadır. Dikkat edilmesi gereken hususlardan bir tanesi aktarım sağlanacak olan veri tabanı ayarlarının doğru yapılandırması olacaktır. Veri tabanına ait bağlantı adresi ve kullanıcı bilgileri gibi yapılandırma ayarları gerçekleştirildikten sonra “DBCPCConnectionPool” etkinleştirilerek veri tabanı bağlantısı kurulmaktadır. Postman uygulaması üzerinden Draco’ya ait sistem sağlığı kontrol çağrısı yapılarak başlıklarda oluşan aktarımlar görüntülenmiş ve Fiware’e ait bu bileşenin sağlıklı çalıştığı test edilmiştir. Yapılan REST çağrısı ve çağrı yanıtı Şekil 4.5’de görüntülenmektedir.



Şekil 4.5. Draco’ya ait sistem sağlığı kontrol REST çağrısı ve çağrı yanıtı.

Başarılı bir sağlık testi sonrasında PostgreSQL veritabanında verilerin akışı kontrol edilmiştir. Aktarımı sağlanan veriler için PgAdmin uygulamasından elde edilen ekran görüntüsü Şekil 4.6’da görüntülenmektedir.

pgAdmin 4

File Object Tools Help

Browser postgres/postgres@PostWork

Dashboard Properties SQL Statistics Dependencies Dependents Processes opcu\_plc.x00... postgres/postgres@F

Query History

1 SELECT \* FROM opcu\_plc.x002fdemo ORDER BY RECVTIMETS DESC

Data Output Messages Notifications

	recvtimets	recvertime	fiwareservicepath	entityid	entitytype	attrname	attrtype	attrvalue	attrmtd
18	672994315470	01/06/2023 08:38:35	plc1	plc	Device	AKIM	Number	2290	
19	1672994310465	01/06/2023 08:38:30	plc1	plc	Device	FREKANS	Number	9830	
20	1672994310465	01/06/2023 08:38:30	plc1	plc	Device	AKIM	Number	2200	
21	1672994303458	01/06/2023 08:38:23	plc1	plc	Device	FREKANS	Number	9830	
22	1672994303458	01/06/2023 08:38:23	plc1	plc	Device	AKIM	Number	2200	
23	1672994296454	01/06/2023 08:38:16	plc1	plc	Device	FREKANS	Number	9830	
24	1672994296454	01/06/2023 08:38:16	plc1	plc	Device	AKIM	Number	2200	
25	1672994290449	01/06/2023 08:38:10	plc1	plc	Device	FREKANS	Number	9830	
26	1672994290449	01/06/2023 08:38:10	plc1	plc	Device	AKIM	Number	2200	
27	1672994280438	01/06/2023 08:38:00	plc1	plc	Device	FREKANS	Number	9830	
28	1672994280438	01/06/2023 08:38:00	plc1	plc	Device	AKIM	Number	2210	
29	1672994272429	01/06/2023 08:37:52	plc1	plc	Device	FREKANS	Number	9830	
30	1672994272429	01/06/2023 08:37:52	plc1	plc	Device	AKIM	Number	2210	
31	1672994265421	01/06/2023 08:37:45	plc1	plc	Device	FREKANS	Number	9830	
32	1672994265421	01/06/2023 08:37:45	plc1	plc	Device	AKIM	Number	2210	
33	1672994262417	01/06/2023 08:37:42	plc1	plc	Device	AKIM	Number	2200	
34	1672994262417	01/06/2023 08:37:42	plc1	plc	Device	FREKANS	Number	9830	
35	1672994255410	01/06/2023 08:37:35	plc1	plc	Device	FREKANS	Number	9830	

Total rows: 138 of 138 Query complete 00:00:00.145 Ln 1, Col 1

Şekil 4.6. PgAdmin uygulamasından elde edilen ekran görüntüsü.

Orion Context Broker'dan Droco'ya iletilen akım ve frekans verileri, sağlıklı bir şekilde PostgreSQL veritabanına aktarılmıştır. Tez çalışmasının odak noktası olan kestirimci bakım için ara katman mimari uygulamasına ait çalışma başarılı bir şekilde gerçekleştirilmiştir. Geliştirilen Fiware tabanlı ara katman, OPC-UA haberleşme protokolünü kullanarak kestirimci bakım çalışmalarında kullanılacak olan makine öğrenmesi algoritmalarına veri sağlayabilmektedir.

Geliştirilen ara katman mimarisinden başarılı bir şekilde verilerin elde edilmesine rağmen iki eleştiri noktası ortaya çıkmıştır. İlki V20 sürücü üzerinde 3 faz olmasına rağmen akım verisi ortalama veri olarak tek faz şeklinde verilmektedir. Bu fazlar üzerinde ayrıntılı çalışmayı engellemektedir. Diğer bir eleştiri noktası ise V20 motor sürücüsünün üzerinde okunan akım verisi istenilen hassasiyette olmamasıdır. Tek faz üzerinden okunan akım verisine ait tam sayıdan sonra gelen ondalıklı kısım iki hane (2.20 Amper gibi) kadar görüntülenebilmektedir. Bu da hassas bir akım verisinin okunamamasına sebep olmaktadır. Oysaki sağlıklı bir kestirimci bakım analizi yapılabilmesi için buradaki hassasiyetin daha fazla olması beklenmektedir. Ayrıca V20 sürücü'den Plc'ye veri aktarımı için kullanılan seri haberleşme protokolü Modbus RTU'da karşılaşılan veri aktarım hataları sırasında

P2014[0] parametresinin 2000ms'ye çıkarılması, birbirini takip ederek gelecek veriler arasında yaklaşık 2 saniyelik bir bekleme süresi olmasını sağlamaktadır. Bu da aktarılan verilere ait örneklemin yeteri kadar hassas olmasını engellemektedir.

#### 4.2. Kestirimci Bakım İçin Gerçekleştirilen Anomali Tespiti Sırasındaki Bulgular

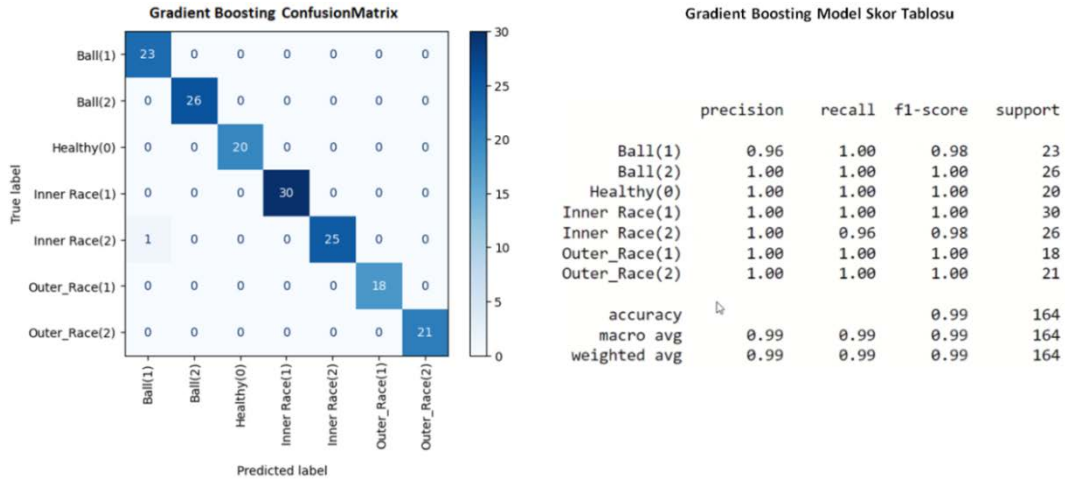
Gerçekleştirilecek anomali tespiti için sürücünden elde edilen verilerin hassasiyetinin düşük olması sebebi ile kestirimci bakım için geliştirilecek makine öğrenmesi algoritmalarında, sürücü aracılığı ile elde edilen veri setinin kullanılmaması gerektiği sonucuna ulaşılmıştır. Bu sebeple, daha önce ESOGÜ Akıllı Fabrika ve Robotik Laboratuvarında elektrik motoru test yatağına ait elektrik motorundan elde edilmiş akım sensör verilerini barındıran offline bir veri seti üzerinde çalışılmaktadır. Verilerin elde edilmesinde hassas bir akım sensörü kullanıldığından 3 faz için ayrı ayrı elde edilen akım verilerinde tam sayıdan sonra gelen ondalıklı kısım altı hane olarak kaydedilmektedir.

