

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**ENDÜSTRİ 4.0 ENTEGRASYONU İÇİN BULUT TABANLI
ÜRETİM YÖNETİM SİSTEMİ GELİŞTİRİLMESİ**

Şebnem Dicle TÜRKMEN
Yüksek Lisans Tezi
YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

HAZİRAN 2022

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

ENDÜSTRİ 4.0 ENTEGRASYONU İÇİN BULUT TABANLI ÜRETİM YÖNETİM SİSTEMİ GELİŞTİRİLMESİ

Tez Yazarı

Şebnem Dicle TÜRKMEN

Danışman

Doç. Dr. Mustafa ULAŞ

HAZİRAN 2022

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yazılım Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı:	Endüstri 4.0 Entegrasyonu İçin Bulut Tabanlı Üretim Yönetim Sistemi Geliştirilmesi
Yazarı:	Şebnem Dicle TÜRKMEN
İlk Teslim Tarihi:	30.05.2022
Savunma Tarihi:	30.06.2022

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

Danışman:	Doç. Dr. Mustafa ULAŞ Fırat Üniversitesi, Mühendislik Fakültesi	<i>İmza</i> Onayladım
Başkan:	Dr. Öğr. Üyesi Oğuz ATA Altınbaş Üniversitesi, Mühendislik ve Mimarlık Fakültesi	Onayladım
Üye:	Prof. Dr. Engin AVCI Fırat Üniversitesi, Teknoloji Fakültesi	Onayladım

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Endüstri 4.0 Entegrasyonu İçin Bulut Tabanlı Üretim Yönetim Sistemi Geliştirilmesi” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

30.06.2022

Şebnem Dicle TÜRKMEN



ÖNSÖZ

Bu tez çalışması, Fırat Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim Dalı Tezli Yüksek Lisans programı için hazırlanmıştır.

Çalışmalarım boyunca beni yönlendiren, sabırla engin birikimlerini paylaşarak akademik hayatıma yön veren, desteğini esirgemeyen değerli hocam Sayın Doç. Dr. Mustafa ULAŞ'a sonsuz saygı ve teşekkürlerimi sunarım.

Yüksek lisans eğitimim süresince beni teşvik eden ve destekleri ile sürekli yanımda olan Hepsiburada.com ve Hepsiburada Ar-Ge merkezine, çalışma arkadaşlarıma sonsuz teşekkür ederim.

Bu süreçte beni her konuda maddi ve manevi destekleyen, sürekli arkamda olan, eğitim sürecinde beni hiçbir zaman yalnız bırakmayan sevgili eşim Aytuğ BOYACI'ya, anneme, kardeşime, yoğun çalışma dönemlerinde ihmal ettiğim arkadaşlarıma, üzerimde hakkı olan ve çalışmalarımda emeği geçen herkese teşekkürü bir borç bilirim.

Şebnem Dicle TÜRKMEN
ELAZIĞ, 2022

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	x
EKLER LİSTESİ	xi
KISALTMALAR	xii
1. GİRİŞ	1
2. GENEL KAVRAMLAR.....	3
2.1. Endüstriyel Veri.....	3
2.1.1. Veri.....	3
2.1.2. Büyük Veri	3
2.1.3. Büyük Veri Analizi	3
2.1.4. Bulut Bilişim	4
2.1.5. Endüstriyel Otomasyon ve Kontrol Sistemleri	4
2.1.6. Akıllı Fabrika	4
2.1.7. Endüstri 4.0	5
2.1.8. Sanayi 4.0.....	6
2.2. Endüstri 4.0 Bileşenleri	6
2.2.1. Nesnelerin İnterneti	7
2.2.2. Büyük Veri ve Analitik	9
2.2.3. Ekl emeli Üretim	9
2.2.4. Bulut Bilişim	9
2.2.5. Gelişmiş Robotlar.....	9
2.2.6. Benzetim	10
2.2.7. Yapay Zeka	10
2.2.8. Artırılmış Gerçeklik	10
2.2.9. Yatay ve Dikey Entegrasyon.....	10
2.2.10. Siber Fiziksel Sistemler.....	11
2.3. Büyük Veri Mimarisi ve Veri Platformları	11
2.3.1. Hadoop	12
2.3.2. Apache Kafka	15
2.3.3. MongoDB.....	15
2.3.4. Elasticsearch.....	16
2.4. Bulut Bilişim Mimarisi	16
2.4.1. Bulut Bilişim Servisleri	17
2.4.2. Bulut Bilişim Dağıtım Modelleri	19
2.4.3. Bulut Bilişimin Uygulama Geliştirme Alanındaki Kullanımı	20
2.4.4. Bulut Tabanlı Platformlar.....	21
2.4.5. Bulut Tabanlı Uygulamalar	21
2.5. Makine Öğrenmesi Teknikleri	23

2.5.1. Perceptron Öğrenme Algoritması.....	23
2.5.2. Ridge Regresyon Algoritması	24
2.5.3. Ensemble (Topluluk) Öğrenme Algoritması	25
2.5.4. Gradyan Artırma (Gradient Boosting - GBM) Algoritması	25
2.5.5. Olasılıksal Dereceli Azalma (Stochastic Gradient Descent – SGD) Algoritması	26
2.5.6. Navie Bayes	26
2.5.7. XGBoost Algoritması.....	26
2.5.8. Rasgele Orman (Random Forest) Algoritması	27
2.5.9. LGBM (LightGBM) Sınıflandırma Algoritması	27
2.5.10.KNN (K-Nearest Neighbors – K-En Yakın Komşu) Algoritması	27
2.5.11.Karar Ağacı Algoritmaları.....	27
2.5.12.CatBoost Sınıflandırma Algoritması	28
2.5.13.AdaBoost Sınıflandırma Algoritması	28
3. SİSTEM MİMARİSİ VE UYGULAMA	29
3.1. Giriş	29
3.2. Mimarinin Genel Yapısı	31
3.3. Gerekli Kabuller	32
3.4. Materyal ve Metot	33
3.5. Veriseti	34
4. BULGULAR VE TARTIŞMA	37
4.1. Kural Tabanlı Modeller	37
4.1.1. Üretim Bandı Mevcut Durum.....	37
4.1.2. Üretim Bandı Sipariş Takibi.....	38
4.1.3. Makine Parkuru İş Yüğü	38
4.1.4. Tezgah Bazlı Performans Takibi	39
4.2. Makine Öğrenmesi Yöntemleri ile Kestirimci Arıza Tespit Modeli	39
4.2.1. Öznitelik Seçimi.....	40
4.2.2. Modellerin Uygulanması ve Sonuçlar	42
5. SONUÇ.....	47
KAYNAKLAR.....	48
EKLER	52
ÖZGEÇMİŞ	

ÖZET

Endüstri 4.0 Entegrasyonu İçin Bulut Tabanlı Üretim Yönetim Sistemi Geliştirilmesi

Şebnem Dicle TÜRKMEN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Haziran 2022, Sayfa: xii + 51

Üretim sürecinin yönetilmesi işletmenin performans ve veriminin artırılması için elzem bir durumdur. Hammadde, personel ve makine gibi kaynakların bir düzen içerisinde üretimin parçasını oluşturması ise ancak iyi yönetilen bir iş akışı ile mümkün olabilmektedir. Kurumsallaşmamış olan birçok KOBİ (Küçük ve Orta Büyüklükteki İşletmeler) ölçeğindeki işletmenin bu konuda başvurduğu çözüm üretim sürecinin doğrudan bir koordinatör tarafından performans takibi göz ardı edilerek yönetilmesinden geçmektedir. Ancak bu geçici çözüm işletmelerin büyüme trendlerine sekte vuran en büyük engel olduğu görülmektedir. İnsan eli ile yapılan kontroller limitlidir. İşletmeler kapasite artışına gittiğinde ise bu insan eli ile yapılan üretim süreç yönetimi, işletme performansının ve karlılığının düşüşüne sebep olmaktadır. Bu konu hakkında yapılacak en önemli çözüm ise üretim süreç yönetiminin makinalar ve yazılımlar ile yapılmasıdır. Endüstri 4.0 entegrasyonu olmamış işletmelerde ürün süreci yönetiminin önünde makine parkurunun bilgisayar kontrollü olmayışı, tedarikten dağıtıma kadar tüm lojistik sürecin manuel yöntemler ile yönetilmesi ve çalışan personelin teknolojik altyapılarının eksikliği gibi faktörler çözülmesi gereken problemler olarak durmaktadır. Bu çalışmada, imalat sanayisinde Endüstri 3.0 üretim modeli ile faaliyet gösteren işletmelerin endüstri 4.0'a geçiş yapabilmelerini sağlayacak bulut tabanlı üretim yönetim sistemi ortaya konmuştur.

Anahtar Kelimeler: Üretim Yönetim Sistemi, Makine Öğrenmesi, Bulut Bilişim, Endüstri 4.0

ABSTRACT

Cloud-Based Manufacturing Execution System Development for Industry 4.0 Integration

Şebnem Dicle TÜRKMEN

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences
Department of Software Engineering

June 2022, Pages: xii + 51

Managing the production process is essential for increasing the performance and efficiency of the enterprise. It is only possible with a well-managed workflow that resources such as raw materials, personnel and machinery form part of the production in an orderly manner. The solution that many non-institutionalized SME-scale businesses resort to is the management of the production process directly by a coordinator, ignoring the performance monitoring. However, this temporary solution seems to be the biggest obstacle to the growth trends of businesses. Man-made controls are limited. When businesses increase their capacity, this human-made production process management causes a decrease in business performance and profitability. The most important solution to this issue is the production process management with machines and software. Factors such as the lack of computer control of the machine park in front of the product process management in enterprises without Industry 4.0 integration, the management of the entire logistics process from supply to distribution by manual methods, and the lack of technological infrastructure of the working personnel stand as problems to be solved. In this study, a cloud-based production management system has been introduced that will enable businesses operating with the Industry 3.0 production model in the manufacturing industry to transition to industry 4.0.

Keywords: Manufacturing Execution System, Machine Learning, Cloud Computing, Industry 4.0

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. Endüstri devrimi.....	5
Şekil 2.2. Endüstri 4.0 bileşenleri.....	6
Şekil 2.3. Hadoop bileşenleri.....	13
Şekil 2.4. Bulut tabanlı uygulamalar	17
Şekil 2.5. Bulut tabanlı uygulamalar	18
Şekil 2.6. Bulut tabanlı uygulamalar	22
Şekil 2.7. Perceptron öğrenme algoritması modeli	24
Şekil 2.8. Lineer regresyon ve Ridge regresyon karşılaştırması	24
Şekil 3.1. Mimari model.....	32
Şekil 3.2. Modelin arayüz taslakları	33
Şekil 3.3. Ham veriseti	35
Şekil 4.1. Üretim bandı mevcut durum.....	37
Şekil 4.2. Üretim bandındaki siparişlerin listesi	38
Şekil 4.3. Makine parkuru anlık iş yükü.....	38
Şekil 4.4. Tezgah bazlı personelin performans takibi.....	39
Şekil 4.5. Üretim İstatistikleri	39
Şekil 4.6. CatBoost modeli ve SHAP tekniği ile öz niteliklerin ağırlıkları	40
Şekil 4.7. LightGBM modeli ve SHAP tekniği ile öz niteliklerin ağırlıkları	41
Şekil 4.8. XGBoost modeli ve SHAP tekniği ile öz niteliklerin ağırlıkları	41
Şekil 4.9. Rastgele Orman Algoritması ve SHAP tekniği ile öz niteliklerin ağırlıkları	42
Şekil 4.10. Modellere ait karmaşıklık matrisleri ve performans metrikleri	45

TABLÖLAR LİSTESİ

	Sayfa
Tablo 3.1. Veriseti öznitelikleri	36
Tablo 4.1. Model başarımları	45



EKLER LİSTESİ

	Sayfa
Ek- 1: Shap ve Rastgele Orman Algoritması Kullanarak Öznitelik Seçimi Kaynak Kodları	52
Ek- 2: Shap ve CatBoost Algoritması Kullanarak Öznitelik Seçimi Kaynak Kodları	53
Ek- 3: AdaBoost Modeli Kaynak Kodları	54
Ek- 4: CatBoost Modeli Kaynak Kodları	55
Ek- 5: Rastgele Orman Modeli Kaynak Kodları.....	56



KISALTMALAR

ML	: Makine Öğrenmesi
AI	: Yapay Zeka
SVM	: Destek Vektör Makinesi
LSTM	: Uzun Kısa Süreli Bellek
CART	: Sınıflandırma ve Regresyon Ağacı
KNN	: K-En Yakın Komşu
IoT	: Nesneleri İnterneti
KOBİ	: Küçük ve Orta Büyüklükteki İşletmeler
M2M	: Makineden Makineye
PLC	: Programlanabilir Lojik Kontrol
AR	: Artırılmış Gerçeklik
NFC	: Yakın Alan İletişimi
BAN	: Vücut Alan Ağı
PAN	: Personel Alan Ağı
CAN	: Kampüs Alan Ağı
MAN	: Metropol Alan Ağı
WAN	: Geniş Alan Ağı
HDFS	: Hadoop Dağıtılmış Dosya Sistemi
IAAS	: Hizmet Olarak Altyapı
PAAS	: Hizmet Olarak Platform
SAAS	: Hizmet Olarak Yazılım
CRM	: Müşteri İlişkileri Yönetimi
ERP	: Kurumsal Kaynak Planlama
AWS	: Amazon Web Servisleri
GBM	: Gradyan Artırma Algoritması
SGD	: Olasılıksal Dereceli Azalma Algoritması
LGBM	: LightGBM Algoritması

1. GİRİŞ

Teknolojinin gelişmesi ile birlikte endüstri alanında da dijital teknolojiler aktif bir şekilde kullanılmaya başlanmıştır. Üretim süreçleri günümüzde tek bir noktada üretim çözümlerinden ziyade dağıtık bir yapıda farklı imalat bantlarından üretim çözümlerine dönüşmüştür. Pazarda rekabet koşullarının artması, ihtiyaçların sürekli değişkenlik göstermesi, kullanıcı odaklı ürün geliştirme ihtiyacı hem mevcut üretim tekniklerini değiştirmiş hem de mevcut verilerden katma değer katacak anlamlar katılmasını zorunlu kılmıştır. Bu da özellikle imalat endüstrisi başta olmak üzere tüm üretim tesislerinin akıllı fabrika süreçlerine adapte olmasına yol açmıştır. Kullanılacak akıllı donanım ve yazılımların seçimi, uçtan uca üretim planlamasının yapılabilmesi, lojistik süreçlerinin anlık takibi, ürün kalitesinin otonom teknikler ile kontrol edilebilmesi, personel verimliliğinin ölçülebilmesi ve makine parkurunun 7/24 uzaktan yönetiminin gerçekleştirilmesi Endüstri 4.0 kavramı ile birlikte ortaya çıkan ihtiyaçlar haline gelmiştir. Bu ihtiyaçların minimum maliyetle ve yüksek doğruluk ile yönetilebilmesi için özellikle sistemin biriktirmiş olduğu büyük verinin analizi ön plana çıkmaktadır. Bir başka deyişle imalat endüstrisi başta olmak üzere tüm üretim tesislerinde rekabetçi ve kaliteli ürünler ortaya çıkarabilmek için verimlilik ve üretkenliği arttıracak çözümlere ihtiyaç duyulmaktadır [1].

Nesnelerin interneti (Internet of Things - IoT cihazlarının endüstride kullanımının yaygınlaşması ile birlikte makine parkurlarından verinin toplanması, toplanan verinin depolanması ve verinin analiz edilerek anlamlandırılması mümkün hale gelmiştir. Toplanan büyük veri analitik yöntemler vasıtasıyla yorumlanarak mevcut sorunlara yeni bakış açıları getirmeye ve tamamen yeni araştırma alanlarının ortaya çıkmasına olanak tanımaktadır.

Bu tez çalışmasında özellikle imalat sanayisindeki makine parkurlarından verilerinin nasıl üretildiği ve oluşan büyük veri için kullanılacak olan altyapı çözümleri irdelenerek özellikle Endüstri 4.0'a geçiş sürecindeki işletmeler için bulut tabanlı bir üretim yönetim sistemi önerilmiştir. Önerilen sistem üzerinde çalışacak kural tabanlı yöntemler ve makine öğrenmesi modelleri ile akıllı karar destek sistemleri oluşturulmuştur.

Tez çalışması ile Endüstri 4.0 tabanlı sistemlerdeki ihtiyaçlara akıllı karar destek sistemlerinin oluşturması için bir başlangıç yapılması planlanmaktadır. Bu bağlamda üretim tesislerinin ihtiyaçları analiz edildiğinde birçok ihtiyacın olduğu görülmekle beraber aşağıdaki sorunların ön plana çıktığı görülmektedir.

- Bakım masraflarının azaltılması
- Hata giderlerinin azaltılması
- Görsel modellerin maliyetinin düşürülmesi ve değerinin yükseltilmesi
- Bakım ve arızaların çözümünde uzaktan müdahale yöntemlerinin kullanılabilmesi

- Makine parkurundaki verilerin fiziksel temasta bulunmadan anlık takibinin yapılabilmesi
- Arıza yönetimi için makine operatörlerinin için gerçek zamanlı görev planlamasının yapılabilmesi [2]

Tez çalışmasında bu problemlerin çözümü için gerekli olan altyapının oluşturulmasına yönelik çözümler irdelenecektir.

