Kestirimci Bakım Sistemlerinde Veri Artırma Yöntemlerinin Geliştirilmesi ve Bir Uygulaması

Sena Kalay

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Kasım 2022

Development of Data Augmentation Methods for Predictive Maintenance Systems and an Application

Sena Kalay

MASTER OF SCIENCE THESIS

Department of Computer Engineering

November 2022

Kestirimci Bakım Sistemlerinde Veri Artırma Yöntemlerinin Geliştirilmesi ve Bir Uygulaması

Sena Kalay

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Bilgisayar Mühendisliği Anabilim Dalı
Yapay Zeka Bilim Dalında
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

Danışman: Dr. Öğr. Üyesi Eyüp Çinar
İkinci Danışman: Prof. Dr. İnci Sarıçiçek

Bu tez çalışması Tübitak 2232 Uluslararası Lider Araştırmacılar Projesi 118C252 no'lu projesi tarafından desteklenmiştir.

Kasım 2022

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Dr. Öğr. Üyesi Eyüp Çinar danışmanlığında hazırlamış olduğum "Kestirimci Bakım Sistemlerinde Veri Artırma Yöntemlerinin Geliştirilmesi ve Bir 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. 24/11/2022

Sena Kalay

İmza

ÖZET

Endüstri 4.0'ın beraberinde getirdiği dijitalleşme, nesnelerin interneti (Internet of Things - IoT) ve büyük veri kavramları ile toplanabilen veri miktarı ve bu verileri kullanan bilimsel araştırmalar günden güne artmaktadır. Toplanan verilerin sürekliliği, niteliği ve istatistiksel analiz gücü, kestirimci bakım sistemlerinin bakım planlama süreçlerinde ve arıza/anomali tespitinde önemli bir role sahiptir. Verilerin eksiksiz ve gerçek zamanlı olarak toplanabilmesi beklenirken ağ ve sensör arızaları, senkronizasyon hataları ve çevresel faktörler gibi çeşitli sebeplerle yazılamayan eksik (kayıp) veriler veri setlerinde boşluklar oluşturabilmektedir. Eksik veri içeren veri setleri yanlı tahminlere, hatalı sonuçlara ve veriye dayalı makine öğrenimi (Machine Learning - ML) modelleri için sorunlara neden olabilmektedir. Tutarlı ve güçlü bir istatiksel veri analizi için eksik veriler ele alınmalı ve araştırmalar eksiksiz bir veri seti üzerinde yürütülmelidir. Bu çalışmada, veri setlerindeki eksik verilerin en uygun regresyon-tabanlı makine öğrenmesi algoritması ile tamamlanması ve bu sürecin Apache Airflow platformu üzerinde Otomatik Makine Öğrenimi (Automated Machine Learning - AutoML) yaklaşımı ile otomatikleştirilmesi sağlanmıştır. Önerilen yöntemin bir uygulaması, Eskişehir Osmangazi Üniversitesi (ESOGÜ) Akıllı Fabrika ve Robotik Laboratuvarında (IFARLAB) tasarlanan IoT sistem platformuna bir veri zenginleştirme modülü olarak entegre edilerek sunulmuştur. Çalışma sonucunda, önerilen yöntemin zaman ve insan gücünden kazanç sağlayarak veri artırma maliyetinin düşmesine olanak sağladığı görülmektedir.

Anahtar Kelimeler: Eksik (Kayıp) veri, veri zenginleştirme, veri imputasyonu, otomatik makine öğrenimi, Apache Airflow.

SUMMARY

The amount of data that can be collected and scientific research using these data are increasing day by day with the concepts of digitalization, the Internet of Things (IoT), and big data brought along by Industry 4.0. The continuity, quality, and statistical analysis power of the collected data have an essential role in maintenance planning processes and fault/anomaly detection of predictive maintenance systems. While it is expected that the data can be collected completely and in real-time, missing data that cannot be written due to various reasons such as network and sensor failures, synchronization errors, and environmental factors can create gaps in the datasets. Datasets with missing data can cause biased predictions, erroneous results, and problems for data-driven machine learning (ML) models. For a consistent and robust statistical data analysis, missing data should be handled, and research should be conducted on a complete dataset. In this study, the missing data in the datasets have been completed with the most appropriate regression-based machine learning algorithm, and this process is automated with the Automated Machine Learning (AutoML) approach on the Apache Airflow platform. An application of the proposed method is presented by integrating it as a data augmentation module into the IoT system platform designed in Eskişehir Osmangazi University (ESOGU) Intelligent Factory and Robotics Laboratory (IFARLAB). As a result of the study, it is seen that the proposed method allows for reducing the cost of data augmentation by saving time and workforce.

Keywords: Missing data, data augmentation, data imputation, automated machine learning (AutoML), Apache Airflow.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	**Açıklama**
null, NaN, NA	Eksik Değer
R^2	Belirlilik Katsayısı
Kısaltmalar	**Açıklama**
API	Uygulama Programlama Arayüzü (**A**pplication **P**rogramming **I**nterface)
ATV	Otonom Taşıyıcı Araç (**A**utonomous **T**ransport **V**ehicle)
AutoML	Otomatik Makine Öğrenimi (**Auto**mated **M**achine **L**earning)
CPS	Siber-Fiziksel Sistem (**C**yber-**P**hysical **S**ystem)
CSV	Virgülle Ayrılmış Değerler (**C**omma **S**eparated **V**alues)
DTR	Karar Ağacı Regresyonu (**D**ecision **T**ree **R**egression)
FSM	Sonlu Durum Makineleri (**F**inite **S**tate **M**achine)
IFARLAB	Akıllı Fabrika ve Robotik Laboratuvarı (**I**ntelligent **Fa**ctory and **R**obotics **Lab**oratory)
IoT	Nesnelerin İnterneti (**I**nternet **o**f **T**hings)
KNN	k-En Yakın Komşu Regresyonu (**k**-**N**earest **N**eighbors)
KPI	Temel Performans Göstergesi (**K**ey **P**erformance **I**ndicator)
LCS	Düşük Maliyetli Sensör (**L**ow-**C**ost **S**ensor)
MAR	Rastgele Kayıp (**M**issing **a**t **R**andom)
MCAR	Tamamen Rastgele Kayıp (**M**issing **C**ompletely **a**t **R**andom)
MF	**M**iss **F**orest

SİMGELER VE KISALTMALAR DİZİNİ (devam)

Kısaltmalar	**Açıklama**
MICE	Zincirleme Denklemlerle Çok Değişkenli Atama (**M**ultivariate **I**mputation by **C**hained **E**quations)
ML	Makine Öğrenimi (**M**achine **L**earning)
MLR	Çoklu Doğrusal Regresyon (**M**ultiple **L**inear **R**egression)
MNAR	Rastgele Olmayan Kayıp (**M**issing **N**ot **a**t **R**andom)
MQTT	Mesaj Kuyruk Telemetri Taşıma (**M**essage **Q**ueuing **T**elemetry **T**ransport)
PdM	Kestirimci Bakım (**P**re**d**ictive **M**aintenance)
RMSE	Hata Kareleri Ortalamasının Karekökü (**R**oot **M**ean **S**quare **E**rror)
ROS	Robot İşletim Sistemi (**R**obot **O**perating **S**ystem)
RUL	Kalan Yararlı Ömür (**R**emaining **U**seful **L**ife)
SVM	Destek Vektör Makinesi (**S**upport **V**ector **M**achine)
SVR	Destek Vektör Regresyonu (**S**upport **V**ector **R**egression)
TPOT	Ağaç Tabanlı Ardışık Düzen Optimizasyon Aracı (**T**ree-based **P**ipeline **O**ptimization **T**ool)
VAE	Varyasyonel Otomatik Kodlayıcı (**V**ariational **A**uto**E**ncoder)
XGB	Ekstrem Gradyan Arttırma Regresyonu (e**X**treme **G**radient **B**oosting)

1. GİRİŞ VE AMAÇ

Üretim endüstrisindeki otomasyon teknolojilerinin mevcut trendini temsil eden ve uçtan uca birbirleri ile haberleşebilen, sensörler yardımıyla ortamı algılayabilen, verileri analiz ederek ihtiyaçları fark edebilen dijital teknoloji ve cihazlara dayanan Endüstri 4.0 Sanayi Devrimi (Xu vd., 2018), endüstride, lojistikte ve insan hayatında köklü bir değişim ve kolaylık sağlamıştır. Temel olarak nesnelerin interneti (Internet of Things - IoT), büyük veri, siber-fiziksel sistemler (Cyber-Physical System - CPS) ve bulut bilişim gibi kavramların üzerine oturtulan bu dijital devrim, birbirleri ile sürekli iletişim halinde olan akıllı üretim sistemlerini, gerçek zamanlı olarak uçtan uca izlenebilen cihazları ve her yönüyle optimize edilmiş imalat süreçlerini kapsamaktadır. Bu kavram ile, yeni dinamik iş akış süreçlerinin tasarımı, kaynak yönetimi, maliyet optimizasyonu ve verimli üretim teknikleri gibi alanlarda çalışmalar oldukça hız kazanmıştır.

IoT'nin sağladığı en önemli hizmetlerden biri, üretim alanındaki cihazlardan gerçek zamanlı veri toplama ve bu verileri işleyerek anlamlı bilgilere dönüştürme kabiliyetidir. Bu katkı, düşük maliyetli bakıma, kaliteli üretime, verimliliğe ve veriye dayalı kestirimci bakıma (Predictive Maintenance - PdM) olanak sağlamaktadır (Aheleroff vd., 2020). IoT sensörleri içeren ekipmanlardan çok büyük miktarda veri toplanabilmekte ve bu veriler hata/anomali tespitinde, bakım planlama süreçlerinde kullanılabilmektedir. Sensörlerden toplanan zaman serisi verilerinin sürekliliği, niteliği ve istatistiksel analiz gücü veri analizinde oldukça kritik bir role sahiptir. Verilerin eksiksiz ve gerçek zamanlı olarak elde edilmesi beklenmektedir. Ancak sensör ve ağ arızaları, senkronizasyon hataları ve çevresel faktörler nedeniyle bazı veriler yazılamamakta ve veri setlerinde boşluklar oluşmaktadır. Çeşitli sebeplerle yazılamayan ve "eksik veri" olarak adlandırılan bu veriler, karar vericiler için hatalı çıktılara, yanlı tahminlere sebep olmakta, sonuçların güvenilirliğini, tutarlılığını azaltabilmekte ve makine öğrenmesi (Machine Learning - ML) modellerinin üretim sistemlerinde çalışmasını engellemektedir. Güçlü ve uygun bir istatistiksel analiz için eksik veriler ele alınmalı ve çalışmalar veri seti eksiksiz hale getirilerek yürütülmelidir.

Birçok araştırmacının karşılaştığı temel sorunlardan biri olan eksik veri problemi çeşitli yöntemlerle ele alınmaktadır. Yaygın olarak kullanılan yöntem, eksik veri içeren kısımların veri setinden çıkarılmasıdır. Bu yöntem düşük maliyetli olmasına rağmen, eksik veri içeren kısımların silinmesi önemli veri kayıplarına sebep olmaktadır. Veri miktarındaki bu azalma araştırmanın analiz gücünü düşürebilmektedir. Doğru bir çıktı elde etmek için eksik verileri mümkün olduğunca gerçeğe en yakın şekilde atamak/impute etmek gerekmektedir. Eksik verileri atamak için birçok yöntem geliştirilmiştir. Makine öğrenimi algoritmaları ve istatistiksel hesaplamalar gibi çeşitli yaklaşımların, veri analistleri ve araştırmacılar tarafından veri ataması/imputasyonu için kullanıldığı ve bu yaklaşımların literatürde birçok çalışmada yer aldığı görülmektedir. Ancak literatür analiz edildiğinde, Otomatik Makine Öğrenimi (Automated Machine Learning - AutoML) yaklaşımının kullanıldığı herhangi bir eksik veri imputasyonu çalışmasına rastlanmamaktadır. Bu tez çalışması, literatürdeki bu eksikliği kapatan yenilikçi bir bilimsel çalışmadır.

Bu çalışmada, öncelikle ESOGÜ Akıllı Fabrika ve Robotik Laboratuvarında (IFARLAB, 2022) elektrik motorları durum izleme (condition monitoring) için tasarlanmış test yatağına ait motorlardan alınmış titreşim ve akım sensörlerinden elde edilen offline veri setleri üzerinde çalışılmıştır. Veri setleri üzerinde farklı konumlarda (rastgele, başlangıç, orta, son) ve farklı oranlarda (%10, %20, %30, %40) boşluklar oluşturulmuştur. Oluşturulan eksik veriler, Support Vector Regresyon (SVR), Decision Tree Regresyon (DTR), Ridge Regresyon, k-Nearest Neighbors Regresyon (KNN), Miss Forest (MF) ve XGBoost Regresyon (XGB) gibi altı regresyon-tabanlı ML algoritması ile impute edilerek doldurulmuştur. ML modellerin veri atamasındaki başarısı ve uygunluğu, tahmin edilen değerler ve bilinen değerler arasındaki Hata Kareleri Ortalamasının Karekökü (Root Mean Square Error - RMSE) ve Belirlilik Katsayısı (R squared – R^2) metrikleri ile analiz edilmiş ve karşılaştırılmıştır.

Gerekli ön hazırlıklar tamamlandıktan sonra veri atama sürecinin gerçek zamanlı akan sensör verileri üzerinde otomatikleştirilmesi hedeflenmiştir. Bu amaca yönelik olarak, mevcut IoT sistem platformuna entegre edilecek bir veri artırma modülü tasarlanmıştır. Bu modül, veri setine en uygun regresyon-tabanlı ML algoritmasının ve hiperparametrelerin

tespitini içeren Otomatik Makine Öğrenimi (AutoML) yaklaşımını ve veri atama süreçlerinin belirli zaman aralıklarında otomatik olarak gerçekleşmesini sağlayan açık kaynaklı iş akışı yönetim platformu Apache Airflow'u kapsamaktadır. Önerilen bu yöntem sayesinde, akan sensör verileri gruplar halinde okunarak eksik veri kontrolünden geçmekte ve eksik veri varlığında en uygun regresyon-tabanlı ML algoritması ile doldurulmaktadır. Tüm bu süreç insan müdahalesine gerek duymadan gerçekleşmektedir.

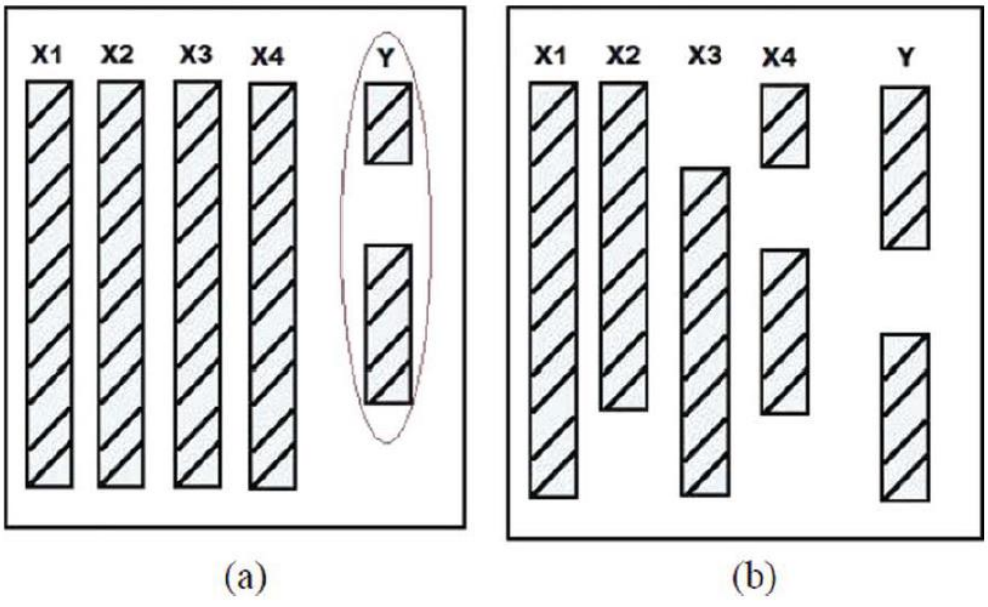
Takip eden bölümde, literatürde eksik veri problemini ele alan çalışmalar ve kullanılan atama/imputasyon yöntemleri gözden geçirilmektedir. Üçüncü bölümde ön hazırlık aşamasında kullanılan materyal ve stratejilere, önerilen yönteme ve tasarlanan iş akış sürecine ayrıntılı olarak yer verilmiştir. Dördüncü bölümde deneysel sonuçlardan elde edilen bulgular ve sonuçların bir değerlendirmesi yer almaktadır. Beşinci bölümde ise tez çalışması ile ilgili sonuçlar ve öneriler bulunmaktadır.

2. LİTERATÜR ARAŞTIRMASI

Bu tez çalışmasında, eksik veri ataması ve veri artırım yöntemleri ile ilgili yapılan çalışmalar ve çalışmalarda yer alan temel kavramlar bu başlık altında incelenmektedir. Çalışmanın temelini oluşturan eksik veri, eksik veri atama yöntemleri ve kestirimci bakım sistemleri başlıklar halinde, literatürde yer alan çalışmalar ışığında detaylı olarak analiz edilmektedir.

1.1. Eksik Veri Nedir?

Eksik/Kayıp veri, veri setinde herhangi bir hücrede değer olmaması yani gözleme dair bir yanıt bulunmaması anlamına gelmektedir. Bu boş değer çoğu kaynaklarda “null”, “NaN” veya “NA” olarak temsil edilmektedir. Eksik verilerin veri setlerindeki dağılımı çeşitli şekillerde olabilmektedir. Eksik veriler, dağılım örüntülerine göre kabaca ikiye ayrılmaktadır. Şekil 2.1’de tek değişkenli (a) ve çok değişkenli (b) eksik veri kalıbının bir temsili bulunmaktadır.



Şekil 2.1. Eksik veri kalıpları (Boukouvala vd., 2010)

Tek değişkenli eksik/kayıp veri kalıbı, veri setinde tek bir özellikten (feature) gelen verilerin "null" değere sahip olduğu durum iken çok değişkenli eksik/kayıp veri kalıbı birçok özellikten gelen verilerin "null" değere sahip olduğu durumdur. Örnek vermek gerekirse, Şekil 2.1 sektör (X1), yaş (X2), cinsiyet (X3) ve göreve (X4) bağlı olarak geliri (Y) içeren bir veri seti olsun. Şekil 2.1 (a)' da sadece geliri ifade eden sütunda veri eksiktir. Şekil 2.1 (b)'de ise yaşı gösteren X2 sütununda, cinsiyeti ifade eden (X3) sütununda, görevi ifade eden (X4) sütununda ve geliri ifade eden (Y) sütununda yer yer eksik veriler olduğu görülmektedir. Eksik verinin tek değişkenli veya çok değişkenli olma durumu veri atama yöntemlerinin tercihinde önemli bir rol oynamaktadır. Tek değişkenli eksik veri kalıbındaki eksik veriler basit atama yöntemleri ile ele alınabilirken eksik veri örüntüsü daha karmaşık hale geldikçe gelişmiş yöntemlerin kullanılması gerekmektedir (Boukouvala vd., 2010).

Eksik veri probleminde kayıp veri miktarı ve oluşum türü atama yönteminin belirlenmesi üzerinde etkin rol oynamaktadır. Eksik veri terminolojisini ilk kez kullanan Little ve Rubin, (2019) eksik verileri oluşumlarına göre üç sınıfta kategorize etmektedir. *Tamamen Rastgele Kayıp* (Missing Completely at Random - MCAR) durumu eksik verilerin birbirlerinden bağımsız ve tamamen rastlantısal olarak kontrol dışı kayıp olmasıdır. Bu tür eksik verilerin veri setindeki diğer verilerle ve herhangi bir değişkenle bir ilgisi bulunmamaktadır. *Rastgele Kayıp* (Missing at Random - MAR) türünde eksik veriler ve gözlenen değerler arasında sistematik bir ilişki bulunmaktadır. Kayıp verilerin rastgele oluşmasının yanı sıra bu veriler ölçülen diğer değişkenlere bağımlı olduğundan tahmin edilebilmesi mümkündür. *Rastgele Olmayan Kayıp* (Missing Not at Random - MNAR) durumu ise eksik verinin tamamen kendisiyle ve veri setindeki ölçülmeyen bir başka değişkenle alakalı olma durumudur. Bu tür eksik veriler, veri setindeki bir başka gözlenen değer ile açıklanamaz ve kesin bir çıkarımda bulunmak mümkün değildir. Üç sınıfta kategorize edilmiş kayıp veri oluşum mekanizmalarının temel özellikleri Çizelge 2.1'de özetlenmektedir.

Çizelge 2.1. Eksik veri mekanizmaları (Rubin, 1976)

Mekanizma Türü	Değişkenlerle Bağlantı	Göz Ardı Edilebilirlik
Tamamen Rastgele Kayıp - MCAR	Bağlantı yok	Edilebilir
Rastgele Kayıp - MAR	Başka değerler ile bağlantılı	Edilebilir
Rastgele Olmayan Kayıp - MNAR	Kendi değerleri ile bağlantılı	Edilemez

Eksik verilerin ele alınmasında temel hedef, kayıp değerlerin yerine mümkün olan en doğru ve tutarlı değerlerin atanmasıdır. Tamamlanmış veri kümeleri oluşturabilmek için öncelikle eksik verilerin hangi türde oluşum gösterdiğini, hangi değişkenlerle ilişkili olduğunu ve miktarını tespit etmek gerekmektedir.

2.2. Eksik Veriyi Ele Alma Yöntemleri

Veri kümesindeki eksik veriler, araştırmalar ve analizlerde kullanılması hedeflenen istatistiksel yöntemlerin neredeyse tamamı için ciddi bir sorun teşkil etmektedir. Tüm bu istatistiksel yöntemler veri setinin eksiksiz olduğu varsayılarak geliştirilmiştir (Pigott, 2001; Allison, 2003; Osborne, 2013). Eksik veri oranının yüksek olduğu veri setlerinde veri madenciliği ve makine öğrenmesi yöntemlerinin düşük performans sergilediği; istatiksel yöntemlerde ise kesin bir tahmin yapabilmenin oldukça zor bir hal aldığı bilinmektedir. Eksik verilerin varlığında gerçekleştirilen çalışmalar ve analizler yanıltıcı ve taraflı sonuçlar üretebilmektedir. Bu sebeple araştırmaların bir ön çalışması olarak, eksik verilerin ele alınması ve doğabilecek sorunların önüne geçilmesi gerekmektedir.

Eksik veriyi ele alma yöntemleri, silme ve atama yöntemleri olarak iki ana başlıkta değerlendirilmektedir. Silme yöntemi akla ilk gelen ve maliyeti düşük olan yöntemlerden biridir. Bu yöntem, eksik verinin yok sayılarak çalışmadan çıkarılmasıdır. Silme yöntemi, hiçbir hesaplama gerektirmeyip tüm istatistiksel analizler için uygun olsa da ciddi veri

kayıplarına yol açmaktadır. Eksik veri şeklinin Tamamen Rastgele Kayıp (MCAR) olduğu ve eksik veri miktarının %5'ten az olduğu senaryolar için önerilmektedir.

Silme yöntemi, satır bazlı (listwise), sütun (column) bazlı ve eşleştirme yoluyla (pairwise) olmak üzere üç farklı şekilde uygulanabilmektedir. Satır bazlı silme yöntemi, gözlenen veride bir veya daha fazla eksik değer bulunması durumunda tüm satırın silinerek çalışmadan çıkarılmasıdır. Sütun bazlı silme işlemi, herhangi bir değişkenin büyük bir çoğunluğunun eksik olması durumunda o değişkenin tamamen silinmesidir. Eşleştirme yoluyla silme yöntemi, eksik olmayan ve erişilebilen tüm verilerin kullanımına dayanmaktadır. Bu yöntemin temel mantığı ortalamalar, standart sapmalar ve korelasyon/kovaryans matrisi gibi özet istatistiksel değerlerin ulaşılabilen tüm veriler kullanarak hesaplamaktır.

Veri/Örneklem kayıplarının önüne geçmek ve veri dağılımında çok büyük değişikliklere sebebiyet vermeden eksiksiz veri setleri elde edebilmek için çeşitli yöntemler geliştirilmiştir. Bu yöntemlerin başında, basit veya model-tabanlı gelişmiş teknikler kullanarak eksik verilere yaklaşık değer atama yöntemi gelmektedir. Atama bir diğer ismiyle imputasyon yöntemi, araştırmacıların zamandan ve emekten tasarruf etmesine imkân sağlayarak veri kaybını ortadan kaldırmayı hedeflemektedir. Eksik veri atamasına yönelik olarak çok fazla sayıda yöntem geliştirilmiş ve geliştirilmeye de devam etmektedir. Bu yöntemler eksik verinin mekanizmasına göre (MCAR, MAR, MNAR), eksik verinin tipine göre (sayısal/sürekli veya kategorik) ve eksiklik oranına göre çeşitlilik göstermektedir. Bu nedenle her eksik veri durumunda her atama yönteminin uygun ve başarılı olması söz konusu değildir.

Tekli atama (Single Imputation - SI) yönteminde bir değişkendeki eksik değerlerin o değişkeni temsil eden tek bir değer ile değiştirilmesi esastır. Eksik değerlerin yerine atanacak temsil değer, eksik veri bulunan değişkende var olan değerlerin ortalaması (Mean Substitution), medyan değeri, mod (tepe) değeri, alt/üst satırındaki komşu değer veya sabit bir değer (sıfır vb.) olabilmektedir. Ortalama, medyan, mod veya sabit bir değer atama yöntemi basit bir yöntem olsa da eksik veri içeren değişkenin varyansını düşürmekte,

merkeze doğru yığılmaya sebep olmakta ve yanlılığa yol açarak veri dağılımını etkilemektedir.

Regresyon atamasında eksik verilerin bulunduğu sütun bağımlı değişken, diğer sütunlar bağımsız değişken olarak tanımlanmaktadır. Temel hedef, bağımlı değişkeni bağımsız değişkenler ile matematiksel olarak tahmin etmektir. Bu yöntemde, eksik değerler dışındaki tüm değerler kullanılarak oluşturulan regresyon modeli ile eksik veriler tahmin edilmektedir. Bu yöntem, yansız tahminler üretse de yalnızca bağımsız değişkenin bağımlı değişkeni açıklama oranının yüksek olduğu durumlarda başarılı çıktılar elde edilebilmektedir.

Stokastik regresyon ataması yöntemi, regresyon ataması yönteminden eksik değerin tahmin edilmesi için oluşturulan doğrusal denkleme normal dağılım gösteren bir hata teriminin eklenmesi yönüyle farklılaşmaktadır. Bu yöntem ile beraber regresyon atamasında karşılaşılan hata varyansının sıfır olması problemi ortadan kalkmaktadır. Hata teriminin eklenmesiyle varyans artmakta ve yanlılık azalmaktadır (Enders, 2022). Bu atama yönteminin regresyonla atama yöntemine göre özellikle MAR ve MCAR eksik veri mekanizmalarında daha başarılı sonuçlar verdiği ve yansız tahminlerde bulunduğu belirtilmektedir (Baraldi ve Enders, 2010).