Veri setinde 7 farklı örneklem grubu yer almaktadır. Her bir örneklem içerisinde 120 saniyelik 6.250 adet akım verisi bulunmaktadır. Bunlardan ilki "Healthy(0)" bilgisi ile etiketlenen ve elektrik motoru üzerinde bulunan sağlam rulman takımı ile elde edilen akım verilerini içeren verilerdir. Diğer 6 örneklem grubu, laboratuvar ortamında bilinçli bir şekilde oluşturulan hatalara sahip rulman takımlarının motor akım verilerini içermektedir. Diğer örneklem grupları ise "Outer Race (1)" bilgisi ile etiketlenen 0,5mm delik dış bileziğe sahip rulman, "Inner Race (1)" bilgisi ile etiketlenen 0,5mm delik iç bileziğe sahip rulman, "Ball (1)" bilgisi ile etiketlenen 0,5mm hatalı bilyeye sahip rulman, "Outer Race (2)" bilgisi ile etiketlenen 1,5mm delik dış bileziğe sahip rulman, "Inner Race (2)" bilgisi ile etiketlenen 1,5mm delik iç bileziğe sahip rulman ve "Ball (2)" bilgisi ile etiketlenen 1,5mm hatalı bilyeye sahip rulman verilerini içermektedir.

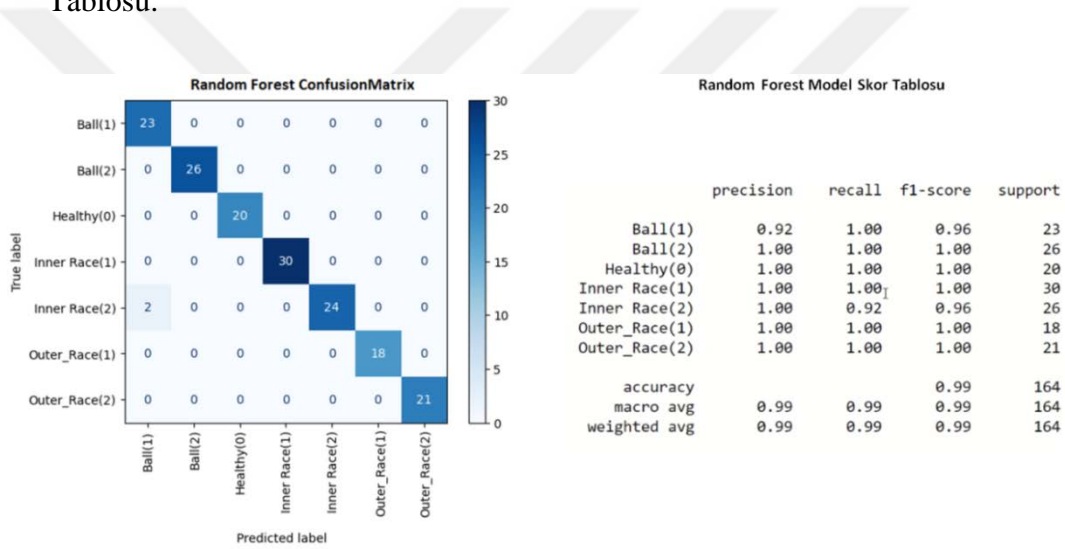
Kestirimci bakım için akım verileri üzerinde yapılacak anomali tespitinde Gradyan Artırma Makinesi (GBM), Rasgele Orman (RF), Karar Ağaçları (DT), Çok Katmanlı Algılayıcılar (MLP), K-En Yakın Komşuluk (K-nN), Hafif Gradyan Artırma Makinesi (LGBM), Lojistik regresyon (LR) ve Kategorik Artırma (CatBoost) makine öğrenmesi algoritmaları olmak üzere sekiz farklı makine öğrenmesi algoritması kullanılmaktadır. Makine öğrenmesi modellerinin geliştirilmesi sırasında veri setinin her hangi bir işleme

tabi tutulmadan direk kullanılması, aşırı öğrenme, eğitim ve test sürelerinin uzaması, veri setinde yüksek ilişkili tekrarlanan verilerin bulunması gibi olumsuz durumlar oluşturabilmektedir. Açıklayıcı olmayan veya gereksiz tahminlere sebep olacak olan bu tarz verilerden kurtulmak ve sağlıklı bir öğrenme gerçekleştirebilmek için veri setinde Öznitelik Seçimi (Feature Selection) yöntemleri uygulanmaktadır. Makine öğrenmesi algoritmalarının çalıştırılmasından önce, veri seti üzerinde Min, Max, Mean, Rms, Var, Std, Kurtosis, P2p öznitelik seçimi yöntemleri kullanılarak veri seti sadeleştirilmekte ve gereksiz görülen verilerden arındırılmaktadır. Bu sayede başarı oranı daha yüksek sonuçlar elde edilmektedir. Ayrıca veri ön işlemede, birbirinden bağımsız verilerin özelliklerine ait aralıkların normalleştirilmesi için Özellik Ölçeklendirme (Feature Scaling) yöntemi kullanılmaktadır. Veri örneklem grupları birleştirilerek elde edilen veri setinin %20'si test için ayrılmaktadır.

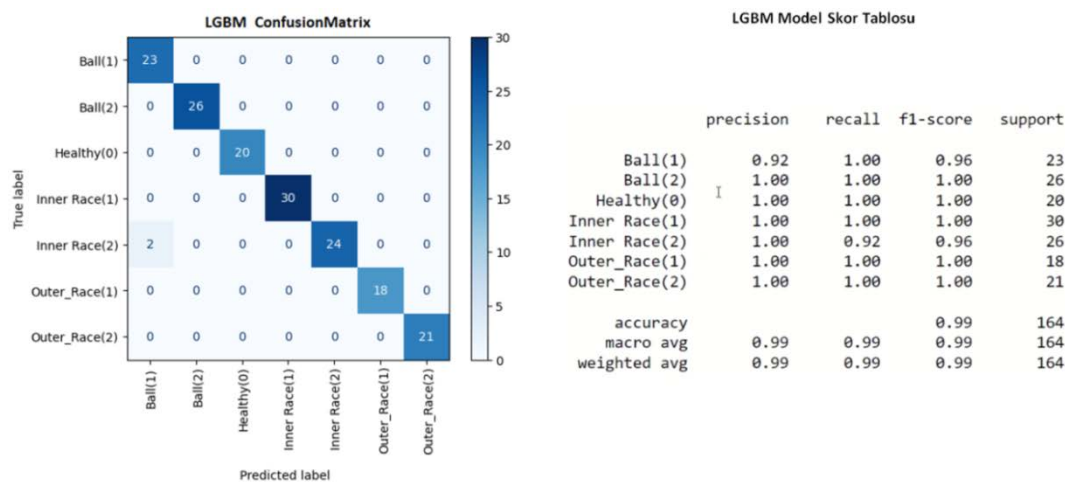
Makine öğrenmesi algoritmalarının çalıştırılmasının ardından her bir modele ait makine öğrenmesi gerçekleştirilmekte ve öğrenme sonuçlarına ulaşılmaktadır. Algoritmalar çalıştırılarak Confusion Matrixleri, Doğruluk Oranları ve Skor Tabloları oluşturulmaktadır. Şekil 4.7'de Gradyan Artırma Makinesi (GBM) algoritması, Şekil 4.8'de Rasgele Orman (RF) algoritması, Şekil 4.9'da Hafif Gradyan Artırma Makinesi (LGBM) algoritması, Şekil 4.10'da Kategorik Artırma (CatBoost) algoritması, Şekil 4.11'de Karar Ağaçları (DT) algoritması, Şekil 4.12'de K-En Yakın Komşuluk (K-nN) algoritması, Şekil 4.13'de Lojistik regresyon (LR) algoritması, Şekil 4.14'de Çok Katmanlı Algılayıcılar (MLP) algoritması Confusion Matrixleri ve Skor Tabloları görüntülenmektedir. Çalıştırılan makine öğrenmesi algoritmaları sonucunda en yüksek doğruluk oranına sahip olan model % 99,39 oran ile Gradyan Artırma Makinesi (GBM) algoritması olurken, en düşük doğruluk oranına sahip olan algoritma % 37,19 oran ile Çok Katmanlı Algılayıcılar (MLP) algoritması olarak görüntülenmektedir. Rasgele Orman (RF), Hafif Gradyan Artırma Makinesi (LGBM), Kategorik Artırma (CatBoost) algoritmaları % 98,78 doğruluk oranları ile en yüksek ikinci doğruluk oranlarına sahip olan algoritmalarlardır. Karar Ağaçları (DT) algoritması %97,56 ve K-En Yakın Komşuluk (K-nN) algoritması % 82,31 doğruluk oranları ile oldukça başarılı sonuçlar üretmektedirler. Lojistik regresyon (LR) ve Çok Katmanlı Algılayıcılar (MLP) makine öğrenmesi algoritmalarına ait başarı oranları ise gerçekleştirilecek anomali tespiti için yeterli düzeyde doğruluk oranlarına ulaşamamışlardır.