Bu tez çalışması toplam beş bölümden oluşmaktadır. Tezin birinci bölümü Giriş bölümüdür. Bu bölümde tezin çıkış noktası, amacı, hedeflenen çıktılar, çalışmanın literatürdeki yeri detayları ile açıklanmıştır. Tezin ikinci bölümü Genel Kavramlar bölümüdür. Bu bölümde tez çalışmasının anlaşılabilmesi için çalışmada kullanılan altyapı, teknikler ve yöntemler ile ilgili konular sunulmuştur. Tez çalışmasının üçüncü bölümü ise sistem mimarisi ve uygulamayı ortaya koymaktadır. Bu bölümde sistem mimarisinin genel yapısı, çalışma gerçekleştirilirken yapılan gerekli kabuller, kullanılan verisetleri, çalışmada kullanılan materyal ve metot verilmiştir. Tezin dördüncü bölümünde ise gerçekleştirilmiş uygulamalara ait bulgular ortaya konularak sistemin başarısı tartışılmıştır. Tez çalışmasının son bölümünde ise gerçekleştirilmiş olan çalışmada elde edilen çıktılar, bu çıktıların teknolojiye ve literatürdeki yeri ortaya konmuştur. Yine son bölümde gelecek çalışmalar için öneriler sunulmuştur.

2. GENEL KAVRAMLAR

Tez çalışmasında kullanılan yöntemlerin iyi şekilde anlaşılabilmesi için tez çalışmasında kullanılan kavramların, yöntemlerin bilinmesi gerekmektedir. Bu bölümde tez çalışmasında kullanılan genel kavram ve yöntemler açıklanmıştır.

2.1. Endüstriyel Veri

2.1.1. Veri

Bilgisayarlar tarafından işlenebilecek sayısal değerler veri olarak adlandırılmaktadır. Günümüzde teknolojinin hızlı gelişimi ve iş süreçlerindeki tüm operasyonların bilgisayarlar tarafından gerçekleştirilme ihtiyacı sayısal verinin analizini son derece önemli hale getirmiştir. İş süreçlerindeki deneysel sonuçlar, gözlemler, toplanan sinyallerin tümü veri kavramını oluşturmaktadır. Günümüzde sağlık sektöründen eğitime, savunma sanayiinden yüksek teknoloji ürün üretimine kadar tüm alanlarda verinin anlık toplanabilmesi, işlenebilmesi ve anlamlı sonuçlar çıkarılabilmesi son derece önemli hale gelmiştir. Bu bağlamda veri kavramını bilişim sistemleri tarafından okunabilen bilgiler olarak tanımlamak mümkündür [3].

2.1.2. Büyük Veri

Teknolojinin gelişmesi ile birlikte iş süreçlerinden günlük yaşantıya kadar birçok alanda sayısal uygulamalar ve sayısallaşmış süreçler karşımıza çıkmaktadır. Bu uygulama ve sistemler üzerinde anlık olarak oluşan veri akışı her geçen gün çok daha hızlı bir şekilde büyümektedir. Bu veriler incelendiğinde hacimli, farklı tiplerde (ses, görüntü, metin tabanlı vb.) anlık ve çok hızla büyümekte olduğu göze çarpmaktadır. Bu verilerin gerçek zamanlı toplanabilmesi, saklanabilmesi, kullanıcılara güvenli bir şekilde sunulabilmesi, görsel raporların oluşturulabilmesi ve görselleştirilebilmesi, silinmesi ya da güncellenebilmesi ve en önemlisi analiz edilerek anlamlı sonuçlar ortaya konulabilmesi oldukça önemli bir problemdir ve bu probleme klasik yöntemler yanıt verememektedir [4]. Oluşan veriye erişim hızından canlı olarak analize birçok konuda anlık çözümlerin ortaya konulması gerekmektedir. Bu bağlamda büyük veri kavramı canlılık ve hızla büyüyen, farklı tiplerde ve dağıtık sistemlerde olan veriler bütünü olarak tanımlanmaktadır.

2.1.3. Büyük Veri Analizi

Büyük verinin anlık olarak analizi yapılmadıkça bu veriler sadece ham veri yığınlarından ibarettir. Oluşan verinin içerisinde doğrudan ya da dolaylı olarak öğrenilebilecek birçok anlamlı bilgi olabilmektedir. Bu bağlamda çeşitli akıllı öğrenme teknikleri barındıran algoritmalar ve

sistemler ile büyük verinin analiz edilmesi ve yüksek güvenilirlikli bilgiler ortaya çıkarılmasına olanak sağlayan yapılar büyük veri analizi olarak tanımlanmaktadır. Büyük veri analizi ile ortaya çıkartılacak doğruluk oranı yüksek bilgiler ışığında karar destek sistemleri oluşturmak mümkün hale gelebilmektedir.

2.1.4. Bulut Bilişim

Teknoloji ile birlikte bilgisayar olarak tanımlanan sistemlerde başkalaşım geçirmiştir. Bilgisayarlar eskiden disk ve monitör gibi öğeleri barındıran cihazlarken verinin çok büyümesi ve farklı alanlarda aynı verinin kullanılabilme ihtiyacı iletişim sistemlerinin yeniden kurgulanmasını sağlamıştır. Bu bağlamda ortaya çıkan bulut bilişim kavramı, bilgisayarlar ve bilgisayar özelliği barındıran sistemlerin disk, hafıza, işlemci gibi ihtiyaçlarının karşılanabilmesini sağlayan ve mekan bağımsız internet üzerinden yönetilebilen sistemler olarak tanımlanmaktadır [6]. Endüstri 4.0 kapsamında oluşan verinin merkezi bir şekilde yönetimi için bulut bilişim teknolojileri yaygın olarak kullanılmaktadır. Bir üretim tesisinin birden fazla farklı yerleşkede bütünsel olarak çalışabilmesi ve tüm sistemlerin mekandan bağımsız çalışabilmesi için bulut bilişim teknolojilerinden yararlanılmaktadır.

2.1.5. Endüstriyel Otomasyon ve Kontrol Sistemleri

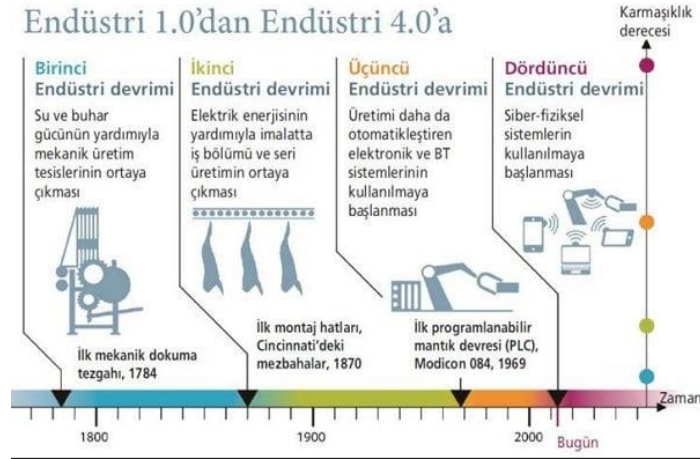
Özellikle üretim yapan işletmeler üretim süreçlerini hammaddenin girişinden nihai son ürünün ortaya çıkmasına kadarki süreçlerini çeşitli otomasyon sistemleri ile yönetmektedirler. Bu sistemler yönetim tesisin içinde kapalı devre olabileceği gibi uzaktan yönetilebilen modern otomasyon sistemleri şeklinde de olabilmektedir. Endüstriyel iş akışlarını yönetmek, kullanılan sistemleri anlık olarak kontrol etmek için kullanılan bu sistemlere Endüstriyel Otomasyon ve Kontrol Sistemleri (EOKS) adı verilmektedir.

2.1.6. Akıllı Fabrika

Uzaktan yönetilebilen ve yapay zeka tabanlı çözümler ile akıllı karar destek sistemleri ile donatılmış EOKS altyapısına sahip fabrikalar akıllı fabrikalar olarak adlandırılmaktadır. Bu fabrikalarda makine – makine etkileşimi, kaynakların optimum verim ile kullanılması ön plana çıkmaktadır. Birden fazla tesisin aynı anda yönetilebildiği, mevcut verilerden öğrenilen kurumsal hafızaya göre geleceğe yöneliktestrimlerinin yapılabildiği teknikler akıllı fabrikalarda sıklıkla kullanılmaktadır.

2.1.7. Endüstri 4.0

Dijitalleşme sürecinde endüstri alanlarının teknolojik altyapı ile birleşmesi, iş akışları ve üretim süreçlerinin otonom hale getirilmesi, sensör ve kullanılan donanımların akıllı hale gelmesi Endüstri 4.0 kavramının temelini oluşturmaktadır. Endüstri 4.0 kavramı ile gelişen imalat endüstrisi ve fabrikalarda kullanılan makine – teçhizatın daha da akıllanması, yapay zeka bileşenleri ile donatılması ve imalatın uçtan uca bütünlük yönetim sistemleri ile yönetilebilmesi ihtiyaçlarını ortaya çıkarmaktadır. Bununla birlikte uçtan uca yönetilen sistemin oluşturduğu büyük verinin anlık olarak işlenmesi, analiz edilmesi ve otonom kararların oluşturulabilmesi gerekmektedir. Bu da Endüstri 4.0 ile beraber gelen önemli bir ihtiyaç olarak karşımıza çıkmaktadır. Bu sayede Endüstri 4.0 ile birlikte rekabet edilebilir ürün geliştirebilmek için üretim bantlarının tamamen sayısallaştırılması, makine parkurundan toplanan sayısal verinin anlık olarak analizi ve makine parkurunun verimli kullanımı için periyodik bakımlarının anlık olarak takip edilmesi gerekmektedir. Makine parkurlarının bakım periyotlarında oluşabilecek aksamalar üretim süreçlerini doğrudan olumsuz yönde etkileyecek ya da üretimin tamamen durmasına yol açacaktır. Bu durum özellikle imalat sektöründe hizmet veren firmaların makine parkurunu optimum düzeyde verimli kullanabilmeleri için tahmine dayalı bakım süreçleri ile entegre olabilecek sistemlere zorlamaktadır [7]. Bir başka deyişle imalata endüstrisinde daha rekabetçi hale gelebilmek için işletmeler verimlilik ve üretkenliklerini artırması gerekmektedir. Endüstri 4.0'a gelen süreç Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Endüstri devrimi

IoT kavramının yaygınlaşması ile birlikte endüstriyel alanlardan verinin toplanması, depolanabilmesi ve anlamlandırılması mümkün olabilmektedir. Özellikle imalat sanayiindeki makine parkurunun verilerin sayısallaştırılması, sayısallaştırılan veriler üzerinden kestirimci bakım metodolojileri ile makine parkurunun daha verimli kullanılması, sayısallaştırılan verinin operasyonel süreçlerle verimli bir şekilde bütünleştirilmesi, analiz ve karar destek sistemleri oluşturulması Endüstri 4.0 kavramının öncelikleri arasında yer almaktadır.

2.1.8. Sanayi 4.0

Sanayi 4.0 olarak ta adlandırılan dördüncü sanayi devrimi kavramı ile birlikte akıllı fabrikaların yönetim biçimlerinden üretim süreçlerine kadar birçok alanda köklü değişiklikler ortaya çıkmaktadır. Bu değişimler sayesinde ürün maliyetlerinde düşüş, ürün kalite süreçlerinin artması ile birlikte daha kaliteli nihai ürün, akıllı lojistik yönetimi, makine – makine etkileşimli otonom süreçlerin ortaya konulması ve rekabetçi fiyatlar ortaya çıkmıştır. Sanayi 4.0'ın en önemli hedeflerinden biride karanlık fabrikalar olarak adlandırılmaktadır. Karanlık fabrikaların ortaya çıkabilmesi için iş akışı ve süreçlerin dinamik bir şekilde planlanmış olması ve anlık olarak üretilen verinin toplanıp işlenmesi ve ortaya çıkan sonuçların sisteme geri besleme olarak sürekli bir iyileştirme yaparak marjinal verimin artırılması beklenmektedir. Bunun için Sanayi 4.0'ın önemli bileşenleri olan akıllı sensörler, yönetilebilen akıllı donanımsal araçlar, MES (İmalat Yönetim Sistemi), bulut tabanlı sistemler ve yapay zeka tabanlı karar destek sistemleridir [8].

Sanayi 4.0 ile akıllı fabrikalarda PLC'ler, Scada sistemler ve diğer donanımsal bileşenler yardımı ile makine parkurundan toplanan veriler merkezi bir bulut sistemine anlık olarak aktarılabilir, aktarılan veri önceki veri kümesi ile veri madenciliği, makine öğrenmesi ve yapay zeka teknikleri ile anlamlandırılarak bir karar destek sistemi oluşturulabilir ve oluşturulan raporlar web, mobil cihazlar (telefon / tablet / akıllı saat / AR gözlüğü) ve IoT sistemlerin ön yüzlerinde son kullanıcıya sunulabilir.

2.2. Endüstri 4.0 Bileşenleri

Endüstri 4.0 kavramı bir ürünün ortaya çıkabilmesi için uçtan uca birçok teknoloji ile entegre çalışan sistemleri barındırmaktadır. Şekil 2.2'de Endüstri 4.0 bileşenleri görülmektedir.



Şekil 2.2. Endüstri 4.0 bileşenleri

2.2.1. Nesnelerin İnterneti

Nesnelerin İnterneti kavramı internet teknolojilerinin yaygınlaşması ile birlikte kablosuz ağ altyapıları, mikro elektromekanik sistemler, mikro servisler gibi teknolojiler ve sensör teknolojilerinin gelişmesi ile birlikte ortaya atılmış bir kavramdır. 1970'lerin başından beri kavramsal olarak karşımıza çıkan IoT kavramı ev, endüstriyel alanlar, arabalar, akıllı şehircilik uygulamaları, akıllı taşımacılık uygulamaları ve robotik sistemlerdeki gelişimler ile birlikte son yıllarda yaygınlaşmaya başlamıştır. IoT, M2M yani makineler arası iletişimi geliştirmiştir. İnsanlara ihtiyaç duymadan böyle bir ağ kurmak nesnelerin internetiyle mümkün hale gelmiştir [5].

Günümüzde elektrikli hemen hemen her nesne internete bağlanabilir ve uzaktan yönetilebilir hale gelmiştir. Bu bağlamda özellikle Endüstri 4.0 kapsamında endüstriyel fabrikalardaki üretim bantları ve bu bantlar ile ilintili tüm donanımlar internet tabanlı uzaktan kontrol altyapısına kavuşmuştur. Bu sayede IoT tabanlı sistemler ile donatılmış fabrika içindeki her anlık aktivite arak takip edilebilmekte ve sensörlerden alınan veriler anlık olarak değerlendirilebilmektedir.

Nesnelerin interneti, birbirinden farklı nesnelerin doğrudan internete bağlandığı bir yaklaşımı tanımlayan bir kavramdır. Bu, çevredeki ortamı algılayabilen, elde edilen verileri iletebilen ve işleyebilen ve daha sonra çevre ile etkileşim için akıllı aktüatörleri kullanabilen akıllı sensörlerin konumlandırılmasına olanak tanır. IoT, birbiriyle ilişkili cihazlar ile sağlanan insanlara ve insandan ağa gerek duymadan bir ağ üzerinden veri aktarabilen sistemlerin tümüne verilen addır. Giderek artan bir şekilde, çeşitli endüstrilerdeki kuruluşlar daha verimli çalışmak, daha iyi müşteri hizmeti sunmak, karar vermeyi geliştirmek ve işin değerini artırmak, müşterileri daha iyi anlamak için IoT teknolojisini kullanırlar. Örneğin bir akıllı araba gidilen mesafeyi ve bu aktiviteleri yaparken aracın düzgün çalışıp çalışmadığını algılayan sensörlere sahiptir. Burada toplanan bu veriler bir istemci (bilgisayar, cep telefonu vb.) tarafından takip edilip hayatı kolaylaştırmada yardımcı olmaktadır. İşte bu iki cihaz arasındaki iletişim, nesnelerin internetini oluşturan en temel özelliktir [9].

Nesnelerin interneti günümüzde birçok alanda kullanılmaktadır. Akıllı ev, sağlık uygulamaları, akıllı hayvancılık uygulaması, akıllı çevre, hasta takip sistemi, lojistik giyilebilir uygulama, araç takip sistemleri, akıllı su, akıllı tarım, akıllı enerji, akıllı şehirler, akıllı ölçüm, endüstriyel kontrol, güvenlik ve acil durumlar, hasta takip sistemi, alışveriş, lojistik, giyilebilir uygulama, araç takip sistemleri gibi uygulamalarda kullanılır. Bu alanlarda daha kaliteli hizmet vermek, verimliliği ve üretkenliği arttırmak için sensörlerden bu veriler toplanır. Bu veriler “Büyük Veriyi” oluşturarak “Bulut Bilişim” sistemlerinde depolanır. “Makine Öğrenimi” yöntemleriyle analiz edilirler ve ilgili iyileştirmelerin yapılmasına katkı sağlarlar.