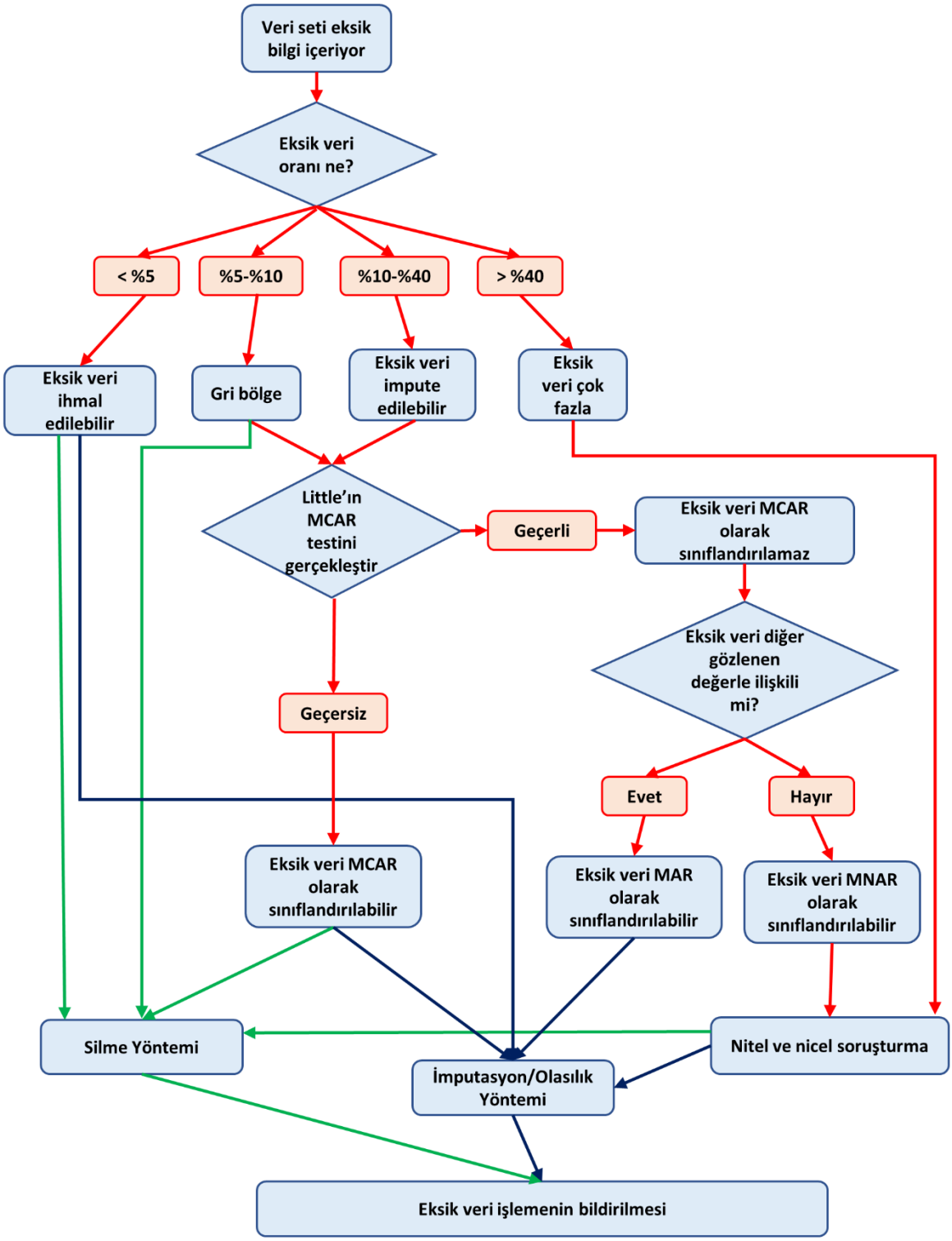
1990'lı yıllarının başlarında çalışmalara başlanması ve sonlarında uygulamaya geçilmesi ile eksik veriyi ele almaya yönelik olarak basit geleneksel yöntemlerin geliştirilmesi ve alternatif yöntemlerin ortaya çıkışı son derece hız kazanmıştır (Allison, 2001). Basit atama yöntemlerinin dezavantajları göz önünde bulundurularak geliştirilen en çok olabilirlik (Maximum Likelihood - ML) ve çoklu veri atama (Multiple Imputation - MI) yaklaşımları eksik veri imputasyonunda yaygın olarak kullanılmaya başlanmıştır.

Beklenti-Maksimizasyon (Expectation–Maximization) yöntemi, eksik verilere bir regresyon denklemi ile yaklaşık değer atama süreci olan beklenti adımı ve oluşan eksiksiz veri seti üzerinden yeniden kurulan bir regresyon denklemi ile tahminlerin yenilenme süreci

olan maksimizasyon adımı olarak yinelemeli (iteratif) iki ayrı aşamadan oluşmaktadır. Beklenti-Maksimizasyon döngüsü, beklenen logaritmik olabilirlik değerinin en yüksek seviyeye ulaştığı ve beklenen değerler arasındaki farkların önemsizleştiği noktaya kadar defalarca tekrarlanmaktadır. Eksik veri oranının yüksek olduğu veri setlerinde yavaş olması bir dezavantaj olarak sayılabilmektedir.

Rubin (2014) tarafından geliştirilen çoklu atama (Multiple Imputation) yöntemi, eksik verilere iki veya daha fazla değer atanması esasına dayanmaktadır. Silme ve basit tekli atama yöntemlerinin dezavantajlarını ortadan kaldırmaya yönelik olarak tasarlanan bu eksik veri atama yöntemi, elde edilen n farklı (n>1 olmak koşuluyla) eksiksiz veri setinin standart istatistiksel tekniklerle analiz edilerek sonuçların birleştirilmesinden oluşmaktadır. Bu atama yöntemi, eksik verinin rastgele oluştuğu MAR mekanizmasında başarılı sonuçlar göstermektedir. Çoklu atama temelinde geliştirilen Markov Zincirleri Monte Carlo gibi özel yaklaşımlar da bulunmaktadır. Markov Zincirleri Monte Carlo yöntemi, üç aşamalı bir çoklu atama tekniğidir. İlk aşama, m ayrı veri setinin simüle edilme sürecidir. İkinci aşamada simüle edilen her bir veri seti üzerinden tam veri setine yönelik tahminleme ve analiz yapılmaktadır. Üçüncü aşamada ise tek bir parametre kümesi elde etmek için tahminlerin ve analizlerin birleştirilmesiyle veri seti tamamlanmaktadır (Hasan vd., 2017).

Bu bölümde açıklanan yöntemlerin dışında, eksik veri varlığında kullanılabilecek çok fazla sayıda yöntem ve teknik bulunmakta ve bu sayı her geçen gün artmaktadır. Bir atama yönteminin diğerine üstünlüğünden söz edilse de “en iyi” olarak nitelendirilebilecek bir yöntemden söz etmek mümkün değildir. Şekil 2.2'de eksik veri çözümleme stratejilerini özetleyen bir görsel yer almaktadır.



Şekil 2.2. Eksik veri çözümleme stratejileri (Mirzaei vd., 2022)

Son yıllarda makine öğrenmesi ve derin öğrenme yöntemlerindeki gelişmeler bu yöntemlerin eksik veri atamasında kullanılmasına imkân sağlamış ve bu alanda başarılı

sonuçlar doğurmuştur. Geleneksel atama yöntemlerinin yanı sıra bilinen karar ağacı regresyonu (DTR), k-en yakın komşu regresyonu (KNN), destek vektör regresyonu (SVR), random forest ve yapay sinir ağları (ANN) gibi yöntemler eksik veri tahminleme alanında yaygın olarak kullanılmaya başlanmıştır. Çizelge 2.2'de literatür araştırması kapsamında incelenen eksik veri imputasyonu çalışmalarının genel bir özeti bulunmaktadır.

Çizelge 2.2. Literatürde eksik veri problemini ele alan başlıca çalışmalar

Referans	**Çalışma**	**Yöntem**
Mohamed vd., 2014	Scalable algorithms for missing value imputation	k-Means tabanlı Classic Imputation (CI), Modification of Classic Imputation (MCI), Enhancement of Modification of Classic Imputation (EMCI)
Noor vd., 2015	Comparison of linear interpolation method and mean method to replace the missing values in environmental data set	Ortalama atama, Lineer interpolasyon
Chong vd., 2016	Imputation of missing values in building sensor data	Sıfır değerini atama, Ortalama atama, Lineer regresyon, KNN, SVM
Peppanen vd., 2016	Handling bad or missing smart meter data through advanced data imputation	Lineer interpolasyon, Historical average (HA), Optimally weighted average (OWA)
Kim vd., 2017	Learning-based adaptive imputation method with kNN algorithm for missing power data	Learning-based Adaptive Imputation (LAI), Extended Learning-based Adaptive Imputation (eLAI), Optimally-Weighted Average (OWA), Probabilistic Principle Component Analysis (PPCA), Lineer interpolasyon

Çizelge 2.2. Literatürde eksik veri problemini ele alan başlıca çalışmalar (devam)

Referans	Çalışma	Yöntem
Demirhan ve Renwick, 2018	Missing value imputation for short to mid-term horizontal solar irradiance data	36 adet imputasyon yöntemi
Bokde vd., 2018	A novel imputation methodology for time series based on pattern sequence forecasting	Ortalama atama, İnterpolasyon, ARIMA (Kalman filtreleme), Random Forest, Last Observation Carried Forward, imputePSF
Hegde vd., 2019	MICE vs PPCA: Missing data imputation in healthcare	Probabilistic principal component analysis (PPCA), Multiple Imputation by Chain Equations (MICE)
Kim vd., 2019	Analysis and impact evaluation of missing data imputation in day-ahead PV generation forecasting	KNN, Lineer interpolasyon, Çoklu atama, Multiple Imputation by Chain Equations (MICE)
Martinez-Luengo vd., 2019	Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation	Artificial Neural Network (ANN)
Izonin vd., 2019	An approach towards missing data recovery within IoT smart system	Ito decomposition, AdaBoost

Çizelge 2.2. Literatürde eksik veri problemini ele alan başlıca çalışmalar (devam)

Referans	Çalışma	Yöntem
Afrifa-Yamoah vd., 2020	Missing data imputation of high-resolution temporal climate time series data	ARIMA (Kalman filtreleme), Structural time series model (Kalman filtreleme), Çoklu lineer regresyon
Velasco-Gallego ve Lazakis, 2020	Real-time datadriven missing data imputation for shortterm sensor data of marine systems. A comparative study	Ortalama atama, STL ayrıştırma (Seasonal-Trend decomposition using LOESS), ARIMA, Holt Winters, PLS regresyon, Ridge regresyon, LASSO regresyon, KNN, SVR (Lineer, RBF kernel), NN (1-2-3 gizli katman), DTR, Vektör otoregresyon, Torbalama ağaçları (SVR, KNN), Random Forest, AdaBoost
Hadeed vd., 2020	Imputation methods for addressing missing data in shortterm monitoring of air pollutants	Ortalama atama, Medyan atama, Last Observation Carried Forward, Kalman Filtresi, Random, Markov, Predictive Mean Matching (PMM), Satır ortalaması atama
Ngueilbaye vd., 2021	Modulo 9 model-based learning for missing data imputation	Modulo 9, Ortalama atama, Silme, SVM, KNN, DTR, Naïve Bayes (NB), Lineer regresyon, Random Forest, Multi-Layer Perceptron (MLP), Support Vector Classifier (SVC), Linear Support Vector Classifier (LSVC)
Wang vd., 2021	Towards missing electric power data imputation for energy management systems	ARIMA, Lineer interpolasyon, KNN, Multi-Layer Perceptron (MLP), SVR

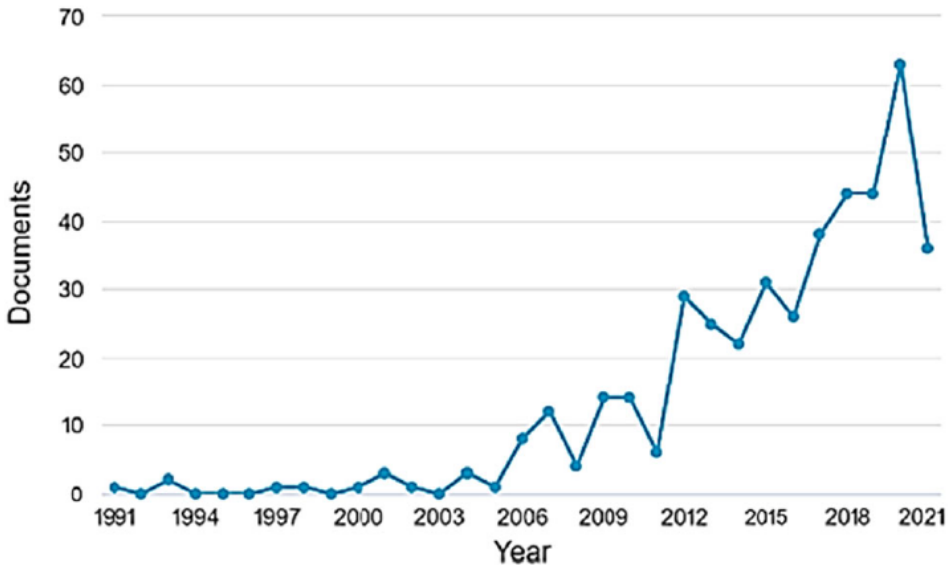
Çizelge 2.2. Literatürde eksik veri problemini ele alan başlıca çalışmalar (devam)

Referans	Çalışma	Yöntem
Wang vd., 2021	Fault detection based on Bayesian network and missing data imputation for building energy systems	EM-BN methodu (Expectation–Maximization (EM) ve Bayesian network (BN) tabanlı)
Okafor ve Delaney, 2021	Missing data imputation on IoT sensor networks: Implications for on-site sensor calibration	Variational Autoencoder (VAE), Neural Network with Random Weights (NNRW), Multiple Imputation by Chain Equations (MICE), Random Forest based imputation (missForest), KNN
Alamoodi vd., 2021	Machine learning-based İmputation soft computing approach for large missing scale and non-reference data imputation	Decision Tree, KNN, Naive Bayes (NB)
Hu vd., 2021	Information granüle-based classifier: A development of granular imputation of missing data	Multi-Layer Perceptron for Imputation and Classification (MPC), Fuzzy weighted KNN Classifier (FW), Fuzzy broad learning classification (FBLC), KNN, Radial basis function kernel Support Vector Machines classifier (RSVM)
Kalay vd., 2022	A comparison of data imputation methods utilizing machine learning for a new IoT system platform	SVR, DTR, Ridge regresyon, KNN, Miss Forest, XGBoost

Çizelge 2.2. Literatürde eksik veri problemini ele alan başlıca çalışmalar (devam)

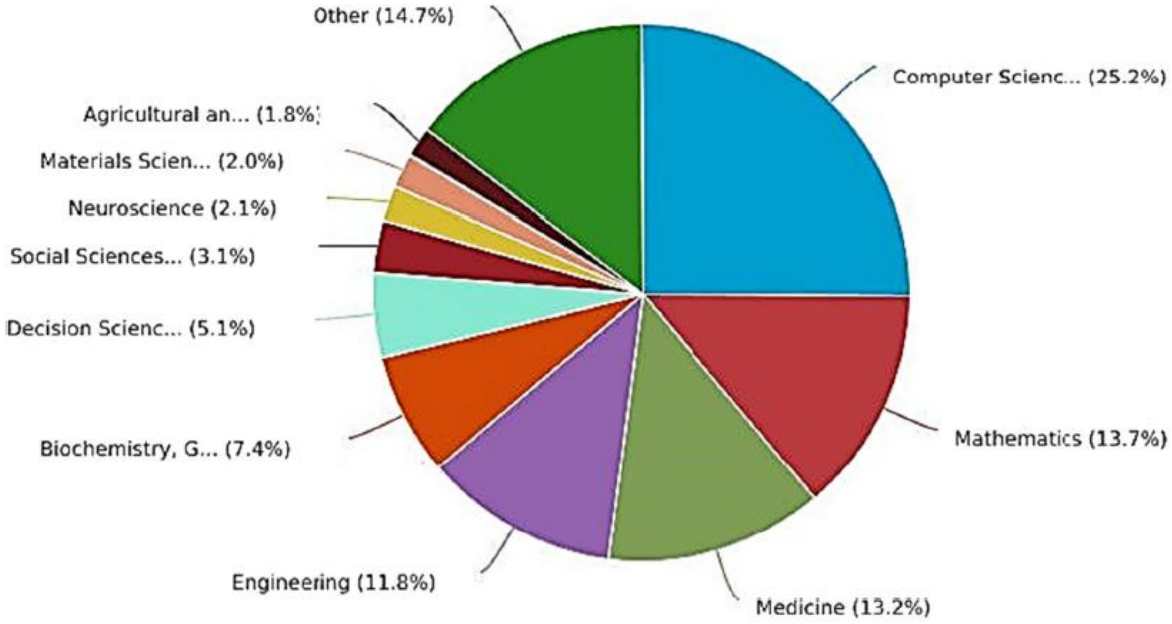
Referans	Çalışma	Yöntem
Awawdeh vd., 2022	EvoImputer: An evolutionary approach for missing data imputation and feature selection in the context of supervised learning	Ortalama atama, Medyan atama, Çoklu atama, Expectation–Maximization (EM), KNN
Zhang ve Thorburn, 2022	Handling missing data in near real-time environmental monitoring: A system and a review of selected methods	Ortalama atama, Last Observation Carried Forward, Lineer interpolasyon, Expectation–Maximization (EM), Multiple Imputation by Chain Equations (MICE), KNN, Sequence-to-Sequence Imputation Model (SSIM), Dual-SSIM, BRITS

Adnan vd. (2022), 1991'den Haziran 2021'e kadar eksik veri imputasyonu üzerine yapılan tüm akademik çalışmaları derinlemesine incelemiş ve kapsamlı bir bibliyometrik analiz ortaya koymuştur. Bu bibliyometrik analiz, Scopus akademik veri tabanında yer alan, "missing data" veya "missing values" veya "missing value" veya "incomplete data" ve "imputation" ve "classification" anahtar kelimeleriyle ilişkili 430 yayını kapsamaktadır. Scopus veri tabanına dayalı olarak, eksik veri imputasyonu alanındaki ilk akademik çalışma 1991 yılında Clogg ve arkadaşları tarafından ortaya konmuştur. Bu alanda en çok alıntı yapılan ilk 20 makaleden biri olan bu dergi makalesinde, yeni bir veri tabanı oluşturmak için Çoklu Atama tabanlı Bayesci Lojistik Regresyon algoritması (Multiple Imputation-based Bayesian Logistic Regression) üzerinde çalışmışlardır. Şekil 2.3'te, eksik veri imputasyonu alanında 1991 ve 2021 yılları arasında ortaya konan akademik çalışmaların yıllara göre bir analizi yer almaktadır.



Şekil 2.3. Eksik veri imputasyonu alanında yıllara göre toplam yayın sayısı (Adnan vd., 2022)

Şekil 2.3'te görüldüğü üzere 1991-2006 yılları arasında eksik veri imputasyonu alanında birkaç çalışma bulunsa da 2006 yılında yayımlanan sekiz akademik çalışma ile beraber bu alanda kayda değer bir gelişme görülmüş ve sonraki yıllarda bu sayı kademeli olarak artmıştır. Eksik verinin veri işlemede kritik bir engel olarak görülmesi ve veri madenciliği yaklaşımının popülerleşmesi ile beraber eksik veri çalışmalarında büyük bir artış yaşanmaktadır. En yüksek yayın sayısı, 63 makale (%14,65) ile birlikte 2020 yılında kaydedilmiştir. Adnan vd. (2022), gerçekleştirdikleri literatür analizinde eksik veri alanında yayınlanan makaleleri konularına göre kategorize etmiş ve çeşitli sonuçlar elde etmişlerdir (Şekil 2.4).



Şekil 2.4. Konularına göre sınıflandırılmış yayınlar (Adnan vd., 2022)

Bilgisayar bilimi, 212 yayın (%25.2) ile birlikte eksik veri imputasyonu alanındaki en çok yayına sahip olan bilim dalı olurken; onu 115 yayın (%13.7) ile matematik ve 111 yayın (%13.2) ile tıp takip etmektedir.

Mevcut literatür analiz edildiğinde, silme yöntemi, istatistiksel teknikler, makine öğrenmesi ve derin öğrenme tabanlı yaklaşımlar olmak üzere çok sayıda eksik veriyi ele alma yöntemi ile karşılaşılmaktadır. Bu yöntemlerin her biri veri setinin büyüklüğüne, eksik verilerin oluşum yapısına, konumuna ve oranına göre çeşitli avantajlar/dezavantajlar içermekte ve eksik veri tahminleme konusunda farklı performanslar sergilemektedir. Bu kapsamda, araştırmacının çeşitli faktörleri dikkate alarak hedef veri setine en uygun yöntemi seçmesi gerekmektedir. Velasco-Gallego ve Lazakis (2020), literatürde öne çıkan eksik veri imputasyon yöntemlerinin avantajlarının, dezavantajlarının ve sınırlamalarının kapsamlı bir analizini sunmaktadır (Çizelge 2.3).

Çizelge 2.3. Eksik veri imputasyon yöntemlerinin avantaj ve dezavantajları (Velasco-Gallego ve Lazakis, 2020)

Yöntem	Avantajlar	Dezavantajlar
Ortalama (Mean) Atama	• Uygulaması ve yorumlaması kolay • Yürütme süresi düşük	• Parametre dağılımının bozulması • Özellikler (features) arasındaki ilişkinin bozulması • Eksik veri mekanizması MAR veya NMAR olduğunda ortalama tahminlerinin yanlılığı
Lineer Regresyon Yöntemleri (PLS, LASSO, Ridge, ElasticNet)	• Yorumlaması kolay • Yürütme süresi düşük • Regülarizasyon modelleri aşırı öğrenmeyi (overfitting) engeller • Doğrusal bir ilişki varlığında yüksek performans	• Yalnızca yanıt (response) ve öngörücüler (predictors) arasındaki doğrusal ilişkileri yakalar • Eksik bir değerin impute edilmesi gerektiğinde her seferinde büyük miktarda veri gerektirir • Tahmin yaparken risk taşır
Sinir Ağları (1-2-3 gizli katman)	• Hem doğrusal hem de doğrusal olmayan ilişkileri yakalar	• Modeli eğitmek için büyük miktarda veri gerektirir • Ağ yapısı tanımındaki karmaşıklık • Yüksek hesaplama maliyeti • Aşırı öğrenmeye (overfitting) karşı hassas

Çizelge 2.3. Eksik veri imputasyon yöntemlerinin avantaj ve dezavantajları (Velasco-Gallego ve Lazakis, 2020) (devam)

Yöntem	Avantajlar	Dezavantajlar
KNN	• Çeşitli uzaklık metrikleri uygulanabilir • Yorumlaması kolay Verilerin yakınlığına bağlı olarak tahmini mesafelere ağırlıklar eklenebilir	• Aykırı değerlere ve gürültülü verilerine karşı hassastır • Örneklem büyük olduğunda veya boyutlar yüksek olduğunda performans düşüşü • Özellik ölçeklendirmesi (Feature scaling) gereklidir • Eksik değeri atamak için gereken komşu sayısı optimal olarak seçilmelidir
SVR (Lineer ve RBF kernel)	• Hem doğrusal hem de doğrusal olmayan ilişkileri yakalar • Aykırı değerlere karşı dirençli • Kolayca uyarlanabilir • Çeşitli kernel fonksiyonları kullanılabilir	• Örneklem gürültü içerdiğinde düşük performans • Hem çekirdek fonksiyonu hem de ayarlama (tuning) parametrelerinin en uygun şekilde seçilmesi gerekir • Özellik ölçeklendirmesi (Feature scaling) gereklidir • Yorumlaması zor • Yüksek hesaplama maliyeti

Çizelge 2.3. Eksik veri imputasyon yöntemlerinin avantaj ve dezavantajları (Velasco-Gallego ve Lazakis, 2020) (devam)

Yöntem	Avantajlar	Dezavantajlar
DTR	• Öznitelik seçimini özünde uygular • Daha az veri ön işleme gerektirir • Yorumlaması kolay • Yürütme süresi düşük • Aşırı öğrenme (overfitting) olasılığı düşük	• Sürekli (continuous) veriler dikkate alındığında optimal olmayan bir değerlendirme performansı göstermesi muhtemel • Eksik değerlere sahip örnekler, modeli eğitmek için kullanılan örneklere benzemiyorsa değerlendirmeler doğru değildir • İstikrarsızlık
Ensemble Yöntemler (Torbalama ağaçları, Random Forets ve AdaBoost)	• Öznitelik seçimini özünde uygular • Daha az veri ön işleme gerektirir • Aşırı öğrenme (overfitting) olasılığı düşük	• Yorumlaması zor ve karışık • DTR'den daha yüksek hesaplama maliyeti • Sürekli (continuous) veriler dikkate alındığında optimal olmayan bir değerlendirme performansı göstermesi muhtemel • Eksik değerlere sahip örnekler, modeli eğitmek için kullanılan örneklere benzemiyorsa değerlendirmeler doğru değildir

Çizelge 2.3'te görüldüğü üzere, istatistik tabanlı, model tabanlı ve sinir ağı tabanlı eksik veri atama yöntemleri, çeşitli avantajları, dezavantajları ve sınırlamaları beraberinde getirmektedir. Özetlemek gerekirse, istatistik tabanlı yöntemlerde eksik değerler belirli bir kural tarafından tanımlanan bir değer ile değiştirilmektedir. Bu yaklaşım, hesaplama açısından basit ve maliyet açısından uygundur ancak veri setindeki değişkenler arasındaki ilişkiyi göz ardı etmektedir (Zhang ve Thorburn, 2022). Model tabanlı yöntemler, eksik olmayan verileri girdi olarak almakta ve eksik veriler için regresyon modelleri oluşturarak farklı değişkenler arasındaki ilişkiyi hesaba katmaktadır. Ancak, özellikle büyük ölçekli veri setleri için ciddi bir hesaplama maliyeti doğurmaktadır. Sinir ağı tabanlı yaklaşımlar ise yüksek başarı oranına sahip olmasına karşın optimal hiperparametre ayarlaması gerektiren maliyetli yaklaşımlardır. Ek olarak, sinir ağı modelleri genellikle açıklanması ve yorumlanması zor olan kara kutu modelleridir ve çıktıları insanlar tarafından izlenemediği için eleştirilmektedir. Araştırmacılar eksik veri içeren veri setleri ile çalışmadan önce, tüm bu yaklaşımların güçlü ve zayıf yönlerini göz önüne almalı, veri setine ve amacına en uygun atama yöntemini seçerek eksiksiz bir veri seti ile çalışmalarını yürütmelidir.

2.3. IoT Sistemlerde Veri Ataması

Literatür incelendiğinde IoT sistemlerden elde edilen çeşitli veri setleri üzerinde eksik veri atamaları/imputasyonu için birçok istatistiksel, makine öğrenmesi ve derin öğrenme yönteminin uygulandığı görülmektedir.