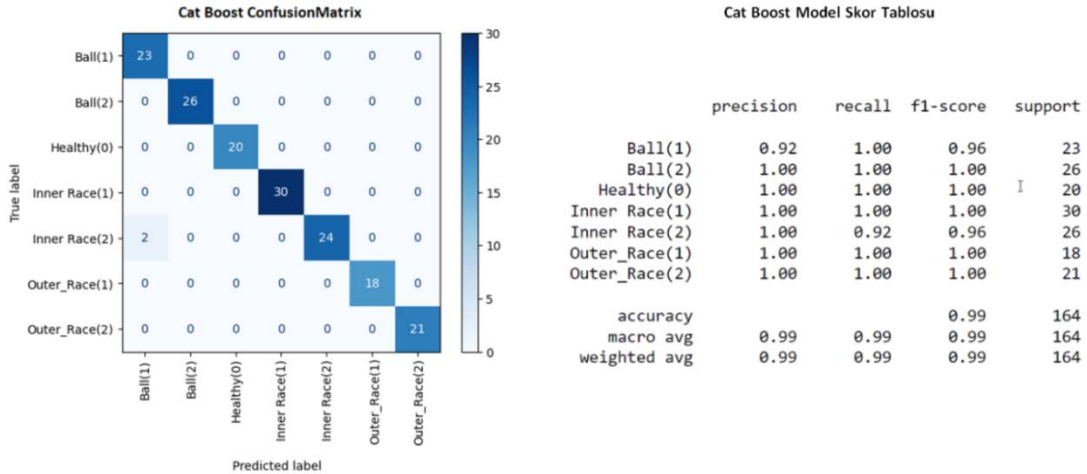
Şekil 4.7. Gradyan Artırma Makinesi (GBM) algoritması Confusion Matrixi ve Skor Tablosu.



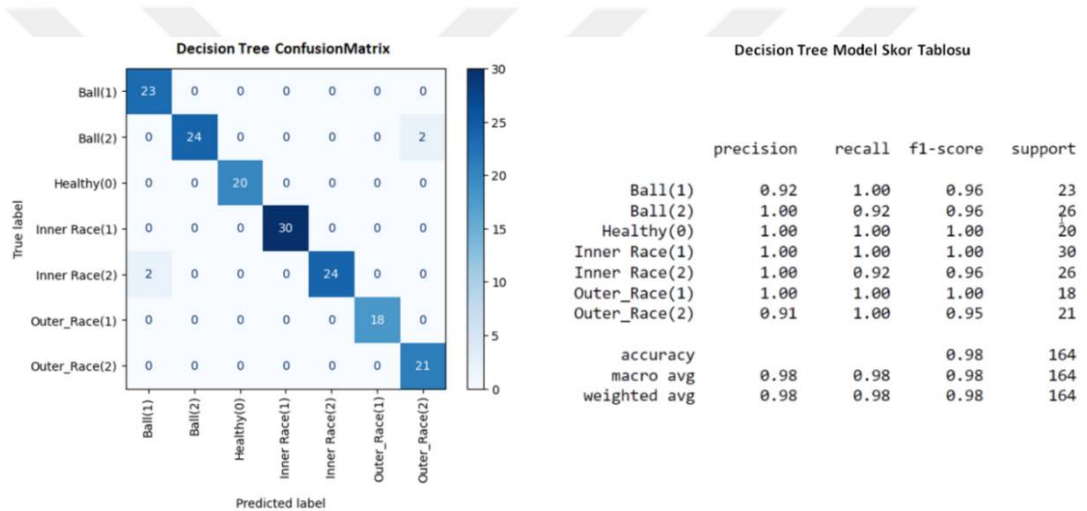
Şekil 4.8. Rasgele Orman (RF) algoritması Confusion Matrixi ve Skor Tablosu.



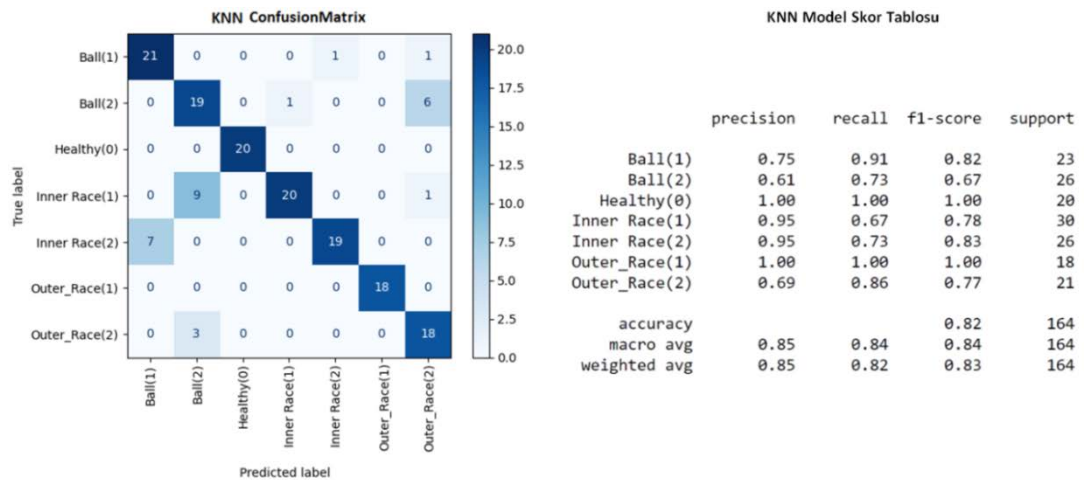
Şekil 4.9. Hafif Gradyan Artırma Makinesi (LGBM) algoritması Confusion Matrixi ve Skor Tablosu.