IoT altyapısı endüstride de kendine karşılık bulmaktadır. Endüstriyel Kontrol Sistemleri kritik altyapıdaki verinin üretilmesinin, işlenmesinin ve denetlenmesinin izlenmesi gibi işlemlerin bilgisayarlar, sensörler ve haberleşme cihazları kullanarak adım adım takibini oluşturulan

sistemlerdir. ICS, standart gömülü sistem platformlarına dayanır ve genellikle ticari olmayan yazılımları kullanılır. İyi bilinen ICS türleri arasında denetimsel kontrol ve veri toplama (SCADA) sistemleri, dağıtılmış kontrol sistemleri (DCS) ve programlanabilir lojik kontrolörler (PLC) bulunur. SCADA sistemleri tarihsel olarak, ICS sistemlerinin en büyük alt grubu ve birden çok yeri ve büyük mesafeleri içerebilen büyük ölçekli süreçler olarak kendilerini diğer ICS'den ayırırlar.

IoT ağlar farklı amaçlarla kullanıldıkları için büyüklüklerine göre farklı büyüklüklerde altyapıları mevcuttur. Buna göre IoT ağlar aşağıdaki şekilde sınıflandırılmaktadırlar.

- Nano
- NFC (Near-Field Communication)
- BAN (Body Area Network)
- PAN (Personal Area Network)
- CAN (Campus/Corporate Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

Bahsi geçen IoT ağlar algılama, hesaplama, depolama ve harekete geçirme gibi özellikler ve işlevselliklerine göre gerçekleştirilecek olan sistem için tercih edilmektedirler [10].

IoT sistem protokolleri incelendiğinde de farklı birçok protokol karşımıza çıkmaktadır. Buna göre yaygın olarak kullanılan IoT protokollerinden bazıları aşağıdaki gibidir.

- Message Queuing Telemetry Transport (MQTT)
- Bluetooth
- Bluetooth LE
- ZigBee
- Z-Wave

Burada kullanılan protokollerde sistemin altyapısına büyüklüğüne ve isterlere bağlı olarak tercih edilmektedir.

IoT kavramı endüstri 4.0 bileşenleri içindeki en önemli bileşenlerdendir. Bir endüstriyel sistemde sensörlerden ve fabrika içindeki uçtan uca donanımsal aygıtlardan bilgi toplanıp anlamlandırılması sürecinin ilk aşaması IoT tabanlı bileşenler ile fabrikanın donatılmasıdır. Bu sayede IoT ile uçtan uca verilerin toplanması, değerlendirilmesi, yönetilmesi ve anlık kararların verilebilmesi mümkün hale gelebilmektedir. Üretim esnasında yaşanan olağan ya da olağan olmayan tüm hareketler incelenebildiğinden anomalinin tespiti ve anlık müdahale kolaylaşacaktır. IoT kullanımı sayesinde üretim tesisindeki makineler bir biri ile haberleşerek ortak hareket edebilme yeteneği kazanılacaktır. Bu da verimin artmasının yanında önemli bir enerji tasarrufunu da beraberinde getirmektedir

2.2.2. Büyük Veri ve Analitik

Büyük veri kavramı için literatürde 42V (V harfi ile başlayan kelimeler ile anlamlandırılan büyük veri tanımı) ile temsil edilen çalışmalar olsa da kabul görmüş 5V ile temsil edilmektedir. Buna göre büyük veriyi tanımlayabilmek için Volume (Hacim), Velocity (Hız), Variety (Çeşitli Tipteki Veriler), Viability (Doğruluk), Value (Anlam katan katma değerli bilgi) özelliklerinin olması gerekmektedir. Bir başka deyişle verinin hızla büyüyen bir hacme sahip olması, ses, görüntü, metin tabanlı içerikler ya da sensör verileri gibi farklı tiplerdeki verileri barındırması, hızla büyümesi, doğruluğu kesin olan veriler olması ve bu verilerden yapılacak gerçek zamanlı analiz sonuçlarından anlamlı ve katma değer yaratacak bilgiler üreten yapılar olarak nitelendirilmektedir.

2.2.3. Eklemeli Üretim

Bir malzemenin seri üretime geçmeden önce Ar-Ge faaliyetleri geçirmesi gerekmektedir. Bu süreçte kalıp maliyetleri çok yüksek olduğundan ürünün 3D yazıcı ile parçalarının üretilmesi çözüm olabilmektedir. Keza eklemeli üretim kavramı da bu ihtiyaç sonrası üretim maliyetlerinin düşmesi ve 3D yazıcılar ile üretilen malzemenin eskisinden çok daha kaliteli sonuçlar vermesi nedeni ile endüstriyel alandaki üretim süreçlerinin arasında yerini almıştır. Bu bağlamda eklemeli üretim nihai bir ürünün çeşitli malzemelerden 3D olarak üretilmesine verilen addır. Endüstriyel alanlarda özellikle basit iş süreçlerinde eklemeli üretim tercih edilmektedir [11].

2.2.4. Bulut Bilişim

Endüstriyel alanlarda bulut bilişimin kullanımı incelendiğinde genel olarak iki farklı bulut bilişim ağı göze çarpmaktadır. Bunlardan birincisi endüstriyel verinin fabrika içinde güvenli bir şekilde tutulduğu ve fabrika içindeki makinelerin bu ağ üzerinde verilerini bulut altyapısı üzerinden paylaştığı yapıdır. Bu sistemde sadece üretim tesisindeki endüstriyel veri bulunmaktadır. Bu veri merkezi sunuculara gönderilmekte ve makineler bir biri ile bu ağ üzerinden konuşabilmektedir. İkinci bulut bilişim ağı ise ürün siparişinden ürünün teslimatına kadar ki tüm süreçlerde kullanılan uygulama ve otomasyonların bulunduğu ve yöneticilerinde karar alabildiği verileri barındıran ağıdır. Genellikle bahsi geçen iki ağ güvenlik kaygıları ile birbirlerinden ayrıştırılmıştır. Ancak son yıllarda modern Endüstri 4.0 altyapılı sistemlerde bütünleşik bulut bilişim çözümlerinin ön plana çıktığı görülmektedir. Fabrika ister dağıtık ister de bütünleşik bir bulut bilişim altyapısı kullansın, önemli olan üretilen büyük verinin anlık analizi ve çok hızlı reaksiyon gösterilebilmesidir [9]

2.2.5. Gelişmiş Robotlar

Akıllı fabrikaların temelinde üretim süreçlerinde insan faktörünün en aza indirilmesi fikri yatmaktadır. Bunun içinde üretim aşamasında uçtan uca robotik teknolojilerinden

yararlanılmaktadır. Fabrika içindeki üretim bantları, bu bantlarda kullanılan robotik kollar ve akıllı depo sistemlerinden lojistik sistemlere kadar tüm alanlarda robotik sistemler kullanılmaktadır. Uçtan uca sürecin yönetilmesi için bu robotik sistemlerin birbirleri ile anlık haberleşmesi gerekmektedir. Bu sayede hem üretim verimi hem de üretilen nihai ürünün kalitesi artacaktır [8].

2.2.6. Benzetim

Üretim sürecinin ilk basamağı Ar-Ge sürecidir. Yüksek teknolojik bir ürünün seri üretime geçmesi için ürün ile ilgili tüm testlerin yapılmış olması ve her türlü koşulda deneysel çalışmaların tamamlanmış olması gerekmektedir. Bunu fiziksel olarak gerçekleştirmek her zaman mümkün olamamaktadır. Bu yüzden ürünün geliştirilme süreçlerinde benzetim teknolojilerinden yararlanılmaktadır. Benzetim sonucu ortaya çıkan ürün başarımını kanıtladığında bu benzetim gerçek bir modele ve seri üretime dönüştürülmektedir.

2.2.7. Yapay Zeka

Yapay zeka, bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri insansı faaliyetlere benzer şekilde yerine getirme kabiliyeti olarak tanımlanmaktadır. Yapay zeka çalışmaları genellikle insanın düşünme yöntemlerini analiz ederek bunların benzeri yapay yönergeleri geliştirmeye yöneliktir. Bu bağlamda fabrikalarda yapay zeka teknikleri üretim süreçlerinin hemen hemen her noktasında kullanılmaktadır. Örneğin üretimden çıkan ürünün defolu olup olmadığını kontrol eden derin öğrenme tabanlı bir karar destek sistemi verilebilir.

2.2.8. Artırılmış Gerçeklik

Artırılmış gerçeklik (Augmented Reality - AR) endüstriyel alanlarda özellikle anlık verinin görselleştirilerek makine parkurunda canlı olarak izlenebilmesi ya da uzaktan bakım süreçlerini planlamak gibi birçok alanda kullanılmaktadır. Artırılmış gerçeklik gerçek dünyadaki görüntünün üzerine ses, görüntü ve konum verileri gibi birçok bileşen ile zenginleştirilmesi işlemini gerçekleştirmektedir. Bu sayede gerçek dünya ile üretilen veri ve sayısal değerlerin aynı anda incelenmesi mümkün olabilmektedir [9].

2.2.9. Yatay ve Dikey Entegrasyon

Geleneksel üretim teknolojilerinde bir ürün genellikle bir tesiste tamamen üretilirken günümüzde bütünün parçaları farklı lokasyonlardaki farklı tesislerde üretilmektedir. Bu üretim sürecinde tesislerin senkronize olabilmesi ve lojistik süreçlerinin yönetilebilmesi gibi birçok problemin çözülmesi gerekmektedir. Ancak bu problemlerin çözümü rekabet edebilir nihai ürünlerin ortaya çıkmasına neden olmaktadır. Bu bağlamda Endüstri 4.0 tabanlı üretimde yatay ve

dikey olarak firmaların entegre olma anlayışı hakim olmaktadır. Bu sayede işletmeler uzmanlıklarını konu odaklı geliştirebilmenin yanında risk paylaşımı yapabilmek, birlikte büyüebilme ivmesini arttırabilmek ve rekabetçi ücretler ortaya çıkarmış olmaktadır.

Yatay entegrasyon işletmelerin aynı amaç doğrultusunda birleşmelerini esas alırken dikey entegrasyon alt sektörlerin ya da alt yüklenicilerin bir işe ait parçaları üretmesini ve bütünün dikey çözümler ile oluşturulmasını baz almaktadır.

2.2.10. Siber Fiziksel Sistemler

Endüstri 4.0 kapsamında anlık ve hızla büyüyen bir veri kümesi bulunmaktadır. Yine aynı şekilde hızla büyüyen verinin anlık olarak ilgili noktalara yayılması gerekmektedir. Bu da çeşitli güvenlik zafiyetlerinin ortaya çıkmasına neden olabilmektedir. Bilginin bütünlüğünün ve gizliliğinin korunması Endüstri 4.0 tabanlı işletmelerin en önemli sorunlarından biridir. Eğer verinin bütünlüğü ya da gizliliği sağlanamaz ise işletme büyük zararlara uğrayabilir. Bu yüzden Siber-fiziksel sistemlerin korunabilmesi gerekmektedir. Bu sistemleri koruyabilmenin ilk adımı anlık gözlemleme yapabilmek ve gözlenen veri üzerinden anomali tespiti yaparak olağan dışı durumlara anlık müdahale edebilmektir [11].

2.3. Büyük Veri Mimarisi ve Veri Platformları

Büyük verinin saklanması, işlenmesi ve istenildiği takdirde arama sonuçlarının çok hızlı bir şekilde getirilmesi işlemi klasik veritabanı yönetim sistemleri ve klasik kurgular ile gerçekleştirilememektedir. Bu yüzden NoSQL (Not-Only SQL) veritabanı sistemleri olarak da adlandırılan sistemlere ihtiyaç duyulmaktadır. İhtiyaçların artması, büyük verinin hızla oluşması, sistemlerin çok daha fazla veri biriktirmesi gibi nedenleri ile NoSQL sistemler çok daha hızlı bir şekilde büyük veri uygulamaları için bir çözüm haline gelmiştir. Birçok NoSQL çözümü içerisinde en iyi NoSQL sistemini kurgulayabilmek için belirli parametreler ve belli ölçeklendirmeler ile kullanılacak olan platform seçilebilir. Kullanılacak alandaki üretilen verinin düşük veya yüksek olması, anlık olan sorgu ihtiyacı, bağlanılacak diğer platformlar ile uyum gibi birçok kritere göre farklı NoSQL çözümlerinin güçlü ve zayıf yönlerinin iyi analiz edilmesi gerekmektedir.

Gün geçtikçe teknolojik çözümlerin artması, ağ üzerindeki cihaz sayısının ve çeşitliliğinin büyümesi, IoT tabanlı bağlanılan uç sayısının giderek artması ve toplanan veri miktarının büyümesi, anlık veri analizine ihtiyaç duyulması, lokasyon bağımsız dağıtık bir sistemdeki tüm verinin birbiri ile korelasyon içinde hareket edebilme gerekliliği gibi nedenlerden ötürü depolanan veriler için NoSQL platformlar tercih edilmektedir. Bu değişim ve ihtiyaçların çeşitliliğinin artması ile sensör ağlardan gelen bilgiler, sosyal medya platformlarından toplanan veriler, dijital arşivler

ya da işletmenin kurumsal hafızasını oluşturan eski tüm verileri, ses ve görüntü gibi multimedya verileri, web ve mobil tabanlı uygulamaların verileri, işletmenin uçtan uca çözüm için kullandığı tüm otomasyon, sistem ya da makine – teçhizatın toplanan veriler büyük veriyi oluşturmaktadır. Hızla büyüyen verinin saklanması, işlenmesi ve yönetilmesi için altyapının çok iyi planlanması ve kullanılacak olan platformların ihtiyaca uygun bir şekilde seçilmesi gerekmektedir. Ortaya konulan bu gereklilikler ve ihtiyaçlar ışığında klasik ilişkisel veritabanı yönetim sistemleri yerine NoSQL sistemler tercih edilmektedir. NoSQL platformlar ve bileşenleri aşağıdaki gibidir [12].

2.3.1. Hadoop

Hadoop, Java altyapısı ile geliştirilmiş ve günümüzde sıklıkla kullanılan bir büyük veri çözümüdür. Hadoop veri kümelerinin paralel olarak işlenebilmesine ve farklı veri kaynaklarından gelen verinin anlık olarak işlenebilmesine imkan sağlayabilmekte olan açık kaynak kodlu bir büyük veri çözümüdür. Bireysel olarak bir PC’de ya da çok daha büyük ölçeklerde dağıtık mimari altyapısı ile ölçeklenebilen bir büyük veri analiz platformudur.

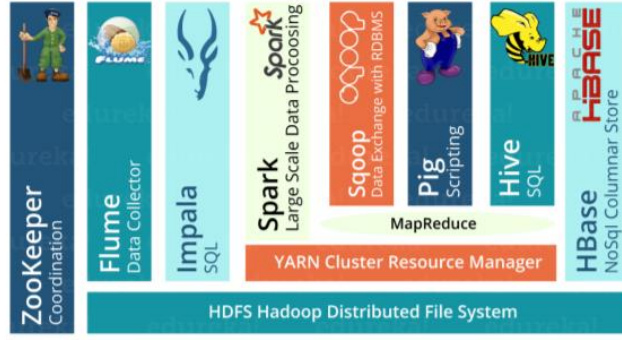
HDFS olarak ta adlandırılan kendisine özel bir dağıtık dosya sistemi bulunmaktadır. Yine MapReduce olarak adlandırılan çözüm ile farklı platformlarda çalışan uygulamaları bir araya getirilebilmektedir.

Apache Hadoop kendine özgü çözümler ile farklı sunucular üzerinde bulunan dağıtık ve canlı olarak akan verilerin analizine imkan tanımaktadır. Bir başka deyişle Hadoop bir kişisel bilgisayardan binlerce sunucuya kadar büyüeyebilen dağıtık yapıdaki büyük altyapılara destek veren ve buradaki büyük verinin anlık olarak analizini yapabilen bir büyük veri platformudur.

Hadoop'un temel özellikleri aşağıdaki gibidir.

- Düşük maliyet,
- Ölçeklenebilirlik,
- Dağıtık verilerin paralel olarak işleyebilme,
- Veri optimizasyonu,
- Veri yükünü dengeleme ve otomatik yük devredebilme.

Hadoop, HDFS ve MapReduce gibi ana bileşenlerden ve birçok alt bileşenden oluşmaktadır. Şekil 2.3’de Hadoop Bileşenleri görülmektedir [9-13].



Şekil 2.3. Hadoop bileşenleri

- **HDFS**

Hadoop'un kendine özgü dosya sisteminin adıdır. HDFS dosya sistemi bir bilgisayar ya da sunucudan binlerce sunucuya kadar büyüyeblen dağıtık sistemlerin kurgulanabilmesine imkan tanımaktadır. HDFS dosya sisteminde büyük boyutlu dosyalar bloklar halinde ayrılarak üç kopyası oluşturulur ve farklı sunucularda dağıtık mimari oluşturulur. Böylece dağıtık mimari oluşturmanın yanı sıra yedekleme ve veri kaybının önüne geçmek adına da çözüm sunulmuş olur [13].

- **MapReduce**

Hadoop, Reduce ve Map'lerden oluşan iş parçacıklarını küme üzerinde dağıtarak aynı anda işlenmesini ve bu işler sonucunda oluşan verilerin tekrar bir araya getirilmesinden sorumludur. HDFS dosya sistemindeki büyük dosya verilerinin işlenmesi için MapReduce fonksiyonları kullanılmaktadır. Büyük verinin analiz edilebilmesi ve verinin süzülebilmesi için Map fonksiyonu, bu veriler üzerinden anlık sonuç üretebilmek için ise Reduce fonksiyonu kullanılmaktadır [14].