Japonya'daki Toshiba Smart Community Center'dan toplanan zaman serisi verileri üzerinde Doğrusal Regresyon, Ağırlıklı k-en yakın komşular (KNN), Destek Vektör Makineleri (SVM), Ortalama (Mean) Atama ve Eksik Değerlerin Sıfır Değeri ile Değiştirilmesi (Replaced by Zero) yöntemleri test edilmiştir (Chong vd., 2016). Yöntemlerin değerlendirilmesinde çeşitli parametre ayarlamaları ve korelasyona dayalı öznitelik (feature) seçimi göz önünde bulundurulmuştur. Hedef ve bağımlı değişkenler arasında doğrusal bir ilişkinin olduğu durumlarda Doğrusal Regresyon algoritmasının daha yüksek doğruluk sağladığı görülürken, Destek Vektör Makineleri'nin doğrusal olmayan bir ilişki olduğunda daha yüksek doğruluk sağladığı görülmektedir.

Güneş enerjisi kullanımında güneş ışınımı tahmininin doğruluğu oldukça önemlidir. Demirhan ve Renwick (2018) güneş ışınımı serilerindeki eksik veriler için doğru tahminleme yapabilen yöntemleri belirlemek amacıyla bir girişimde bulunmuştur. Avustralya'dan toplanan dakikalık, saatlik, günlük ve haftalık zaman serisi verilerini içeren gerçek ve eksiksiz bir güneş ışınımı veri seti üzerinde Monte Carlo yaklaşımıyla eksiklikler oluşturulmuştur. Dört farklı frekansta otuz altı atama yöntemi değerlendirilmiş ve MASE, rMAE, rRMSE değerleri [0, 1] aralığında yeniden ölçeklendirilerek ısı haritaları (heatmap) sunulmuştur.

Velasco-Gallego ve Lazakis (2020) bugüne kadar literatürdeki birçok çalışmanın eksik veri atama yöntemlerini ele aldığını, ancak hiçbir çalışmanın kayıp verileri gerçek zamanlı olarak atamadığını savunmaktadır. Bu nedenle, sadece mevcut algoritmaların eksik veri atamalarının doğruluğunu değil, aynı zamanda gerçek zamanlı atamaları gerçekleştirme yeteneklerini de değerlendirmişlerdir. Bu alanda en yaygın olarak kullanılan yirmi makine öğrenimi ve zaman serisi tahmin algoritmasının avantajlarını, dezavantajlarını ve sınırlamalarını gelecek çalışmalara ışık tutması amacıyla listelemişlerdir. Modellerin performansını gerçek bir senaryo ile ilişkilendirmek adına, bir kargo gemisine yerleştirilmiş sensörlerden elde edilen toplam yedi makine sistem parametresi üzerinde bir çalışma gerçekleştirmişlerdir. Tahmin modellerinin performansını ortaya çıkarmak için yürütme süresi (execution time), MSE, MSLE, RMSE, MAPE, MedAE ve Max Error gibi standart performans metriklerini kullanmışlardır.

Bir başka çalışma (Ngueilbaye vd., 2021), eksik verilerin makine öğrenimi algoritmalarının performansını ve veri analizine dayalı karar vermeyi nasıl etkileyebileceğini göstermektedir. Ayrıca, eksik veri sorununa verimli bir çözüm niteliğinde olan yeni bir eksik veri imputasyon yöntemi Modulo 9'u tanıtmışlardır. Önerilen yeni yöntemin on bir popüler makine öğrenimi algoritmasından daha iyi performans sergilediğini göstermişlerdir.

Düşük Maliyetli Sensör (Low-Cost Sensor - LCS) verileri üzerinde eksik veri atamasını en başarılı şekilde gerçekleştiren algoritmayı bulmayı hedefleyen Okafor ve

Delaney (2021), çeşitli stratejilerin atama/imputasyon yeteneğini değerlendirmiştir. Bu değerlendirme sırasında farklı kayıp veri oranları (%5, %10, %30, %50, %70) ve farklı kayıp veri süreleri (1 gün, 1 hafta, 2 hafta, 1 ay) dikkate alınmıştır. Bu farklı periyotlar ve oranlar için algoritmaların Hata Kareleri Ortalamasının Karekökü (RMSE) değerleri kıyaslanmıştır. Ek olarak, çeşitli kalibrasyon yöntemlerini kullanarak impute edilen veriler ile sensör kalibrasyon performansını analiz etmişlerdir. Eksik veri atama performans değerlendirmesinde Variational Autoencoder (VAE), Neural Network with Random Weights (NNRW), Multiple Imputation by Chain Equations (MICE), Random Forest tabanlı imputasyon (Miss Forest) ve k-Nearest Neighbors (KNN) algoritmaları kullanılırken; sensör kalibrasyon performans değerlendirmesi için Çoklu Doğrusal Regresyon (Multiple Linear Regression - MLR), Karar Ağacı (Decision Tree - DT), Rastgele Orman (Random Forest - RF) ve XGBoost algoritmalarını tercih etmişlerdir. Eksik veri atama aşamasında VAE algoritmasının diğer algoritmalardan daha iyi bir performans sergilediği gözlemlenmiştir.

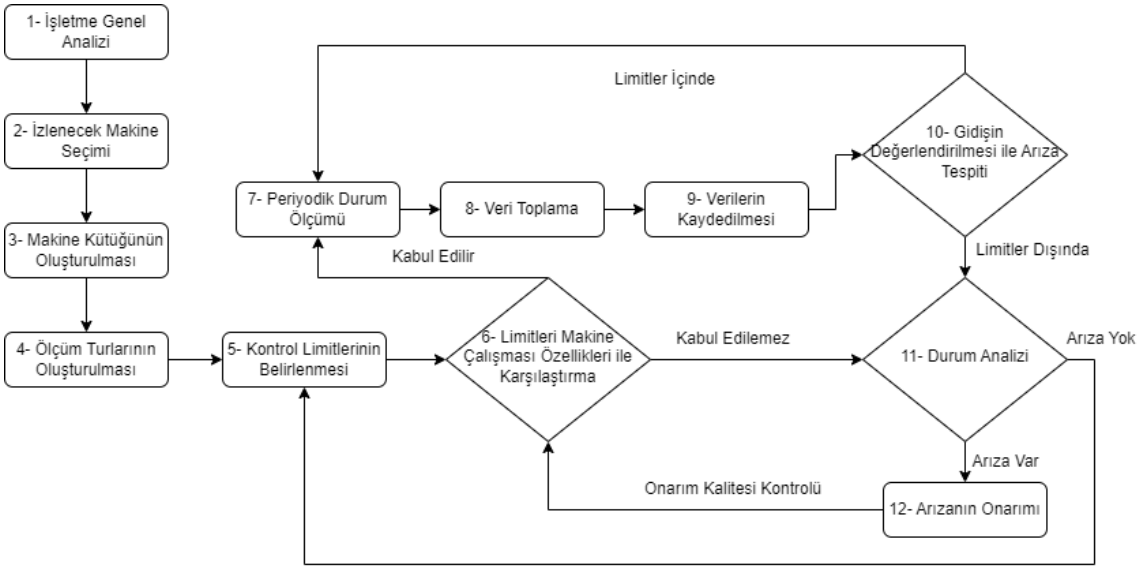
2.4. Kestirimci Bakım Sistemleri

Bakım yönetimi ve verimliliği odaklı araştırmalar, bakım maliyetlerinin büyük bir kısmının gereksizce yapılan veya zamanında yapılmayan bakımlardan oluştuğunu ve bu durumun işletmeler için büyük bir kaynak israfı olduğunu göstermektedir. Periyodik bakımın öne çıkan dezavantajlarından biri, sistem işleyişine normal seviyede devam edebilecek durumda olsa dahi bakım yapılmasıdır. Bu durum üretim faaliyetlerinin duraksamasına ve verimliliğin düşmesine neden olmak ile birlikte bakım maliyetlerinin artmasına sebep olmaktadır. Kestirimci bakımın temel ilkesi ise bakımın bir arıza sonucu veya bakım planı dahilinde değil yalnızca gerektiğinde yapılmasıdır. Bu yaklaşım, her arızanın en az bir ön habercisi olduğunu varsayarak yalnızca bu ön haberci belirtilerinin izlenmesini ve veri analiz teknikleri ile arıza tahmininde bulunulmasını vurgulamaktadır. PdM, sağladığı çok yönlü katkılar sebebiyle son yıllarda endüstriyel ve akademik çalışmaların odağı haline gelmektedir.

Bakım stratejisi, üretim sistemlerinin verimliliği ve maliyet/kaynak planlaması için kritik bir öneme sahiptir (Khan vd., 2022). Son yıllarda, geleneksel bakım yaklaşımlarının

yanı sıra Endüstri 4.0 ve onun getirdiği teknolojik yenilikler ile beraber bakım maliyetlerinin azaltılması işletmeler için bir zorunluluğa dönüşmüş ve kestirimci bakım (Predictive Maintenance - PdM), bir diğer ismiyle öngörücü bakım, yaklaşımı ilgi çeken bir disiplin haline gelmiştir. Kestirimci bakım (PdM), geçmiş zaman serisi verileriyle makine arızalarını önceden tahmin etmeye yardımcı olan durum-tabanlı izleme (condition-based monitoring) ve prognostik (Kalan Yararlı Ömür - RUL tahmini vb.) stratejilerini içeren şemsiye bir terimdir (Dündar vd., 2021). Büyük veri, IoT, dijitalleşme ve bulut gibi gelişen teknoloji konseptleri sayesinde kazanılan üretim ortamından büyük miktarlarda veri toplama, depolama ve işleme yeteneği, kestirimci bakım uygulamalarının kritik ihtiyaçlarını karşılayarak etkin bakım planlamasını mümkün kılmıştır.

Koşul temelli bakım yaklaşımının gelişmiş bir versiyonu olan kestirimci bakım (Zonta vd., 2020), sistemlerin olası arıza ve anomalilerini tespit etmek ve kalan yararlı kullanım ömrünü öngörebilmek için sensör veya akıllı kontrol cihazları ile elde edilen durum verilerini makine öğrenmesi algoritmaları ve istatistiksel yöntemlerle modelleyerek bakım planlaması ile ilgili en iyi kararı verme süreci olarak ifade edilebilmektedir (Zhang vd., 2019). PdM, geçmiş verilere dayalı olarak ekipman arızalarını önceden tahmin eden duruma dayalı bir stratejidir ve gelecekte meydana gelmesi muhtemel arızaları önceden tahmin etmek için sistemin normal işleyişi sırasında ekipman performansının doğrudan izlenmesine odaklanmaktadır. Bu sayede beklenmeyen arıza sayısı en aza indirilmekte, bakım için ayrılan süre ve maliyet azalmakta ve verimlilik artmaktadır (Cinar vd., 2022). Bu bakım yaklaşımında ekipman bakımı yalnızca çalışma süresine göre planlanmamaktadır. Olası arızaları önceden tahmin etmek için akım, titreşim, basınç, sıcaklık ve diğer değişkenler toplanarak modelleme ve analiz yöntemleri ile ekipman koşulları izlenmektedir. Bu noktada devreye giren büyük veri analiz teknikleri, birden fazla kaynaktan gelen büyük ölçekli veri akışlarını gerçek zamanlı olarak işleyerek PdM yaklaşımının kullanılmasına olanak sağlamaktadır (Su ve Huang, 2018). Kestirimci bakım stratejisi, bir makinenin veya süreç bütünlüğünün sürekli olarak izlenmesini sağlamakta ve bakımın yalnızca ihtiyaç duyulduğunda yapılmasına izin vermektedir (Carvalho vd., 2019). Şekil 2.5'te kestirimci bakım sistemlerinin işleyişini özetleyen bir akış şeması bulunmaktadır.



Şekil 2.5. Kestirimci bakım akış şeması (Mobley, 2002)

Başarılı bir kestirimci bakım çalışması için ilk aşama, kaliteli bir veri setinin oluşturulmasıdır. Bu veri seti, yapay zeka (AI) modellerine ve istatistiksel modellere tahmin edilmesi istenen olayların çeşitli örneklerini sunmalıdır. Tüm koşulları içermeyen, yanlı (biased) veri setlerinden doğan modellerin iyi bir tahmin yürütmesi mümkün olmamaktadır. PdM çalışmaları için ideal bir veri seti, ekipmanın tüm durumlarına ait gözlemlenmiş zaman serilerinden oluşmalıdır (Ramos vd., 2014).

Kestirimci bakım sistemlerinde, veriler IoT sensörleri ve çeşitli veri toplama donanımları aracılığıyla uçta toplanabilmektedir. Veriler, merkezi sunucularda programlanmış iş süreçlerine göre analiz edilebilmekte ve ekipman arızalarını tahmin etmek için yapay zeka (AI) modellerinde kullanabilmektedir. Bu tez çalışmasında, ilk olarak teknik ekipmandan alınan titreşim, akım, sıcaklık, ses gibi sensör verileri eksik veri varlığı açısından kontrol edilmektedir. Teknik ekipmandan toplanan ve çeşitli nedenlerle yazılamayan eksik sensör verileri, tasarlanan IoT platformunun bir bileşeni olan veri artırma modülü ile impute edilmekte ve tamamlanmaktadır. Eksiksiz veriler sistemde çevrimiçi (online) ve çevrimdışı (offline) veriler olarak kullanılmaktadır. Geçmiş çevrimdışı veriler, makine öğrenmesi ve derin öğrenme modellerine eğitim verisi olarak verilmektedir. Bu çevrimdışı veriler, sistem üzerinde tasarlanan AI modelleri eğitmek ve doğrulamak için

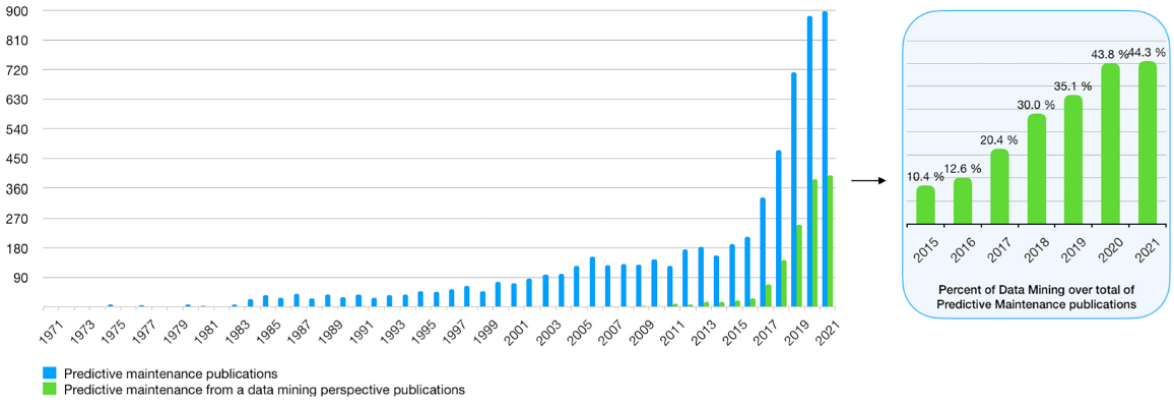
kullanılmaktadır. ML ve DL modeller, arıza tespiti/teşhisi ve kalan yararlı ömür (Remaining Useful Life - RUL) tahmini gibi amaçlar için hazır hale getirilmektedir. Öte yandan, kaba (coarse) izleme için ekipmandan elde edilen çevrimiçi veriler kullanılmaktadır. Çevrimiçi sensör verileri, ekipman/sistem sağlığı ve performansı hakkında bilgiler içerdiğinden bu verilerin eksiksiz ve gerçek zamanlı olarak elde edilmesi beklenmektedir.

Jimenez vd. (2020) kestirimci bakım sistemlerini istenmeyen arızaların önüne geçmek için arızaların/anomalilerin kaynağını ve sistemin sağlık durumunu belirlemeyi amaçlayan tanısal (diagnostics) yöntemler ve sistemin gelecekteki durumunun öngörülebilmesi ve RUL tahmini için kullanılan belirtisel (prognostics) yöntemler olarak iki ayrı kapsamda incelemektedir. Literatür incelendiğinde, son yıllarda her iki kestirimci bakım uzantısının bilimsel çalışmalarda ve endüstride oldukça yaygın olarak kullanıldığı görülmekte ve bu yöntemlerin fırsatlarını, sınırlarını ve zorluklarını belirlemeye yönelik kapsamlı araştırmalara rastlanmaktadır.

Ekipman bakımı, endüstride kritik bir anahtardır ve ekipmanın çalışma süresini ve sistem verimliliğini büyük ölçüde etkilemektedir. Bu nedenle, üretim faaliyetlerinde aksamalardan ve durmalardan kaçınarak teknik ekipman arızalarının tanımlanması ve çözülmesi gerekmektedir (Wan vd., 2017). Bu kapsamda, Vafaei vd. (2019) bir araba üretim hattında yüksek maliyete sebep olan ani duruşların önüne geçmek amacıyla ekipman arızalarını önceden tahmin eden bir bulanık (fuzzy) alarm sistemi önermektedir. Benzer yaklaşımla, Dong vd. (2019) bakım maliyetlerini azaltmak, gereksiz hizmet kesintilerini önlemek ve bakım planlamalarını optimize etmek amacıyla üretim sistemlerindeki sensör arızalarını öngörebilmek için bir prognostik ve sağlık yönetimi (Prognostics Health Management - PHM) çerçevesi geliştirmiştir. Angelopoulos vd. (2019), Carvalho vd. (2019) ve Dündar vd., (2021) kestirimci bakım sistemlerinde uygulanan ML tekniklerine ve bu alandaki son gelişmelere yönelik sistematik bir literatür incelemesi sunmaktadır. Bu incelemeler sonucunda: arıza teşhisi/tespiti, RUL tahmini ve ekipman sağlığı/performansı izlenmesinde teknik ekipman olarak çoğunlukla rulman, motor ve rüzgar türbinleri tercih edildiği ve bu ekipmanlar üzerinde deneyler gerçekleştirildiği, kestirimci bakım alanında ML tekniklerini içeren uygulamaların artış eğiliminde olduğu ve incelenen çalışmalarda

derin öğrenme, KNN, Random Forest, SVM, karar ağaçları (DT), yapay sinir ağları (ANN) ve lojistik regresyon (LR) algoritmalarının kestirimci bakım uygulamalarında en çok tercih edilen algoritmalar olduğu sonucuna varılmaktadır.

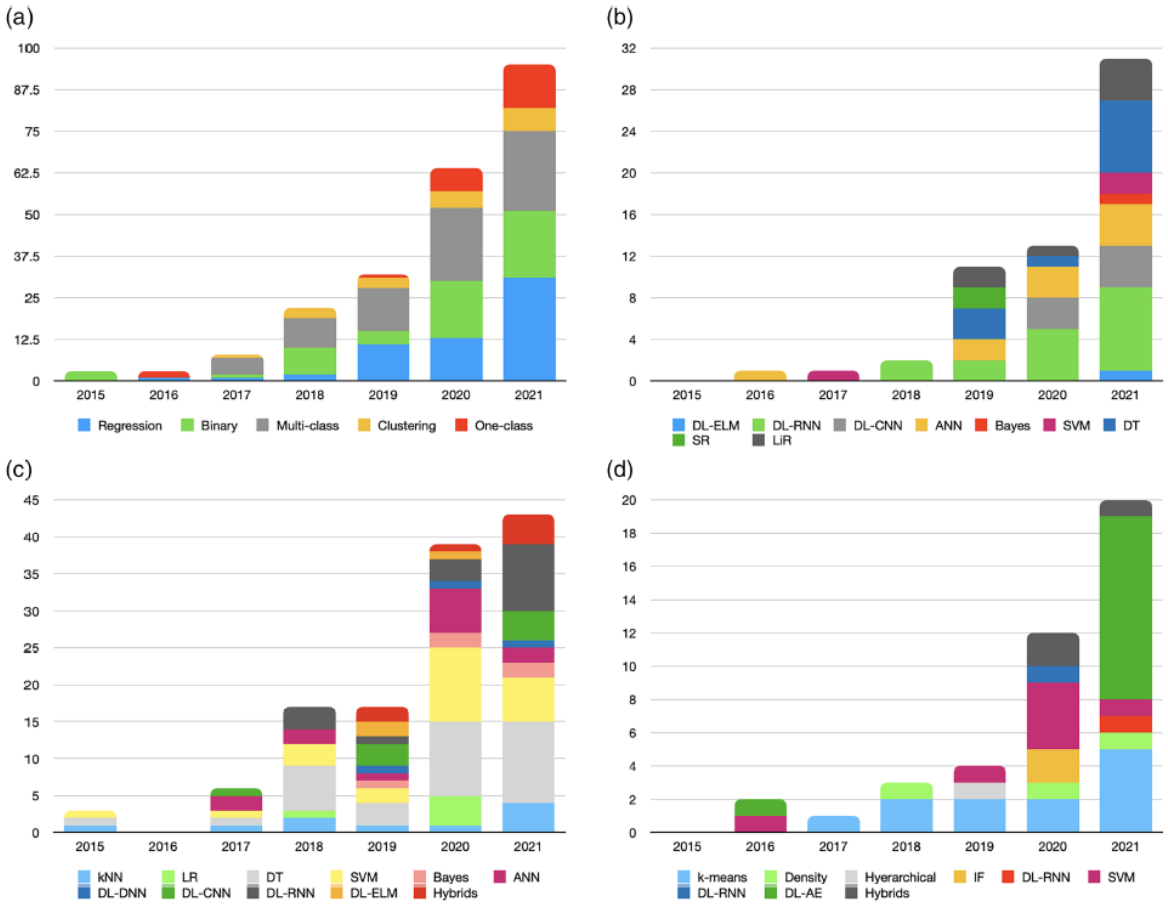
Esteban vd., (2022) kestirimci bakım ile ilgili mevcut durumu veri madenciliği perspektifinden ele alarak kapsamlı bir literatür taraması sunmakta ve kestirimci bakıma uygulanan veri madenciliğinin potansiyeli ve zorluklarına değinmektedir. Şekil 2.6, Scopus akademik veritabanına göre 1970-2021 yılları arasında yayımlanan kestirimci bakım ile ilişkili tüm akademik yayınlara ve kestirimci bakım alanındaki makine öğrenmesi, derin öğrenme veya veri madenciliği anahtar kelimeleriyle ilişkili akademik yayınlara genel bir bakış sunmaktadır.



Şekil 2.6. Scopus'ta yer alan yıllık akademik yayın sayısı: PdM ve makine öğrenmesi ile ilişkili PdM (Esteban vd., 2022)

Esteban vd. (2022) tarafından gerçekleştirilen literatür analizi sonucunda, PdM alanındaki ilk çalışmaların 1970'li yıllara dayandığı ve bu alandaki çalışmaların 2016 yılından bu yana katlanarak büyüdüğü söylenebilmektedir. PdM uygulamalarındaki derin öğrenme ve makine öğrenmesi tabanlı veri madenciliği yaklaşımının 2015 yılından itibaren gözle görülür bir artış gösterdiği ve 2021 yılında yayımlanan kestirimci bakım alanındaki makine öğrenmesi, derin öğrenme ve veri madenciliği ile ilişkili yayınların, yalnızca PdM ile ilişkili yayınların yaklaşık üçte birini oluşturduğu görülmektedir. Kestirimci bakım kapsamındaki arıza/anomali tespiti, RUL tahmini ve kademeli bozulma durumlarının

öngörülmesi gibi problemler çeşitli yöntemlerle ele alınmaktadır. Şekil 2.7'de PdM uygulamalarında kullanılan bu yöntem ve yaklaşımların yıllara göre bir analizi bulunmaktadır. Bu analiz grafiği, PdM alanındaki en son gelişmeler dikkate alınarak hazırlanmış ve dört ayrı başlıkta kategorize edilmiştir.



Şekil 2.7. PdM uygulamalarında kullanılan yöntemlerin yıllara göre analizi: (a) veri madenciliği görevinde kullanılan yöntemler, (b) regresyon görevinde kullanılan yöntemler, (c) sınıflandırma görevinde kullanılan yöntemler, (d) denetimsiz ve yarı-denetimli öğrenme görevinde kullanılan yöntemler (Esteban vd., 2022)

Şekil 2.7 (a), veri madenciliği çözüm tekniğinin kullanıldığı PdM çalışmalarının yıllara göre dağılımını temsil etmektedir. Bu tekniğin 2015 ve 2016 yıllarında üç, 2017 yılında sekiz, 2018 yılında yirmi iki, 2019 yılında otuz iki, 2020 yılında altmış dört ve 2021 yılında doksan beş akademik çalışmada tercih edilerek son yıllarda büyük bir artış eğilimi gösterdiği gözlenmektedir. Ek olarak, regresyon ve çok sınıflı sınıflandırma yaklaşımlarının şu anda PdM alanındaki veri madenciliği görevinde kullanılan en popüler teknikler olduğu

söylenebilmektedir. Şekil 2.7 (b), PdM problemlerinde, özellikle RUL tahmininde, regresyon görevini kullanan akademik yayınların yıl bazında bir analizini göstermektedir. Bu grafikte sinir ağlarına doğru olan kayma dikkat çekmektedir. Şekil 2.7 (c), PdM problemlerinde, özellikle sağlık durumu ve başarısızlık tahmininde, sınıflandırma görevini kullanan akademik yayınlara genel bir bakış sunmaktadır. Daha açıklanabilir sonuçlar elde edilebilmesi sebebiyle, son yıllarda DT ve KNN gibi klasik modellere olan talebin arttığı görülmektedir. Öte yandan, yüksek doğruluk değeri sebebiyle CNN ve RNN gibi derin öğrenme modelleri PdM alanında etkisini sürdürmektedir. Şekil 2.7 (d), PdM uygulamalarında denetimsiz ve yarı-denetimli öğrenme yaklaşımlarının kullanıldığı çalışmaların sayılarını yıl bazında göstermektedir. K-Means ve OCSVM gibi ilk ortaya çıkan yöntemlerin ilerleyen yıllarda da tercih edilmeye devam etmesi dikkat çekmektedir. Özellikle otomatik kodlayıcı (Auto-Encoder) tabanlı derin öğrenme modeli 2021 yılı boyunca şaşırtıcı derecede ilerleme kaydetmektedir. Kümeleme yöntemleri arasından Hiyerarşik kümelenin aksine K-Means kümeleme algoritmasının, yorumlanabilirliği ve yanıt verme hızı sebebiyle, 2017 yılından beri en tutarlı şekilde kullanılan yöntem olduğu söylenebilmektedir.

3. MATERYAL VE YÖNTEM

Bu tez çalışmasında ele alınan kestirimci bakım sistemlerinde veri artırma yöntemlerinin geliştirilmesi kapsamında çeşitli veri setleri, makine öğrenmesi yöntemleri ve teknolojiler kullanılmıştır. Bu bölümde akıllı fabrika ortamı, tasarlanan IoT sistem platformu ve bileşenleri, çalışmada kullanılan veri seti, makine öğrenmesi yöntemleri, kullanılan araç ve platformlar detaylı olarak açıklanmaktadır.

3.1. Materyal

Çalışmada kullanılan akıllı fabrika test ortamının ve IoT sistem platformunun genel tanıtımı, kullanılan veri seti, Python programlama dili ve kütüphanelerine ait bilgiler izleyen alt bölümlerde verilmektedir.

3.1.1. Akıllı fabrika ortamı ve IoT sistem platformu

Akıllı fabrika, tüm süreçlerin otomasyon kullanılarak gerçekleştiği, sanal ve fiziksel dünyanın entegrasyonunu sağlayan ve günümüz pazarının ihtiyaçlarını karşılamak için daha esnek, kaliteli ve verimli üretime olanak sağlayan bir üretim çözümü olarak tanımlanabilmektedir (Hozdić, 2015). Endüstri 4.0'ın getirdiği yeniliklerle beraber, fabrika ortamındaki makine/ekipman ve otonom robotların hem birbirleriyle hem de insanlarla haberleşebilmesi ve veri alışverişinde bulunabilmesi mümkün hale gelmiştir.