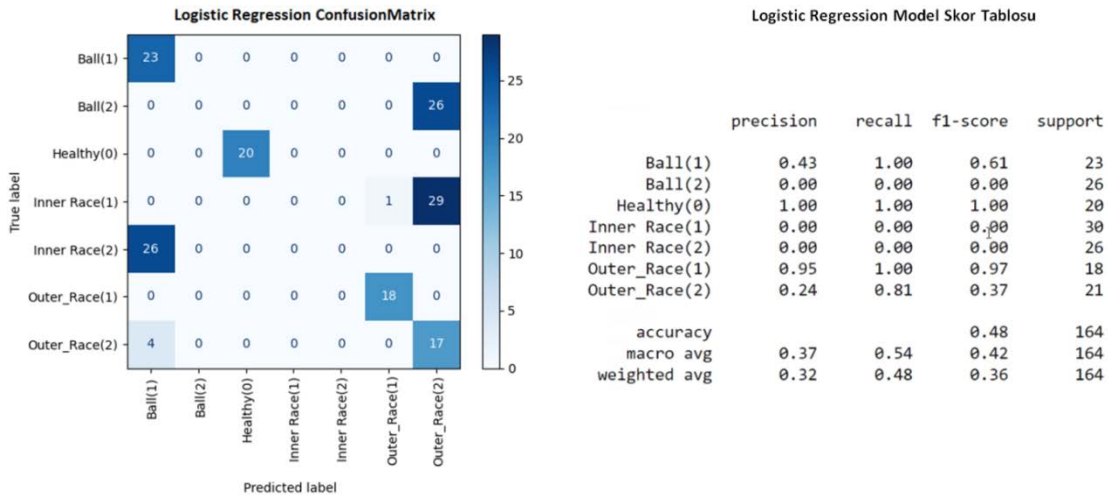
Şekil 4.10. Kategorik Artırma (CatBoost) algoritması Confusion Matrixi ve Skor Tablosu.



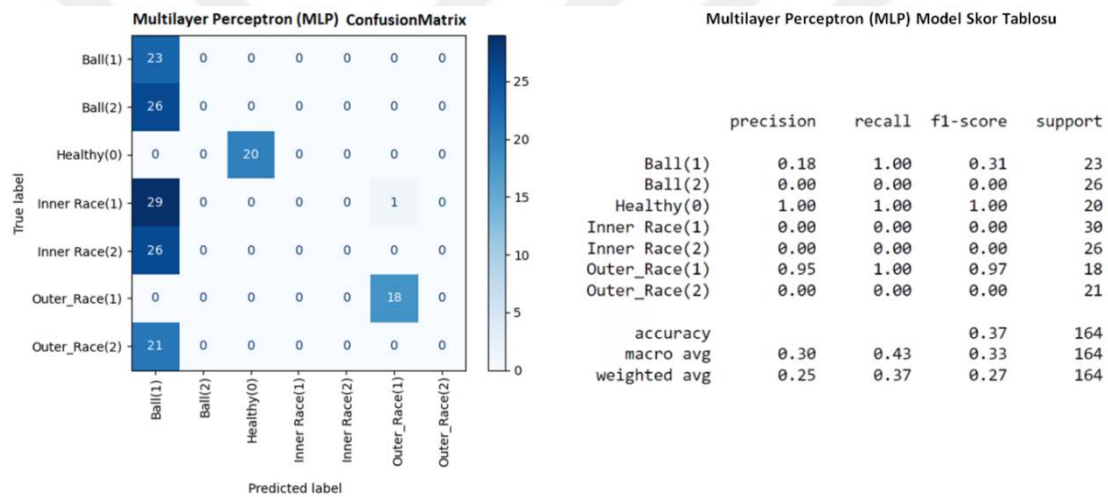
Şekil 4.11. Karar Ağaçları (DT) algoritması Confusion Matrixi ve Skor Tablosu.



Şekil 4.12. K-En Yakın Komşuluk (K-nN) algoritması Confusion Matrixi ve Skor Tablosu.



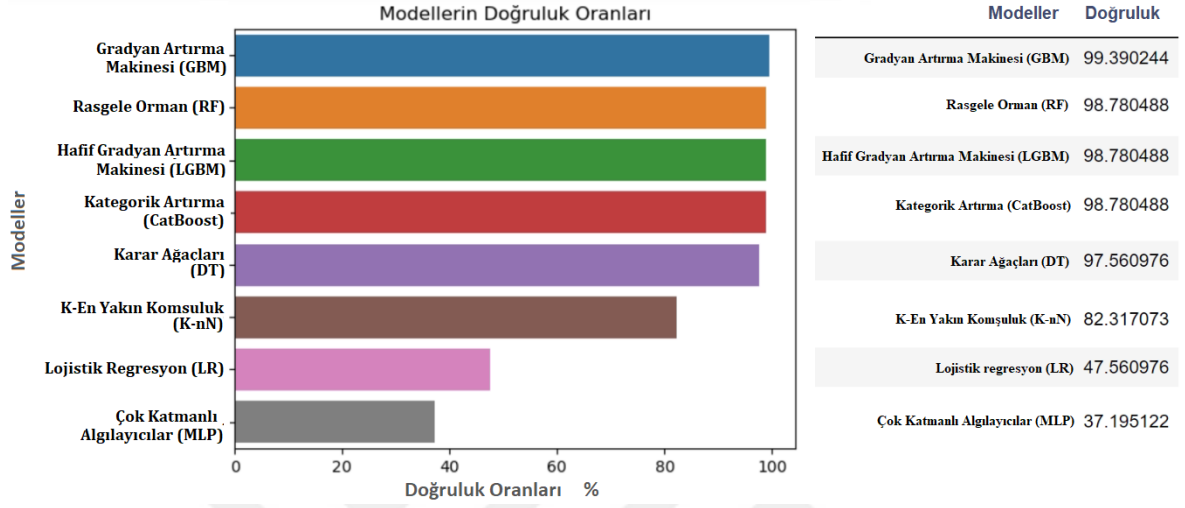
Şekil 4.13. Lojistik regresyon (LR) algoritması Confusion Matrixi ve Skor Tablosu.



Şekil 4.14. Çok Katmanlı Algılayıcılar (MLP) algoritması Confusion Matrixi ve Skor Tablosu.

Kesitimci bakım çalışmalarının yapılabilmesi için mevcut bulunan offline veri seti ile gerçekleştirilen anomali tespiti, 6 makine öğrenmesi modelinden elde edilen doğruluk oranı yüksek sonuçlarla tamamlanmaktadır. Bu sayede, akım verileri üzerindeki hassas değişimler takip edilerek motor rulman takımlarında oluşabilecek olan olası sorunlar için önceden kestirimler yapılabilir. Çalışma özelinde, Makine öğrenmesi modellerinin eğitimi ve test sonuçları, anomali tespiti için yeterli düzeyde tatmin edici sonuçlanmaktadır.

Şekil 4.15’de makine öğrenmesi modellerine ait en yüksek doğruluk oranından en düşük doğruluk oranına doğru sıralanmış olan grafiksel gösterimi ve modellerin sayısal değerler ile sıralanışı yer almaktadır.



Şekil 4.15. Makine öğrenmesi modellerinin doğruluk oranlarının grafiksel gösterimi ve sayısal değerleri.

## 5. SONUÇ VE ÖNERİLER

Yürütülen tez çalışması kapsamında, kestirimci bakım için Fiware tabanlı yeni bir ara katman mimarisi geliştirilerek anomali tespiti uygulaması gerçekleştirilmektedir. Donanım bileşenlerinin fiziksel bağlantı ve kurulumlarını takiben gerçekleştirilen düzenlemeler ile V20 motor sürücüsü ve Plc arasında ModbusRTU seri haberleşme sağlanmaktadır. Plc üzerinde bulunan dahili OPC-UA sunucusu sayesinde, akım verilerinin sunucu üzerinden yayınlanması sağlanmaktadır. İstemci olarak kullanılan OPC-UA IoT aracı ile sunucuya abone olunarak yayın bilgileri toplanmaktadır. OPC-UA haberleşme protokolü ile başarılı bir iletişim hattı kurularak, tasarımı gerçekleştirilen ara katman mimarisi üzerinde toplanan akım verileri işlenmekte ve PostgreSQL veri tabanına aktarılmaktadır.