- **Apache Spark**

Hadoop verileri saklamak için HDFS dosya sistemini kullanmaktadır. Dağıtık bir dosya sistemi olan HDFS ile büyük veri kolaylıkla istenilen platformlarda saklanabilmektedir. Büyük veri analizinin ikinci aşaması saklanan veri üzerinde analizler yapabilmek ya da çeşitli filtreler uygulayarak veri üzerinden çeşitli raporlar oluşturabilmektir. Bunun için Hadoop altyapısında Map & Reduce çözümü bulunmaktadır. Ancak Map & Reduce basit analizler yapabilmek için kurgulanmıştır. Map & Reduce, HDFS dosya sisteminden veriyi okuduktan sonra sorgulama ve analiz işlemlerini gerçekleştirmekte ve sonrasında ise çıkan sonuçları tekrar HDFS dosya formatında yazmaktadır. Bu da yazma ve okuma işlemlerinin yoğunluğunu arttırmakta ve büyük analizlerde sistemi hantallaştırmaktadır. Bu yüzden Spark sistemi çözüm olarak sunulmuştur. Spark çok daha büyük veri kümelerini dağıtık bir yapıda ve çok hızlı bir şekilde inceleyerek analiz edebilmektedir. Spark veri analizi ile birlikte makine öğrenmesi altyapısını da bünyesinde barındırmaktadır. Spark Machine Learning(MLlib) kütüphanesi ile makine öğrenmesi algoritmalarının çalışmasına olanak sağlanmaktadır. Yine aynı şekilde grafik işleyebilmek için

GraphX kütüphanesi bulunmaktadır. Spark üzerinde R, E, Scala, Java, Python gibi birçok dilde geliştirme yapılabilmektedir [15].

- **RDD (Resilient Distributed Datasets)**

Spark işlemleri için verilerin verisetleri halinde dağıtılması gerekmektedir. Bu işlem için RDD kullanılmaktadır. Spark, HDFS dosya sistemi ile birlikte kullanılabileceği gibi Cassandra, MongoDB, HBase, MapR-DB, ve Amazon S3 gibi farklı dağıtık veri saklama çözümleri ile birlikte de kullanılabilen bağımsız bir çözüm olarak karşımıza çıkmaktadır. RDD, HDFS dosya sistemi ya da diğer çözümlerde saklanan verinin belleğe depolanarak işlem yapılabilmesini sağlamaktadır. RDD sayesinde dağıtık olarak diskler üzerinde saklanan büyük veri bellekler üzerine alınarak çok hızlı bir şekilde işlem yapılabilmesini sağlamaktadır [16].

- **Apache Pig**

Pig yüksek seviyeli bir dil çözümüdür. Uygulama geliştiricilerin ya da analistlerin kod yazma gereksinimi duymadan HDFS dosya sistemi üzerinde saklanan veriyi çıkarma, kullanma ve dönüştürülmesine olanak sağlayan çözüm olarak karşımıza çıkmaktadır.

- **Apache Hive**

Apache Hive temel anlamda SQL sorgu diline benzerlik göstermektedir. Apache Hive elimizdeki büyük veri üzerinden yapılacak sorgu işlemleri ile verilerin özetlenmesine olanak tanıyan, sorgulama ve analiz işlemlerinin kolaylıkla yapılabilmesini sağlayan veri ambarı çözümüdür.

- **Ambari**

Ambari web tabanlı kullanılan bir test çözümüdür. Sadece HDFS dosya sistemi ile kurgulanmış çözümler için değil MapReduce, Hive , HCatalog, HBase, ZooKeeper, Oozie, Pig ve Sqoop gibi altyapılara da destek veren büyük veri kümelerinin yönetilmesi ve rest işlemlerinin yapıldığı platformdur. Ambari aynı zamanda veri üzerinde performans analiz de yapabilmektedir. Yapmış olduğu analizleri göstermek ve yönetim işlemleri için web tabanlı görsel arayüzleri bulunmaktadır.

- **Cassandra**

Cassandra, dağıtık mimaride ve çok sunuculu sistemlerde kullanılan açık kaynaklı bir NoSQL çözümüdür. Çok büyük miktardaki verinin işlenmesi yönetilebilmesi ve ölçeklendirilmesine imkan tanıyan bir platformdur. Büyük veri çözümlerinde sıklıkla kullanılmaktadır.

- **Flume**

Büyük ve dağıtık bir mimaride anlık veri akışı çok büyük boyutlara çıkabilmektedir. Bu verileri geleneksel yöntemlerle toplamak gecikmelere ve kayıplara neden olabilmektedir. Akan verinin anlık olarak toplanıp HDFS dosya sistemine aktarılması için Flume çözümü kullanılmaktadır.

- **HBase**

Hbase, büyük veri altyapısında kullanılan ilişkisel olmayan veritabanı olarak tanımlanmaktadır. Hbase özellikle Map&Reduce hizmeti için servis altyapısı sağlamaktadır.

- **HCatalog**

Büyük veri mimarisi ile geliştirilen sistemler canlı sistemler olduğu için, yeni ihtiyaçlar olduğunda sistem üzerinde anlık geliştirme yapılabilmesi gerekmektedir. Bu yüzden geliştiricilerin verilere erişebilmesi gerekmektedir. HCatalog uygulama geliştiricilerin veriye belli yetkiler çerçevesinde erişebilmesini ve yönetebilmesini sağlayan bir ara katman çözümüdür.

- **Hadoop Sqoop**

Büyük verinin bir noktadan başka bir noktaya aktarılması ya da yeni bir sisteme taşınabilmesi oldukça zahmetlidir. Hadoop Sqoop, veritabanları arasında bu büyük verinin aktarılması için gerekli olan altyapıyı sağlayan platformdur.

- **Hadoop Zookeeper**

Büyük veri genellikle dağıtık mimaride çalışmaktadır. Dağıtık mimaride farklı sunucularda farklı servisler çalışacağı gibi oluşan büyük veri farklı sunucularda da tutulabilir. Aynı zamanda sistem ve verileri canlı yedekleme çözümleri de (Replikasyon) bu dağıtık mimari içinde yer almaktadır. Hadoop Zookeeper dağıtık mimarideki sistemleri ya da verileri senkronize edebilmek için kullanılan bir çözümdür.

2.3.2. Apache Kafka

Apache Kafka da Hadoop gibi büyük veri çözümü sunan bir platformdur. Kafka, Apache altyapısı üzerine oluşturulmuştur. Büyük verinin sistemler üzerinden anlık ve hızlı bir şekilde toplanmasına ve dağıtık mimari altyapısındaki ilgili sunuculara bu verilerin aktarılmasını sağlayan bir çözümdür. LinkedIn firması tarafından Java altyapısı kullanılarak geliştirilen Kafka, Apache platformu üzerinde açık kaynak bir altyapıya dönüşmüştür. Hali hazırda LinkedIn ile birlikte Twitter, Uber, Netflix gibi büyük veri üreten ve kullanan birçok firma Kafka çözümünü kullanmaktadır.

Kafka, büyük verinin temelini teşkil eden klasik veritabanlarından NoSQL sistemlere kadar birçok teknolojik altyapı ile sorunsuz çalışabilmektedir. Büyük verinin hareketi protokol ve sistemlerden bağımsız yani bir başka deyişle platform bağımsız olarak sorunsuz bir şekilde gerçek zamanlı ve çok hızlı gerçekleştirebilmektedir [17].

2.3.3. MongoDB

MongoDB klasik veritabanında tutulan verileri NoSQL verilerine dönüştürerek doküman şeklinde saklayarak kullanıcılarına sunmaktadır. Bu yüzden MongoDB geliştirme ve ölçekleme

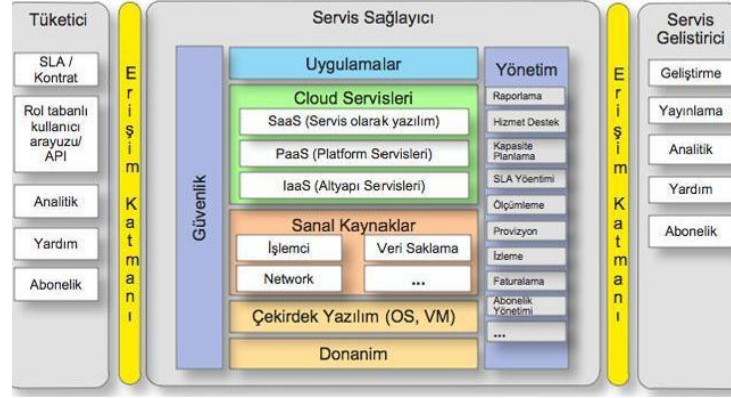
açısından önemli kolaylıklar sağlamaktadır. MongoDB’de yukarıda anlatılan diğer platformlar gibi açık kaynaklı bir çözüm olarak karşımıza çıkmaktadır. MongoDB klasik veritabanlarında tablolara satır bazında kaydedilen verileri doküman haline dönüştürerek kaydetmektedir. Bu özellikle arama algoritmalarının işini kolaylaştırmakta ve dosyaya çok hızlı erişim sağlanabilmektedir MongoDB ile oluşturulan kayıtlar JSON (JavaScript Object Notation) formatındadır. Bazı durumlarda MongoDB JSON yerine Binary JSON olarak ta adlandırılan BSON formatında saklanmaktadır [18].

2.3.4. Elasticsearch

Elasticsearch, Java dilinde yazılmış açık kaynak kodlu bir platformdur. Asıl amacı büyük veri içinde arama yapabilmektir. Bu bağlamda geliştirilmiş olan Elasticsearch bir arama motorudur ve birçok büyük kurum tarafından sistem altyapılarında kullanılmaktadır. Gerçek zamanlı performans gösteren Elasticsearch metin tabanlı arama yapabildiği gibi veri analizi içinde kullanılmaktadır. Dağıtık ve ölçeklenebilir bir yapısı vardır. Elasticsearch’i diğer platformlara göre öne çıkarak en önemli özelliği doğrudan arama yapmak yerine yapmış olduğu indekslemeler üzerinden arama işlemlerini gerçekleştirmektir. Bu tüm dosya ya da tüm büyük veri yerine öncelikle oluşturulan indeksin aranması sayesinde istenilen veriye çok daha kısa sürede erişilebilmesini sağlamaktadır. Yine Elasticsearch yapmış olduğu sorgular ve çıkan sonuçlar üzerinden istatistiki analizler ve sonuçlar ortaya koymaktadır.

2.4. Bulut Bilişim Mimarisi

Bulut bilişim kavramı ile birlikte bir işletmenin teknolojik altyapısını oluşturan donanım ve yazılım öğeleri ile birlikte kurumsal hafızanın oluştuğu verilerin depolanacağı birimler için yapılması gereken yatırım maliyetleri minimize edilmiştir. Bulut bilişim mimarilerinin kurulması maliyet etkin yaklaşım ile birlikte hız, esneklik, ölçeklenebilirlik, performans ve güvenlik gibi birçok avantajı beraberinde getirmiştir. Kurulan mimariye erişimin lokasyon bağımsız şekilde gerçekleştirilmesi ile ekiplerin uzak noktalardan birlikte çalışabilme olanağını doğurmuştur. Bulut bilişim mimari yapısı Şekil 2.4’de görülmektedir.



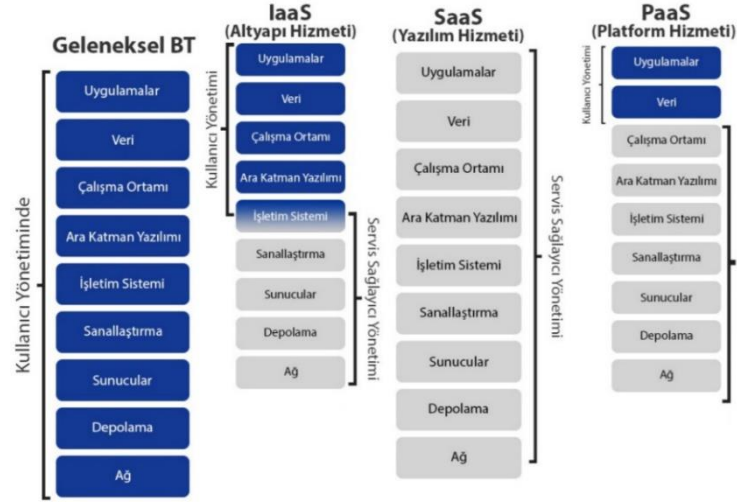
Şekil 2.4. Bulut tabanlı uygulamalar [19]

Bilişim altyapısında gerekli olan periyodik yapılandırmalar, güncelleştirme ya da yedekleme gibi ihtiyaçlar otomatize edilmiştir. Bulut tabanlı mimarilerin merkezi ya da dağıtık mimaride kurgulanabilme kabiliyetleri sayesinde operasyonel sürece uygun topolojilerin oluşturulmak kolaylaşmıştır. Bu sayede teknolojinin gelişimi ile birlikte doğan yeni ihtiyaçlar etkin bir şekilde çözümlenebilir hale gelmiştir [19].

2.4.1. Bulut Bilişim Servisleri

Bulut bilişim mimari yapısını geleneksel bilişim teknoloji servislerinden farklı olarak Hizmet Olarak Altyapı (IaaS), Hizmet Olarak Platform (PaaS) ve Hizmet Olarak Yazılım (SaaS) olmak üzere üç bulut bilişim servisi oluşturmaktadır. Şekil 2.5’de geleneksel bilişim teknoloji servisleri ile bulut bilişim servislerinin karşılaştırılması gösterilmiştir.

Geleneksel yapıda bilişim teknoloji altyapısında ihtiyaç duyulan ağ, depolama, sunucular, sanallaştırma, işletim sistemi, ara katman yazılımları, çalışma ortamları, veri ve uygulamaların tamamı kullanıcının yönetimindedir. Bu yaklaşımda ihtiyaç duyulan hizmete göre yatırımdan yapılandırmaya, yönetimden güvenliğe kadar tüm çözümlerin yerinde sunulması gerekmektedir. Bulut bilişim teknolojilerinde ise ihtiyaca uygun olarak mimari yapı esnek bir şekilde kurgulanabilmekte, altyapının önemli bir kısmı servis sağlayıcının görev, yetki ve sorumluluğunda, diğer kısmı ise kullanıcı yönetimindedir. Bu sayede kullanılan teknolojinin sevk ve idaresi hem maliyet etkin bir şekilde çözülmekte hem de son kullanıcının yetkileri kontrol altına alınabilmektedir [20].



Şekil 2.5. Bulut tabanlı uygulamalar [20]

- **Hizmet Olarak Altyapı (IaaS - Infrastructure as a Service):** IaaS bulut teknolojilerindeki temel hizmetlerden biri olarak karşımıza çıkmaktadır. Bu hizmet türünde servis sağlayıcı kullanıcılarına temel bilgi işlem altyapı hizmeti, ağ hizmeti, depolama hizmeti, sunucu ya da temel donanım hizmeti ve sanallaştırma hizmeti vermektedir. Bu sayede işletmelerin BT (Bilişim Teknolojileri) altyapı yatırımları minimize edilebilmektedir. IaaS çözümü uygulama geliştirme noktasından incelendiğinde ise web ve mobil yazılımların geliştirme ve test süreçlerinde, yüksek performansa dayalı hesaplama ya da analiz süreçlerinde kullanımının yaygın olduğu görülmektedir.
- **Hizmet Olarak Platform (PaaS - Platform as a Service):** PaaS, bulut teknolojilerinde IaaS'ın sağladıklarının yanı sıra başta işletim sistemi olmak üzere ara katman yazılımları, talep edilen uygulamalar, bulut yönetim uygulamaları, çeşitli yazılım ve donanım çözümleri, depolama ve yedekleme çözümleri, uygulama geliştirme araçları, iş zekası uygulamaları, veritabanı yönetim sistemleri ve uçtan uca güvenlik çözümleri sunmaktadır. Bu bağlamda PaaS uygulama geliştirmek için gerekli tüm altyapıyı sunmaktadır. PaaS, uygulama geliştirmenin yanında uygulamanın barındırılması, 7/24 kesintisiz hizmetin verilebilmesi, geliştirilen uygulamaya yönelik test ortamlarının oluşturulması, bakım süreçlerinin yönetilmesi gibi çözümler de sunarak yazılım yaşam döngüsünün tamamını kapsar.
- **Yazılım Olarak Altyapı (SaaS - Software as a Service):** SaaS ise yukarıdaki hizmetlere ek olarak kullanıcılara sunulacak yazılım hizmetlerini kapsamaktadır. SaaS hizmeti kullanıcıların uygulamalara internet üzerinden erişimini ve kullanımını mümkün kılmaktadır. E-Posta hizmeti, çeşitli ofis uygulamaları, ortak takvim araçları, doküman yönetim sistemleri, belge yönetim sistemleri, ERP (Kurumsal Kaynak Planlama) ya da CRM (Müşteri İlişkileri Yönetimi) gibi yazılımları satın almadan kiralama modeli ile

kullanımı SaaS'ın sunmuş olduđu çözümleri arasındadır. Hizmet kullandığın kadar ödeme yöntemi sunduğundan maliyet etkindir. Aynı zamanda uygulamaların platform bağımsız şekilde web ve mobil tabanlı cihazlar üzerinden erişiminin mümkün olması, sürekli olarak güncel versiyonlarının hizmete sunulması, uygulamalara ve içeriğe her yerden güvenli erişimin sağlanabilmesi SaaS hizmetlerinin avantajları arasında sayılabilir.