Bu çalışmada, test ortamı olarak son teknoloji otonom sistemlerin entegre edildiği bir akıllı fabrika ortamı olan Eskişehir Osmangazi Üniversitesi (ESOGÜ) Akıllı Fabrika ve Robotik Laboratuvarı (IFARLAB, 2022) tercih edilmiştir. Bu laboratuvar, gerçek zamanlı olarak uçtan uca iletişim kurabilen ekipmanlar ve yapay zeka tabanlı bakım sistemleri ile

çalışma imkanı sağlamaktadır. Akıllı fabrika test ortamı Şekil 3.1'de yer almaktadır. Test ortamı, gerçek bir fabrikadaki malzeme taşıma araçları için oluşturulan yolları ve üretim alanlarını içermektedir. Bu test ortamında Otonom Taşıyıcı Araç (Autonomous Transport Vehicle - ATV) farklı rotalar ve koşullar altında test edilmektedir. Şekilde mavi ile temsil edilen test alanı, Otonom Taşıyıcı Araç (ATV) testlerinin yapıldığı zemindir.

Şekil 3.1. ESOGÜ Akıllı Fabrika ve Robotik Laboratuvarı (IFARLAB, 2022)

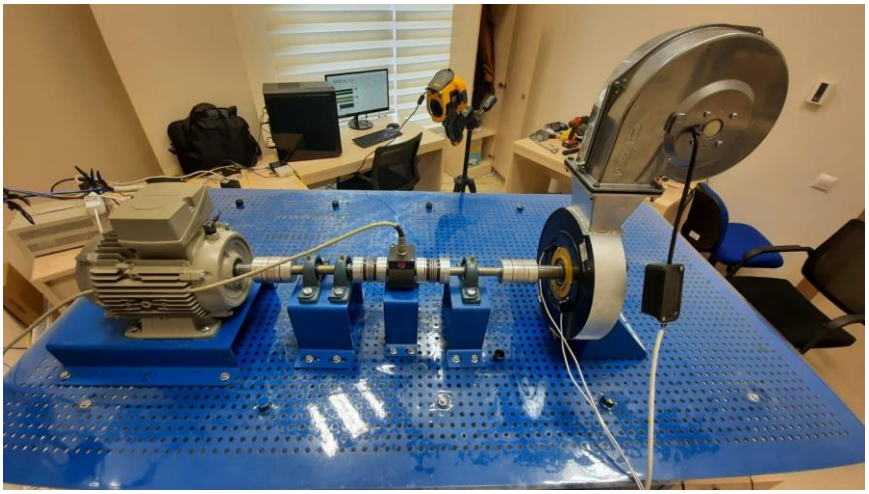
Otonom Taşıyıcı Araç (ATV), sensörler, aktüatörler, internet bağlantısı ve otomatik kontrol sistemi donanımından oluşan, insan müdahalesine ihtiyaç duymadan hareket edebilen ve fiziksel çevresini algılayabilen yeni nesil bir araçtır (Garretson vd., 2016).

Akıllı fabrika ortamının bir kullanım senaryosu olan ATV, maksimum 100 kilogram kapasiteli yükü alıp istenilen hedef konuma taşımak üzere tasarlanmış, diferansiyel tahrikli bir taşıyıcı araçtır. ATV'de açık kaynaklı bir arakatman yazılımı olan Robot İşletim Sistemi (Robot Operating System - ROS) kullanılmaktadır. Görev bazlı operasyonlarda, araç

istenilen hedef konuma doğru hareket ederken çarpışmalardan ve engellerden kaçınmak için haritalama sistemi, navigasyon sistemi ve Sonlu Durum Makinesi (Finite State Machine - FSM) sistemi kullanılmaktadır. ATV dış boyutları 1,026 x 0,728 x 0,325 m'dir. Testlerde ATV'nin maksimum doğrusal hızı 0,4 km/h ve açısal hızı 0,55 rad/s olarak belirlenmiştir. Şekil 3.2'de otonom taşıyıcı aracın bir görseli bulunmaktadır.

Şekil 3.2. Otonom Taşıyıcı Araç (Autonomous Transport Vehicle - ATV)

Ek olarak, akıllı fabrika laboratuvarında, endüstride oldukça yaygın olarak kullanılan elektrik motoru kullanım senaryosuna da yer verilmektedir. Asenkron/İndüksiyon motorlarının en önemli bileşeni olan rulmanlardaki arızaların tespitine, titreşim, akım ve tork sensör verilerinin analizine dayalı yapay zeka tabanlı bakım planlamasına yönelik birçok çalışma gerçekleştirilmektedir. Şekil 3.3'te elektrik motorunun bir görseli yer almaktadır.

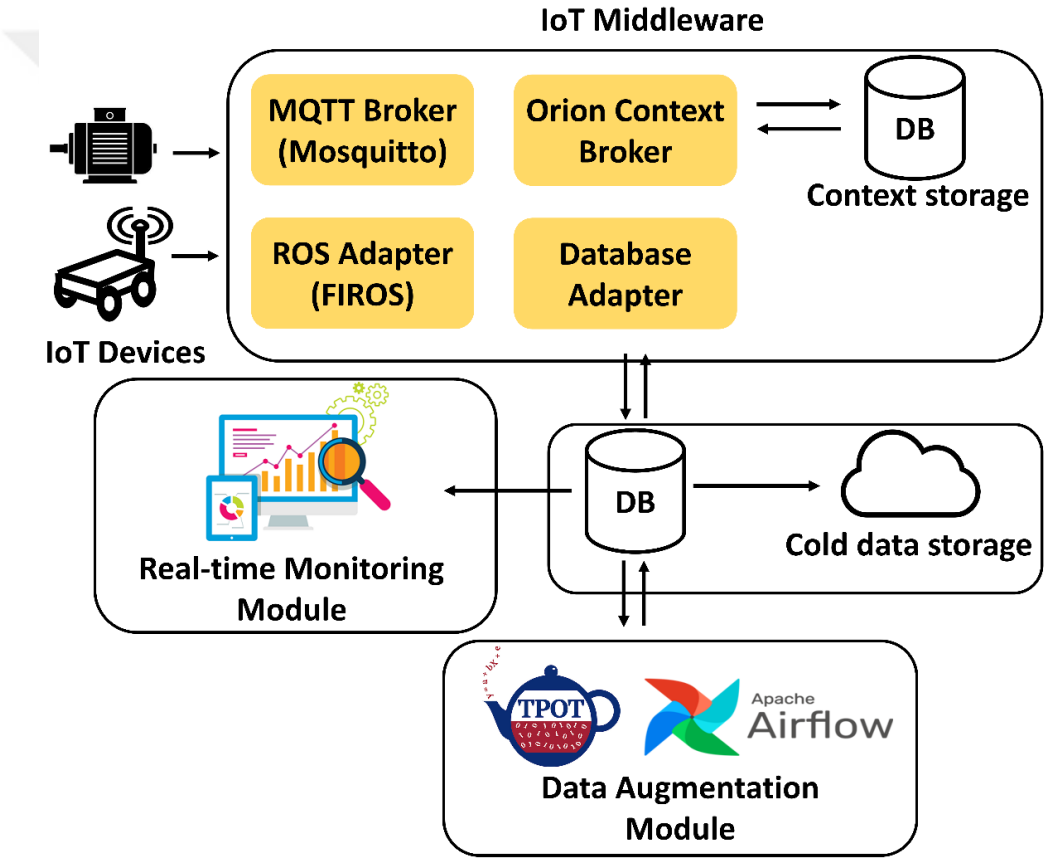


Şekil 3.3 IFARLAB'ta yer alan elektrik motoru

PdM, gelecekteki arızaları tahmin etmek için normal işlemler sırasında ekipman performansının izlenmesini vurgulamaktadır. Böylelikle beklenmeyen arıza sayısı en aza indirilir, bakım için ayrılan süre ve maliyet azalır, verimlilik artar. Bu anlayış, ekipman bakımını yalnızca çalışma süresine göre planlamamaktadır. Bu noktada devreye giren büyük veri analiz teknikleri, birden çok kaynaktan gelen büyük ölçekli veri akışlarını gerçek zamanlı olarak işleyerek PdM yaklaşımının kullanılmasına olanak sağlamaktadır (Su ve Huang, 2018). Siber-fiziksel sistemlerin kestirimci bakımı, bir makinenin veya süreç bütünlüğünün sürekli olarak izlenmesini sağlamaktadır ve yalnızca gerektiğinde bakım gerçekleştirmektedir (Carvalho vd., 2019). Siber-fiziksel sistemlerde endüstriyel büyük verileri yönetebilen PdM uygulamalarını faaliyete sokabilmek için genel ve fonksiyonel bir mimari tasarlamak oldukça kritik bir gereksinim haline gelmiştir (Lee vd., 2015).

Çoğu PdM mimarisinin temel yapı taşını oluşturan IoT ara katman (middleware) yazılımı, izlenen ekipmanlar ve sistem arasında gerçek zamanlı iletişim ve veri iletimini sağlamaktadır. Çeşitli haberleşme protokolleri ve arayüzleri aracılığıyla teknolojik altyapı farklılıklarını ortadan kaldırmaktadır. Bu çalışmada, ölçeklenebilirliği ve modüler yapısı nedeniyle IoT ara katman yazılımı olarak Fiware adlı açık kaynaklı bir yapı tercih edilmiştir ve tüm yazılım yığını, tasarlanan yeni PdM sisteminin ihtiyaçlarına yönelik olarak optimize edilmiştir. Fiware, farklı IoT senaryolarında çeşitli API'ler (Uygulama Programlama Arayüzü) kullanarak bağlam verilerini belirli standartlar çerçevesinde yönetmeyi amaçlayan

açık kaynak kodlu bir platformdur (Fiware Foundation, 2022). Fiware ekosistemi, son kullanıcıların farklı alanlardaki uygulamalarda belirli ihtiyaçlara göre özelleştirebileceği Genel Etkinleştiriciler (Generic Enablers - GE) adı verilen açık kaynaklı yazılım bileşenlerinden oluşmaktadır. Her bir GE, belirli bir görevde uzmanlaşmıştır ve merkezi bir Bağlam Aracısı (Context Broker) üzerinden iletişim kurabilmektedir. Bu çalışmada, kestirimci bakım uygulamaları için özel olarak tasarlanmış yeni bir IoT mimarisi önerilmektedir. Tasarlanan IoT sistemin mimarisi genel hatlarıyla Şekil 3.4'te gösterilmektedir.



Şekil 3.4. Kestirimci bakım uygulamaları için tasarlanan IoT sistem mimarisi

Şekil 3.4'te yer alan mimari, genel veri adaptörleri ve IoT sistemlerinde yaygın olarak kullanılan iletişim protokolleri ile çeşitli yazılım ve donanım bileşenlerinden oluşmaktadır. Genel olarak bu bileşenler, IoT sensörleriyle donatılmış uç cihazları, IoT ara

katman yazılımı, sıcak ve soğuk veri depolama bileşenleri, veri zenginleştirme modülü ve sistemin gerçek zamanlı izlenmesini sağlayan veri analitiği modülü olarak gruplandırılabilmektedir.

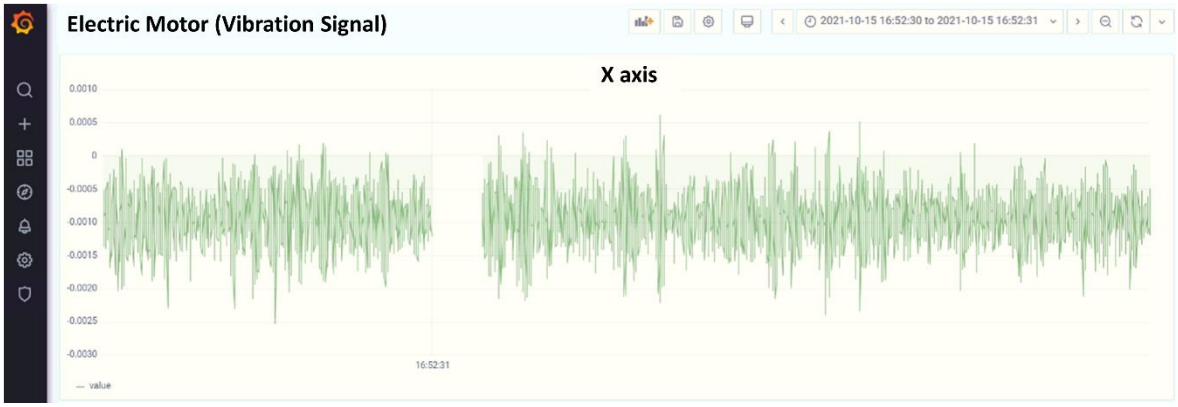
IoT uç cihazları, tasarlanan PdM sisteminde elektrik motorları için MQTT (Message Queuing Telemetry Transport) gibi veri iletişim protokolleri ve otonom taşıyıcı araçlar için ROS gibi ara katman yazılımları aracılığıyla kesintisiz veri toplamayı sağlayan IoT sensörleri ile donatılmıştır. Sistem, çalışma süresi boyunca birçok süreç ve ekipman filosu içeren akıllı bir fabrika ortamından zaman serisi verileri toplamakta ve bu verileri gerçek zamanlı olarak izlemektedir.

Tasarlanan sistem, birçok ekipman arayüzü ve iletişim protokolü için özelleşmiş farklı adaptörlere sahiptir. Örneğin, bir programlanabilir mantıksal denetleyici (Programmable Logic Controller - PLC) ekipmanı için endüstriyel bir veri toplama arayüzü gerekiyorsa, ekipmanı sisteme bağlamak için OPC/UA adaptörü kolayca entegre edilebilmektedir. Bu adaptörler, veri formatlarını birbirlerine dönüştürerek iki farklı bileşen arasında çevirmen görevi görmektedir (Fiware, 2022). Sistem mimarisinde kullanılan Firos adındaki ROS adaptörü, üretim katındaki ATV filolarında yüklü olan ROS ile Orion bağlam aracısı arasına konumlandırılan mikroservis-tabanlı bir paket çeviricisidir. Firos, ROS ve küresel olarak standartlaştırılmış NGSI (Next Generation Service Interface) mesaj formatları arasında gerekli dönüşümü sağlamak ile yükümlüdür. Öte yandan Mosquitto, MQTT protokolü ile haberleşen ekipmanlar için açık kaynaklı bir mesaj broker uygulamasıdır. Bu MQTT adaptörü, elektrik motoru ve IoT ara katman yazılımı olan Fiware arasında bir köprü görevini üstlenerek iletişim kurulmasına olanak sağlamaktadır. Elektrik motoruna ait çeşitli sensör verileri, MQTT protokolü üzerinden ara katman yazılımına aktarılmaktadır. Aktarılan veriler, titreşim, akım ve tork gibi ham sensör verilerini veya uç birimlerde elde edilen istatistiksel özellikleri içermektedir.

Orion bağlam aracısı (Context Broker), PdM platformunun çekirdek/temel bloğu olarak ifade edilebilmektedir. Bu bileşen sayesinde bağlam (context) verileri kaydedilmekte, sorgulanmakta ve güncellenebilmektedir. Orion Context Broker, tüm bağlam verilerini

(varlıklar (entities), abonelikler (subscriptions) vb.) tarihsel olarak kalıcı hale getirmek için MongoDB gibi bir veritabanı kullanmaktadır. Ara katman yazılımı Fiware ve veritabanı arasındaki bir dönüştürücü adaptör yardımıyla, sıcak veriler olarak adlandırılan akan sensör verileri (akım, titreşim, tork, ses vb.) evrensel bir zaman parametresi ile bir veritabanına iletilerek kaydedilmektedir. Bu dönüştürücü veritabanı adaptörü, seçilen veritabanına göre değişiklik gösterebilmektedir. Örneğin ElasticSearch kullanan bir sistemde bu adaptör Logstash olarak tercih edilmektedir.

Nadiren kullanılan veriler periyodik olarak buluta yedeklenmektedir. Bu, birincil depolama kaynağının aşırı yüklenmesini önleyerek depolama maliyetini düşürmektedir. Sık kullanılan ve anında erişilen verileri içeren veritabanı, gerçek zamanlı bir monitörleme aracıyla (Grafana, Kibana vb.) etkileşime girmektedir. Sıcak sensör verileri ve sistem performansını ve sağlığını temsil eden Temel Performans Göstergeleri (Key Performance Indicators - KPI), gösterge panolarında anında izlenebilmektedir. Örnek bir gösterge panosu Şekil 3.5'te yer almaktadır.



Şekil 3.5. Grafana Veri Analitiği Aracı aracılığıyla gerçek zamanlı monitörleme

Sensör verilerinin ve tanımlanmış KPI'ların gerçek zamanlı izlenmesi sırasında bir sorunla karşılaşıldığı görülmektedir. Bu çalışmanın temel nedeni haline gelen bu sorun, veri toplama sırasında oluşan eksik verilerdir. Ağ arızaları, sensör arızaları vb. nedenlerle yazılamayan veriler, veri setlerinde ve gösterge tablolarında boşluklar oluşturmaktadır. Şekil 3.5'te elektrik motorunun titreşim sinyalinin (X ekseni) bir kısmının yazılamadığı ve veri

setinde boşluk oluştuğu görülmektedir. Gün sonunda elde edilen veri setindeki bu tür boşlukların, bu veri setinden elde edilecek analizlerin kalitesini ve gücünü düşürdüğü bilinmektedir. Bu nedenle mevcut IoT sistemine yeni bir veri zenginleştirme modülü eklenmiştir. Bu modül, sıcak verinin tutulduğu veritabanı ile etkileşime girerek veri setlerindeki eksik veriyi en gerçekçi ve olağan şekilde impute etmeyi amaçlamaktadır. Modül, veritabanındaki verileri belirli periyotlarda gruplar halinde okuyarak eksik verileri tespit etmektedir. Eksik veri durumunda, veri seti için en iyi performansı gösteren regresyon-tabanlı ML model ile eksik/kayıp değerler, gerçek değerlere en yakın şekilde impute edilerek tamamlanmaktadır.

3.1.2. Veri Seti

Ön hazırlık çalışmasında kullanılan veri seti, bir önceki başlıkta tanıtılan sistemin bir kullanım senaryosu olan IoT sensörleriyle donatılmış bir elektrik motorunun titreşim ve akım sensörlerinden gelen zaman serisi verilerini içermektedir. Bu veriler, elektrik motoru normal koşullar altında çalışırken toplanmıştır. Şekil 3.6'da, elektrik motoruna ait titreşim ve akım sensör verilerinin Grafana Veri Analitiği Aracı ile gerçek zamanlı olarak izlendiği bir gösterge paneli (dashboard) örneği yer almaktadır.

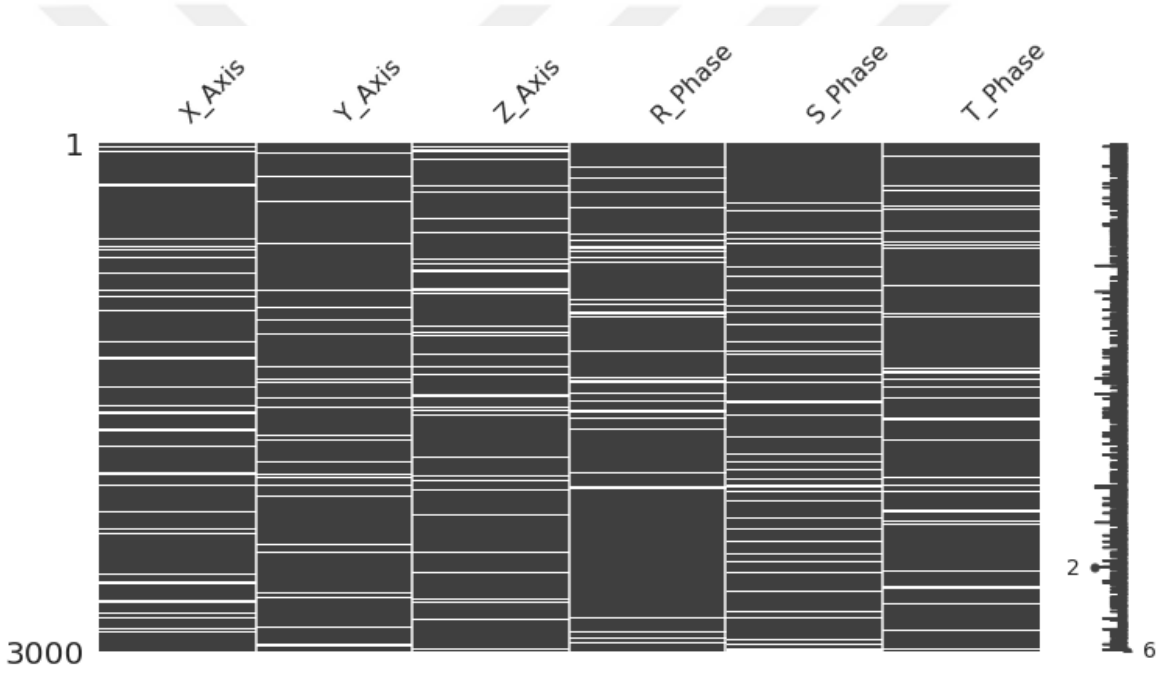
Şekil 3.6. Elektrik motoru sensör verilerine ilişkin bir gösterge paneli örneği

Veri setinde 3000 kayıt ve altı öznitelik bulunmaktadır: Titreşim sinyallerinin üç eksenli verileri (X ekseni, Y ekseni, Z ekseni) ve akım sinyallerinin üç fazlı verileri (R fazı, S fazı, T fazı). Herhangi bir eksik kaydı olmayan veri setinde çeşitli konum ve oranlarda eksik veriler oluşturulmuştur. Oluşturulan eksiklerin konumları şu şekilde tanımlanmıştır:

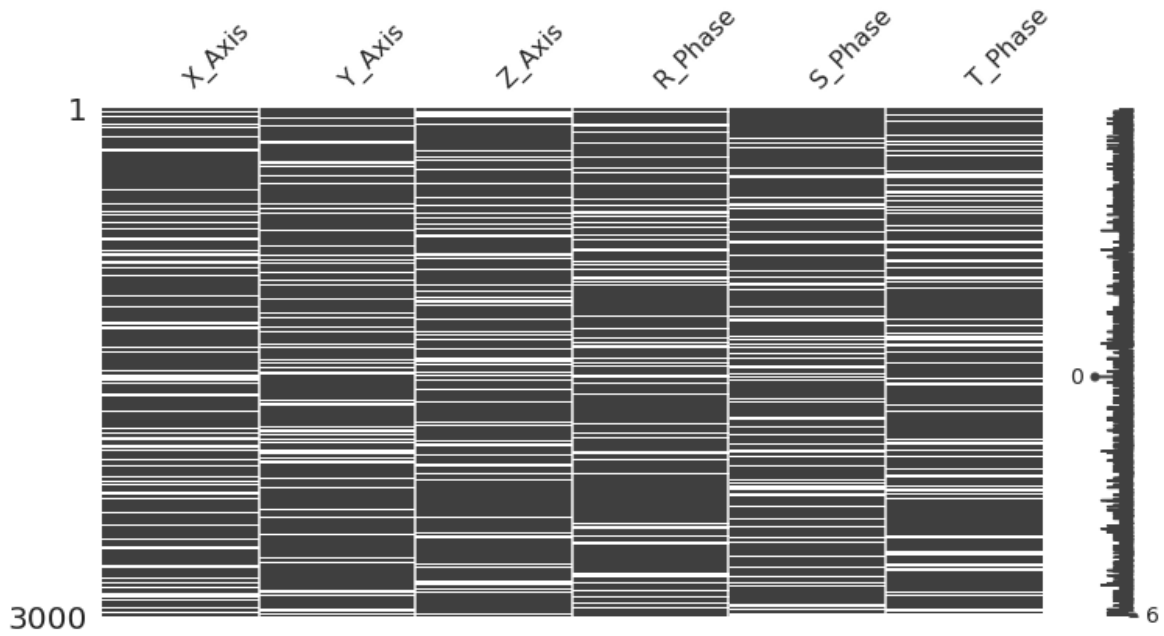
1. Tüm veri setine dağılan boşluklar "rastgele (random)",
2. İlk yüzde 33,3'lük dilim "başlangıç (beginning)",
3. İkinci yüzde 33,4'lük dilim "orta (center)",
4. Son yüzde 33,3'lük dilim "bitiş (end)".

Eksiklik oranları %10, %20, %30, %40 olarak belirlenmiştir. Benzer şekilde performans göstermeleri beklendiği için %10'dan az boşluklar dahil edilmemiş (Dong ve Peng, 2013) ve

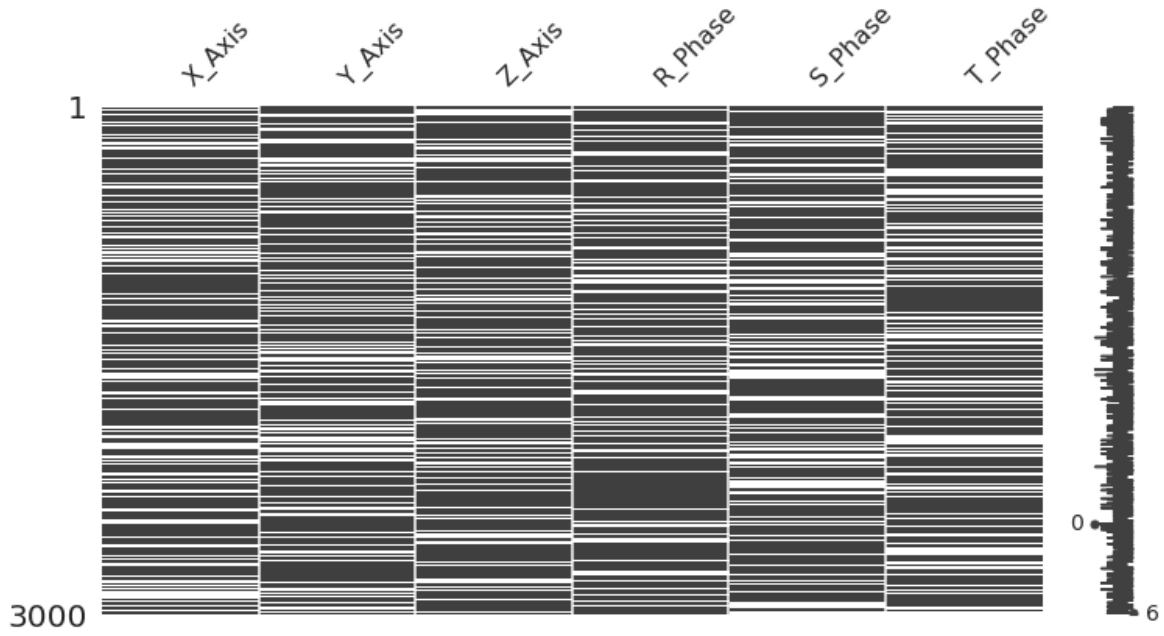
deneme sonuçları yalnızca hipotez üreten sonuçlar olarak değerlendirilebileceğinden maksimum kayıp veri oranı %40 olarak belirlenmiştir (Jakobsen vd., 2017). Toplamda, on altı kombinasyon için veri kümeleri türetilmiş ve tüm durumlar için altı ML modeli tarafından atama/imputasyon gerçekleşmiştir. Şekil 3.7'de rastgele (random) durum için %10 eksik veri, Şekil 3.8'de rastgele (random) durum için %20 eksik veri, Şekil 3.9'da rastgele (random) durum için %30 eksik veri, Şekil 3.10'da rastgele (random) durum için %40 eksik veri içeren veri görselleme matrisi bulunmaktadır. Burada beyaz kısımlar eksik (null) değerleri temsil etmektedir. Eksik veri görselleştirmesi, veri setlerindeki boşlukların dağılımı hakkında bilgi sağlamaktadır.