Kestirimci bakım için ara katman mimarisi geliştirilmesi için gerçekleştirilen çalışmada, V20 sürücü aracılığı ile okunan verilerin hassasiyetinin düşük olması ve iletişim hattının kurulmasında haberleşme sorunları yaşanması, çalışma için olumsuzluklar oluşturmaktadır. Karşılaşılan hataların çözümü için uygulanan yöntemler ise veri kaynağından alınan verilerin hassasiyetini düşürürken, veriler arasındaki örneklem sürelerini artırmaktadır. PostgreSQL veritabanında kaydedilen akım verilerinin düşük hassasiyete sahip olması sebebi ile anomali tespiti uygulamasında daha önce IFARLAB motor test yatağında elde edilen offline veri seti kullanılmaktadır. Kullanılan veri setinin, motor üzerindeki her faz için daha hassas ve bağımsız ölçümler içermesi sayesinde daha hassas sonuçlar elde edilmektedir. Uygulamada sekiz farklı makine öğrenmesi modeli kullanılarak yüksek oranda başarıya sahip anomali tespiti gerçekleştirilmektedir.

Ara katman mimarisinin geliştirilmesinde Fiware platformuna ait bileşenlere yer verilmektedir. NGSİ-V2 veri modeline dayalı Fiware platformu ve bileşenleri, kullanılan donanımlar ve diğer yazılım bileşenleri ile tam uyum içerisinde çalışmaktadır. Fiware ve bileşenlerinin, ücretsiz açık kaynak kod sistemine sahip olması ve diğer bileşenler ile entegrasyonunun kolay olması, sağladığı en önemli avantajlar arasında yer almaktadır. Bu sebeple, hem endüstriyel hem de akademik alanda, gelecekte gerçekleştirilecek çalışmalarda daha fazla tercih sebebi olacağı düşünülmektedir.

Fiware bileşenleri ile geliştirilecek gelecek ara katman çalışmalarında, V20 motor sürücüsünün Modbus RTU iletişim yeteneğinin sınırlılıkları ve ürettiği akım verilerinin hassas değerler olmaması sebebi ile sensör verilerinin toplanmasında, V20 motor sürücüsü yerine daha hassas algılayıcılar kullanılarak ayrıntılı verilerin toplanması önerilmektedir. Bu sayede kestirimci bakım çalışmalarında daha başarılı sonuçlar elde edileceği düşünülmektedir. Ayrıca, kullanılan makine öğrenmesi modelleri dışındaki diğer modellerin kullanılması ile anomali tespiti için farklı yaklaşımlar gerçekleştirilebileceği düşünülmektedir.



## KAYNAKLAR DİZİNİ

- Agarwal, P., & Alam, M. (2020). *Investigating IoT Middleware Platforms for Smart Application Development*. *Smart Cities—Opportunities and Challenges*, pp.231-244. Doi:10.1007/978-981-15-2545-2\_21.
- Aizaga-Villon, X., Alarcon-Ballesteros, K., Cordova-Garcia, J., Padilla, V. S. & Velasquez, W. (2022). FIWARE-Based Telemedicine Apps Modeling for Patients' Data Management. *IEEE Engineering Management Review*, Vol. 50, No. 2, pp. 173-188. Doi: 10.1109/EMR.2022.3169991.
- Akarsu, Y. & Dilbaz Alacahan, N. (2021). İct Kullanımının İş Gücü Piyasalarına Etkisi: Türkiye Analizi. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 26(4), 445-456. <https://dergipark.org.tr/tr/pub/sduiibfd/issue/65632/976546>
- Akkaya, S., Akbatı, O., & Ergenc, A. (2015). *Endüstriyel Sistemler için FPGA Tabanlı Modbus Ağ Geçidi Tasarımı ve Uygulaması - Design and Application of a FPGA-based Modbus Gateway for Industrial Systems*. TOK'2015 Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Denizli, Turkey, Sayfa: 445-451
- Alan, A., & Karabatak, M. (2020). *Veri Seti - Sınıflandırma İlişkisinde Performansa Etki Eden Faktörlerin Değerlendirilmesi*. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, Cilt. 32 (2) , 531-540. Doi: 10.35234/fumbd.738007
- Alonso, A., Pozo, A., Cantera, J. M., Vega, F. & Hierro, J. (2018). *Industrial Data Space Architecture Implementation Using FIWARE*. *Sensors*, Vol. 18(7). 2226. Doi: 10.3390/s18072226.
- Aslam, A., & Curry, E. (2021) *A Survey on Object Detection for the Internet of Multimedia Things (IoMT) using Deep Learning and Event-based Middleware: Approaches, Challenges, and Future Directions*. *Image and Vision Computing*, Vol.106. <https://doi.org/10.1016/j.imavis.2020.104095>
- Araujo, V., Mitra, K., Saguna, S., & Ahlund,C. (2019). *Performance evaluation of FIWARE: A cloud-based IoT platform for smart cities*. *Journal of Parallel and Distributed Computing*, Vol. 132, Pages 250-261. <https://doi.org/10.1016/j.jpdc.2018.12.010>.
- Armando, N., Sinche, S., Rodrigues, A., Silva, J. S. & Boavida, F. (2022). *IoT Management Services: A Comparative Assessment of Popular FIWARE Agents*. *IEEE Ninth International Conference on Communications and Electronics*, Nha Trang, Vietnam, pp. 81-86, Doi: 10.1109/ICCE55644.2022.9852047.

## KAYNAKLAR DİZİNİ

- Barriga, J. A., Clemente, P. J., Hernandez J., & Perez-Toledano M. A. (2022), *SimulateIoT-FIWARE: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments on FIWARE*. IEEE Access, vol. 10, pp. 7800-7822. Doi: 10.1109/ACCESS.2022.3142894.
- Benayache, A., Bilami, A., Barkat, S., Lorenz, P., & Taleb, H. (2019). *MsM: A microservice middleware for smart WSN-based IoT application*. Journal of Network and Computer Applications, Vol. 144, pp 138-154. <https://doi.org/10.1016/j.jnca.2019.06.015>.
- Baca Gomez Y. R., Esquivel H. E., Rebollar A. M., & Vasquez D. V. (2019). *A Novel Air Quality Monitoring Unit Using Cloudino and FIWARE Technologies*. Mathematical and Computational Applications. Vol. 24(1) 15. <https://doi.org/10.3390/mca24010015>
- Bauer, M., Cirillo, F., Fürst, J., Solmaz, G., & Kovacs, E. (2021). *Urban Digital Twins – A FIWARE-based model*. at – Automatisierungstechnik, 69 (12), 1106-1115. <http://doi.org/10.1515/auto-2021-0083>.
- BitDegree. (2023). *Python Kütüphaneleri: Hangisi Ne İşe Yardımcı Olur?* Erişim: <https://tr.bitdegree.org/tutorial/python-kutuphaneleri/>, Erişim Tarihi: 01.05.2023
- Brownlee J. (2019). *How to Choose a Feature Selection Method For Machine Learning*. Erişim: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>, Erişim: 01.05.2023
- Bouloukakakis, G., Georgantas, N., Ntumba, P., & Issarny, V. (2019). *Automated synthesis of mediators for middleware-layer protocol interoperability in the IoT*. Future Generation Computer Systems, Vol. 101, pp. 1271-1294, <https://doi.org/10.1016/j.future.2019.05.064>.
- Cardoso, J., Pereira, C., Aguiar, A., & Morla, R. (2017). *Benchmarking IoT middleware platforms*. IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks, Macau, China, pp. 1-7, doi: 10.1109/WoWMoM.2017.7974339
- Cavalcanti, D., Carvalho, R., & Rosa, N. (2021). *Adaptive Middleware of Things*. 2021 IEEE Symposium on Computers and Communications, Athens, Greece, pp. 1-6. Doi:10.1109/ISCC53001.2021.9631408.