2.4.2. Bulut Bilişim Dağıtım Modelleri

Bulut bilişim kullanımının sunduğu avantajların yanında bilgi güvenliği, mahremiyet gibi taleplerden dolayı çekinceleri de beraberinde getirmektedir. Bu sorunları yok etmek için bulut bilişim teknolojileri farklı dağıtım modelleri ile kullanılmaktadır. Genel Bulut Çözümleri, Özel Bulut Çözümleri ve Hibrit Bulut Çözümleri olmak üzere üç ayrı kategoride bu modelleri incelemek mümkündür.

- **Genel Bulut Modeli (Public Cloud Model):** Genel bulut modelinde hizmeti veren servis sağlayıcı, altyapının oluşturulmasından yönetilmesine, sistem kaynaklarından ihtiyaç duyulan bant genişliğine kadar ki tüm süreçten sorumludur. Bulut sistemini kullanan kurum ya da kişiler ise hazır olan altyapının üzerine inşa edecekleri sistemden ve bu sistemin yönetilmesinden sorumlu olmaktadır. Genellikle büyük işletmeler özellikle uygulama geliştirme alanındaki süreçlerini genel bulut modellerinde kurgulamaktadır. Genel bulut modelinde son kullanıcılar lokasyon bağımsız bir şekilde internete bağlanabildikleri her platformdan bir web tarayıcısı yardımı ile bağlanabilmektedir. Genel bulut modeli ile özellikle kaynakların etkin kullanımı ve iş süreçlerindeki verimliliğin artışı hedeflenmiştir.
- **Özel Bulut Modeli (Private Cloud Model):** Genel bulut modelinde kaynakların yönetimi servis sağlayıcıdadır ve bu bazı çekinceleri de beraberinde getirmektedir. Kritik bilginin olduğu iş süreçlerinde bu bilginin kurum dışındaki ve yönetimi üçüncü şahıslarda olan altyapılarda tutulması sakıncalı olabilmektedir. Nitekim ülkemizde de Resmi Gazetede yayımlanan genelge ile kamu kurum ve kuruluşlarının verilerini ve sistemlerini genel bulut hizmetlerinde saklamaları yasaklanmıştır. Kurum ya da kişilerin bulut sistemlerini kendi lokal ağları içerisinde oluşturmaları ve uçtan uca tüm hizmetin kendi kontrolünde olduğu sistemlere özel bulut modeli adı verilmektedir. Bu modelde ihtiyaç duyulan bilişim altyapısı, veri depolama ve yedekleme birimleri, sistem kaynakları üzerindeki yazılımsal yapılandırmalar, bakım ve destek süreçleri ile ilgili tüm akış, alınması gereken güvenlik önlemleri, sistem için yapılacak ek yatırımlar ve sistemin uçtan uca idame ettirilmesi kurum içinde kurumun kendi personelleri tarafından yapılmaktadır. Yine bu modelde de bulut sisteminin kullanıcıları sisteme bir web tarayıcısından yetkisi dahilindeki süreçlere erişebilmektedir. Bu tarz modeller özellikle büyük ölçekli işletmeler ve kamu kurumları

tarafından tercih edilmektedir. Bu model genel bulut modeli ile kıyaslandığında yatırım maliyetlerinin kurum tarafından yapılacak olması ve kesintisiz hizmet için kurum bünyesinde ekipler kurulması gerekliliği dezavantajları olarak görülebilir. Ancak bilginin mahremiyeti ve kritik altyapılar noktasından incelendiğinde özel bulut modelinin avantajlarının ortaya çıktığı görülmektedir.

- **Melez Bulut Modeli (Hybrid Cloud Model):** Melez bulut modeli adından da anlaşılacağı gibi genel ve özel bulut modellerinin ihtiyaçların analiz kapsamında bir kısmının özelden bir kısmının ise genel bulut modelinde kullanıldığı yapılardır. Kurum ya da kuruluşlar kritik altyapılarını kendi özel bulutlarında diğer hizmetlerini ise genel bulut sistemlerinde kurgulamaktadır [21].

Yukarıda anlatılan bulut bilişim modelleri ile kurum ya da kuruluşun ihtiyacına göre iş modellerini modüler bir şekilde kurgulamak mümkün hale gelmektedir. Bu sayede kurumun bilişim altyapısından ihtiyaç duymuş olduğu performans, güvenilirlik, ölçeklenebilirlik, süreklilik, her yerden ve her platformdan erişim, birlikte çalışabilme, bakım - destek süreçleri ve maliyet konularında önemli avantajlar elde edilebilmektedir. Bu avantajların yanında sistemlere bağlanabilmek için internet erişiminin ihtiyaç olması, oluşabilecek güvenlik problemleri, gizlilik ve mahremiyet, bilişim sistemlerine ve verilere yetkisiz erişim riskleri ve yasal sorunları da bulut sistemlerinin dezavantajları olarak saymak mümkündür.

2.4.3. Bulut Bilişimin Uygulama Geliştirme Alanındaki Kullanımı

Yazılım yaşam döngüsüne ait süreçlerin tamamını bulut tabanlı modeller üzerinden kurgulayabilmek mümkün hale gelmiştir. Bulut sistem altyapısı kurulduktan sonra yazılım yaşam döngüsündeki tüm süreçleri yürütebilecek birçok uygulama ile yönetebilmek mümkün hale gelmiştir. Bulut üzerinde geliştirilecek olan uygulamanın planlanması, ölçümleme, efor ve diğer analizlerin yapılabilmesi, uygulamaya ait ekran arayüzlerinin eskiz (mockup) çalışmalarının yapılabilmesi geliştirme süreçlerinin yürütülmesi, test süreçlerinin oluşturulması, test platformlarının ve ihtiyaç olan sistem kaynaklarının planlanabilmesi, bakım süreçlerinin projenin yaşam ömrü boyunca yönetilebilmesi süreçleri gerçekleştirilebilmektedir. Bu bağlamda bulut sistemleri üzerinde web ve mobil tabanlı uygulamalar geliştirmek, api ve sdk geliştirmek, geliştirilen uygulamalara ait versiyonlama ve yedekleme mimarilerini oluşturabilmek, ses ve video gibi yüksek bant genişliğine ve büyük depolama alanlarına ihtiyaç duyulan yapıları kurgulayabilmek, geliştirilecek olan uygulama için ihtiyaç olan yazılımların dağıtımının sağlanabilmesi, uygulamanın test edilmesi için ortamların oluşturulabilmesi büyük verinin analizi ve yapay zekaya dayalı tekniklerin oluşturulması, sistemlerin eğitilmesi mümkün olabilmektedir [22].

2.4.4. Bulut Tabanlı Platformlar

Google Drive, Microsoft One Drive, Amazon Drive, Apple iCloud, Dropbox, Yandex Disk, TTNET Bulut, Turkcell Bulut, SpiderOak, Tresoit vb gibi depolama ve yedekleme hizmeti veren birçok servis sağlayıcı olmasına rağmen uçtan uca bütünleşik bulut hizmeti veren servis sağlayıcı platform olarak karşımıza AWS (Amazon Web Servisleri), Microsoft Azure, Google Cloud, Huawei Cloud, IBM Cloud, Oracle Cloud, Alibaba Cloud, WMvare Cloud gibi büyük firmalar çıkmaktadır.

Bulut hizmeti veren platformlar incelendiğinde ihtiyaca uygun olarak farklı hizmetler verdikleri görülmektedir. Buna göre bulut platformları;

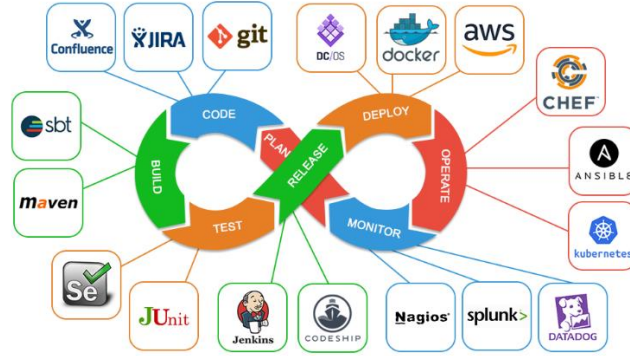
- Analiz,
- Uygulama Entegrasyonları
- İş Zekası Uygulamaları,
- Blok Zinciri Uygulamaları,
- Finansal Yönetim Uygulamaları,
- Makine Öğrenmesi ve Yapay Zeka Uygulamaları,
- Web ve Mobil Tabanlı Ön Yüz Geliştirme (Front-End) ve Arka Plan Geliştirmeler (Back-End),
- Oyun Geliştirme Uygulamaları,
- Nesnelerin İnterneti (IoT) ve Endüstri 4.0 Tabanlı Uygulama Geliştirme,
- Güvenlik Uygulamaları Geliştirme,
- Artırılmış ve Sanal Gerçeklik Uygulamaları Geliştirme,
- Ses ve Video gibi Medyaları Kullanan Uygulamalar Geliştirme,
- Müşteri İlişkileri Yönetimi,
- Kuantum Teknolojileri,
- Robotik Teknolojileri
- Uydu Teknolojileri

gibi birçok uygulama geliştirme ihtiyacı olan teknolojilere hazır altyapılar sundukları görülmektedir. Geliştirme yapacak kurum ya da kuruluşlar bu platformları kendi ölçeklerine uygun olarak kurgulayıp hemen kullanabilir hale gelmektedir [23].

2.4.5. Bulut Tabanlı Uygulamalar

Bulut sistemleri üzerinden uygulama geliştirme süreçleri incelendiğinde ekiplerin birlikte çalışması için gerekli olan yazılımsal altyapı çözümlerinden proje yönetim platformlarına, uygulamanın test edileceği test yazılımlarından geliştirilen uygulamanın versiyon takibini ve

yedeklenme süreçlerini yönetebilecek versiyon takip platformlarına kadar birçok çözümün geliştirildiği görülmektedir. Şekil 2.6’da bulut tabanlı uygulamalardan bazıları görülmektedir [23].



Şekil 2.6. Bulut tabanlı uygulamalar [23]

- **Jira:** Atlassian firması tarafından java dilinde geliştirilmiş web tabanlı proje yönetim yazılımıdır. Uygulama geliştirme ekipleri tarafından sıklıkla kullanılmaktadır.
- **Git:** Yaygın kullanılan bir kaynak yönetim ve versiyon kontrol platformudur. Uygulama geliştirme ekipleri tarafından sıklıkla kullanılan dağıtık mimari altyapısına sahip web tabanlı bir yönetim aracıdır.
- **Kubernetes:** Geliştirilen uygulamaya yönelik olarak iş yüklerini ve hizmetlerin yönetimini sağlayan açık kaynak bir yazılımdır.
- **Monit:** UNIX sistemlerin yönetiminde kullanılan açık kaynaklı bir yazılımdır.
- **Datadog:** Bulut tabanlı sistemlerdeki sunucu kaynakları, veritabanı yükleri ve aktif hizmetlerin analizini yapabilen bir görüntüleme aracıdır.
- **WhiteSource:** Güvenlik açıklarını analiz eden açık kaynaklı bir yazılımdır. Genellikle Microsoft Azure DevOps, GitHub gibi platformlardaki analiz için kullanılmaktadır.
- **Jenkins:** Yedekleme ya da benzer takvime bağlanması gereken işleri otomatize eden bir platformdur.
- **Selenium:** Geliştirilen uygulamanın testlerinin yapılması için kullanılan bir platformdur. Yazılım test uzmanları bu uygulama üzerinde geliştirdikleri fonksiyonlar ile uygulamanın testini kolaylıkla bulut üzerinden gerçekleştirebilmektedir.
- **Vagrant:** İş akışı yada süreçlerinin oluşturulması ve yönetilmesi için kullanılmaktadır.
- **Trello:** Atlassian firmasının bir ürünü olan Trello, proje ekiplerinin birlikte çalışabilmesine, Agile gibi yazılım geliştirme ve süreç yönetim modellerinin uygulanmasına ve proje ekibinin performans takibinin yapılmasına olanak sağlamaktadır.
- **Snort:** Sisteme yapılabilecek saldırıların gerçek zamanlı tespitine yarayan bir görüntüleme yazılımıdır.
- **Splunk:** Veri yönetiminin yapılması için kullanılan bir yazılımdır.
- **Nagios:** Ağ ve altyapı yönetimi yazılımıdır.

- **Consul:** Nagios gibi ağın yapılandırılması için kullanılan bir araçtır.
- **Sumo:** Basit bir log analiz yazılımıdır.
- **Docker:** Dağıtık mimaride uygulamalar kurulumak için kullanılan bir araçtır.
- **CFEngine:** Geliştirilen uygulamaların en önemli yönetilmesi gereken konularından biriside konfigürasyon yönetimidir. Bu yazılım geliştirilen uygulamanın konfigürasyon yönetimini otomatize edebilmektedir.
- **Supervisor:** UNIX tabanlı sistemlerde kullanılan süreç yönetim yazılımıdır.

Yukarıda bahsi geçen uygulamalar bulut tabanlı mimarilerde sıklıkla tercih edilen uygulamalardır. Bu uygulamalar gibi birçok uygulama bulunmakla beraber her gün yenileri de eklenmektedir [24].

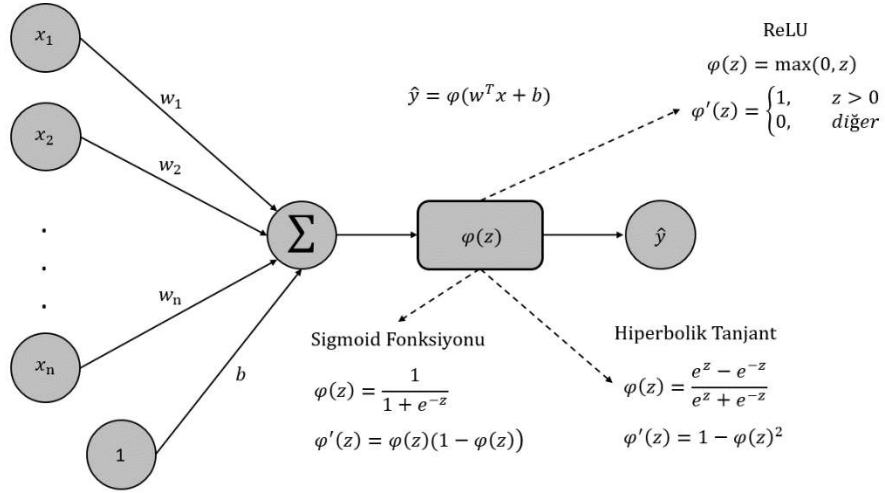
Teknolojinin gelişimi, farklı lokasyonlarda bulunan proje ekiplerinin birlikte çalışma ihtiyacı ve bilişim altyapı yatırımlarındaki artan maliyet iş geliştirme, yönetme ve hizmete sunma süreçlerinin bulut tabanlı modellere dönüşmesine neden olmuştur. Gelişen teknoloji ile birlikte artan rekabet ve ihtiyaç olan hız bulut bilişim mimarilerinin kullanımını doğrudan tetikleyen etmenlerden biri haline gelmiştir. Yine proje bazlı bilişim teknolojilerine olan ihtiyacın çeşitliliği, müşteri tarafındaki yeni doğan ihtiyaçlara adaptasyon, iş süreçlerinin ölçeklendirilebilmesi ve ölçümlenebilir olması, artan güvenlik ihtiyacına hızlı adaptasyon gibi sebeplerde bulut tabanlı mimarilere dönüşüm sürecini hızlandıran diğer etmenler olduğu görülmüştür.

2.5. Makine Öğrenmesi Teknikleri

Bu bölümde tez çalışmasında kullanılan makine öğrenmesi teknikleri ele alınmıştır. Kullanılan yöntemlerin matematiği ve algoritması ile beraber tez çalışmasında neden tercih edildiği açıklanmıştır.

2.5.1. Perceptron Öğrenme Algoritması

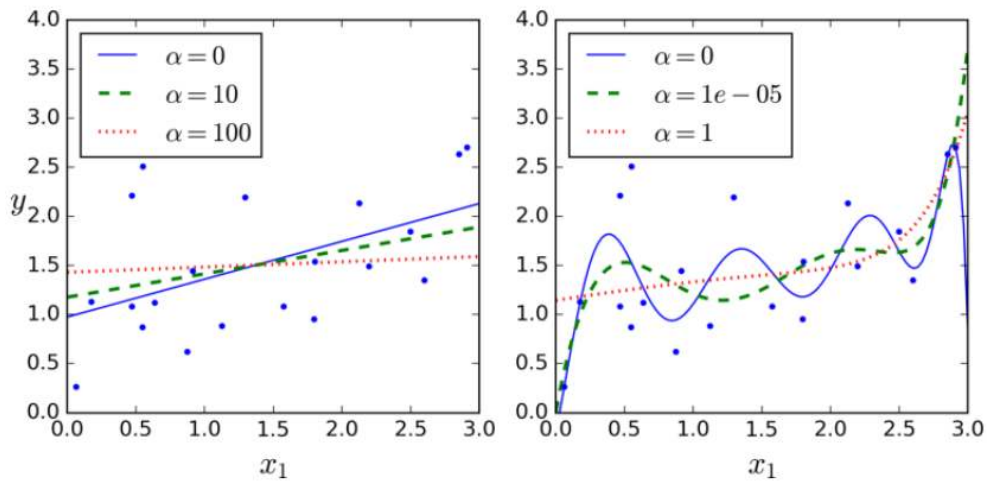
Perceptron öğrenme algoritması bir yapay sinir ağı ile verisetinde bulunan giriş verilerinin özelliklerinin tespit edilmesi için kullanılmaktadır. Doğrusal bir sınıflandırma algoritması olan Perceptron öğrenme algoritması girdi verilerinin ağırlık katsayılarını belirleyebilmekte ve belirlenen ağırlık katsayıları ile sinir ağının sonuçlarını ortaya koyabilmektedir [25]. Şekil 2.7’de Perceptron öğrenme algoritmasına ait bir model gösterilmiştir.