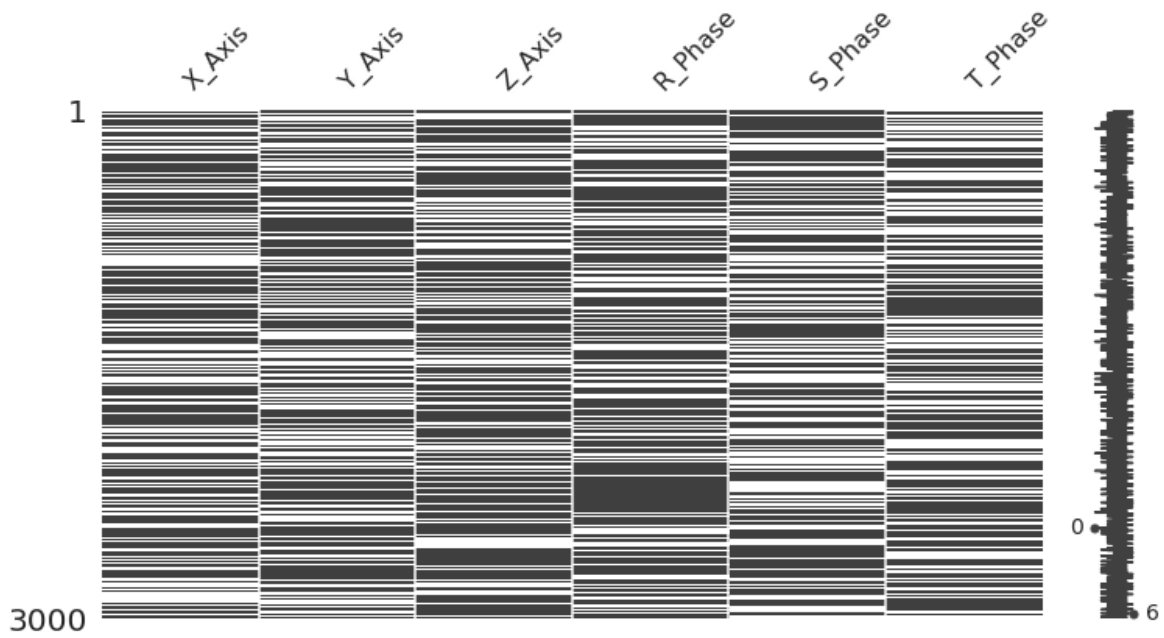
Şekil 3.7. Rastgele %10 durumu için eksik veri görselleme matrisi



Şekil 3.8. Rastgele %20 durumu için eksik veri görselleme matrisi

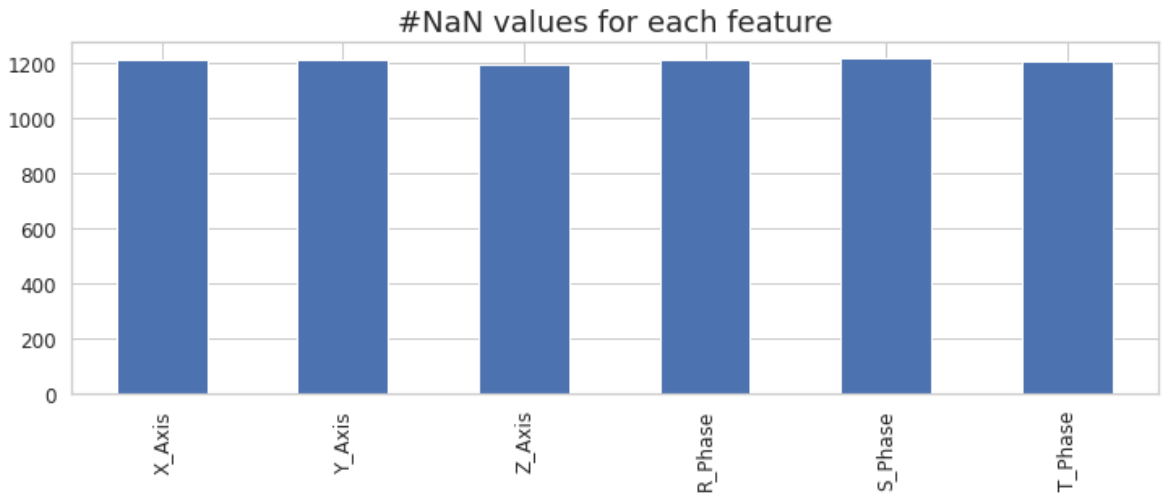


Şekil 3.9. Rastgele %30 durumu için eksik veri görselleme matrisi



Şekil 3.10. Rastgele %40 durumu için eksik veri görselleme matrisi

Eksik değerlerin miktarı, Şekil 3.11'e benzer şekilde satır veya sütun bazında matris, çubuk grafik, dendrogram ve ısı haritası ile görselleştirilebilmektedir.



Şekil 3.11. Rastgele %40 durumu için eksik veri miktarını temsil eden çubuk grafiği

Deneysel çalışmaya başlamadan önce eksik verilerin çeşitli yöntemlerle görselleştirilmesi araştırmacıya bu veri seti ile elde edilen sonuçların yanlılığı ve tutarlılığı hakkında ön bilgi sağlamaktadır.

3.1.3. Python programlama ve yararlanılan kütüphaneler

1989 yılında Guido Van Rossum tarafından geliştirilen Python, açık kaynak kodlu, nesne yönelimli programlamayı destekleyen, ücretsiz bir programlama dilidir. Yüksek düzeyde etkileşimli doğası ve olgunlaşan bilimsel kütüphane ekosistemi sayesinde, algoritma geliştirme ve veri analizi için en iyi seçimlerden biri olduğu söylenebilmektedir (Dubois, 2007). Genel amaçlı bir dil olarak sadece akademik çalışmalarda değil, endüstri alanında da oldukça fazla kullanılmaktadır (Pedregosa vd., 2011). Python için tasarlanmış olan NumPy, Pandas, Matplotlib, SciPy ve Scikit-learn kütüphaneleri çok hızlı ve dinamik bilimsel programlamalar yapabilmeyi ve veriyi analiz ederek görselleştirebilmeyi mümkün kılmaktadır. Bu sebeple, özellikle son yıllarda yapay öğrenme, web programlama, finans, biyoloji ve fizik gibi çok geniş bir yelpazede oldukça tercih edilen bir programlama dili haline gelmiştir (Arslan, İ., 2019). Bu tez çalışması kapsamında tüm geliştirmeler Python programlama dili ile yapılmış ve başlıca Scikit-learn, Pandas, NumPy ve Matplotlib kütüphaneleri kullanılmıştır.

Scikit-learn, Python'daki en kapsamlı ve açık kaynaklı makine öğrenimi kütüphanesidir (Hao ve Ho, 2019). Bu kütüphane, makine öğrenimini akademide ve endüstride bulunan herkes için erişilebilir kılmayı amaçlayan topluluk destekli güçlü bir yazılım projesi olarak tanımlanabilmektedir (Varoquaux vd., 2015). Scikit-learn, temel olarak makine öğreniminin dört ana başlığını kapsamaktadır. Bunlar veri dönüştürme (data transformation), denetimli (supervised) öğrenme, denetimsiz (unsupervised) öğrenme ve model değerlendirme/seçme (model evaluation/selection) olarak sıralanabilmektedir.

Scikit-learn kütüphanesinin eksik veri atamasına yönelik olarak geliştirilmiş çeşitli paketleri bulunmaktadır. “SimpleImputer” paketi, tek bir değişkende bulunan eksik verileri

ortalama atama, medyan atama, mod atama veya sabit bir değer atama gibi basit atama yöntemleri ile doldurmayı hedeflemektedir. “IterativeImputer” paketi, her bir değişkendeki eksik veriyi diğer değişkenler yardımıyla modelleyerek çoklu atamayı sağlamaktadır. “KNNImputer” paketi ise her bir eksik değeri en yakın komşulardan alınan ortalama değerler ile doldurarak eksiksiz bir veri setine olanak tanımaktadır. Bu paketler “from sklearn.impute import IterativeImputer” gibi tek bir komut satırı ile projeye kolayca dahil edilebilmektedir.

2008 yılında finansal veri analizi uygulamaları için geliştirilen Pandas kütüphanesi veri yükleme, veri ön işleme, veri temizleme ve veri analizinde kullanılan açık kaynaklı güçlü bir Python kütüphanesidir (Mckinney vd., 2010). Veri setlerinin tekrar boyutlandırılması, döndürülmesi, birleştirilmesi ve etiket bazlı gruplandırılması gibi işlevleriyle öne çıkan bu kütüphane veri analistleri için bir vazgeçilmez haline gelmektedir (Mckinney, 2012). Tek boyutlu seriler ve iki boyutlu veri yapısı (DataFrame) olarak adlandırılan iki temel veri yapısı ile .CSV (Comma Separated Values) uzantılı dosyaların kolaylıkla okunmasını ve işlenmesini sağlamaktadır. Pandas kütüphanesi, eksik/kayıp verilerin kolayca tespit edilebilmesi için “isnull” ve “notnull” isminde özel iki API fonksiyonuna sahiptir (Mckinney vd., 2011). Bu fonksiyonlar veri setlerindeki eksik değerlerin kolayca tespit edilmesine ve yönetilmesine imkân tanımaktadır.

NumPy (Numerical Python) çok boyutlu diziler ve matrisleri destekleyen, üst düzey bilimsel hesaplamaların hızlı bir şekilde yapılabilmesini sağlayan açık kaynaklı ve topluluk destekli bir Python kütüphanesidir. NumPy'ın temelini oluşturan NumPy dizileri sayısal veriler için standart bir temsildir (Van Der Walt vd., 2011) ve çok boyutlu dizilerin verimli bir şekilde depolanmasını ve onlara kolayca erişilebilmesini sağlayarak çeşitli bilimsel hesaplamaları mümkün kılan bir veri yapısıdır (Harris vd., 2020).

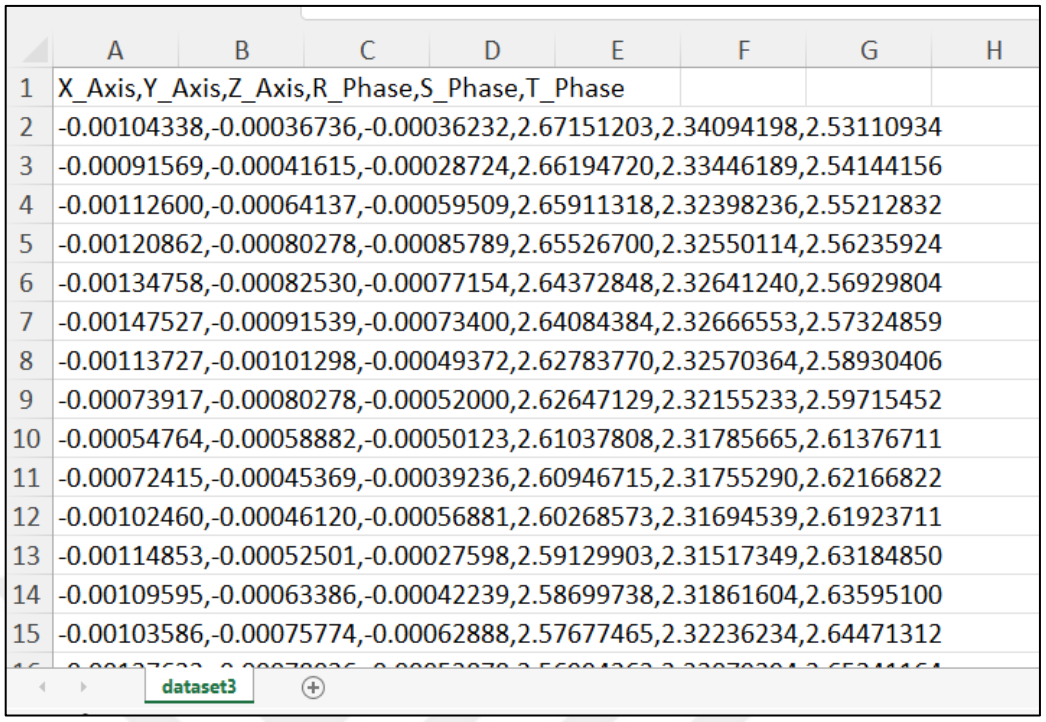
Matplotlib, temel bir veri görselleştirme ve 2D/3D grafik çizim kütüphanesidir. John D. Hunter tarafından kullanıcının yalnızca bir veya birkaç komutla çeşitli grafikler oluşturabilmesi felsefesi ile ücretsiz ve açık kaynaklı olarak tasarlanmıştır (Barrett vd., 2005). Matplotlib sayesinde birkaç satır kod ile grafikler, histogramlar, çubuk grafikleri, hata çizelgeleri, güç spektrumları ve dağılım grafikleri üretebilmek mümkündür. Bu

kütüphanenin temel hedeflerinden biri yayın kalitesinde görseller sağlamaktır. Sağladığı çözünürlük (dpi) ve kalite (quality) parametreleri sayesinde birçok yüksek çözünürlüklü grafik jpg, png gibi formatlarda dışa aktarılabilmektedir.

TPOT (Tree-based Pipeline Optimization Tool), denetimli (supervised) sınıflandırma ve regresyon algoritmalarını içeren ve en iyi tahminleme performansı gösteren modeli, hiperparametreleri ve veri hattını (pipeline) otomatik olarak keşfeden açık kaynaklı bir Python kütüphanesidir. Otomatik Makine Öğrenimi (AutoML) yaklaşımıyla tüm olası veri hatlarını deneyerek en iyi sonuca ulaşmayı hedefleyen, genetik programlama tabanlı bu kütüphane çok kısa zamanda tahminleme başarısı yüksek ML modeller sunabilmektedir. Bu tez çalışmasında kullanılan TPOT kütüphanesi, Bölüm 3.2.2'de detaylı olarak anlatılmaktadır.

3.2. Yöntem

Çalışmanın ilk evresinden son evresine kadar çeşitli yöntemler uygulanarak ML modellerin veri atamadaki performansı analiz edilmiştir. Ön hazırlık çalışması kapsamında öncelikle seçilen altı regresyon-tabanlı ML algoritmasının veri imputasyon performansı, ESOGÜ Akıllı Fabrika ve Robotik Laboratuvarında tasarlanan IoT sistem platformunun bir bileşeni olan elektrik motoruna ait titreşim ve akım sensörlerinden elde edilen .CSV formatındaki offline veri setleri üzerinde manuel olarak test edilmiştir. Şekil 3.12'de normal şartlar altında toplanmış, tamamı eksiksiz, 3000 kayıt ve altı özniteliğe sahip veri setinden bir kesit yer almaktadır.

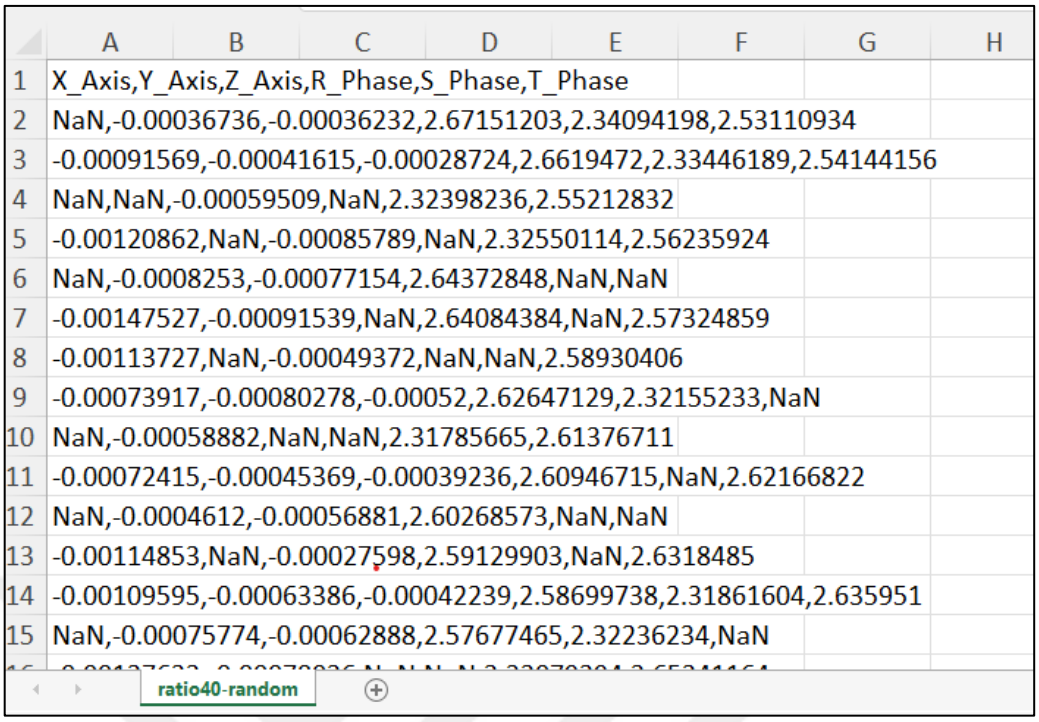


	A	B	C	D	E	F	G	H
1	X_Axis,Y_Axis,Z_Axis,R_Phase,S_Phase,T_Phase							
2	-0.00104338,-0.00036736,-0.00036232,2.67151203,2.34094198,2.53110934							
3	-0.00091569,-0.00041615,-0.00028724,2.66194720,2.33446189,2.54144156							
4	-0.00112600,-0.00064137,-0.00059509,2.65911318,2.32398236,2.55212832							
5	-0.00120862,-0.00080278,-0.00085789,2.65526700,2.32550114,2.56235924							
6	-0.00134758,-0.00082530,-0.00077154,2.64372848,2.32641240,2.56929804							
7	-0.00147527,-0.00091539,-0.00073400,2.64084384,2.32666553,2.57324859							
8	-0.00113727,-0.00101298,-0.00049372,2.62783770,2.32570364,2.58930406							
9	-0.00073917,-0.00080278,-0.00052000,2.62647129,2.32155233,2.59715452							
10	-0.00054764,-0.00058882,-0.00050123,2.61037808,2.31785665,2.61376711							
11	-0.00072415,-0.00045369,-0.00039236,2.60946715,2.31755290,2.62166822							
12	-0.00102460,-0.00046120,-0.00056881,2.60268573,2.31694539,2.61923711							
13	-0.00114853,-0.00052501,-0.00027598,2.59129903,2.31517349,2.63184850							
14	-0.00109595,-0.00063386,-0.00042239,2.58699738,2.31861604,2.63595100							
15	-0.00103586,-0.00075774,-0.00062888,2.57677465,2.32236234,2.64471312							

dataset3

Şekil 3.12. Elektrik motoru kullanım senaryosunda toplanan titreşim ve akım verileri

IoT sensörler yardımıyla anlık olarak toplanan ham veriler üzerinde birtakım değişiklikler yapılarak yeni veri setleri oluşturulmuştur. Eksiklik oranları %10, %20, %30, %40 ve konumları rastgele, başlangıç, orta, bitiş olacak şekilde toplamda on altı adet veri seti hazırlanmıştır. Şekil 3.13'te rastgele konumda ve %40 oranında eksik veriye sahip veri setinden bir kesit yer almaktadır. Burada "NaN" ile ifade edilen değerler eksik/kayıp verileri, bir diğer ifadeyle ham veri setinden manuel olarak silinmiş verileri temsil etmektedir.



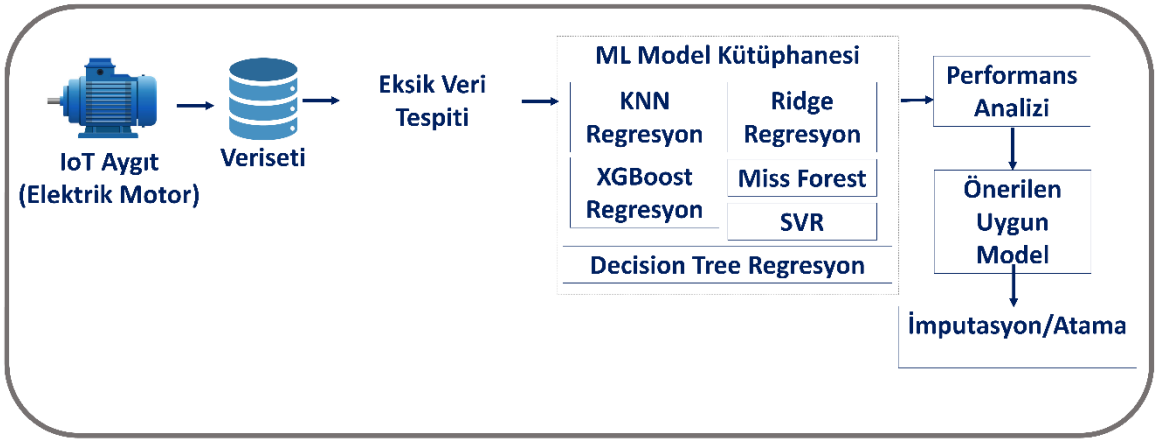
	A	B	C	D	E	F	G	H
1	X_Axis,Y_Axis,Z_Axis,R_Phase,S_Phase,T_Phase							
2	NaN,-0.00036736,-0.00036232,2.67151203,2.34094198,2.53110934							
3	-0.00091569,-0.00041615,-0.00028724,2.6619472,2.33446189,2.54144156							
4	NaN,NaN,-0.00059509,NaN,2.32398236,2.55212832							
5	-0.00120862,NaN,-0.00085789,NaN,2.32550114,2.56235924							
6	NaN,-0.0008253,-0.00077154,2.64372848,NaN,NaN							
7	-0.00147527,-0.00091539,NaN,2.64084384,NaN,2.57324859							
8	-0.00113727,NaN,-0.00049372,NaN,NaN,2.58930406							
9	-0.00073917,-0.00080278,-0.00052,2.62647129,2.32155233,NaN							
10	NaN,-0.00058882,NaN,NaN,2.31785665,2.61376711							
11	-0.00072415,-0.00045369,-0.00039236,2.60946715,NaN,2.62166822							
12	NaN,-0.0004612,-0.00056881,2.60268573,NaN,NaN							
13	-0.00114853,NaN,-0.00027598,2.59129903,NaN,2.6318485							
14	-0.00109595,-0.00063386,-0.00042239,2.58699738,2.31861604,2.635951							
15	NaN,-0.00075774,-0.00062888,2.57677465,2.32236234,NaN							

ratio40-random

Şekil 3.13. Rastgele konumda ve %40 oranında eksik veri içeren veri seti

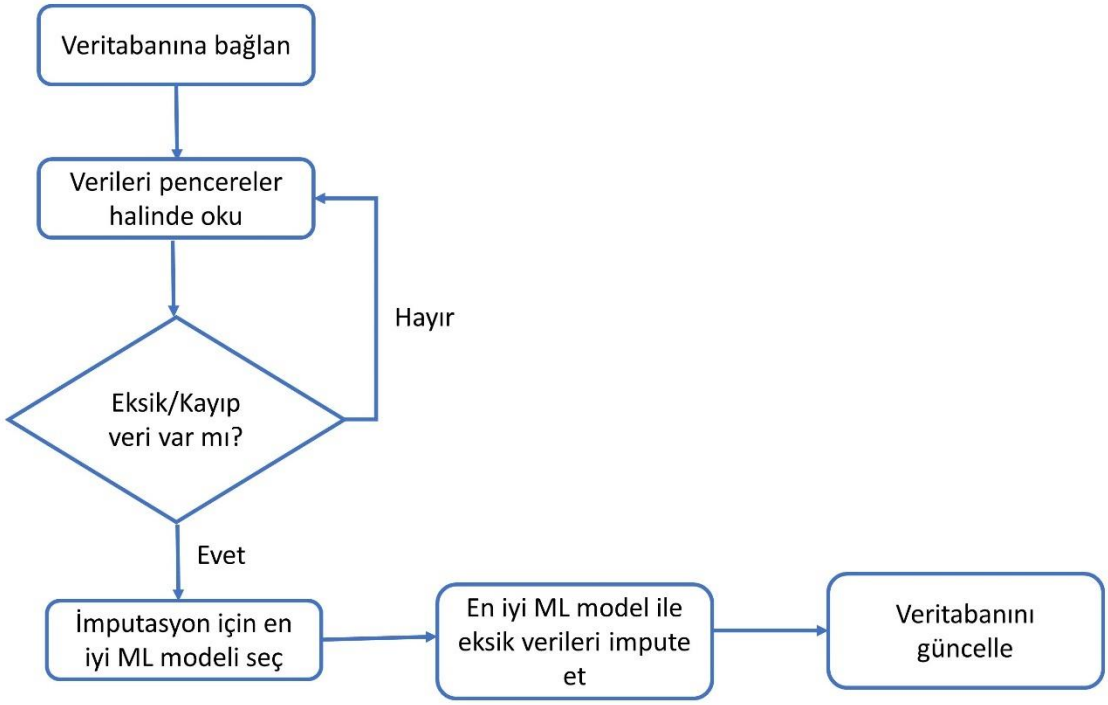
Tasarlanan IoT sistem platformundan toplanan ve eksik veriler oluşturularak hazırlanan veri setlerinin, seçilen altı regresyon-tabanlı ML algoritma arasından en iyi performans gösteren algoritma ile impute edilerek eksiksiz hale getirilmesi hedeflenmiştir. Bu hedefe ilişkin izlenen süreçler, bir akış şeması halinde Şekil 3.14'te yer almaktadır. Seçilen ML algoritmaları ise şu şekilde sıralanmaktadır:

1. Support Vector Regresyon (SVR),
2. Decision Tree Regresyon (DTR),
3. Ridge Regresyon,
4. k-Nearest Neighbors Regresyon (KNN),
5. Miss Forest (MF),
6. XGBoost Regresyon (XGB).



Şekil 3.14. Ön hazırlık çalışmasına ilişkin akış şeması

Tez çalışmasının ilerleyen aşamasında, ön çalışmada söz edilen eksik veri imputasyon süreçlerinin manuel olarak değil de otomatik olarak ilerlemesine ve kullanıcı müdahalesine ihtiyaç duyulmadan gerçekleşmesine ilişkin çalışmalar yapılmıştır. Yapılan literatür araştırmaları sonucunda Apache Airflow (2022a) platformu üzerinde, belirli periyotlarda ve koşullarda otomatik olarak harekete geçen, dinamik bir ML veri hattı (pipeline) tasarlanmıştır. Apache Airflow platformuna yönelik detaylar Bölüm 3.2.3'te yer almaktadır. Ek olarak, eksik veri atamasında en iyi performansı gösteren algoritma ve hiperparametre kombinasyonunun bulunmasının da manuel yoldan değil otomatik olarak yapılmasının daha verimli sonuçlar doğuracağı, zaman ve maliyet yönüyle tasarruf sağlayacağı düşünülmüştür. Bu sebeple, AutoML teknolojileri ML veri hattına dahil edilmiştir. Çalışmanın temel yöntemine ilişkin akış şemasına Şekil 3.15'te yer verilmektedir.