## KAYNAKLAR DİZİNİ

- Cavalieri, S., Salafia, M. G., & Scropo, M. S. (2019). *Integrating OPC UA with web technologies to enhance interoperability*. Computer Standards & Interfaces, Vol. 61, pp. 45-64. <https://doi.org/10.1016/j.csi.2018.04.004>.
- Celesti A., Fazio M., Marquez F. G., Glikson A., Mauwa H., Bagula A., Celesti F., & Villari M. (2019). *How to Develop IoT Cloud e-Health Systems Based on FIWARE: A Lesson Learnt*. Journal of Sensor and Actuator Networks, Vol. 8(1):7. <https://doi.org/10.3390/jsan8010007>
- Cirillo, F., Solmaz, G., Berz, E. L., Bauer, M., Cheng B., & Kovacs, E. (2019). *A Standard-Based Open Source IoT Platform: FIWARE*. IEEE Internet of Things Magazine, Vol. 2, No. 3, pp. 12-18, Doi: 10.1109/IOTM.0001.1800022.
- Coito, T., Martins, M., Viegas, J., Firme, B., Figueiredo, J., Vieira, S., & Sousa, J., (2020). *A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation*. Computers in Industry. Vol. 123. <https://doi.org/10.1016/j.compind.2020.103329>.
- Conde, J., Munoz-Arcentales, A., Alonso, A., Lopez-Pernas S., & Salvachua, J. (2022). *Modeling Digital Twin Data and Architecture: A Building Guide With FIWARE as Enabling Technology*. IEEE Internet Computing, Vol. 26, No. 3, pp. 7-14. Doi: 10.1109/MIC.2021.3056923.
- Cruz, M. A. A., Rodrigues, J. J. P. C., Al-Muhtadi, J., Korotaev, V. V., & Albuquerque, V. H. C. (2018). *A Reference Model for Internet of Things Middleware*. IEEE Internet of Things Journal, Vol. 5 No. 2, pp. 871-883, Doi: 10.1109/JIOT.2018.2796561.
- Cruz, M. A. A., Rodrigues, J. J. P. C., Al-Muhtadi, J., Korotaev, V. V., & Albuquerque, V. H. C. (2021). *In.IoT—A New Middleware for Internet of Things*. IEEE Internet of Things Journal, Vol. 8 No. 10, pp. 7902-7911, Doi: 10.1109/JIOT.2020.3041699.
- Çetin, F., & Yörükeren, N. (2020). *Belirli Güç Kalitesi Problemlerinin Nesnelere İnterneti Teknolojileri ile Tespiti ve Bulut Tabanlı Sunuculara Entegrasyonu*. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, Cilt.8 Sayı.4, ss.2321-2336. Doi: 10.29130/dubited.689757
- Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). *Comparison of Discrimination Methods for the Classification of Tumors using Gene Expression Data*. Journal of the American Statistical Association, 97(457), pp. 77-87.

## KAYNAKLAR DİZİNİ

- Duman, H. S., & Başaraner, M. (2022). *Şekil göstergeleri ve topluluk öğrenmesi sınıflandırma algoritmaları ile bina detaylarının şekil karmaşıklık analizi*. *Geomatik*, 7(3) , 197-208. Doi:10.29128/geomatik.947334
- Dündar, D. R., Sarıçiçek, İ., Çınar, E., & Yazıcı, A. (2021). *Kestirimci Bakımda Makine Öğrenmesi: Literatür Araştırması*. *Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 29 (2) , 256-276. Doi: 10.31796/ogummf.873963
- Eken, S., Kaya, F., Sayar, A., & Kavak, A. (2014). *Doküman Tabanlı NoSQL Veritabanları: MongoDB ve CouchDB yatay ölçeklenebilirlik karşılaştırması*. *Mühendislik ve Teknoloji Sempozyumu, Ankara, Türkiye*
- Erdoğan, H., Küçük, K., & Khan, S. A. (2020). *Endüstriyel IoT Bulut Uygulamaları için Düşük Maliyetli Modbus/MQTT Ağ Geçidi Tasarımı ve Gerçekleştirilmesi*. *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, Cilt.7, Sayı.1, ss.170-183. Doi: 10.35193/bseufbd.708445
- Farahzadi, A., Shams, P., Rezazadeh, J., & Farahbakhsh, R. (2017). *Middleware Technologies for Cloud of Things - a survey*. *Digital Communications and Networks Elsevier*, Vol 4. Doi:10.1016/j.dcan.2017.04.005.
- Fernandez P., Santana J.M., Ortega S., Trujillo A., Suarez J.P., Dominguez C., Santana J., & Sanchez A. (2016). *SmartPort: A Platform for Sensor Data Monitoring in a Seaport Based on FIWARE*. *Sensors*. Vol.16(3) 417. <https://doi.org/10.3390/s16030417>
- Ferreira, D., Corista, P., Gıao, J., Ghimire, S., SarraipJ., & Jardim-Gonçalves, R. (2017). *Towards smart agriculture using FIWARE enablers*. *International Conference on Engineering, Technology and Innovation, Madeira, Portugal, 2017*, pp. 1544-1551, doi: 10.1109/ICE.2017.8280066.
- Fiware – Orion. (2023). *Orion Context Broker*. Erişim: <https://fiware-orion.readthedocs.io/en/master/>, Erişim Tarihi: 01.05.2023.
- Fiware – OPC UA Agent. (2023). *OPC UA Agent*. Erişim: <https://iotagent-opcu.readthedocs.io/en/latest/index.html>, Erişim Tarihi: 01.05.2023.
- Fiware – Draco. (2023). *Fiware Draco*. Erişim: <https://fiware-draco.readthedocs.io/en/latest/>, Erişim Tarihi: 01.05.2023

## KAYNAKLAR DİZİNİ

- Fonseca, J. M. C., & Marquez, F. G. (2018). *FIWARE-NGSI v2 Specification*. Erişim: <https://fiware.github.io/specifications/ngsiv2/stable/>, Erişim Tarihi: 01.05.2023.
- Gonzalez, I., Calderon, A. J., Figueiredo, J., & Sousa, J.M.C. (2019). *A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments*. *Electronics*, 8(5), 510. <https://doi.org/10.3390/electronics8050510>.
- Hail, M. A., Pösse, I., & Fischer, S. (2022). *Integration of FIWARE and IoT based Named Data Networking (IoT-NDN)*. In *Proceedings of the 11th International Conference on Sensor Networks*, Vol.1 pp. 184-190 <https://doi.org/10.5220/0010936200003118>.
- Hosmer Jr., D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. (3rd Edition), John Wiley & Sons, 510s.
- IFARLAB. (2023). *Eskisehir Osmangazi University Intelligent Factory and Robotics Laboratory*. Erişim: <https://ifarlab.ogu.edu.tr/>, Erişim tarihi: 01.05.2023.
- Iaknomakis, E., Kotsiantis, S., & Tampakas, V. (2005). *Text Classification Using Machine Learning Techniques*. *WSEAS transactions on computers*. 4(8), pp. 966-974.
- Joseph, T., Jenu, R., Assis, A. K., Kumar, V. A. S., Sasi, P. M., & Alexander, G. (2017). *IoT middleware for smart city: (An integrated and centrally managed IoT middleware for smart city)*. *IEEE Region 10 Symposium*, Cochin, India, pp. 1-5, Doi:10.1109/TENCONSpring.2017.8070054.
- Jupyter. (2023). *JupyterLab: A Next-Generation Notebook Interface*. Erişim: <https://jupyter.org/>, Erişim Tarihi: 01.05.2023
- Kalay, S., Çınar E., & Sarıçiçek, İ. (2022). *A Comparison of Data Imputation Methods Utilizing Machine Learning for a New IoT System Platform*. *8th International Conference on Control, Decision and Information Technologies*, Istanbul, Turkey, pp. 69-74, Doi: 10.1109/CoDIT55151.2022.9804113.
- Kamienski, C., Soininen, J.-P., Taumberger, M., Dantas, R., Toscano, A., Salmon Cinotti, T., Filev Maia, R., & Torre Neto, A. (2019). *Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture*. *Sensors*, Vol. 19, 276. <https://doi.org/10.3390/s19020276>