Şekil 2.7. Perceptron öğrenme algoritması modeli [25]

2.5.2. Ridge Regresyon Algoritması

Regresyon tabanlı bir sınıflandırma algoritması olan Ridge Regresyon Algoritması verisetinde çok değişkenin olduğu durumlarda tercih edilmektedir. Algoritmanın temel amacı lineer (doğrusal) regresyon algoritmalarında yaşanan sistemin aşırı öğrenilmesi (overfitting) problemini ortadan kaldırabilmektir. Bunun için Ridge algoritması maliyet fonksiyonuna (cost function) bir düzeltme eklemesi gerçekleştirilerek model ağırlıklarının minimum tutmaktadır. Model ağırlıkları çok küçük olduğu için sistemin öğrenmesi diğer lineer regresyon modellerine göre daha yavaş olsa da hem aşırı öğrenme probleminin önüne geçilebilmekte hem de daha sağlıklı sonuçlar elde edilebilmektedir [26]. Şekil 2.8’de bir veri kümesine Lineer Regresyon ve Ridge Regresyon’un öğrenme süreçleri kıyaslanmaktadır.



Şekil 2.8. Lineer regresyon ve Ridge regresyon karşılaştırması [26]

2.5.3. Ensemble (Topluluk) Öğrenme Algoritması

Bir sınıflandırma algoritması olan Ensemble öğrenme algoritması farklı skorlara göre cevap veren sınıflandırıcıların sonuçları üzerinden karar vermektedir. Ensemble öğrenme algoritması doğruluk oranı yüksek ve tüm verisetleri için genellenebilir bir modeldir. Model performansını arttırmak için Torbalama (Bagging) ve Yükseltme (Boosting) tekniklerini kullanmaktadır.

- **Torbalama (Bagging) Yöntemi:** Veriseti üzerinden alt örneklemelerin oluşturulduğu ve bu örneklemeler üzerinden sınıflandırıcıların alt kararlar oluşturduğu yöntemidir. Bu yöntemde oluşturulan alt kararların birleştirilmesi ile ana sınıflandırıcı kararı belirlenmektedir. Bu yöntemde öncelikle veriseti üzerinden alt verisetleri oluşturulmaktadır. Oluşan alt verisetleri üzerinden modeller oluşturularak birbirinden tamamen bağımsız şekilde kararlar üretilmektedir. Sonuçta oluşan sınıflandırıcı tahmini ise alt sınıflandırıcıların kararları üzerinden gerçekleştirilmektedir.
- **Yükseltme (Boosting) Yöntemi:** Bu yöntemde ise verisine farklı ağırlıklar verilerek elde edilen sınıflandırıcı kararları üzerinden sonuç kararı belirlenmektedir. Yükseltme yönteminde tüm nitelikler başlangıçta eşit ağırlıkta iken niteliklerin ağırlıkları değiştirilerek yeni sonuçlar elde edilmekte ve bu sonuçlar kıyaslanmaktadır. Yükseltme yönteminde öncelikle veriseti üzerinden alt kümeler oluşturulmaktadır. Oluşan alt kümeler üzerindeki niteliklere eşit ağırlıklar verilerek başlatılan algoritma ağırlıkların değiştirilmesi ile alt sınıflandırıcı tahminleri yapılması ve bu tahminler üzerinden nihai sınıflandırıcı kararının verilmesini sağlamaktadır.
- **Rastgele Altuzaylar (Random Subsample):** Veriseti öz nitelikleri rastgele seçilerek alt kümelerin oluşturulduğu yöntemidir. Bu yöntemde eğitimin gerçekleştirileceği veri setinde oluşturulacak olan alt kümeler için öz niteliklerin bazılarının rastgele seçimi ile yeni alt veri setleri oluşturulmaktadır. Rastgele oluşturulan yeni verisetleri üzerinden sınıflandırıcılar bir karar verir. Bu kararların ortalaması alınarak nihai sınıflandırıcı kararı üretilmektedir [27].

Ensemble öğrenme algoritması ile kullanılan torbalama yöntemleri Bagging meta-estimator, Random Forest gibi algoritmalarda da kullanılırken yükseltme yöntemleri AdaBoost, (Gradient Boosting (GBM), XGBM algoritmalarında kullanılmaktadır. Tez çalışmasında oluşturulan verisetinde bahsi geçen algoritmaların başarımları ortaya konulmuştur.

2.5.4. Gradyan Artırma (Gradient Boosting - GBM) Algoritması

Topluluk (Ensemble) öğrenme algoritmalarından biri olan Gradyan artırma algoritması regresyon ve sınıflandırma temelli verisetlerinde başarılı sonuçlar verebilmektedir. Algoritmanın temelinde performanslı ve doğru sonuçlar verebilmek için alt verisetleri üzerinde yükseltme

yöntemleri kullanılmaktadır [28]. Regresyon ağacında veriseti önceki ağaçta hesaplanan hatalar baz alınarak başarımın artmasına yönelik bir kurgu oluşturulmuştur. Bu yüzden her adımda kayıp fonksiyonundaki hata oranları minimum olacak şekilde güncellemeler gerçekleştirilerek sınıflandırıcı çalıştırılmaktadır.

2.5.5. Olasılıksal Dereceli Azalma (Stochastic Gradient Descent – SGD) Algoritması

Doğrusal regresyon tabanlı bir sınıflandırıcı yöntemi olan SGD algoritması seyreltilmiş verisetlerinde ve metin tabanlı sınıflandırma problemlerinde başarılı sonuçlar veren bir makine öğrenme algoritmasıdır. Algoritma uygulama kolaylığı ve verimliliği açısından tercih edilmektedir. SGD algoritmasında verisetinden alınan rastgele veri üzerinden işlem yapılarak optimuma ulaşılması hedeflenmektedir [29]. Algoritma iyi bir sonuca ulaşabilir ancak düzenlileştirme ve yineleme sayısı gibi birçok parametrenin optimizasyonu gerektiğinden optimum çözüme ulaşmayabilmektedir.

2.5.6. Navie Bayes

Navie Bayes algoritması olasılık tabanlı bir sınıflandırma modelidir. Özellikle dengesiz veri kümelerinde başarılı sonuçlar veren algoritma verisetindeki her durumun olasılığı üzerinden olasılığın en büyük olma durumuna göre karar vermektedir. Eğitim verisinin yetersiz olduğu durumlarda Navie Bayes Algoritması tercih edilebilmektedir. Bu durumda test kümesinde hesaplanamayan olasılıklar için oluşacak Sıfır Frekans (Zero Frequency) durumu düzeltme teknikleri ve optimizasyon yöntemleri ile çözülebilmektedir [30].

2.5.7. XGBoost Algoritması

XGBoost (eXtreme Gradient Boosting) algoritması Chen, T. ve arkadaşlarının 2016 yılında yapmış oldukları çalışmada Gradyan Artırma (Gradient Boosting) algoritmasını geliştirmesi ile ortaya çıkmıştır [31]. Algoritma, Gradyan Artırma algoritmasında yaşanan aşırı öğrenmenin (overfitting) önüne geçecek yöntemler sunmuştur. Ayrıca verisetindeki boş alanların yönetimi için optimizasyon yöntemleri ortaya koymuştur. Yapılan geliştirmeler ile birlikte XGBoost algoritması Gradyan Artırma algoritmasından çok daha etkin tahmin gücü ve hız sağlamıştır. Karar ağacı tabanlı bir algoritma olan XGBosst algoritması sistem kaynaklarını da çok etkin kullanabilmektedir [31].

2.5.8. Rasgele Orman (Random Forest) Algoritması

Sınıflandırma ve regresyon tabanlı modellere uygun verisetlerinde başarılı sonuçlar veren Rasgele Orman (Random Forest) Algoritması [32] oluşturduğu karar ağaçlarının üzerinden sınıflandırma başarımını arttırmayı hedeflemektedir. Algoritma, üretmiş olduğu karar ağaçlarının arasından en yüksek başarıımı olan en yüksek skorlu değeri seçmektedir. Algoritma kök düğümü bulmak ve karar ağacını en optimum şekilde oluşturmak için rastgele düğümleri bölmekte ve oluşan alternatifler üzerinden en iyi çözümü seçmektedir. Eğer verisetinden makul sayıda ağaç oluşturulabiliyor ise algoritma aşırı öğrenme probleminin de önüne geçebilmektedir. Bu yüzden algoritmanın başarıımı uygun verisetinin oluşturulmasına bağlıdır.

2.5.9. LGBM (LightGBM) Sınıflandırma Algoritması

Histogram tabanlı bir algoritma olan LGBM (LightGBM) algoritması diğer boostig algoritmaları olan Catboost, XGBoost gibi Gradient Boosting (Gradyan Arttırma) algoritmasını iyileştirmek üzere kurgulanmıştır. LGBM algoritması 2017 yılında Microsoft DMTK (Distributed Machine Learning Toolkit) tarafından geliştirilmiş ve muadillerine göre çok daha hızlı eğitim süreçleri ve yüksek doğruluk oranları elde edilebilen bir algoritmadır [33].

2.5.10. KNN (K-Nearest Neighbors – K-En Yakın Komşu) Algoritması

Makine öğrenmesi alanında en bilinen ve yaygın olarak kullanılan sınıflandırma temelli olan KNN algoritması komşular arasındaki uzaklık ve en yakın komşu sayısı üzerinden karar vermektedir. Algoritmada öncelikle tahmin edilecek verinin özelliklerini koordinat düzlemi üzerinde bir nokta olarak belirlemektedir. Bu noktanın verisetindeki diğer noktalar ile arasındaki mesafeler ölçülmektedir. Ölçülen mesafelere göre komşuluklar belirlenmektedir. Komşuluklar üzerinden tahmin edilecek olan verinin sınıfı belirlenmektedir. Burada optimum komşu sayısı bulunarak belirlenen komşuların sınıflarındaki oranda hangi sınıf en çok ise tahmin edilmesi gereken veri o sınıfa dahil edilmektedir. KNN algoritması basitliği, hızlı ve doğru sonuçlar üretmesinden dolayı tercih edilmektedir [34].

2.5.11. Karar Ağacı Algoritmaları

Karar ağacı algoritmalarının temelinde verisetinde bulunan öznelitliklere bir başka deyişle tüm öğrenme kümesine göre ağacın oluşturulması ve ağaçtaki gereksiz dalların budanması ile en optimum çözümü sağlayan ağacın belirlenmesi bulunmaktadır. Belirlenen ağaç üzerinden de tahmin edilecek olan verinin sınıflandırması gerçekleştirilmektedir. Optimum ağacın oluşması için farklı teknikler kullanılmaktadır. Karar ağacı yöntemleri aşağıdaki gibidir.

- **Entropi Tabanlı Karar Ağacı Algoritmaları:** ID3 ve C4.5 algoritmalarını entropi tabanlı karar algoritmalarına örnek verilebilir. Entropi temelli algoritmaların temel mantığı verisetindeki özniteliklerden bilgi kazancının en yüksek olanı seçmek ve o öznitelik üzerinden ağacın dallanmasını sağlamak üzerine kuruludur. Bilgi kazancı en yüksek olan öznitelige göre tüm dallanma tamamlanana kadar algoritma çalışmakta ve karar ağacını oluşturabilmektedir.
- **Sınıflandırma ve Regresyon Temelli Karar Ağacı Algoritmaları:** Twoing ve Gini algoritmaları regresyon temelli karar ağacı algoritmalarının içerisinde sıklıkla tercih edilen algoritmalar olarak karşımıza çıkmaktadır. Bu algoritmaların temeli verisetindeki özniteliklere göre karar ağacı oluşturulurken karar ağacının sadece iki dala ayrılabilmesi sağlanmaktadır. Bir düğümde seçim yapılırken hangi dala doğru dallanmanın gerçekleşeceği ise algoritmada her dala ait uygunluk ölçüsünün belirlenmesi ile sağlanmaktadır.
- **Bellek Tabanlı Karar Ağacı Algoritmaları:** K-En Yakın Komşu algoritmasını bellek tabanlı karar ağacı algoritmasına örnek vermek mümkündür [35].

2.5.12. CatBoost Sınıflandırma Algoritması

1999 yılında ortaya atılan Gradyan Arttırma (Gradient Boosting) algoritması, boosting temelli makine öğrenme algoritmaların birçoğunun çıkış noktası olarak görülmektedir. Bu algoritmadan sonra 2016 yılında XGBoost algoritması 2017 yılında LGBM algoritması ve bu algoritmayı takiben yine 2017 yılında CatBoost algoritması önerilmiştir. CatBoost algoritması Yandex tarafından önerilen açık kaynaklı bir algoritmadır. CatBoost adını Category ve Boostig kelimelerinin baş harflerinden almıştır. CatBoost algoritması GPU desteği ile çok daha hızlı sonuçlar üretebilmektedir. Bu özelliği sayesinde metin tabanlı içerikleri ve nümerik verisetlerini işleyebilme imkanı sağlamaktadır [36].

2.5.13. AdaBoost Sınıflandırma Algoritması

1997 yılında Freund ve arkadaşları tarafından önerilen Adaboost algoritması zayıf algoritmaları bir araya getirilmesi ve daha gelişmiş yeni algoritmalar ile oluşturulmuş karma bir modeldir. Oluşturulan model ile çok daha güçlü sonuçlar ortaya çıkmıştır. [37]

3. SİSTEM MİMARİSİ VE UYGULAMA

Tez çalışmasında Endüstri 4.0 entegrasyonu tamamlanamamış işletmelerde oluşan büyük verinin işlenmesi, anlamlı hale getirilmesi ve üretim yönetim sistemlerine entegrasyonu için bulut tabanlı bir model geliştirilmiştir. Bu bölümde geliştirilen modelin mimari yapısı, oluşturulan kural tabanlı ve makine öğrenmesi modelleri tanıtılmıştır. Yine bu bölümde oluşturulan model ile ilgili her aşama adım adım anlatılacaktır. Modelin başarımı ise Bölüm 4 'te Bulgular ve Tartışma başlığı altında verilmiştir.

3.1. Giriş

Yeni sanayi devrimi olarak anılan Endüstri / Sanayi 4.0, 2011 yılında ilk kez Hannover Fuarı'nda ortaya atılmıştır. Bilişim çağının modern bir hal alarak üretim teknolojisini bir üst seviyeye taşınmakta olduğu uzmanlar tarafından ifade edilmiştir. Tüm dünyada olduğu gibi Türkiye'de de Endüstri 4.0'a geçiş süreci özellikle KOBİ ölçeğindeki firmalarda sorunlu ilerlemektedir. Türkiye İstatistik Kurumu verilerine göre Türkiye'deki toplam işletmelerin %99,8 'i KOBİ ölçeğindedir. Bu işletmelerin 1-19 kişi çalıştıran adedi 303.590 iken 20-49 arasında personel çalıştıran işletme sayısı 18.076'dır. Ciroyu yüksek ve çok personel çalıştıran işletmelerin önemli bir çoğunluğu üretim süreçlerinin yönetimi için büyük bütçeler ayırabilirken özellikle 1-19 ve 20-49 arası personel çalıştıran KOBİ'ler, üretim süreci yönetimini insan eli ile yapmayı daha düşük maliyetli olduğu için tercih etmektedirler [38]. Bu işletmeler ihtiyaç duydukları muhasebe, satış, stok takibi gibi elzem işler için yazılımlara ciddi bütçeler ayırabilirler de üretim yönetimi için bu bütçeleri maliyet problemlerinden dolayı ayıramamaktadırlar. Ayrıca piyasada var olan üretim yönetim sistemlerinin yine bu küçük boyuttaki KOBİ'ler için entegre edilmesi istendiğinde üretimde kullanılan tezgahların veya makinelerin Endüstri 4.0'a uyumlu olmadığı görülmektedir.