Şekil 3.15. Çalışmanın geliştirilmiş akış şeması

Çalışmada yer verilen ML yöntemlerine, kullanılan state-of-the-art teknolojilere ve tercih edilen açık kaynak kodlu platformlara ait detaylı bilgiler aşağıdaki alt bölümlerde verilmektedir.

3.2.1. Makine öğrenmesi yöntemleri

1959 yılında Arthur Samuel tarafından ortaya konan makine öğrenmesi terimi, bilgisayar sistemlerinin açıkça programlanmadan, varolan veriden bir öğrenim sağlayarak ileriye dönük tahminler ve bilinmeyene dair çıkarımlar yapabilme kabiliyeti olarak tanımlanabilmektedir. Bu öğrenimi sağlamak için matematiksel ve istatistiksel hesaplamalar içeren çeşitli öğrenme algoritmaları geliştirilmiştir. Bu algoritmalar görüntü işleme, tahmine dayalı analitik ve veri madenciliği gibi çeşitli alanlarda kullanılmaktadır (Mahesh, 2020) ve sınıflandırma, olağandışı veri noktalarını bulma, eksik değerleri tahmin etme ve benzerlikleri tespit etme gibi birçok görevi bir insana benzer şekilde yerine getirmektedir. Yapay zekanın bir alt dalı olan makine öğrenimi, hastalıklara tanı koyma ve teşhis etme, plaka tanıma, yüz

tanımlama, e-posta spam filtreleme, otonom araç, sesli asistan ve optik karakter tanıma (OCR) gibi birçok örnekle günlük yaşantımızda yerini almaktadır.

Makine öğreniminde temel süreç eğitim verilerinin algoritmaya verilmesi ile başlamaktadır. Algoritma çeşitli matematiksel hesaplamalarla belirli örüntüleri (pattern) yakalayarak veriyi öğrenmeye çalışmaktadır. Öğrenimini tamamladığı düşünülen ML algoritması daha önce hiç karşılaşmadığı test verileri ile test edilmektedir. Böylece algoritmanın doğru çalışıp çalışmadığı ve istenilen seviyede başarıya ulaşıp ulaşmadığı kontrol edilmektedir. Algoritma, yüksek doğruluk oranını yakalayana kadar veri artırımı ile yeniden eğitilmektedir.

Makine öğrenimi, öğrenme türlerine göre dört ayrı grupta incelenmektedir:

1. Denetimli (Supervised) Öğrenme
2. Denetimsiz (Unsupervised) Öğrenme
3. Yarı Denetimli (Semi-Supervised) Öğrenme
4. Pekiştirmeli (Reinforcement) Öğrenme

Denetimli (Gözetimli) öğrenme, algoritmaya eğitim verisi olarak verilen etiketli (labelled) girdiler ve çıktılar arasındaki eşlemeyi sağlayan ve aralarındaki ilişkileri ve bağımlılıkları modelleyen fonksiyonu üreten bir makine öğrenmesi tekniği olarak tanımlanabilmektedir. Bu teknikte, geliştirici etiketlenmiş girdiler ve bu girdilerin çıktılarını eğitim (train) kümesi olarak algoritmaya verir. Bu algoritmalar, eğitim veri setinden bir örüntü (pattern) yakalar, öğrenir ve bunları tahmin veya sınıflandırma için test veri setine uygular (Mahesh, 2020). Bu öğrenme tekniğinin denetimli (gözetimli) olarak nitelendirilmesinin sebebi, eğitim veri kümesi üzerinde bir öğrenme sürecinin gerçekleşmesi ve bu öğrenme sürecinin test veri kümesi ile denetlenmesidir. Denetimli öğrenme, sınıflandırma ve regresyon olarak iki ayrı başlıkta ele alınmaktadır. Bu tez çalışmasında, eksik verilerin sayısal (numerik) bir değer olarak tahmin edilmesine yönelik regresyon-tabanlı denetimli makine öğrenmesi algoritmalarına yer verilmiştir.

Regresyon, bir bağımlı değişken ile diğer bağımsız değişkenler arasındaki ilişkiyi belirlemeye çalışan ve bu ilişkiyi nicel değişkenlerin tahmin ve kestirimi için kullanan denetimli bir öğrenmedir. Regresyon-tabanlı algoritmaların başında Doğrusal (Linear) Regresyon, Destek Vektör Regresyonu (SVR), Karar Ağacı Regresyonu (DTR), K-En Yakın Komşu Regresyonu (KNN), Polinom Regresyon, Ridge Regresyon ve Rastgele Orman (Random Forest) Regresyonu gelmektedir.

Doğrusal Regresyon, *X* girdi değişkeni ile *Y* çıktı değişkeni arasındaki doğrusal ilişkinin bir doğru denklemi yardımıyla tanımlanmasını ve bir değişkenin bilindiği durumlarda diğer değişken değerinin tahmin edilebilmesini sağlayan bir algoritmadır. SVR, yaygın olarak kullanılan bir denetimli makine öğrenimi tekniğidir. Bu algoritma, bir tolerans (margin) aralığında en küçük hata ile maksimum noktayı alabilen bir regresyon fonksiyonunu doğru bir şekilde oluşturmayı amaçlamaktadır. DTR, IF-THEN koşullarıyla hedef değişkenin sayısal sonucunu tahmin eden ağaç tabanlı bir ML modeldir. Ridge Regresyon, minimum varyans ile çoklu regresyon modellerinin katsayılarını tahmin etme yöntemidir. K-En Yakın Komşu Regresyonu, eksik verilerin k-en yakın komşulardan alınan ortalama değerle değiştirilmesini sağlayan basit bir mesafe tabanlı yöntemdir. MissForest, Random Forest Regresyon algoritmasını kullanarak eksik verileri tamamlayan bir atama yöntemi olarak ifade edilebilmektedir. Bu algoritma, her değişken için yinelemeli olarak rastgele bir orman modeli oluşturur ve eksik değeri tahmin eder. XGBoost (eXtreme Gradient Boost) Regresyon, gradyan artırma algoritmasının optimize edilmiş yüksek performanslı bir sürümüdür. Sağladığı daha yüksek doğruluk ve hız ile mevcut gradyan artırma tekniklerine kıyasla oldukça popüler olduğu söylenebilmektedir.

Denetimsiz (Unsupervised) öğrenme, yalnızca etiketsiz (unlabelled) girdi değişkenleri üzerinden bilinmeyen bir yapıyı veya dağılımı modellemeyi sağlayan bir ML öğrenme türüdür. Bu öğrenme modelinde, girdi verisinin sınıfı belirsizdir ve modelin denetlenmesine ihtiyaç duyulmamaktadır. K-Means gibi kümeleme algoritmaları ve Temel Bileşen Analizi (Principal Component Analysis - PCA) gibi boyut indirgeme algoritmaları denetimsiz makine öğrenmesine örnek olarak verilebilmektedir. Bu öğrenme türü, genel olarak anomalilerin tespitinde ve düzenlerin belirlenmesinde kullanılmaktadır. Denetimsiz

öğrenme etiketli bir eğitim veri setine ihtiyaç duymadığından denetimli öğrenmeye göre çok daha az zaman ve efor gerektirse de denetimli algoritmalar doğru ve yeterli bir eğitim veri seti ile çok daha başarılı sonuçlar üretebilmektedir.

Yarı denetimli (Semi-Supervised) öğrenme, hem denetimli hem de denetimsiz öğrenmenin bir arada kullanıldığı karma bir öğrenme türüdür. Küçük miktardaki etiketli veri ile büyük miktardaki etiketsiz verinin birlikte kullanılmasıyla öğrenmede önemli ölçüde iyileşme sağlandığı bilinmektedir (Van Engelen ve Hoos, 2020). Büyük miktarda etiketli veriye ihtiyaç duymaması ile öne çıkan bu öğrenme türü, genellikle web sayfası ve genetik sıralamada kullanılmaktadır.

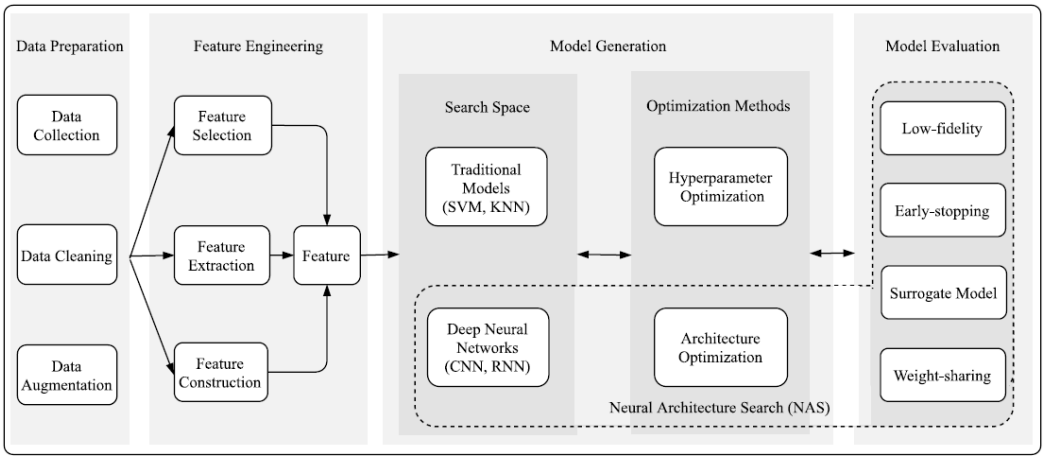
Bir ödül (reward) ve ceza (penalty) sistemi üzerine kurulu olan pekiştirmeli (reinforcement) öğrenme, deneme-yanılma yöntemiyle maksimum ödüle ulaşmayı hedefleyen bir öğrenme çeşidi olarak tanımlanabilmektedir. Bu öğrenmede temel hedef, en az ceza ile ödülleri maksimuma çıkaracak davranışları öğrenerek ajanın (agent) sıradaki en doğru eylemi belirleyebilmesidir. Genellikle robotik alanında, tekrarlayan sinir ağlarında (Recurrent Neural Network - RNN) ve modern video oyunlarında sıklıkla kullanılmaktadır.

3.2.2. Otomatik Makine Öğrenimi yaklaşımı

Son yıllarda giderek popülerleşmeye başlayan derin öğrenme ve makine öğrenmesi yöntemleri nesne algılama, görüntü sınıflandırma ve dil modelleme gibi birçok alanda zorlu yapay zeka görevlerini çözmek üzere kullanılmaktadır (He vd., 2021). Kullanım alanlarının yaygınlaşması ve teknolojinin gelişmesi, giderek karmaşıklaşan derin sinir ağı modellerini ve ML algoritmalarını da beraberinde getirmektedir. Bu durum, en iyi performans gösteren modelleri ve hiperparametre kombinasyonlarını deneme-yanılma yoluyla keşfedebilmek için uzmanların dâhi ciddi zamana ve kaynağa ihtiyaç duyması anlamına gelmektedir. Öğrenme oranı (learning rate), parça boyutu (batch size), optimizasyon yöntemi (optimizer), aktivasyon fonksiyonu ve gizli katman (hidden layer) sayısı gibi hiperparametrelerin eğitimden önce optimize edilmesi, model yapısının doğru oluşturulmasında, eğitim süresi ve

hızının belirlenmesinde, aşırı öğrenme (overfitting) ve eksik öğrenme (underfitting) problemlerinin yaşanmasında kritik öneme sahiptir. Yüksek doğruluk oranına sahip optimal modeller tasarlayabilmek için hiperparametrelerin optimize edilmesi gerekmektedir. Hiperparametre optimizasyonu oldukça zaman alıcı bir süreçtir ve bu durum üretim ssitemlerinde pratik değildir. Tam bu noktada, Otomatik Makine Öğrenimi (AutoML) bu zahmetli geliştirme maliyetlerini azaltmak için makine öğreniminin tüm veri hattını (pipeline) otomatikleştirmeye yönelik yeni bir fikir olarak ortaya çıkmıştır (He vd., 2021).

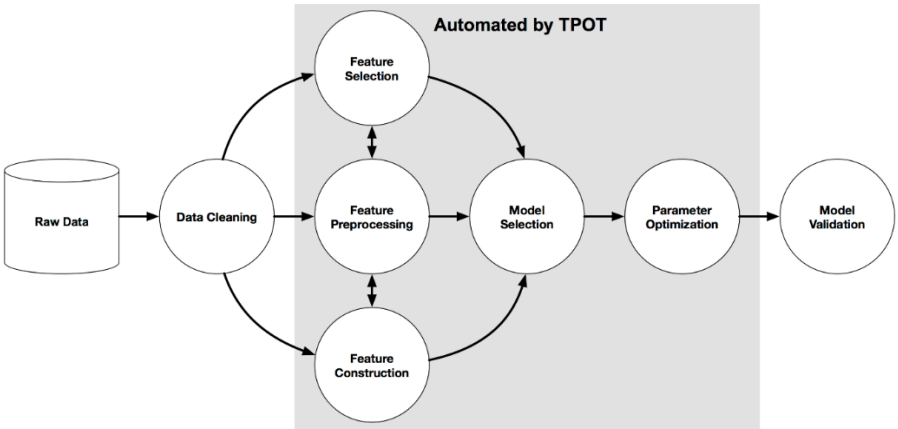
Günümüzde makine öğreniminin hemen hemen tüm sektörlerde yerini alması, istenilen düzeyde bilgi ve beceriye sahip bir veri bilimi ekibini işe alacak kaynaklara sahip olmayan şirketler için erişilebilir tekniklere olan ihtiyacı giderek arttırmaktadır (Balaji ve Allen, 2018). Erişilebilir makine öğrenimi tekniklerine yönelik talebe yanıt olarak, verilerden mümkün olan en kısa sürede ve en az çabayla değer elde etmek için çeşitli açık kaynaklı AutoML çerçeveleri oluşturulmaktadır. Bu AutoML çerçeveleri, bir veri bilimcinin sorumluluğundaki makine öğrenimi veri hattını oluşturma ve uygulama, kombine algoritma seçimi ve hiper parametre optimizasyonu (Combined Algorithm Selection and Hyper-parameter Optimization - CASH) gibi birçok zaman alıcı görevi otomatikleştirerek yerine getirmektedir. Şekil 3.16, veri hazırlama, özellik mühendisliği, model oluşturma ve model değerlendirme süreçlerini kapsayan uçtan uca bir AutoML veri hattına genel bir bakış sunmaktadır.



Şekil 3.16. AutoML veri hattına genel bir bakış (He vd., 2021)

Yıllar içinde gelişen AutoML yapıları, genel itibariyle veri ön işleme adımından en iyi modelin uygulanmasına kadarki standart süreçleri kapsamaktadır. Evrensel olarak en iyi AutoML yaklaşımı bulunmamakla birlikte, çeşitli yaklaşımları baz alan MLBox, H2O AutoML, TPOT, Auto-WEKA, Auto-Sklearn ve Cloud AutoML gibi birçok AutoML aracı mevcuttur. Bu AutoML araçları nispeten standart süreçler ve teknikler uygulasa da bu tekniklerin uygulanmasını ve değerlendirilmesini otomatikleştirmek için kullanılan yöntemler büyük ölçüde farklılık göstermektedir (Balaji ve Allen, 2018). Bu sebeple mevcut literatürdeki AutoML araçların performansını gerçek veri setleri kullanarak kıyaslayan birçok benchmarking çalışması bulunmaktadır (Balaji ve Allen, 2018; Gijsbers vd., 2019; Erickson vd., 2020; Zöller ve Huber, 2021; Ferreira vd., 2021; Gijsbers vd., 2022; A Romeo vd., 2022).

Ağaç Tabanlı Ardışık Düzen Optimizasyon Aracı (TPOT), uçtan uca ML veri hatları oluşturan, genetik programlama tabanlı bir optimize edici olarak tanımlanabilmektedir (Balaji ve Allen, 2018). Scikit-learn kütüphanesinin sınıflandırıcı ve regresyon opsiyonu ile ayrı bir uzantısı olan TPOT, bir AutoML ardışık düzen aramasında algoritma ve hiperparametre kombinasyonu ihtimalleri arasından veri setine en uygun ardışık düzeni seçebilmektedir. TPOT, süreçleri temsil etmek için ağaç tabanlı bir yapı kullanmaktadır. Şekil 3.17'de TPOT tarafından otomatikleştirilen ML veri hattı yer almaktadır.



Şekil 3.17. TPOT tarafından otomatikleştirilen ML süreçleri (Olson vd., 2016)

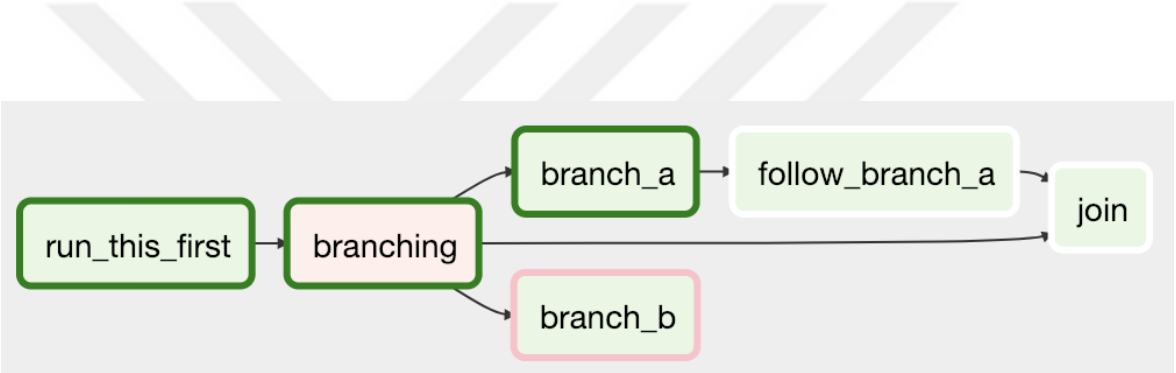
Literatürde TPOT'un 150 adet denetimli sınıflandırma veri seti üzerinde diğer ML yaklaşımlar ile karşılaştırıldığı ve bunların 21'inde diğer temel makine öğrenimi analizlerinden önemli ölçüde daha iyi performans sergilediği bilinmektedir (Olson ve Moore, 2016). En iyi ML veri hattı keşfinin herhangi bir alan bilgisi veya insan girdisi olmadan gerçekleşmesine dikkat çekilerek genetik programlama tabanlı TPOT'un AutoML alanında ciddi bir umut vaat ettiği söylenebilmektedir. Bu tez çalışmasında, AutoML araçları arasından performans kıyaslamalarında öne çıkması ve açık kaynaklı olması sebebiyle Ardışık Düzen Optimizasyon Aracı TPOT tercih edilmiştir.

3.2.3. Apache Airflow platformu

Son yıllarda veri analitiği ve makine öğrenimine odaklanan yeni İş Akışı Yönetim Sistemi (Workflow Management System - WMS) uygulamaları ortaya çıkmış ve bu durum iş akışı yönetim modellerindeki gelişimi hızlandırmıştır (Mitchell vd., 2019). Ölçeklenebilir ve çeşitli teknolojilerle gelişmiş iş birliği içinde olan bu platformlar arasında en popülerleri Apache Airflow, Makeflow, Luigi, Kubeflow ve MLflow olarak sıralanabilmektedir. Bu tez çalışmasında iş akış yönetim platformu olarak, modülerliği, diğer platformların yanı sıra Amazon S3, Google Cloud veya HDFS gibi yaygın bulut teknolojilerine ve veritabanı sistemlerine önceden oluşturulmuş birçok arabirim (operatör) sağlaması ve iş akışının bir dalının yalnızca belirli bir koşul yerine getirildiğinde yürütüldüğü koşullu yürütme desteği sebebiyle Apache Airflow tercih edilmiştir.

Apache Airflow, Airbnb tarafından dahili iş akışlarını verimli bir şekilde yönetmek için geliştirilen ve 2016 yılında piyasaya sürülen açık kaynaklı bir Apache Software Foundation projesidir (Mitchell vd., 2019). Temel olarak Airflow, birbirine bağlı veya birbirinden bağımsız birden fazla görevin yer alabileceği çeşitli işleri yürütmek, planlamak, dağıtmak ve izlemek amacıyla tasarlanan bir çerçevedir (Singh, 2019). Airflow, Python komut dosyaları aracılığıyla iş akışları oluşturmakta ve onları kolayca yönetebilmektedir. Python programlama dilini kullandığından, iş akışlarının daha kolay oluşturulması ve yönetilmesi için kütüphanelerin ve sınıfların kolayca içe aktarılabilmesi mümkün hale gelmektedir.

Airflow'da tüm görevler DAG (Directed Acyclic Graph) ismindeki iş akışlarında tanımlanmaktadır. DAG'lar mevcut görevlerin, bağımlılıkların ve ilişkilerin bir bütün olarak nasıl yürütülmesi gerektiğini söylemektedir. Görevlerin ve bağımlılıkların yürütülme sırası, iş akışı Airflow ile yürütülmeden önce, DAG yapısında açık bir şekilde tanımlanmalıdır. DAG ile ilgili tüm konfigürasyonlar, Python uzantılı bir DAG dosyasında yer almaktadır. Bu ".py dosyası", hata durumunda gönderilecek e-posta, başlangıç ve bitiş zamanı, yeniden deneme sayısı gibi tüm bağımlılıkları ve yapılandırma parametrelerini içermektedir (Singh, 2019). Şekil 3.18, Apache Airflow platformu üzerinde tanımlanmış örnek bir DAG yapısını temsil etmektedir.



Şekil 3.18. Örnek bir DAG yapısı (Apache Airflow, 2022b)

Bir DAG birden fazla görevden sorumlu olabilmekte ve bu görevler birbirinden tamamen farklı olabilmektedir. Apache Airflow, farklı görev türleri için bazıları çekirdekte yerleşik veya önceden kurulmuş sağlayıcılarla birlikte çok geniş bir operatör yelpazesine sahiptir (Apache Airflow, 2022c). Bash (Shell script), Python (Python script), Email, SimpleHttp, MySql, Postgres, MsSql, Oracle, Jdbc, Docker, Hive, S3FileTransform, Slack operatörleri popülerler arasında yer almaktadır. Airflow, temel üç ayrı servisten oluşmaktadır. Bu servisler şu şekilde sıralanabilmektedir:

1. Webserver: Web sunucusu, kullanıcıların DAG'ları ve bireysel görevleri görselleştirebileceği, takip edebileceği, yürütebileceği ve durdurabileceği zengin bir kullanıcı arayüzü sağlamaktadır.
2. Scheduler: DAG'lar ile açıkça tanımlanan görevlerin yürütülmesini zamansal olarak koordine etmektedir (çizelgelemektedir).

3. Worker: Zamanlayıcının sıraya koyduğu görevleri gerçekleştiren süreçtir.

Bu çalışmada, Apache Airflow İş Akışı Yönetim platformu üzerinde kullanılan TPOT AutoML aracı sayesinde eksik veri imputasyon süreci otomatize edilmiştir.

4. BULGULAR VE TARTIŞMA

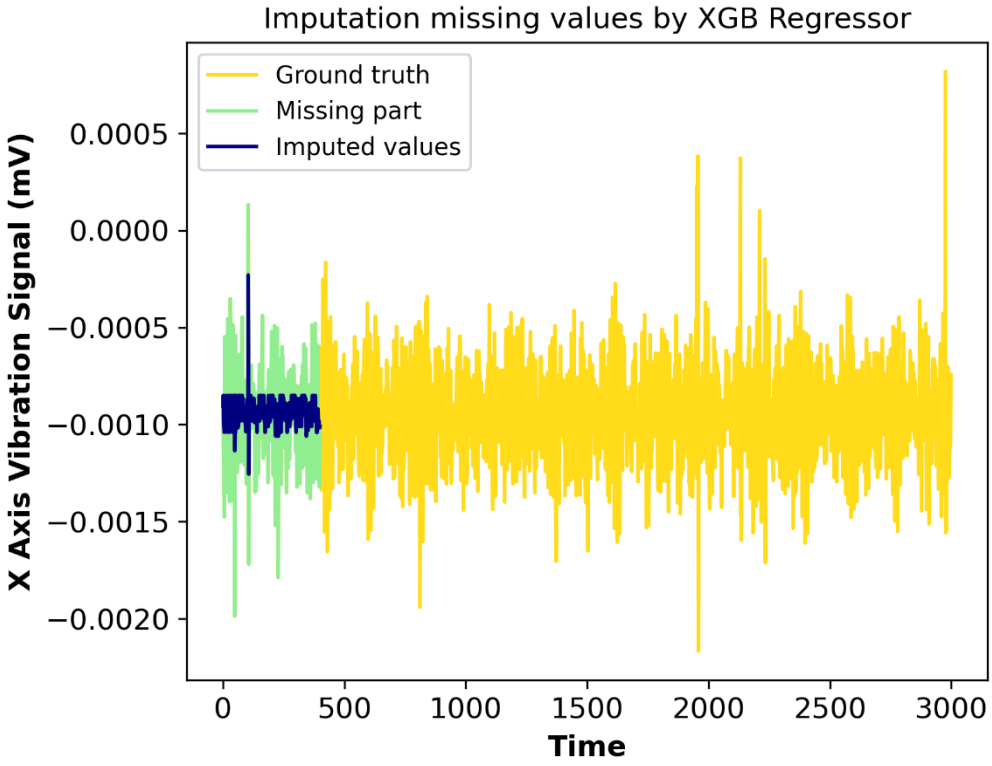
Kestirimci bakım sistemlerinde veri artırma yöntemlerinin geliştirilmesine yönelik olarak önerilen yöntemler farklı koşullarda çeşitli veri setlerine uygulanmıştır. Ön hazırlık çalışmasına ve otomatik eksik veri imputasyonu çalışmasına ilişkin bulgular bu bölümde detaylı olarak analiz edilmekte ve tartışılmaktadır.

4.1. Ön Hazırlık Çalışmasına İlişkin Bulgular

Ön hazırlık çalışmasında, Bölüm 3.1.2'de detaylı olarak anlatılan veri setleri üzerinde çeşitli performans analizleri gerçekleştirilmiştir. IoT sistem platformunun bir kullanım senaryosu olan elektrik motoruna ait titreşim ve akım verileri üzerinde farklı konum ve oranlarda eksik veriler oluşturularak seçilen ML modellerin bu eksik verileri tamamlama yeteneği test edilmiştir. Bu kapsamda, eksiksiz olan ham veri setindeki değerler (bilinen değer) ile ML modellerin atadıkları değerler (tahmin edilen değer) arasındaki korelasyona dikkat çekilmektedir. Değerlendirmeler ve karşılaştırmalar, Hata Kareleri Ortalamasının Karekökü (Root Mean Square Error - RMSE) ve Belirlilik Katsayısı (R squared – R^2) metriklerine dayalı nicel analiz ile yapılmaktadır.