## KAYNAKLAR DİZİNİ

- Karabulut, Y. E., & Küçüksille, E. U. (2018). *Twitter Profesyonel İzleme ve Analiz Aracı*. Teknik Bilimler Dergisi, Cilt.8, Sayı.2, ss. 17-24
- Karasoy B., & Çınar S. (2014). *Dağıtık Sistemler İçin Haberleşme Otomasyon Ara Katmanı: ULAK*. Proceedings of the 8th Turkish National Software Engineering Symposium, Güzelyurt, KKTC, Turkey, September 8-10. Vol.1221
- Khandelval P. (2017). *Which algorithm takes the crown: Light GBM vs XGBoost?* Erişim adresi: <https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/>, Erişim Tarihi: 01.05.2023.
- Kılıç, A. (2022). *Nesnelerin İnterneti Kapsamında Kullanılan Ara Katman Yazılımlarına Yönelik Ağ Benzetimi*. Konya Mühendislik Bilimleri Dergisi, Cilt. 10, 52-60. <https://doi.org/10.36306/konjes.1076950>
- Kuş, İ., Bozkurt Keser, S., & Yolaçan, E. (2021). *Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması*. Avrupa Bilim ve Teknoloji Dergisi, Sayı 31 Sayfa 725-734. Doi: 10.31590/ejosat.971875
- Lee, B., Kim, D., Yang, H., & Oh, S. (2017) *Model transformation between OPC UA and UML*. Computer Standards & Interfaces, Vol. 50, pp. 236-250. <https://doi.org/10.1016/j.csi.2016.09.004>.
- Liu, C., Vengayil, H., Lu, Y., & Xu, X. (2019). *A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect*. Journal of Manufacturing Systems, Vol. 51, pp. 61-74. <https://doi.org/10.1016/j.jmsy.2019.04.006>.
- Llopis, J. A., Mena, M., Criado J., & Iribarne, L. (2021). *MI-FIWARE: A web component development method for FIWARE using microservices*. IEEE 45th Annual Computers, Software, and Applications Conference, Madrid, Spain, pp. 1058-1065, Doi: 10.1109/COMPSAC51774.2021.00144.
- Lopez-Riquelme, J.A., Pavon-Pulido, N., Navarro-Hellin, H., Soto-Valles, F., & Torres-Sanchez, R. (2017). *A software architecture based on FIWARE cloud for Precision Agriculture*. Agricultural Water Management, Vol. 183, Pages 123-135, <https://doi.org/10.1016/j.agwat.2016.10.020>.

**KAYNAKLAR DİZİNİ**

- Loss, S., Singh, H. P., Cacho N., & Lopes, F. (2021). *Using FIWARE and Blockchain in Post Pandemic Vaccination Scenario*. Third International Conference on Blockchain Computing and Applications, Tartu, Estonia, pp. 143-150, Doi: 10.1109/BCCA53669.2021.9656972.
- Loss, S., Singh, H.P., Cacho, N., & Lopes, F. (2022). *Using FIWARE and blockchain in smart cities solutions*. Cluster Computing. <https://doi.org/10.1007/s10586-022-03732-x>
- Marguglio, A., & Aiello G. (2018). *OPC UA IoT Agent and Facility Enabler: Bringing Manufacturing Data into Context*. Erişim: <https://www.fiware.org/2018/11/05/opc-ua-iot-agent-and-facility-enabler-bringing-manufacturing-data-into-context/>, Erişim Tarihi: 01.05.2023.
- Martinez R., Pastor J.A., Alvarez B., & Iborra A. (2016). *A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture*. Sensors, Vol. 16 (11), 1979. <https://doi.org/10.3390/s16111979>
- Medeiros, R., Fernandes, S. R., & Queiroz, P. (2022). *Middleware for the Internet of Things: a systematic literature review*. Journal of Universal Computer Science. 28. 54-79. DOI: 10.3897/jucs.71693
- Muniraj, S.P. & Xu, X. (2021). *An Implementation of OPC UA for Machine-to-Machine Communications in a Smart Factory*. Procedia Manufacturing, Vol. 53, pp. 52-58. 10.1016/j.promfg.2021.06.009.
- Mourtzis, D., Milas, N., & Athinaios, N. (2018). *Towards Machine Shop 4.0: A General Machine Model for CNC machine-tools through OPC-UA*. Procedia CIRP, Vol. 78, pp. 301-306. <https://doi.org/10.1016/j.procir.2018.09.045>.
- Namba, Y., & Nishi, H. (2022). *Personal Data Access and Distribution Management Extension to FIWARE*. IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, pp. 1-6, Doi: 10.1109/IECON49645.2022.9968512.
- Ngu, A. H., Gutierrez, M., Metsis, V., Nepal S., & Sheng, Q. Z. (2017). *IoT Middleware: A Survey on Issues and Enabling Technologies*. IEEE Internet of Things Journal, Vol. 4, No. 1, pp. 1-20. Doi:10.1109/JIOT.2016.2615180.

## KAYNAKLAR DİZİNİ

- Özdemir, E., & Ballı, S. (2020). *Türkiye Erkekler Basketbol Ligi Maç Sonuçlarının Makine Öğrenmesi Yöntemleri İle Tahmini*. Mühendislik Bilimleri ve Tasarım Dergisi, Cilt.8, Sayı.3, ss. 740-752 . Doi: 10.21923/jesd.723109
- Özdoğan, E., & Erdem. O. A. (2020). *Nesnelerin İnterneti İçin Hibrit Uygulama Katmanı Protokol Tasarımı*. Mühendislik Bilimleri ve Tasarım Dergisi, Cilt.8, Sayı.1, s. 285–304. Doi: 10.21923/jesd.530295
- Özlüer Başer, B., Yangın, M., & Sarıdaş, E. S. (2021). *Makine Öğrenmesi Teknikleriyle Diyabet Hastalığının Sınıflandırılması*. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 25 (1) , 112-120. Doi: 10.19113/sdufenbed.842460
- Postgresql. (2023). *About-What is PostgreSQL?* Erişim: <https://fiware-draco.readthedocs.io/en/latest/>, Erişim Tarihi: 01.05.2023
- Postman. (2023). *What is Postman?* Erişim: <https://www.postman.com/>, Erişim Tarihi: 01.05.2023
- Pozo, A., Alonso, A., & Salvachua, J. (2020). *Evaluation of an IoT Application-Scoped Access Control Model over a Publish/Subscribe Architecture Based on FIWARE*. Sensors, Vol. 20. 4341. 10.3390/s20154341.
- Pradeep, P., Krishnamoorthy, P., & Vasilakos, A. V. (2021). *A holistic approach to a context-aware IoT ecosystem with Adaptive Ubiquitous Middleware*. Pervasive and Mobile Computing, Vol. 72. <https://doi.org/10.1016/j.pmcj.2021.101342>.
- Preventis, A., Stravoskoufos, K., Sotiriadis, S., & Petrakis, E. (2016). *IoT-A and FIWARE: Bridging the Barriers between the Cloud and IoT Systems Design and Implementation*. Proceedings of the 6th International Conference on Cloud Computing and Services Science – Vol.1, 146-153. Doi: 10.5220/0005912001460153.
- Profanter, S., Tekat, A., Dorofeev, K., Rickert, M., & Knoll A. (2019). *OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols*. IEEE International Conference on Industrial Technology, Melbourne, VIC, Australia, pp. 955-962, Doi: 10.1109/ICIT.2019.8755050
- Razzaque, M. A., Milojevic-Jevric, M., Palade A., & Clarke, S. (2016). *Middleware for Internet of Things: A Survey*. IEEE Internet of Things Journal, Vol. 3, No. 1, pp.70-95. Doi: 10.1109/JIOT.2015.2498900.