Literatür ve endüstride kullanılan ürünler incelendiğinde Üretim Yönetim Sistemi (Manufacturing Execution System - MES) olarak ta adlandırılan ve hedefi bir fabrikadaki üretim süreçlerini anlık olarak takip edebilmek olan birçok sistemin olduğu görülmektedir [39]. Bu sistemlerin çoğunluğu bir üretim reçetesi üzerinden gerçekleşen üretimin akışını takip etmek ve bu akış üzerinden anlamlı raporlar sunmak şeklindedir. Fabrikalarda ya da üretim tesislerinde kullanılan üretim yönetim sistemleri imalat süreçlerinin etkinliğini arttırmak, olağan dışı durumlara müdahaleyi hızlandırmak ve en önemlisi bir üretim yaşam döngüsü üzerinden nihai ürünü optimum verim ile ortaya çıkmasını sağlamaktır [40]. Gelişen teknoloji ve ortaya yeni çıkan ihtiyaçlar doğrultusunda verimin artması için klasik yöntemlerin yetersiz kalması sonucu MES platformlarında da akıllı tekniklerin kullanılmaya başlandığı görülmektedir [41]. Yapay zeka temelli bir üretim yönetim sisteminin önündeki temel problemin üretim bantlarından toplanacak

olan gerçek zamanlı verinin yetersizliği ve çeşitliliğidir. Özellikle KOBİ ölçeğindeki işletmelerde kullanılan üretim tezgahlarının önemli bir kısmında bilgisayar kontrollü üretim altyapısının bile bulunmaması bu tezgahlardan veri toplanmasını güçleştirmektedir. Bu da tüm üretim sürecinin takibini zorlaştırmaktadır. Üretim süreçlerindeki standartların oluşabilmesi için Amerikan Ulusal Standartlar Enstitüsü (ANSI), ISA-95 standardını ortaya koyarak uçtan uca bir üretim için gerekli olan operasyonel modeller standartlaştırılmıştır [42]. ISA-95 ve sonrasında geliştirilen standartlar ile üretim yönetim sistemleri;

- Bir sürece ait uçtan uca zamanlamanın yapılabilmesi ve ölçümlenebilmesi,
- Kalite yönetim ve ölçümlenmesinin yapılabilmesi,
- Veri toplama ve analiz süreçlerinin planlanması,
- Personel başta olmak üzere işletme kaynaklarının ölçümleme ve yönetiminin gerçekleştirilebilmesi,
- Personel, üretim bandı ve uçtan uca süreç performansının analizinin gerçekleştirilebilmesi,
- Süreç ve geliştirmelere uygun dokümantasyon yönetiminin sağlanabilmesi,

fonksiyonlarını içermektedir [43]. Standartlar olmasına rağmen gelişen teknoloji ve ihtiyaçlar ile standartlara bağlı kalarak yeni ihtiyaçlara cevap verebilecek sistemlere ihtiyaç duyulmaktadır. Bu tez çalışmasında ortaya konulan bulut tabanlı üretim yönetim sistemi uygulaması bahsi geçen standartlara ek olarak;

- Hibrit yazılım altyapısına uyumlu olabilmek (var olan tüm yazılımlar ile çalışabilme),
- Kare kod teknolojisiyle Endüstri 3.0'dan 4.0'a dönüşüm ile ilgili IoT çözüm önerileri sunmak ve bu sayede bilgisayar kontrollü olmayan üretim tezgahlarındaki verilerinde sayısallaştırılması, toplanması ve analizinin yapılabilmesini sağlamak,
- Makine parkuru ve personel performanslarının takibini üretim reçetesinden bağımsız bir şekilde yapabilmek,
- Yapay Zeka altyapısı ile iş zekası çözümü ve sayısallaştırılmış veriler üzerinden üretim yönetimi ile alakalı Risk / Fırsat Analizi yapabilmek,
- Web / Mobil tüm platformlardan erişim ile yöneticilerin üretim orkestrasyonunu yakından takip etme kabiliyetini kazandırmak

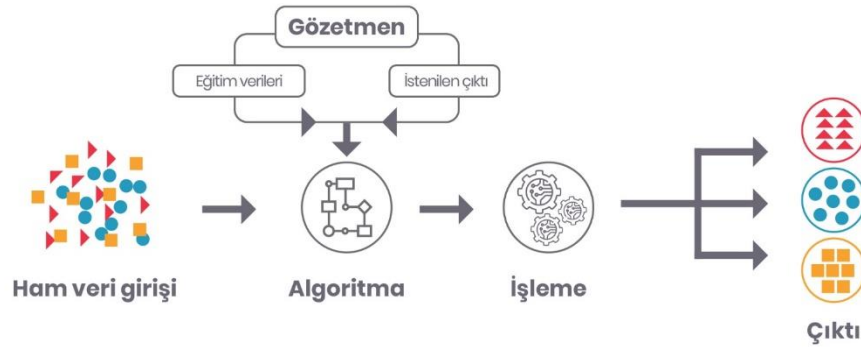
fonksiyonlarını sunmayı hedeflemiştir. Birçok alanda olduğu gibi işletmelerinde üretim esnasında elde edilen verilerinin boyutu ve kontrolü, Endüstri 4.0'a geçiş sağlanıp veriler sayısallaştıkça zorlaşmaktadır. İşletmeler büyüdükçe üretim süreçlerinin yönetimi de kâğıt ortamında insan eli ile yapılamayacak boyutlara ulaşmaktadır. İşletmeler tam bu geçiş evresinde gerekli olan teknolojik entegrasyonları tamamlayamadıkları için performans ve verimlikte düşüş görülmektedir [44].

Tez çalışmada Endüstri 4.0 Entegrasyonu için bir Bulut tabanlı üretim yönetim sistemi ile makinelerin ve personellerin Endüstri 4.0 uyumu sağlanarak üretim süreçlerinin sayısallaştırılması hedeflenmiştir. Çalışma ile donanımsal entegrasyon, Tezgâh / Makine / Personel 'in ürettiği

verilerin sayısallaştırılması, veri madenciliği ve yapay zekâ algoritmaları ile fabrikanın risk / fırsat analizi yapılması aşamalarına çözüm olunması hedeflenmiştir.

3.2. Mimarinin Genel Yapısı

Teknolojinin gelişmesi ve rekabetin artması ile birlikte imalat sanayisinde üretim gerçekleştiren işletmeler, malzeme tedarikinden, makine ve personellerin performans takibine kadar tüm süreçleri uçtan uca ve anlık olarak kontrol etmek istemektedir. Üretim sürecinin yönetilmesi işletmenin performans ve veriminin artırılması için son derece önemlidir. Hammadde, personel ve makine gibi kaynakların bir düzen içerisinde üretimin parçasını oluşturması ise ancak iyi yönetilen bir iş akışı ile mümkün olabilmektedir. Özellikle KOBİ ölçeğindeki işletmeler incelendiğinde uçtan uca kontrol için makine – teçhizatın ve altyapının yetersiz olduğu görülmektedir. Önerilen tez çalışması ile imalat sanayisinde üretim gerçekleştiren ve Endüstri 4.0 entegrasyonu sağlanmamış işletmeler için iş süreçlerini uçtan uca yönetebilecekleri bir üretim yönetim sistemi geliştirilmesi hedeflenmiştir. Endüstri 4.0 entegrasyonu olmayan işletmelerde imalat parkurundaki veri sayısallaştıramadığı için süreçleri anlık olarak gözlemlemek, ölçümlemek ve yönetebilmek mümkün olamamaktadır. Bu işletmeler üretim reçeteleri ile süreçleri manuel yöntemler ile işletmekte ve anlık olarak makine parkurundan bilgi akışı olamamaktadır. Bunun sebepleri incelendiğinde makine parkurunun bilgisayar kontrollü olmayışı, tedarikten dağıtıma kadar tüm lojistik sürecin manuel yöntemler ile yönetilmesi ve çalışan personelin teknolojik altyapılarının eksikliği gibi faktörlerin olduğu görülmektedir. Bu tez çalışmasında imalat sanayisinde Endüstri 3.0 üretim modeli ile faaliyet gösteren KOBİ'lerin hibrid yazılım altyapılarına uyumlu bir şekilde Endüstri 4.0'a geçiş yapabilmelerini sağlayacak üretim yönetim sistemi uygulaması ile ilgili çalışma yapılmıştır. Geliştirilen sistem ile makine parkurunun verilerinin anlık olarak sayısallaştırılması, sayısallaştırılan verinin büyük veri mimarisine uygun olarak bulut tabanlı sistemlerde depolanması ve bu veri üzerinden anlık akıllı çıkarımlar yapılması mümkün hale gelmiştir. Tez çalışması ile geliştirilen sistemin bulut tabanlı olması sayesinde üretim sürecinin anlık olarak takibi ve idamesi, sistem verileri üzerinden akıllı karar destek süreçlerinin oluşturulması sağlanmıştır. Tez çalışmasında ortaya konan mimari model Şekil 3.1'de gösterilmektedir.



Şekil 3.1. Mimari model

Tez çalışması ile geliştirilen model iki farklı veri kümesi ile beslenmektedir. Birinci veri kümesi fabrikadaki tezgahlardan alınan IoT verileri, üretim bandı verileri, tezgah iş yükü bilgisi, personel görev ve iş yükü bilgileri, üretim verileri gibi anlık olarak akan ham verilerdir. Bu veriler üzerinden tez çalışmasında kural tabanlı çözümler ortaya konmuştur. Tez çalışmasının 4. Bölümünde bu veriler üzerinden oluşturulan kural tabanlı çözümler sunulmuştur. İkinci veri kümesi tezgah üzerinden alınmış sıcaklık, çalışma süreleri vb. gibi sensör verilerini barındırmaktadır. Bu veri kümesinden ise makine öğrenmesi teknikleri ile karar destek sistemi oluşturulmuştur. Yine oluşturulan model sonuçlar tez çalışmasının 4. Bölümünde tartışılmıştır. Tez çalışmasında fabrika içinden akan veri ve hazır verisetlerinden oluşan veri kümesi bulut tabanlı bir sistem üzerinde işlemektedir. Kullanıcılar sistem üzerine web veya mobil tabanlı teknolojiler ile erişim sağlayabilmektedir.

3.3. Gerekli Kabuller

Tez çalışmasında ortaya konmaya çalışılan çalışma büyük bir sistemin bir parçası olarak tasarlanmıştır. Fabrikalarda kullanılmakta olan üretim yönetim sistemleri, ürünün siparişi, üretim süreçleri, stok süreçleri, kurumsal kaynak yönetim süreçleri, teslimat süreçleri ve bakım yönetim süreçleri gibi büyük bir sistemi barındırmaktadır. Bu yüzden tez çalışması için belli kabuller yapılarak geliştirme yapılmıştır. Oluşturulan mimari için yapılan kabuller aşağıdaki gibidir.

- Geliştirme yapılırken verinin anlık olarak geldiği kabul edilen hazır verikümesi üzerinden çalışma gerçekleştirilmiştir.
- Veri kümesi ve bulut sunucuya servis tabanlı mimari ve API'ler üzerinden etkileşim yapıldığı kabul edilmiştir.
- Üretim bandından gelen sensör verilerinin doğru olduğu kabul edilmiştir.
- Üretim yönetim sistemi diğer servisleri ile etkileşimin olduğu varsayılarak sadece var olan bütün veri üzerinden analiz gerçekleştirilmiştir.

3.4. Materyal ve Metot

Gerçekleştirilen tez çalışmasında fabrika içinde üretilen verinin bulut tabanlı sistem üzerine alınması, alınan verinin kural tabanlı ve makine öğrenmesi tabanlı teknikler ile işlenmesi ve işlenen sonuçların sistem kullanıcılarına web ve mobil platformlar aracılığıyla paylaşılması gerekmektedir. Bu yüzden tez çalışmasında mobilya sektöründe üretim gerçekleştiren işletmeler pilot uygulama için seçilmiştir. Mobilya imalatı yapan makinalardan elde edilen veriler üzerinden bir çalışma yapılmıştır.

Tez çalışmasında web ve mobil tabanlı yazılım geliştirme süreçlerinin yanında veriseti ile yapılan makine öğrenmesi teknikleri süreçleri bulunmaktadır. Bu yüzden tez çalışmasında gerçekleştirilen yöntemleri uygulama geliştirme süreçleri ve veri analiz süreçleri olarak iki aşamada değerlendirmek gerekmektedir. Gerekli web ve mobil uygulama servislerinin geliştirilmesi için öncelikle uygulamaların eskizleri (mockups) oluşturulmuştur. Şekil 3.2’de geliştirilen uygulamanın arayüzlerine ait taslak çalışmaları görülmektedir.



Şekil 3.2. Modelin arayüz taslakları

Tez çalışması ile ortaya konan servisler: üretim yönetim sistemine ait uygulama web ve mobil platformlar için ayrı uygulamalar olarak gerçekleştirilmiştir. Uygulama makine

parkurundaki tezgahlardan IoT sensörleri ile elde ettiği sayısal veriler üzerinde üretim esansında her bir personel ve makine başına yük analizi ve performans takibi yapabilmektedir. Üretimdeki ürünlerin hangi siparişe ait olduğu ve bu siparişin olası teslim edilme süreleri için geliştirilen uygulamanın toplanan veriler ve önceki deneyimler aracılığıyla akıllı tahminleme yapabilmesi sağlanmıştır. Çeşitli makine öğrenmesi teknikleri ile karar destek sistemi oluşturabilmek için bir yöntem geliştirilmiştir. Bu amaç doğrultusunda tez çalışmasında elde edilen üretim verileri ve eğitim için hazır veri kümeleri kullanılarak verisetleri oluşturulmuştur. Bu verisetleri her bir iş istasyonundan elde edilen üretim verilerini içermektedir. Elde edilen verisetine makine öğrenmesi algoritmaları uygulanarak, üretim riski analizi, personel ve makine/tezgâh bazından verimlilik analizi, makine/tezgâh ve personelin günlük olarak hedeflerine ait analizler yapılmıştır.

Tez çalışmasında kural tabanlı yöntemlerin haricinde Perceptron Öğrenme, Ridge Regresyon, Ensemble Öğrenme, GBM, SGD, Navie Bayes, XGBoost, Random Forest, LGBM, KNN, Entropi ve Regresyon Temelli Karar Ağacı, CatBoost ve AdaBoost algoritmaları kullanılmıştır. Kullanılan algoritmaların başarımı tez çalışmasının 4. bölümü olan “Bulgular ve Tartışma” bölümünde sunulmuştur.