ML modellerinin uygulanması ve eksik veri ataması için Python programlama dili ve ağırlıklı olarak Scikit-learn ve Missingpy kütüphaneleri kullanılmıştır. Eksik değerler içeren verisetleri, tam ve eksik parçalardan oluşmaktadır. Tam kısmın tamamı ML modellerinin eğitimi için kullanılmıştır. Eksik verilerin olduğu kısım, eğitilen ML modellerine test verisi olarak verilmiş ve ML modelleri tarafından ayrı ayrı impute edilmiştir. Her bir ML modelin farklı hiperparametre kombinasyonları denenerek en iyi model seçilmiştir. En uygun hiperparametrelere sahip ML modelleri, diğer bir tabirle en iyi modeller, daha kararlı sonuçlar elde etmek için on altı kez çalıştırılarak analiz edilmiş ve karşılaştırılmıştır. ML modellerin tahmin yeteneğini analiz etmek için birçok yöntem bulunmaktadır. İlk olarak, eksik veri tahmin doğruluğunu daha iyi görebilmek için atanan veriler ve gerçek veriler üst

üste çizilmektedir. Şekil 4.1, başlangıç durumunda %40 oranında eksik veri için XGBoost Regresyon algoritması tarafından impute edilen ve bilinen (gerçek) verilerin bir grafiğini göstermektedir. Şekil 4.2 ise rastgele durumunda %10 oranında eksik veri için XGBoost Regresyon algoritması tarafından impute edilen ve bilinen (gerçek) verilerin bir grafiğini göstermektedir. Gerçek veriler sarı, oluşturulan boşluklar yeşil ve ML modeli tarafından impute edilen/doldurulan veriler mavi ile temsil edilmektedir. Bu noktada, mavi kısmın yeşil kısım üzerindeki dağılımı doğru tahmin yeteneği açısından belirleyicidir (Kalay vd., 2022).



Şekil 4.1. Başlangıç %40 durumunda gerçek ve tahmin değerlerinin bir karşılaştırması

Şekil 4.2. Rastgele %10 durumunda gerçek ve tahmin değerlerinin bir karşılaştırması

Grafiklerde titreşim değerlerinin ani yükselişlerinin yakalandığı, tahmin edilen değerlerin genel itibariyle gerçek değerlere yakınlık gösterdiği görülmektedir. Nitel gözleme dayalı grafikler incelendikten sonra ham veri setinin ve ML algoritmalar tarafından ayrı ayrı tamamlanan veri setlerinin tanımlayıcı istatistikler tablosu arasında karşılaştırmalar ve çıkarımlar yapılmaktadır. Orijinal veri setinin istatistiksel analizi Çizelge 4.1'de yer alırken, başlangıç durumunda %40 oranında eksik veri içeren veri setinin Ridge Regresyon algoritması tarafından imputasyonu sonrası tanımlayıcı istatistikleri Çizelge 4.2'de yer almaktadır.

Çizelge 4.1. Orijinal veri setinin tanımlayıcı istatistikleri

İmputasyondan Önce					
	Count	***Mean***	***Std***	***Min***	***Max***
X axis	3000	-0.00094	0.00024	-0.00217	0.00082
Y axis	3000	-0.00062	0.00018	-0.00163	0.00136
Z axis	3000	-0.00042	0.00020	-0.00158	0.00291
R phase	3000	2.51638	0.14558	2.30425	2.73098
S phase	3000	2.52081	0.14476	2.31153	2.73117
T phase	3000	2.50414	0.14488	2.29063	2.71116

Çizelge 4.2. İmputasyon sonrası veri setinin tanımlayıcı istatistikleri

Ridge Regresyon ile İmputasyon Sonrası					
	Count	***Mean***	***Std***	***Min***	***Max***
X axis	3000	-0.00094	0.00023	-0.00217	0.00082
Y axis	3000	-0.00062	0.00017	-0.00163	0.00136
Z axis	3000	-0.00042	0.00019	-0.00158	0.00291
R phase	3000	2.51544	0.13576	2.30425	2.73098
S phase	3000	2.52196	0.13440	2.31153	2.73117
T phase	3000	2.50388	0.13503	2.29185	2.71116

Çizelge 4.1'deki veri karakteristiği Çizelge 4.2'deki ile karşılaştırıldığında, verilerin gerçek karakteristiğine uygun olarak başarılı bir şekilde impute edildiği sonucuna varılmaktadır. ML algoritma tarafından atanan verinin standart sapmasının daha düşük olduğu ve orijinal veri setinin karakteristiğinin korunduğu görülmektedir. Ek olarak, veri setlerindeki değişkenlerin imputasyon öncesi ve sonrası ortalama (Mean) ve standart sapma (Std) değerlerindeki sapma miktarları (%) her bir ML algoritma için incelenmiştir. Çizelge 4.3'te X axis değişkeninin ortalama değerinde, Çizelge 4.4'te ise standart sapma değerinde meydana gelen sapma miktarının bir karşılaştırma tablosu yer almaktadır.

Çizelge 4.3. Ortalama (Mean) değerinde meydana gelen sapma miktarının karşılaştırma tablosu

		Ridge Regresyon		KNN		SVR		DTR		MF		XGB	
Veri Seti	**Orijinal Mean**	Mean	Delta (%)	Mean	Delta (%)	Mean	Delta (%)	Mean	Delta (%)	Mean	Delta (%)	Mean	Delta (%)
Başlangıç %10 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,00093	0,958466454	-0,000937	0,212992545	-0,000938	0,106496273	-0,000939	0
Orta %10 eksik veri	-0,000939	-0,000939	0	-0,000939	0	-0,00093	0,958466454	-0,00095	1,171458999	-0,000937	0,212992545	-0,000939	0
Bitiş %10 eksik veri	-0,000939	-0,000936	0,319488818	-0,000939	0	-0,00093	0,958466454	-0,00094	0,106496273	-0,000939	0	-0,00094	0,106496273
Rastgele %10 eksik veri	-0,000939	-0,000938	0,106496273	-0,000933	0,638977636	-0,000903	3,833865815	-0,000938	0,106496273	-0,000939	0	-0,000939	0
Başlangıç %20 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,00092	2,02342918	-0,000939	0	-0,00094	0,106496273	-0,000939	0
Orta %20 eksik veri	-0,000939	-0,000939	0	-0,000939	0	-0,000921	1,916932907	-0,000937	0,212992545	-0,000941	0,212992545	-0,00094	0,106496273
Bitiş %20 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,000921	1,916932907	-0,000926	1,384451544	-0,000928	1,171458999	-0,00094	0,106496273
Rastgele %20 eksik veri	-0,000939	-0,000939	0	-0,000945	0,638977636	-0,000868	7,561235357	-0,00094	0,106496273	-0,00094	0,106496273	-0,000939	0
Başlangıç %30 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,000921	1,916932907	-0,000931	0,851970181	-0,000942	0,319488818	-0,00094	0,106496273
Orta %30 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,000912	2,875399361	-0,000935	0,425985091	-0,000943	0,425985091	-0,00094	0,106496273
Bitiş %30 eksik veri	-0,000939	-0,00094	0,106496273	-0,00094	0,106496273	-0,000913	2,768903088	-0,00095	1,171458999	-0,000936	0,319488818	-0,000941	0,212992545
Rastgele %30 eksik veri	-0,000939	-0,000937	0,212992545	-0,000942	0,319488818	-0,000883	5,963791267	-0,000936	0,319488818	-0,00094	0,106496273	-0,000935	0,425985091
Başlangıç %40 eksik veri	-0,000939	-0,000936	0,319488818	-0,000936	0,319488818	-0,000901	4,04685836	-0,000984	4,792332268	-0,000938	0,106496273	-0,000939	0
Orta %40 eksik veri	-0,000939	-0,000938	0,106496273	-0,000938	0,106496273	-0,000903	3,833865815	-0,000918	2,236421725	-0,000942	0,319488818	-0,00094	0,106496273
Bitiş %40 eksik veri	-0,000939	-0,00094	0,106496273	-0,00094	0,106496273	-0,000904	3,727369542	-0,00096	2,236421725	-0,000941	0,212992545	-0,000944	0,532481363
Rastgele %40 eksik veri	-0,000939	-0,000935	0,425985091	-0,000941	0,212992545	-0,000881	6,176783813	-0,000937	0,212992545	-0,00094	0,106496273	-0,000934	0,532481363
ORTALAMA			**0,14**		**0,19**		**3,21**		**0,97**		**0,24**		**0,15**

Çizelge 4.4. Standart Sapma (Std) değerinde meydana gelen sapma miktarının karşılaştırma tablosu

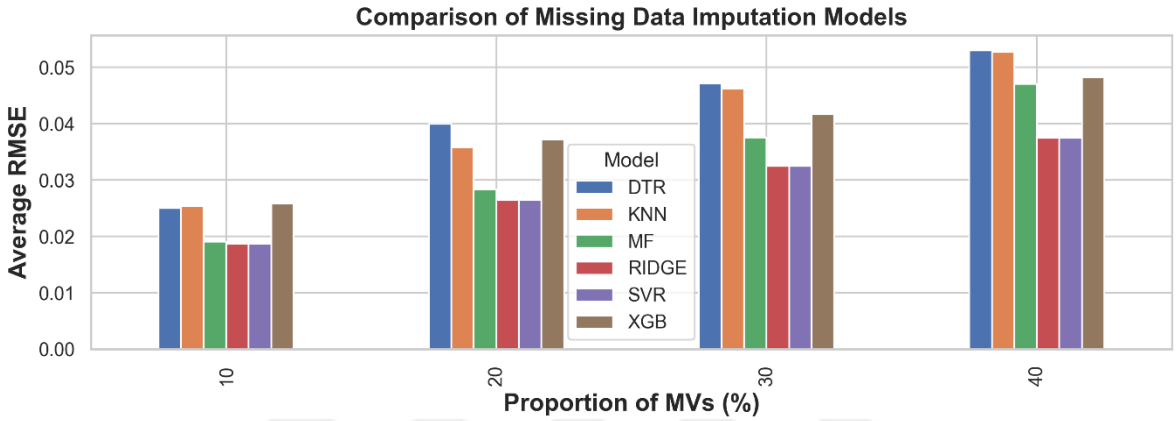
		Ridge Regresyon		KNN		SVR		DTR		MF		XGB	
Veri Seti	**Orijinal Std**	Std	Delta (%)	Std	Delta (%)	Std	Delta (%)	Std	Delta (%)	Std	Delta (%)	Std	Delta (%)
Başlangıç %10 eksik veri	0,000241	0,000236	2,074688797	0,000236	2,074688797	0,000241	0	0,000236	2,074688797	0,000236	2,074688797	0,000236	2,074688797
Orta %10 eksik veri	0,000241	0,000239	0,829875519	0,000239	0,829875519	0,000244	1,244813278	0,000247	2,489626556	0,000239	0,829875519	0,000239	0,829875519
Bitiş %10 eksik veri	0,000241	0,000225	6,639004149	0,000239	0,829875519	0,000244	1,244813278	0,00024	0,414937759	0,000239	0,829875519	0,000239	0,829875519
Rastgele %10 eksik veri	0,000241	0,00023	4,564315353	0,000232	3,734439834	0,000253	4,979253112	0,00024	0,414937759	0,000232	3,734439834	0,00023	4,564315353
Başlangıç %20 eksik veri	0,000241	0,000233	3,319502075	0,000233	3,319502075	0,000242	0,414937759	0,000233	3,319502075	0,000233	3,319502075	0,000233	3,319502075
Orta %20 eksik veri	0,000241	0,000236	2,074688797	0,000236	2,074688797	0,000245	1,659751037	0,000236	2,074688797	0,000236	2,074688797	0,000236	2,074688797
Bitiş %20 eksik veri	0,000241	0,000233	3,319502075	0,000233	3,319502075	0,000242	0,414937759	0,000233	3,319502075	0,000236	2,074688797	0,000233	3,319502075
Rastgele %20 eksik veri	0,000241	0,000216	10,37344398	0,000225	6,639004149	0,000259	7,468879668	0,000241	0	0,000223	7,468879668	0,000219	9,128630705
Başlangıç %30 eksik veri	0,000241	0,000229	4,979253112	0,000229	4,979253112	0,000242	0,414937759	0,00023	4,564315353	0,000229	4,979253112	0,000229	4,979253112
Orta %30 eksik veri	0,000241	0,000233	3,319502075	0,000233	3,319502075	0,000246	2,074688797	0,000233	3,319502075	0,000233	3,319502075	0,000233	3,319502075
Bitiş %30 eksik veri	0,000241	0,000227	5,809128631	0,000227	5,809128631	0,000241	0	0,000229	4,979253112	0,000227	5,809128631	0,000227	5,809128631
Rastgele %30 eksik veri	0,000241	0,000202	16,18257261	0,000212	12,03319502	0,000259	7,468879668	0,00024	0,414937759	0,000216	10,37344398	0,000209	13,2780083
Başlangıç %40 eksik veri	0,000241	0,000225	6,639004149	0,000225	6,639004149	0,000242	0,414937759	0,000256	6,22406639	0,000225	6,639004149	0,000225	6,639004149
Orta %40 eksik veri	0,000241	0,000229	4,979253112	0,000229	4,979253112	0,000246	2,074688797	0,000234	2,904564315	0,000229	4,979253112	0,000229	4,979253112
Bitiş %40 eksik veri	0,000241	0,000221	8,298755187	0,000221	8,298755187	0,000239	0,829875519	0,000227	5,809128631	0,000221	8,298755187	0,000221	8,298755187
Rastgele %40 eksik veri	0,000241	0,000185	23,23651452	0,000197	18,25726141	0,000196	18,67219917	0,000238	1,244813278	0,000211	12,44813278	0,000196	18,67219917
ORTALAMA			**6,66**		**5,45**		**3,09**		**2,72**		**4,95**		**5,76**

Çizelge 4.3'te yer alan ortalama değerleri karşılaştırma tablosunda Ridge Regresyon algoritmasının %0,14 değeri ile en düşük ortalama sapma miktarına (%) sahip olduğu görülmektedir. Bu durumda Ridge Regresyon algoritmasının orijinal veri setindeki verilerin ortalamasına en yakın tahminlerde bulunduğu ve atadığı verilerle veri setinin ortalama (Mean) istatistiksel karakterini bozmadığı sonucuna varılmaktadır. XGB algoritması %0,15 sapma oranıyla Ridge Regresyon algoritmasına benzer bir performans sergilemektedir. SVR algoritması ise %3,21 oranında sapma göstererek diğer algoritmalara kıyasla veri setinin ortalama değerinde daha büyük farklılıklara neden olmaktadır.

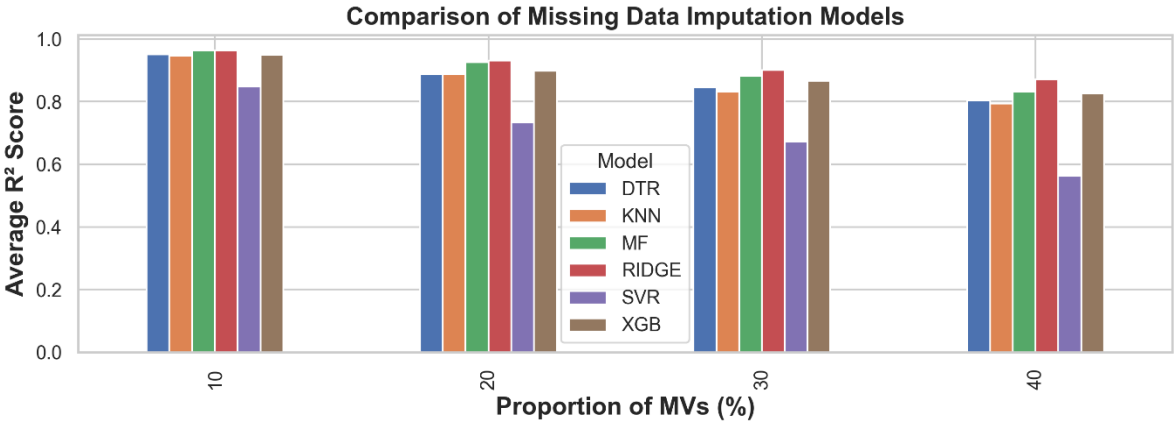
Çizelge 4.4'te yer alan standart sapma değerinde meydana gelen değişiklikler incelendiğinde, DTR algoritmasının %2,72 oranıyla en düşük sapma miktarına sahip olduğu ve diğer atama yöntemlerinden daha iyi bir performans sergilediği görülmektedir. %3,09 oranıyla SVR algoritması standart sapma farkı açısından DTR algoritmasına benzer performans göstermektedir. Ridge Regresyon algoritması en düşük ortalama sapmasına sahip olmasına rağmen %6,66 oranıyla en yüksek standart sapma farkına sahiptir. Bu durum standart sapmanın korunması gereken durumlarda daha iyi sonuçlar elde edebilmek için DTR ve SVR algoritmalarının tercih edilebileceği sonucunu ortaya koymaktadır.

RMSE ve R^2 metrikleri, ML modellerin atama/imputasyon performanslarını değerlendirmek için kullanılmaktadır. RMSE, zaman serisi verileri üzerindeki tahmin performanslarını değerlendirmek için yaygın olarak kullanılan bir doğruluk ölçüsüdür (Yozgatligil vd., 2013) ve literatürde model performans kıyaslama çalışmalarında sıklıkla görülmektedir. RMSE, tahmin edilen değerler ve gerçek değerler arasındaki hata oranını temsil etmektedir (Okafor ve Delaney, 2021). Daha düşük bir RMSE değerinin daha doğru bir değerlendirmeyi temsil ettiği bilinmektedir (Vastrad vd., 2013). Belirlilik katsayısı olarak da adlandırılan R^2, regresyon modeli tarafından açıklanan varyans oranı olarak tanımlanmaktadır (Alamoodi vd., 2021). Bağımlı değişkendeki değişkenliğin bağımsız değişkenler tarafından açıklanma oranını ifade etmektedir. Atama performansının değerlendirilmesinde sıklıkla kullanılan, tahmin edilen değerler ve gerçek değerler arasındaki korelasyonu ölçen bir metriktir. 1'e yaklaşan R^2 değeri mükemmel bir uyum anlamına gelmektedir (Alamoodi vd., 2021). Bu çalışmada, her ML algoritması on altı kez

çalıştırılmakta ve ML modellerinin ortalama performans ölçüleri grafiklere yansıtılmaktadır. Altı ML modelin ortalama RMSE ve R^2 değerleri dört duruma (rastgele, başlangıç, orta, bitiş) göre analiz edilmektedir. Başlangıç durumundaki tüm eksiklik oranları (%10, %20, %30, %40) için ortalama RMSE ve R^2 değerlerinin karşılaştırılması Şekil 4.3 ve Şekil 4.4'te gösterilmektedir.



Şekil 4.3. Başlangıç durumundaki eksik veriler için ML modellerin ortalama RMSE değerleri



Şekil 4.4. Başlangıç durumundaki eksik veriler için ML modellerin ortalama R^2 değerleri

Şekil 4.3'te, her modelde eksik veri oranı arttıkça, RMSE değeri de dört durum için (rastgele, başlangıç, orta, bitiş) artmaktadır. Ridge Regresyon modeli, dört durum ve tüm eksiklik oranları için en düşük RMSE değerini vermektedir. DTR, KNN ve XGB modelleri, diğer modellere kıyasla tüm durumlar için önemli ölçüde daha yüksek bir RMSE değerine

sahiptir. Ridge Regresyon ve SVR modelleri başlangıç, orta ve bitiş durumları için aynı RMSE değerini gösterirken, SVR ve Ridge Regresyonunun atama/imputasyon performansı rastgele durumda farklılık göstermektedir. Ridge Regresyon modelinin RMSE değeri, diğer modellere göre daha düşük kalmaktadır.

Şekil 4.4'te, Ridge Regresyon modeli tüm durumlar için diğer ML modellerinden daha iyi bir performans göstermektedir. Ridge Regresyon ile oluşturulan modelin bu veri setine daha uyumlu/fit olduğu görülmektedir. SVR modeli, diğer modellere kıyasla tüm eksik veri oranları için daha düşük bir performans sergilemektedir. Özellikle kayıp veri oranı %10'dan fazla olduğunda diğer ML modellerinden önemli ölçüde daha kötü olduğu söylenebilmektedir.

4.2. Otomatik Eksik Veri İmputasyonu Çalışmasına İlişkin Bulgular

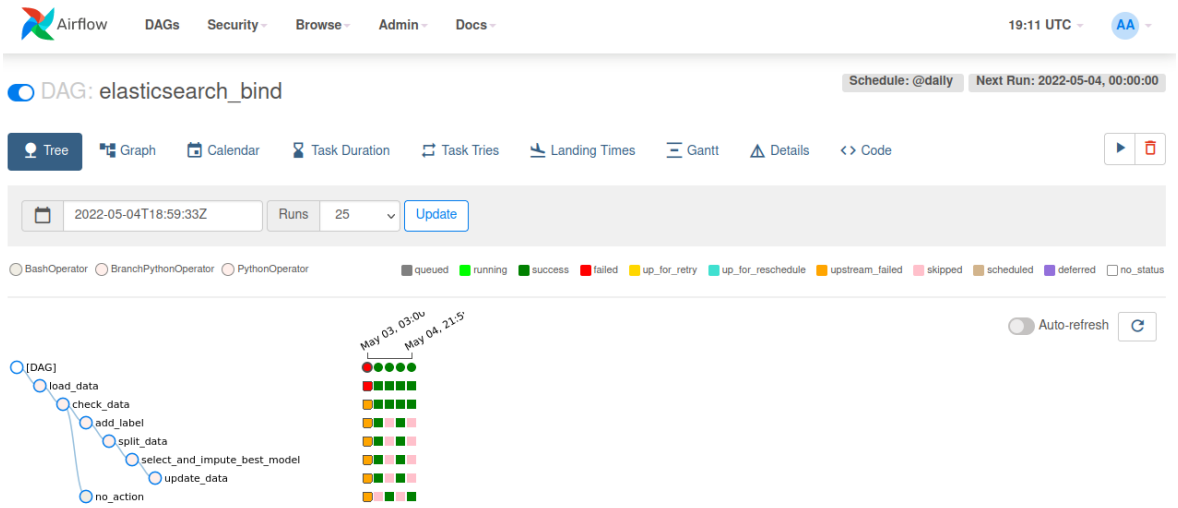
IoT sensörleri içeren ekipmanlardan çok büyük miktarda veri toplanabilmekte ve bu veriler arıza/anomali tespiti ve bakım planlama süreçlerinde kullanılabilmektedir. Sensörlerden toplanan zaman serisi verilerinin sürekliliği, niceliği ve niteliği veri analizinde kritik bir rol oynamaktadır. Ağ arızaları, sensör arızaları, senkronizasyon sorunları gibi çeşitli nedenlerle yazılamayan bazı sensör verileri, Grafana (2022) ve Kibana (2022) gibi veri analitiği araçlarının gerçek zamanlı olarak izlenen gösterge panellerinde gözle görülür boşluklar oluşturmaktadır. Eksik veriler içeren veri kümeleri, veri analistleri ve karar vericiler için hatalı çıktılar üretebilmekte, taraflı tahminler yapabilmekte ve sonuçların tutarlılığını azaltabilmektedir. Başarılı ve güvenilir bir analiz çalışması için, eksik sensör verileri ele alınmalı ve veri setleri eksiksiz hale getirilmelidir.

Bu tez çalışmasında kestirimci bakım sistemleri için tasarlanan veri artırma modülü, IoT sensörleriyle donatılmış ekipmanlardan toplanan verilerin tutulduğu veritabanı ile etkileşime girerek eksik verileri otomatik olarak orijinal değerine en yakın bir şekilde atamayı amaçlamaktadır. Bu kapsamda, iş akışı süreçlerini otomatikleştirmek için açık kaynaklı bir iş akışı yönetim platformu olan Apache Airflow ve eksik veri ataması için en

uygun ve başarılı algoritma ve hiperparametre kombinasyonunu belirlemek için Ağaç Tabanlı Ardışık Düzen Optimizasyon Aracı (TPOT) (Le vd., 2020) tercih edilmiştir.

Bir İş Akışı Yönetim Sistemi (Workflow Management System - WMS) olan Apache Airflow, Python programlama dilini kullanarak iş akışları oluşturulabilmesini ve bu akışların kolayca planlanabilir ve izlenebilir hale getirilmesini mümkün kılmaktadır. DAG (Directed Acyclic Graph) ile temsil edilen iş akışları her saat, her gün, her ay, her yıl gibi belirli zaman aralıklarında ve oluşturulan çeşitli IF-THEN koşullarında tetiklenerek otomatik olarak çalışmaktadır. Bu platform, süreçlerin birbirlerine bağımlılıklarının, tetikleme mekanizmalarının ve görevlerin başarılı/başarısız sonuçlarının görsellenmesine ve iş akışının daha kolay takip edilmesine olanak sağlar.

Bu çalışmada, Şekil 3.14'teki akış diyagramına benzer şekilde, kestirimci bakım sisteminin ihtiyaçları doğrultusunda TPOT AutoML yaklaşımı ile belirli periyotlarda eksik verileri impute eden DAG yapısı oluşturulmuştur. Şekil 4.5'te, Apache Airflow'da otomatik olarak gerçekleşen eksik veri atama süreçlerinin kolayca takip edildiği ve yönetildiği bir arayüz yer almaktadır.