## KAYNAKLAR DİZİNİ

- Rodriguez, M.A., Cuenca, L., & Ortiz, A. (2018). *FIWARE Open Source Standard Platform in Smart Farming - A Review*. IFIP Advances in Information and Communication Technology, Vol 534, pp. 581-589 [https://doi.org/10.1007/978-3-319-99127-6\\_50](https://doi.org/10.1007/978-3-319-99127-6_50)
- Sang, G., Xu, L., De Vrieze, P., & Bai, Y. (2020). *Towards Predictive Maintenance for Flexible Manufacturing Using FIWARE*. Systems Engineering Workshops. Lecture Notes in Business Information Processing, Vol 382. [https://doi.org/10.1007/978-3-030-49165-9\\_2](https://doi.org/10.1007/978-3-030-49165-9_2)
- Saygın, E., & Baykara, M. (2021). *Karaciğer Yetmezliği Teşhisinde Özellik Seçimi Kullanarak Makine Öğrenmesi Yöntemlerinin Başarılarının Ölçülmesi*. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 33(2), 367-377. Doi: 10.35234/fumbd.832264
- Scipy. (2023). *SciPy User Guide*. Erişim: <https://docs.scipy.org/doc/scipy/tutorial/index.html>, Erişim Tarihi: 01.05.2023
- Sevli, O. (2022). *Farklı Sınıflandırıcılar ve Yeniden Örnekleme Teknikleri Kullanılarak Kalp Hastalığı Teşhisine Yönelik Karşılaştırmalı Bir Çalışma*. Journal of Intelligent Systems: Theory and Applications, 5 (2), 92-105. Doi: 10.38016/jista.1069541
- Shershneu, M., & Oskin, A. (2020). *Postman Platform For Api Development In The Mobile Application "Musicians Of Russia"*. Materials Of XII Junior Researchers Conference
- Siemens. (2023a). *Industry Mall, 1LE1003-1AC42-2AA4 DataSheet*. Erişim: <https://support.industry.siemens.com/cs/WW/en/ps/1LE1003-1AC42-2AA4/>, Erişim Tarihi: 01.05.2023.
- Siemens. (2023b). *Sinamics V20 Datasheet*. Erişim: [www.siemens.com.tr/SinamicsV20](http://www.siemens.com.tr/SinamicsV20), Erişim Tarihi: 01.05.2023.
- Siemens. (2023c). *Sinamics V20 Devreye Alma Dökümanı*. Erişim: [www.siemens.com.tr/SinamicsV20](http://www.siemens.com.tr/SinamicsV20), Erişim Tarihi: 01.05.2023.
- Silva, D., Carvalho, L.I., Soares, J., & Sofia, R.C. (2021) *A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA*. Applied Sciences, 11, 4879. <https://doi.org/10.3390/app11114879>

## KAYNAKLAR DİZİNİ

- Sinanovic, A., Meskovic, E., Mujcic A., & Suljanovic, N. (2022). *Smart city use case development based on FIWARE technology*. 30th Telecommunications Forum Belgrade, Serbia, pp. 1-4. Doi: 10.1109/TELFOR56187.2022.9983683.
- Schneider Electric. (2023a). *EcoStruxure Machine Expert (SoMachine)*. Erişim: <https://www.se.com/ww/en/product-range/2226-ecostruxure-machine-expert-somachine/>, Erişim Tarihi: 01.05.2023.
- Schneider Electric. (2023b). *Vijeo Designer*. Erişim: <https://www.se.com/ww/en/product-range/1054-vijeo-designer/>, Erişim Tarihi: 01.05.2023.
- Schneider Electric. (2023c). *M241 kontrolör 40 GÇ transistör PNP Ethernet - TM241CE40T* . Erişim: <https://www.se.com/tr/tr/product/TM241CE40T/>, Erişim Tarihi: 01.05.2023.
- Schneider Electric. (2023d). *HMIGXU3500 - 7 inch wide screen*. Erişim: <https://www.se.com/id/en/product/HMIGXU3500/7-inch-wide-screen-basic-model-1-serial-port-embedded-rtc/>, Erişim Tarihi: 01.05.2023.
- Sousa, P.R., Magalhaes, L., Resende, J. S., Martins, R., & Antunes, L. (2021). *Provisioning, Authentication and Secure Communications for IoT Devices on FIWARE*. Sensors, Vol. 21, 5898. <https://doi.org/10.3390/s21175898>
- Tamboli, S., Rawale, M., Thoraiet, R., & Agashe, S. (2015). *Implementation of Modbus RTU and Modbus TCP communication using Siemens S7-1200 Plc for batch process*. International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, Avadi, India, pp. 258-263, Doi: 10.1109/ICSTM.2015.7225424.
- Tütüncü, T. E., & Gürsakal, S. (2023). *Kredi Temerrüt Riskini Tahmin Etmede Makine Öğrenme Algoritmalarının Karşılaştırılması*. Avrupa Bilim ve Teknoloji Dergisi, (50) , 14-22 . Doi: 10.31590/ejosat.1171611
- Urrea, C., Morales, C., & Kern,J. (2016) *Implementation of error detection and correction in the Modbus-RTU serial protocol*. International Journal of Critical Infrastructure Protection, Vol. 15, pp. 27-37, <https://doi.org/10.1016/j.ijcip.2016.07.001>.
- Vaclavova, A., Strelec, P., Horak, T., Kebisek, M., Tanuska, P., & Huraj, L. (2022). *Proposal for an IIoT Device Solution According to Industry 4.0 Concept*. Sensors, Vol.22, 325. <https://doi.org/10.3390/s22010325>

**KAYNAKLAR DİZİNİ**

- Velasquez, W., Tobar-Andrade L., & Cedeno-Campoverde, I. (2021). *Monitoring and Data Processing Architecture using the FIWARE Platform for a Renewable Energy Systems*. IEEE 11th Annual Computing and Communication Workshop and Conference, NV, USA, pp. 1383-1387, Doi: 10.1109/CCWC51732.2021.9376026.
- Vikash & Mishra, Lalita & Varma, Shirshu. (2021). *Middleware Technologies for Smart Wireless Sensor Networks towards Internet of Things: A Comparative Review*. Wireless Personal Communications, 116. Doi: 10.1007/s11277-020-07748-7.
- Villalon, D., Jimenez, L., Campos, Iglesia M., Aguirre, M., & Delgado Fernandez, T. (2022). *An IoT architecture for smart cities based on the FIWARE platform*. Revista de Ciencia y Tecnología. Pp.20-27. Doi: 10.36995/j.recyt.2022.38.003.
- Zamora-Izquierdo, M. A., Santa, J., Martinez, J. A., Martinez, V., Skarmeta, & Antonio F. (2019). *Smart farming IoT platform based on edge and cloud computing*. Biosystems Engineering, Vol. 177, pp 4-17. <https://doi.org/10.1016/j.biosystemseng.2018.10.014>.
- Zhang, J., Ma, M., Wang, P., & Sun, X. (2021). *Middleware for the Internet of Things: A survey on requirements, enabling technologies, and solutions*. Journal of Systems Architecture, Vol. 117. <https://doi.org/10.1016/j.sysarc.2021.102098>.