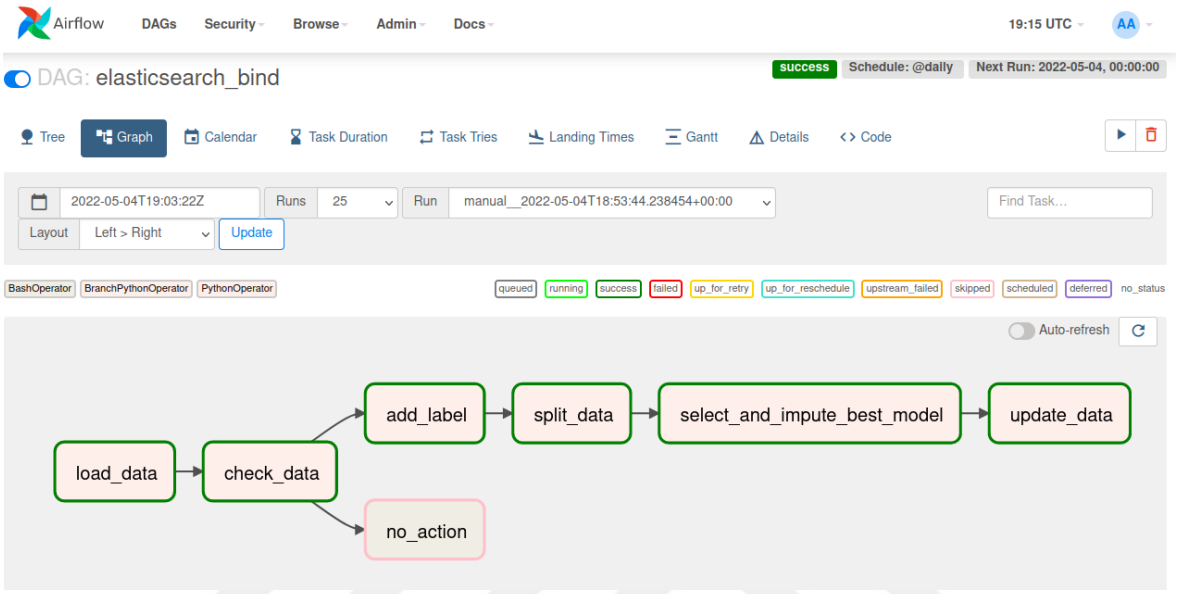


Şekil 4.5. DAG'ların çalışma durumunu gösteren Airflow arayüzü

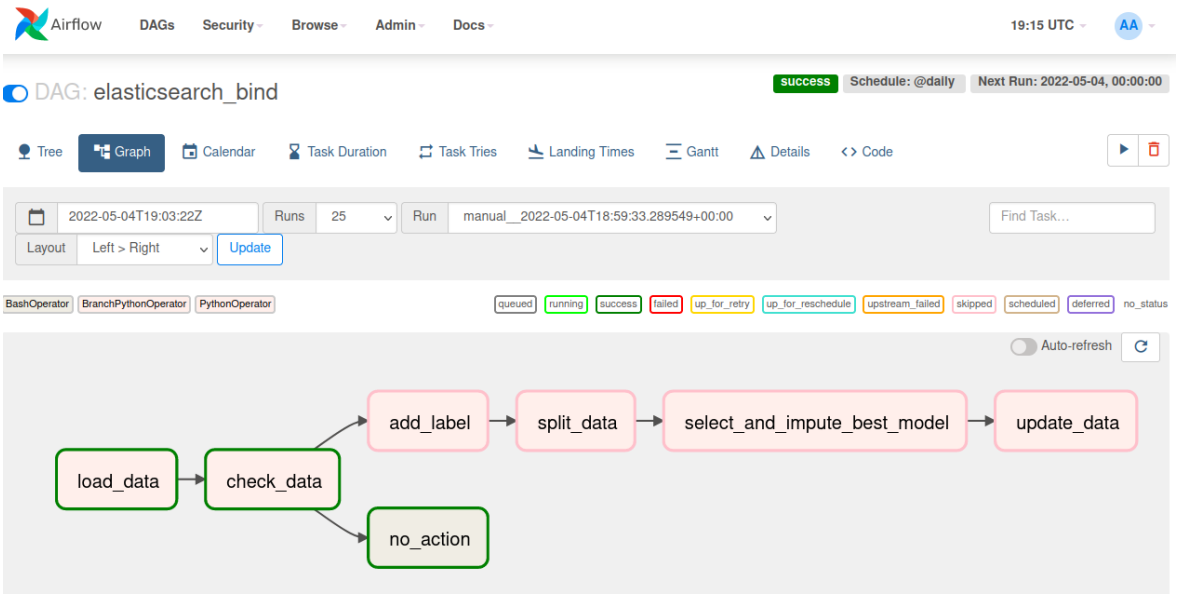
DAG'lar bazı durumlarda başarı (success) ile görevini yerine getirirken; bazı durumlarda başarısız (failed), bazı durumlarda bağımlı olduğu sürecin başarısızlığına bağlı olarak başarısız (upstream failed) veya yoğunluktan kaynaklı olarak kuyrukta bekliyor (queued) olabilmektedir. Apache Airflow'un sunduğu arayüz ile kullanıcı, çeşitli ağaç yapıları ve renklendirmeler sayesinde mevcut süreci kolayca anlayabilmekte ve duruma müdahale edebilmektedir. Bu çalışmada, her biri farklı bir görevi yerine getirmek üzere yedi adet DAG oluşturulmuştur. Bu DAG'lar birbirlerine çeşitli şekillerde bağımlılık göstermekte ve birbirlerinin çalışmasını tetiklemektedir. DAG'lar ve görevleri şu şekilde sıralanmaktadır:

1. **Load data:** IoT sistemden toplanan verilerin anlık olarak yazıldığı veritabanı ile etkileşime girilir ve sensör verileri Airflow'da belirlenen zaman periyotlarında gruplar halinde okunur.
2. **Check data:** Okunan veriler içerisinde eksik (null) değer olup olmadığı kontrol edilir.
3. **Add label:** Eksik veri varlığında var olan veriler eğitim (train) verisi, eksik veriler ise test verisi olarak etiketlenir.
4. **Split data:** Gerekli ön işlemeler doğrultusunda *x-train, y-train, x-test, y-test* değişkenleri oluşturulur.
5. **Select and impute best model:** Temel amaç, ML modelini eğitmek için mevcut verileri kullanmak ve eğitimli ML modeli ile null değerleri impute ederek verisetini eksiksiz hale getirmektir. TPOT AutoML aracı, test ve train olarak ayrılan veri kümelerini, en uygun ML algoritmasını ve algoritmanın en yüksek doğruluk değerine sahip hiperparametrelerin kombinasyonunu belirlemek için kullanır. TPOT, eksik verilerin orijinaline en yakın şekilde atanmasını sağlayacak ML algoritmasını belirler ve eksik verileri bu en iyi algoritma ile doldurur.
6. **Update data:** Eksik (null) değerler, ML algoritması tarafından atanan değerler ile değiştirilerek veritabanında güncellenir. Böylece boş değer içermeyen tam bir veriseti elde edilir.
7. **No action:** Check data işleminden sonra okunan veriler içerisinde eksik (null) değer yoksa bir sonraki çalıştırma planına kadar herhangi bir işlem yapılmaz.

Şekil 4.6 ve Şekil 4.7'de tasarlanan DAG yapısının eksik veri varlığında ve yokluğundaki çalışma düzeni yer almaktadır. DAG'ların boyandıkları renkler, süreçlerin çalışma durumu hakkında bilgiler içermektedir.



Şekil 4.6. Eksik veri varlığında gerçekleşen süreçleri temsil eden ağaç grafiği



Şekil 4.7. Eksik veri yokluğunda gerçekleşen süreçleri temsil eden ağaç grafiği

Her beş dakikada bir, her saat başı gibi belirlenen zaman aralıklarında gruplar halinde okunan verilerde eksik değer varsa ve veri imputasyonuna gerek duyuluyorsa Şekil 4.6 gibi; okunan toplu veride eksik değer yoksa ve veri imputasyonuna gerek duyulmuyorsa Şekil 4.7'deki gibi bir iş akışı görmek mümkündür. Burada yeşil ile temsil edilen süreçlerin problemsiz bir şekilde çalışarak görevlerini başarı ile yerine getirdikleri, pembe ile temsil edilen süreçlerin gerekli IF koşulunu sağlamadığı için atlanarak çalıştırılmadığı bilinmektedir.

Süreçlere ilişkin gerekli veritabanı bağlantıları ve kodlamalar yapılmış ve bu tez çalışmasının nihai hedefine ulaşılmıştır. Şekil 4.8'de Elasticsearch veritabanında, ATV kullanım senaryosuna ilişkin toplanan ivmeölçer (accelerometer) sensör verilerinden okunan değerler görülmektedir. Burada Y ekseni verisinin yazılamadığı ve null değer olarak boş bırakıldığı dikkat çekmektedir.

Şekil 4.8. Otomatik eksik veri imputasyonu öncesi sensör verisi

Tam bu noktada Apache Airflow platformunda tasarlanan DAG yapısı devreye girmektedir. Elasticsearch ile gerekli bağlantıyı kuran **Load data** DAG'ı verileri okumakta, **Check data** DAG'ı ise eksik veri varlığını tespit etmekte ve imputasyon için birbirlerine bağımlı süreçler tek tek çalıştırılmaktadır. Böylece Şekil 4.9'da, süreç sona erdiğinde Elasticsearch veritabanındaki verinin impute edilerek güncellendiği görülmektedir.

Şekil 4.9. Otomatik eksik veri imputasyonu sonrası sensör verisi

Kullanıcı müdahalesine ihtiyaç duymadan, belirli zaman aralıklarında AutoML yaklaşımı ile otomatik olarak veri artırımı sağlayan bu çalışmanın, zamandan ve maliyetten tasarruf sağlayarak araştırmacıların mevcut veri setleri ile daha tutarlı ve güvenilir bilimsel çalışmalar yapabilmesine ve veri biliminin önündeki engellere ışık tutması beklenmektedir.

5. SONUÇ VE ÖNERİLER

Yürütülen tez çalışması kapsamında, kestirimci bakım sistemlerine yönelik veri artırma yöntemleri geliştirilmiş ve IoT sistem platformu üzerinde bir uygulaması sunulmuştur. Ön hazırlık çalışması aşamasında regresyon-tabanlı altı ML algoritmasının eksik veri atama/imputasyon performansı, dört farklı eksiklik konumu (rastgele, başlangıç, orta, bitiş) ve dört farklı eksiklik oranı (%10, %20, %30, %40) içeren veri setleri üzerinde karşılaştırılmıştır. Orijinal veri setlerinin ve her bir ML modeli tarafından tamamlanan veri setlerinin tanımlayıcı istatistikleri incelenmiş ve kıyaslanmıştır. Modellerin performansları karşılaştırıldığında, Ridge Regresyon modelinin diğer ML modellere göre daha iyi bir imputasyon performansı sergilediği sonucuna varılmıştır. Öte yandan, eksik veri oranı arttıkça DTR modelinin diğer modellere kıyasla daha kötü bir performans gösterdiği gözlenmektedir.

IoT sistemlerinde, zaman serisi verilerinin gerçek zamanlı olarak sürekli toplanması ve bu verilerin kestirimci bakım sistemlerinde anomali/arıza tespitinde ve bakım planlamalarında kullanılması karar vericiler için oldukça kritik bir öneme sahiptir. Bu çalışmada, sensör verilerinin akışında herhangi bir eksik veri tespit edilmesi durumunda eksik verileri otomatik olarak atayarak eksiksiz veri setleri oluşturan bir veri artırma modülü tasarlanmıştır. Açık kaynaklı Apache Airflow iş akışı yönetim platformunda, AutoML yaklaşımıyla, insan müdahalesine ihtiyaç duymadan gerçekleşen süreçler bütünü, eksik verilerin en iyi makine öğrenmesi modeli ve hiperparametre kombinasyonu ile orijinaline en yakın şekilde doldurulmasını sağlamaktadır. Modüler, açık kaynak tabanlı ve izlenebilir bir yapıda olan bu veri artırma modülünün mevcut IoT sistem tasarımlarına kolayca entegre edilebilmesi ve zaman, maliyet ve insan gücünden tasarruf sağlaması beklenmektedir.

Gelecek çalışmalarda, önerilen veri artırma modülünün kullanım senaryosu ve içerdiği state-of-the-art teknolojiler arttırılabilir ve karar vericinin beklentisi doğrultusunda zamandan mı yoksa doğruluk değerinden mi kazanım sağlamak istediği opsiyonu sunulabilir.

KAYNAKLAR DİZİNİ

A Romero, R. A., Y Deypalan, M. N., Mehrotra, S., Jungao, J. T., Sheils, N. E., Manduchi, E., & Moore, J. H. (2022). *Benchmarking AutoML frameworks for disease prediction using medical claims.* BioData mining, 15(1), 1–13.

Adnan, F. A., Jamaludin, K. R., Wan Muhamad, W. Z. A., & Miskon, S. (2022). *A review of the current publication trends on missing data imputation over three decades: direction and future research*. Neural Computing and Applications, 1-16.

Afrifa-Yamoah, E., Mueller, U. A., Taylor, S., & Fisher, A. (2020). *Missing data imputation of high-resolution temporal climate time series data.* Meteorological Applications, 27(1), e1873.

Aheleroff, S., Xu, X., Lu, Y., Aristizabal, M., Velásquez, J. P., Joa, B., & Valencia, Y. (2020). *IoT-enabled smart appliances under industry 4.0: A case study.* Advanced engineering informatics, 43, 101043.

Alamoodi, A., Zaidan, B., Zaidan, A., Albahri, O., Chen, J., Chyad, M., Garfan, S., & Aleesa, A. (2021). *Machine learning-based imputation soft computing approach for large missing scale and non-reference data imputation*. Chaos, Solitons & Fractals, 151, 111236.

Allison, P. D. (2001). *Missing data.* Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-136. Thousand Oaks, CA: Sage.

Allison, P. D. (2003). *Missing data techniques for structural equation modeling*. Journal of abnormal psychology, 112(4), 545.

Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2019). *Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects.* Sensors, 20(1), 109.

KAYNAKLAR DİZİNİ (devam)

Apache Airflow. (2022a). *Airflow is a platform created by the community to programmatically author, schedule and monitor workflows.* Erişim: https://airflow.apache.org/, Erişim tarihi: 06.10.2022.

Apache Airflow. (2022b). *DAGs.* Erişim: https://airflow.apache.org/docs/apache-airflow/stable/concepts/dags.html, Erişim tarihi: 12.10.2022.

Apache Airflow. (2022c). *Operators.* Erişim: https://airflow.apache.org/docs/apache-airflow/stable/concepts/operators.html, Erişim tarihi: 12.10.2022.

Arslan, İ. (2019). *Python ile Veri Bilimi* (2. Baskı). Pusula Yayıncılık.

Awawdeh, S., Faris, H., & Hiary, H. (2022). *EvoImputer: An evolutionary approach for Missing Data Imputation and feature selection in the context of supervised learning.* Knowledge-Based Systems, 236, 107734.

Balaji, A., & Allen, A. (2018). *Benchmarking automatic machine learning frameworks.* arXiv preprint arXiv:1808.06492.

Baraldi, A. N., & Enders, C. K. (2010). *An introduction to modern missing data analyses.* Journal of school psychology, 48(1), 5–37.

Barrett, P., Hunter, J., Miller, J. T., Hsu, J.-C., & Greenfield, P. (2005). *matplotlib–A Portable Python Plotting Package.* Astronomical data analysis software and systems XIV, 347, 91.

Bokde, N., Beck, M. W., Álvarez, F. M., & Kulat, K. (2018). *A novel imputation methodology for time series based on pattern sequence forecasting.* Pattern recognition letters, 116, 88–96.

KAYNAKLAR DİZİNİ (devam)

Boukouvala, F., Muzzio, F. J., & Ierapetritou, M. G. (2010). *Predictive modeling of pharmaceutical processes with missing and noisy data*. AIChE Journal, 56(11), 2860–2872.

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. (2019). *A systematic literature review of machine learning methods applied to predictive maintenance.* Computers & Industrial Engineering, 137, 106024.

Chong, A., Lam, K. P., Xu, W., Karaguzel, O. T., & Mo, Y. (2016). *Imputation of missing values in building sensor data.* ASHRAE and IBPSA-USA SimBuild, 6, 407–14.

Clogg, C. C., Rubin, D. B., Schenker, N., Schultz, B., & Weidman, L. (1991). *Multiple imputation of industry and occupation codes in census public-use samples using Bayesian logistic regression.* Journal of the American statistical association, 86(413), 68-78.

Cinar, E., Kalay, S., & Saricicek, I. (2022). *A Predictive Maintenance System Design and Implementation for Intelligent Manufacturing.* Machines, 10(11). https://doi.org/10.3390/machines10111006

Demirhan, H., & Renwick, Z. (2018). *Missing value imputation for short to midterm horizontal solar irradiance data.* Applied Energy, 225, 998–1012.

Dong, Y. [Yifan], Xia, T., Fang, X., Zhang, Z., & Xi, L. (2019). *Prognostic and health management for adaptive manufacturing systems with online sensors and flexible structures.* Computers & Industrial Engineering, 133, 57–68.

Dong, Y. [Yiran], & Peng, C.-Y. J. (2013). *Principled missing data methods for researchers.* SpringerPlus, 2(1), 1–17.

KAYNAKLAR DİZİNİ (devam)

Dubois, P. F. (2007). *Guest editor's introduction: Python: Batteries included*. Computing in Science & Engineering, 9(3), 7–9.

Dündar, D. R., Sarıçiçek, İ., Çinar, E., & Yazıcı, A. (2021). *KESTİRİMCİ BAKIMDA MAKİNE ÖĞRENMESİ: LİTERATÜR ARAŞTIRMASI*. Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, 29(2), 256–276.

Enders, C. K. (2022). *Applied missing data analysis.* Guilford Publications.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). *Autogluon-tabular: Robust and accurate automl for structured data.* arXiv preprint arXiv:2003.06505.

Esteban, A., Zafra, A., & Ventura, S. (2022). *Data mining in predictive maintenance systems: A taxonomy and systematic review*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12(5), e1471.

Ferreira, L., Pilastri, A., Martins, C. M., Pires, P. M., & Cortez, P. (2021). *A comparison of AutoML tools for machine learning, deep learning and XGBoost.* 2021 International Joint Conference on Neural Networks (IJCNN), 1–8.

Fiware. (2022). *Interface with IOT, Robots, and Third-Party Systems.* Erişim: https ://www.fiware.org/developers/catalogue/, Erişim tarihi: 03.10.2022.

Fiware Foundation. (2022). *A curated framework of Open Source Platform components to accelerate the development of Smart Solutions.* Erişim: https://www.fiware.org/aboutus/, Erişim tarihi: 03.10.2022.

KAYNAKLAR DİZİNİ (devam)

Garretson, I. C., Mani, M., Leong, S., Lyons, K. W., & Haapala, K. R. (2016). *Terminology to support manufacturing process characterization and assessment for sustainable production.* Journal of Cleaner Production, 139, 986–1000.

Gijsbers, P., Bueno, M. L., Coors, S., LeDell, E., Poirier, S., Thomas, J., Bischl, B., & Vanschoren, J. (2022). *AMLB: an AutoML Benchmark.* arXiv preprint arXiv:2207.12560.

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019). *An open source AutoML benchmark.* arXiv preprint arXiv:1907.00909.

Grafana. (2022). *Grafana: The open observability platform.* Erişim: https://grafana.com/, Erişim tarihi: 27.09.2022.

Hadeed, S. J., O'Rourke, M. K., Burgess, J. L., Harris, R. B., & Canales, R. A. (2020). *Imputation methods for addressing missing data in short-term monitoring of air pollutants.* Science of The Total Environment, 730, 139140.

Hao, J., & Ho, T. K. (2019). *Machine learning made easy: a review of scikit-learn package in python programming language.* Journal of Educational and Behavioral Statistics, 44(3), 348–361.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., vd. (2020). *Array programming with NumPy.* Nature, 585(7825), 357–362.

Hasan, H., Ahmad, S., Osman, B. M., Sapri, S., & Othman, N. (2017). *A comparison of model-based imputation methods for handling missing predictor values in a linear regression model: A simulation study*. AIP Conference Proceedings, 1870(1), 060003.

KAYNAKLAR DİZİNİ (devam)

He, X., Zhao, K., & Chu, X. (2021). *AutoML: A survey of the state-of-the-art.* Knowledge-Based Systems, 212, 106622.

Hegde, H., Shimpi, N., Panny, A., Glurich, I., Christie, P., & Acharya, A. (2019). *MICE vs PPCA: Missing data imputation in healthcare.* Informatics in Medicine Unlocked, 17, 100275.

Hozdić, E. (2015). *Smart factory for industry 4.0: A review.* International Journal of Modern Manufacturing Technologies, 7(1), 28-35.

Hu, X., Pedrycz,W.,Wu, K., & Shen, Y. (2021). *Information granule-based classifier: A development of granular imputation of missing data.* Knowledge-Based Systems, 214, 106737.

IFARLAB. (2022). *Eskisehir Osmangazi University Intelligent Factory and Robotics Laboratory.* Erişim: https://ifarlab.ogu.edu.tr/, Erişim tarihi: 10.10.2022.

Izonin, I., Kryvinska, N., Tkachenko, R., & Zub, K. (2019). *An approach towards missing data recovery within IoT smart system.* Procedia Computer Science, 155, 11–18.

Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). *When and how should multiple imputation be used for handling missing data in randomized clinical trials–a practical guide with flowcharts.* BMC medical research methodology, 17(1), 1–10.

Jimenez, J. J. M., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). *Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics.* Journal of Manufacturing Systems, 56, 539–557.

KAYNAKLAR DİZİNİ (devam)

Kalay, S., Çinar, E., & Sarıçiçek, İ. (2022). *A Comparison of Data Imputation Methods Utilizing Machine Learning for a New IoT System Platform*. 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), 1, 69–74.

Khan, M., Ahmad, A., Sobieczky, F., Pichler, M., Moser, B. A., & Bukovský, I. (2022). *A Systematic Mapping Study of Predictive Maintenance in SMEs*. IEEE Access, 10, 88738-88749.

Kibana. (2022). *Your window into the Elastic Stack.* Erişim: https://www.elastic.co/kibana/, Erişim tarihi: 27.09.2022.

Kim, M., Park, S., Lee, J., Joo, Y., & Choi, J. K. (2017). *Learning-based adaptive imputation method with kNN algorithm for missing power data.* Energies, 10(10), 1668.

Kim, T., Ko, W., & Kim, J. (2019). *Analysis and impact evaluation of missing data imputation in day-ahead PV generation forecasting.* Applied Sciences, 9(1), 204.

Le, T. T., Fu, W., & Moore, J. H. (2020). *Scaling tree-based automated machine learning to biomedical big data with a feature set selector.* Bioinformatics, 36(1), 250–256.

Lee, J., Ardakani, H. D., Yang, S., & Bagheri, B. (2015). *Industrial big data analytics and cyber-physical systems for future maintenance & service innovation.* Procedia cirp, 38, 3–7.

Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data*. (Volume 793). John Wiley & Sons.

KAYNAKLAR DİZİNİ (devam)

Mahesh, B. (2020). *Machine learning algorithms-a review*. International Journal of Science and Research (IJSR), 9, 381–386.

Martinez-Luengo, M., Shafiee, M., & Kolios, A. (2019). *Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation.* Ocean Engineering, 173, 867–883.

McKinney, W. (2010). *Data structures for statistical computing in python.* In S. van der Walt & J. Millman (Eds.), Proceedings of the 9th Python in Science Conference, 445(1), 51–56. https://doi.org/10.25080/Majora-92bf1922-00a

McKinney, W. (2011). *pandas: a foundational Python library for data analysis and statistics.* Python for high performance and scientific computing, 14(9), 1–9.

McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* O'Reilly Media, Inc.

Mirzaei, A., Carter, S. R., Patanwala, A. E., & Schneider, C. R. (2022). *Missing data in surveys: Key concepts, approaches, and applications*. Research in Social and Administrative Pharmacy, 18(2), 2308-2316.

Mitchell, R., Pottier, L., Jacobs, S., da Silva, R. F., Rynge, M., Vahi, K., & Deelman, E. (2019). *Exploration of workflow management systems emerging features from users perspectives*. 2019 IEEE International Conference on Big Data (Big Data), 4537–4544.

Mobley, R. K. (2002). *An introduction to predictive maintenance*. Elsevier.

KAYNAKLAR DİZİNİ (devam)

Mohamed, M. H., Abdel-rahiem, A. H., & Abdelsamea, M. M. (2014). *Scalable algorithms for missing value imputation.* International Journal of Computer Applications, 87(11).

Ngueilbaye, A., Wang, H., Mahamat, D. A., & Junaidu, S. B. (2021). *Modulo 9 model-based learning for missing data imputation.* Applied Soft Computing, 103, 107167.

Noor, N. M., Al Bakri Abdullah, M. M., Yahaya, A. S., & Ramli, N. A. (2015). *Comparison of linear interpolation method and mean method to replace the missing values in environmental data set.* Materials Science Forum, 803, 278–281.

Okafor, N. U., & Delaney, D. T. (2021). *Missing data imputation on IoT sensor networks: Implications for on-site sensor calibration*. IEEE Sensors Journal, 21(20), 22833–22845.

Olson, R. S., & Moore, J. H. (2016). *TPOT: A tree-based pipeline optimization tool for automating machine learning.* Workshop on automatic machine learning, 66–74.

Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., & Moore, J. H. (2016). *Automating biomedical data science through tree-based pipeline optimization.* In European conference on the Applications of Evolutionary Computation, 123–137.

Osborne, J. W. (2013). *Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data.* Los Angeles, CA: Sage.

KAYNAKLAR DİZİNİ (devam)

Peppanen, J., Zhang, X., Grijalva, S., & Reno, M. J. (2016). *Handling bad or missing smart meter data through advanced data imputation.* 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 1–5.

Pigott, T. D. (2001). *A review of methods for missing data.* Educational research and evaluation, 7(4), 353–383.

Ramos, P., Oliveira, J. M. S., & Silva, P. (2014). *Predictive maintenance of production equipment based on neural network autoregression and ARIMA.* 21st International EurOMA Conference-Operations Management in an Innovation Economy.

Rubin, D. B. (1976). *Inference and missing data*. Biometrika, 63(3), 581–592.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys* (Volume 81). John Wiley & Sons.

Singh, P. (2019). *Supervised machine learning.* Learn PySpark, 117–159.

Su, C.-J., & Huang, S.-F. (2018). *Real-time big data analytics for hard disk drive predictive maintenance.* Computers & Electrical Engineering, 71, 93–101.

Vafaei, N., Ribeiro, R. A., & Camarinha-Matos, L. M. (2019). *Fuzzy early warning systems for condition based maintenance.* Computers & Industrial Engineering, 128, 736–746.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). *The NumPy array: a structure for efficient numerical computation.* Computing in science & engineering, 13(2), 22–30.

KAYNAKLAR DİZİNİ (devam)

Van Engelen, J. E., & Hoos, H. H. (2020). *A survey on semi-supervised learning.* Machine Learning, 109(2), 373-440.

Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., & Mueller, A. (2015). *Scikit-learn: Machine learning without learning the machinery.* GetMobile: Mobile Computing and Communications, 19(1), 29–33.

Vastrad, C., vd. (2013). *Performance analysis of neural network models for oxazolines and oxazoles derivatives descriptor dataset.* arXiv preprint arXiv:1312.2853.

Velasco-Gallego, C., & Lazakis, I. (2020). *Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study*. Ocean Engineering, 218, 108261.

Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). *A manufacturing big data solution for active preventive maintenance.* IEEE Transactions on Industrial Informatics, 13(4), 2039–2047.

Wang, M.-C., Tsai, C.-F., & Lin, W.-C. (2021). *Towards missing electric power data imputation for energy management systems.* Expert Systems with Applications, 174, 114743.

Wang, Z., Wang, L., Tan, Y., & Yuan, J. (2021). *Fault detection based on Bayesian network and missing data imputation for building energy systems.* Applied Thermal Engineering, 182, 116051.

Xu, L., Da, Xu, E.L. & Li, L. (2018). *Industry 4.0: state of the art and future trends.* International Journal of Production Research, Vol. 56 No. 8, pp. 2941-2962. DOI: 10.1080/00207543.2018.1444806

KAYNAKLAR DİZİNİ (devam)

Yozgatligil, C., Aslan, S., Iyigun, C., & Batmaz, I. (2013). *Comparison of missing value imputation methods in time series: the case of Turkish meteorological data*. Theoretical and applied climatology, 112(1), 143–167.

Zhang, W., Yang, D., & Wang, H. (2019). *Data-driven methods for predictive maintenance of industrial equipment: A survey.* IEEE Systems Journal, 13(3), 2213-2227.

Zhang, Y., & Thorburn, P. J. (2022). *Handling missing data in near real-time environmental monitoring: A system and a review of selected methods.* Future Generation Computer Systems, 128, 63–72.

Zonta, T., Da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). *Predictive maintenance in the Industry 4.0: A systematic literature review*. Computers & Industrial Engineering, 150, 106889.

Zöller, M.-A., & Huber, M. F. (2021). *Benchmark and survey of automated machine learning frameworks.* Journal of artificial intelligence research, 70, 409–472.

Ek Açıklama-A: Bu Tez Çalışmasından Üretilen Bilimsel Yayınlar

1. Kalay, S., Çinar, E., & Sarıçiçek, İ. (2022, May). A Comparison of Data Imputation Methods Utilizing Machine Learning for a New IoT System Platform. In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)* (Vol. 1, pp. 69-74). IEEE.

2. Cinar, E., Kalay, S., & Saricicek, I. (2022). A Predictive Maintenance System Design and Implementation for Intelligent Manufacturing. *Machines*, 10(11), 1006.