3.5. Veriseti

Tez çalışmasında iki adet veriseti kullanılmıştır. Bunlardan birincisi mobilya fabrikası iç sistemleri, personel verileri, iş süreçleri gibi ham verilerden oluşan veritesidir. Veriseti Elazığ Organize Sanayi Bölgesinde faaliyeti gösteren bir mobilya fabrikasındaki tezgahlardan manuel yöntemler ile toplanmıştır. Veri tezgahların ürettiği sayısal veri ve personellerin tabletler üzerinden ürettiği veriler toplanarak oluşturulmuştur. Bu sayede fabrika içerisindeki tezgahların mevcut durumu anlık olarak analiz edilebilir hale gelmiştir. Veriseti oluşturulurken tezgahın çalışmaya başlama ve çalışma bitiş süreleri, duraklama süreleri otomatik olarak oluşturulurken, tezgahdaki işin kime ait olduğu bilgisi üretim için oluşturulan kare kodlar aracılığı ile takip edilmiş ve verisetine manuel olarak eklenmiştir. Bu veriseti üzerinden üretim bandındaki sipariş süreçleri, makine parkuru anlık iş yükü, tezgah bazlı performans süreçleri, üretim tabanlı süreçler ile ilgili kural tabanlı çıkarımlar gerçekleştirilmiştir. Üretim verilerinden oluşan ham veriseti Şekil 3.3’de görülmektedir.

```

33, 333333333333, 43, 939393939394, 34, 939393939394, 40, 243902439024, 43, 103448275862, 0, 50, 75, 40, 28, 571428571429, 68, 181818181818, 37, 037037037037, 40, 25, 3
20, 37, 5939846962406, 23, 770491803279, 40, 47619047619, 46, 551724137931, 0, 25, 20, 40, 28, 571428571429, 31, 818181818182, 11, 111111111111, 43, 333333333333, 0, 38,
40, 38, 931297709904, 29, 838709677419, 43, 20987654521, 47, 457627118644, 75, 0, 50, 20, 33, 333333333333, 39, 130434782609, 22, 222222222222, 48, 27586206896, 25, 37
33, 333333333333, 40, 769230769231, 24, 34, 567901234568, 44, 827586206897, 0, 0, 40, 66, 666666666667, 58, 333333333333, 17, 857142857143, 31, 03448275862, 12, 3, 4
20, 41, 98473284427, 30, 645161290323, 45, 679012345679, 39, 655172413793, 25, 25, 20, 33, 333333333333, 47, 82608956522, 30, 769230769231, 32, 258064518125, 37,
0, 41, 791044776119, 33, 884297520661, 42, 352941176471, 40, 51724137931, 25, 50, 25, 60, 42, 837142857143, 40, 909090909091, 23, 076923076923, 44, 827586206897, 25, 39
20, 33, 8345864661654, 31, 967213114754, 37, 34939759036, 41, 379310344828, 25, 25, 20, 33, 333333333333, 26, 086956521739, 25, 929292929292, 24, 137931034483, 25,
0, 38, 343846661654, 27, 808852459016, 43, 373493975904, 41, 379310344828, 25, 25, 0, 66, 666666666667, 45, 454545454545, 25, 20, 88965517241, 37, 5, 45, 454545454545
0, 40, 151515151515, 30, 894308943089, 32, 926629266293, 39, 655172413793, 0, 25, 50, 40, 50, 31, 818181818182, 29, 62962962963, 54, 818709677419, 12, 5, 45, 454545454545
33, 333333333333, 40, 35, 2, 36, 585305355609, 45, 689055172414, 50, 25, 0, 80, 16, 666666666667, 45, 454545454545, 35, 714285714286, 43, 333333333333, 25, 60, 666666666667
20, 43, 609022596391, 25, 20325203252, 41, 463414634146, 49, 565217391304, 20, 50, 25, 0, 33, 333333333333, 45, 833333333333, 25, 929292929292, 60, 12, 5, 51, 5151515151
20, 48, 484848484848, 29, 032258064516, 40, 243902439024, 44, 347826086957, 0, 0, 25, 20, 16, 666666666667, 40, 909090909091, 21, 428571428571, 66, 666666666667, 55, 55
20, 38, 345864661654, 21, 487603305785, 36, 585305355609, 43, 589743589744, 75, 50, 25, 16, 666666666667, 16, 666666666667, 40, 909090909091, 19, 230769230769, 41, 93
40, 40, 909090909091, 25, 32, 926629266293, 45, 217391304348, 0, 16, 666666666667, 25, 20, 50, 45, 454545454545, 7, 407407407407, 45, 161296822581, 25, 50, 45, 56, 25, 38
20, 43, 795620437566, 33, 613445378151, 45, 78313253012, 44, 347826086957, 25, 50, 75, 66, 666666666667, 50, 23, 076923076923, 41, 935483870968, 12, 5, 42, 837142857
40, 46, 989898989897, 26, 229508196721, 45, 679012345679, 42, 735042735043, 25, 40, 50, 0, 54, 545454545455, 18, 518518518519, 37, 931034482759, 12, 5, 48, 4848484848
0, 33, 587788259542, 27, 5, 32, 926629266293, 44, 166666666667, 0, 25, 25, 80, 33, 333333333333, 54, 545454545455, 19, 230769230769, 37, 931034482759, 25, 35, 2941176470
40, 35, 338345864662, 27, 049180327869, 51, 130476190476, 42, 341379310345, 50, 75, 25, 20, 50, 47, 82608956522, 19, 230769230769, 23, 333333333333, 44, 444444444444
40, 40, 151515151515, 39, 516129012258, 43, 373493975904, 50, 434782608956, 40, 50, 50, 40, 66, 666666666667, 47, 82608956522, 35, 714285714286, 46, 273820089866, 25,
50, 35, 114503816794, 40, 43, 90243902439, 40, 849665217391, 0, 50, 25, 80, 33, 333333333333, 36, 363636363636, 30, 769230769231, 34, 137931034483, 25, 35, 294117647059
0, 40, 151515151515, 28, 45518451846, 45, 78313253012, 49, 137931034483, 75, 25, 0, 20, 50, 45, 454545454545, 29, 62962962963, 37, 931034482759, 20, 45, 714285714286, 4
20, 41, 17647058235, 26, 666666666667, 43, 373493975904, 45, 217391304348, 20, 0, 0, 50, 56, 23, 076923076923, 44, 827586206897, 12, 5, 38, 235294117647, 47, 5, 43, 73, 3
20, 38, 345864661654, 29, 508196721311, 36, 470588235294, 37, 931034482759, 0, 25, 25, 60, 16, 666666666667, 36, 363636363636, 21, 428571428571, 37, 931034482759, 12, 5
20, 33, 834586466165, 31, 967213114754, 37, 34939759036, 44, 827586206897, 0, 25, 25, 40, 33, 333333333333, 65, 217391304348, 34, 615384615385, 33, 333333333333, 22, 2
0, 39, 69485648855, 29, 032258064516, 45, 679012345679, 43, 103448275862, 0, 0, 100, 33, 333333333333, 16, 666666666667, 54, 545454545455, 39, 285714285714, 40, 12, 5, 2
0, 42, 4242424242, 30, 527868852459, 35, 365853658537, 37, 80683796838, 50, 25, 25, 60, 33, 333333333333, 39, 130434782609, 25, 929292929292, 30, 25, 42, 424242424242
20, 41, 890465126279, 24, 8, 40, 243902439024, 34, 188034188034, 0, 0, 40, 20, 0, 40, 909090909091, 32, 142857142857, 27, 586208956522, 35, 393939393939, 48, 7048704
66, 666666666667, 33, 587788259542, 33, 605557177049, 28, 829268292683, 31, 35593220339, 50, 25, 0, 60, 33, 333333333333, 20, 088956521739, 14, 814814814815, 37, 9310
0, 39, 84962406015, 30, 327868852459, 41, 176470588235, 42, 341379310345, 50, 0, 25, 33, 333333333333, 16, 666666666667, 40, 909090909091, 30, 769230769231, 37, 93103
0, 33, 333333333333, 27, 808852459016, 31, 325301204819, 47, 863247863248, 0, 25, 50, 0, 50, 50, 090909090909, 25, 929292929292, 44, 827586206897, 0, 31, 428571428571, 37,
40, 35, 384613334615, 32, 539682539683, 41, 483414834146, 46, 95652173913, 0, 0, 50, 20, 50, 31, 034482758621, 41, 379310344828, 37, 5, 27, 272727272727, 56, 410256410
0, 44, 360902259639, 26, 829268292683, 44, 444444444444, 38, 260895652173, 50, 0, 40, 25, 47, 82608956522, 11, 111111111111, 34, 137931034483, 25, 33, 333333333333,
20, 38, 638363636364, 34, 714285714286, 34, 52380952381, 42, 372881355932, 50, 100, 75, 0, 14, 285714285714, 39, 130434782609, 23, 076923076923, 43, 333333333333, 37, 5
0, 36, 434108527132, 35, 714285714286, 33, 333333333333, 39, 655172413793, 0, 50, 40, 60, 16, 666666666667, 54, 545454545455, 17, 857142857142, 27586206896, 25, 29
40, 41, 666666666667, 32, 520325203252, 49, 397590361446, 38, 793103448276, 25, 50, 25, 60, 16, 666666666667, 39, 130434782609, 22, 222222222222, 41, 379310344828, 37,
20, 42, 222222222222, 33, 057851239669, 44, 578313253012, 46, 95652173913, 25, 50, 25, 0, 16, 666666666667, 40, 909090909091, 23, 076923076923, 33, 333333333333, 37, 5
20, 39, 89465648855, 23, 387096774194, 38, 55421686747, 37, 068965517241, 25, 0, 40, 33, 333333333333, 36, 363636363636, 29, 62962962963, 37, 931034482759, 22, 22222
20, 38, 401538481538, 27, 777777777777, 37, 80487804878, 42, 80893652174, 25, 100, 50, 20, 86, 866666666667, 45, 454545454545, 17, 261379310345, 24, 137931034483, 25

```

Şekil 3.3. Ham veriseti

İkinci veriseti tezgah verilerinden oluşan ve makine öğrenmesi teknikleri ile çıkarımlar gerçekleştirilen verisetidir. Czuriaga tarafından oluşturulan veriseti toplam 28 öznitelik ve 8774 ölçüm verisinden oluşmaktadır [45]. Veriseti üretim parkurundaki makinelerden alınan gerçek zamanlı verilerden oluşturulmuştur. Veriseti öznitelikleri Tablo 3.1’de gösterilmektedir. Veriseti makinelerden periyodik zamanlarda toplanan 15 adet farklı sensör bilgisi, ortamın sıcaklık ve nem bilgileri, makinenin çalışma ve durma zamanları ve oluşan arızalar ile ilgili bilgilerden oluşmaktadır. Tez çalışmasında verisetindeki arıza var / yok verisi üzerinden makine öğrenmesi modelleri uygulanarak sınıflandırma gerçekleştirilmiştir. Verisetine öncelikle ön işleme adımları uygulanarak tutarsız ve eksik bilgiler düzenlenmiştir. Eksik bilgiler ilgili öznitelik değerinin ortalaması alınarak doldurulmuştur. Veri ön işleme sürecinden sonra özniteliklerin ağırlıklarının tespit edilmesi ve etkisi bulunmayan özniteliklerin temizlenmesi için öznitelik seçim yöntemleri denenmiştir. Bu aşamadan sonra ise modeller verisetine uygulanmıştır. Çalışmada verisetinde bulunan 8774 verinin %80’i eğitim verisi olarak %20’si ise test verisi olarak kullanılmıştır. Uygulanan modellerde elde edilen başarımlar tez çalışmasının 4. Bölümü olan “Bulgular ve Tartışma” bölümünde değerlendirilmiştir.

Tablo 3.1. Veriseti öznitelikleri

Öznitelik Adı	Veri Tipi	Açıklama
Tarih	Tarih (Date-Time)	Tezgahın çalışma ve durma zamanı
Sıcaklık	Nümerik	Tezgahın çalışma esnasındaki ısı değeri
Nem	Nümerik	Tezgahın çalışma esnasındaki nem değeri
Operatör	Karakter	Çalışan operatör bilgisi (8 farklı operatör)
Sensör Verisi 1	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 2	Karakter	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 3	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 4	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 5	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 6	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 7	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 8	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 9	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 10	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 11	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 12	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 13	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 14	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Sensör Verisi 15	Nümerik	Periyodik zamanlarda yapılan ölçüm verileri
Arızadan Sonra Geçen Süre	Nümerik	Tezgahın arızadan arızanın giderilmesine kadar geçen süre
Arıza	Karakter	Arıza var/yok bilgisi
Tarih (Yıl)	Tarih	Arıza yıl bilgisi
Tarih (Ay)	Tarih	Arıza ay bilgisi (12 ay)
Tarih (Gün)	Tarih	Arıza gün bilgisi (31 gün)
Tarih (Hafta)	Tarih	Arıza haftanın gün bilgisi (7 gün)
Saat	Tarih	Arıza saat bilgisi (24 saat)
Dakika	Tarih	Arıza dakika bilgisi
Saniye	Tarih	Arıza saniye bilgisi

4. BULGULAR VE TARTIŞMA

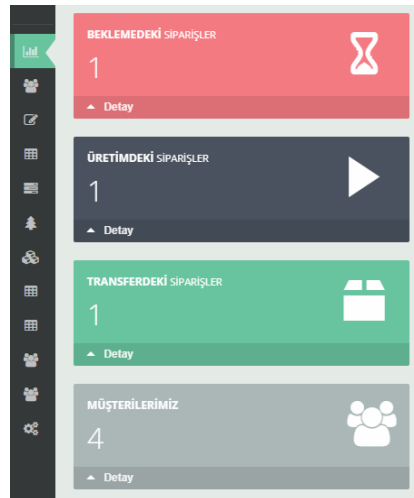
Bu bölümde tez çalışması sonucu elde edilen bulgular ortaya konularak başarımlar değerlendirilmiştir. Gerçekleştirilen çalışmada fabrika verileri ile oluşturulan 1. veriseti üzerinden kural tabanlı çıkarımlar ve tezgah verilerinin barındırıldığı veriseti üzerinden makine öğrenmesi teknikleri ile üretim süreçleri ile ilgili arıza kestirimi yapılmıştır. 2. veriseti üzerinden ise öncelikle özellik çıkarımı yapılarak önemli öznitelikler belirlenmiş ve modele makine öğrenmesi algoritmalarından Perceptron Öğrenme, Ridge Regresyon, Ensemble Öğrenme, GBM, SGD, Navie Bayes, XGBoost, Random Forest, LGBM, KNN, Entropi ve Regresyon Temelli Karar Ağacı, CatBoost ve AdaBoost algoritmaları uygulanarak sonuçları karşılaştırılmıştır. Verisetlerinde aşırı öğrenme problemi yaşanmaması için önlemler alınmış ve aşırı öğrenmeyi engelleyecek algoritmalar tercih edilmiştir. Aşağıdaki bölümlerde tez çalışmasının sonuçları sunulmaktadır.

4.1. Kural Tabanlı Modeller

Yapılan tez çalışmasında öncelikle üretim ve fabrika süreç verilerinden kural tabanlı çalışmalar gerçekleştirilmiştir. Gerçekleştirilen çalışmalar aşağıda sunulmaktadır.

4.1.1. Üretim Bandı Mevcut Durum

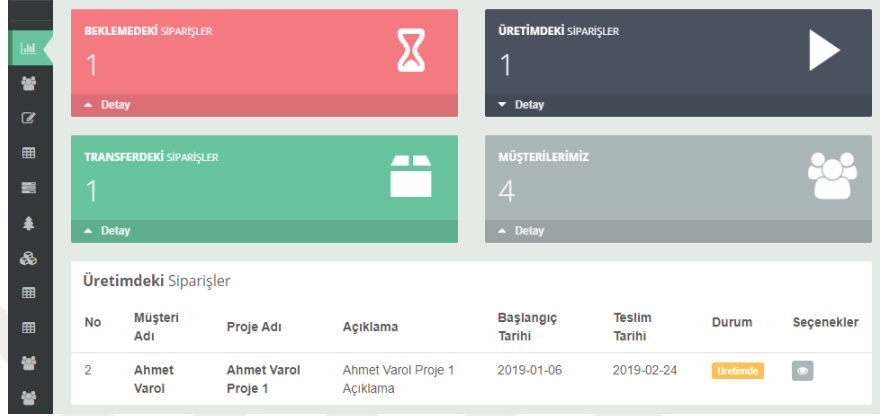
Özellikle üretim yapan işletmelerdeki önemli sorunlardan biride üretilen ürünün hangi tezgahta ya da üretimin hangi aşamasında olduğunun anlık takip edilememesidir. Geliştirilen uygulama ile bu problem ortadan kaldırılmıştır. Şekil 4.1’de işletmenin makine parkuruna ait üretim bandı mevcut durumu görülmektedir. Bu arayüzde: bekleyen, üretimde olan ve tamamlanıp transfer sürecinde olan tüm ürünler anlık ve periyodik olarak görüntülenebilmektedir.



Şekil 4.1. Üretim bandı mevcut durum

4.1.2. Üretim Bandı Sipariş Takibi

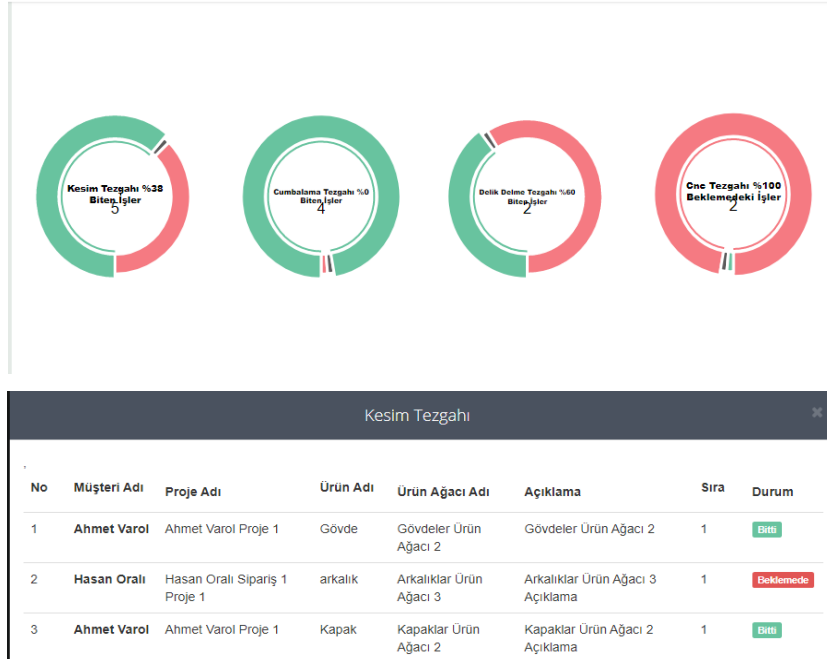
Üretim sürecinde makine parkurundaki tezgahlarda yapılmış, yapılacak ya da devam eden işlerin anlık takibi ve anlık olarak buradaki sürecin yönetilmesi istenmektedir. Şekil 4.2’de Üretim Bandındaki Siparişlerin Listesi görülmektedir. Bu sayede tezgah başındaki personel hangi ürünü ne zaman üreteceğini bilmekte, yönetici ise anlık olarak üretimi takip edebilmekte ve ölçümleyebilmektedir.



Şekil 4.2. Üretim bandındaki siparişlerin listesi

4.1.3. Makine Parkuru İş Yüğü

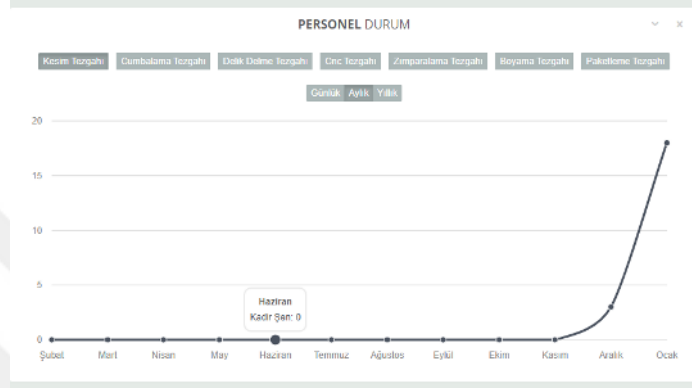
Şekil 4.3’de makine parkurundaki tüm tezgahların anlık iş yükleri görülmektedir. Geliştirilen algoritma tezgahın bir ürün için işe başladığı ve işin tamamlandığı süreleri, makinenin çalışmadığı ya da arızalı olduğu süreleri, geçmiş dönemdeki periyotlar ile karşılaştırarak tahminlemelerde ve iş yükünün yönetimini gerçekleştirmektedir.



Şekil 4.3. Makine parkuru anlık iş yükü

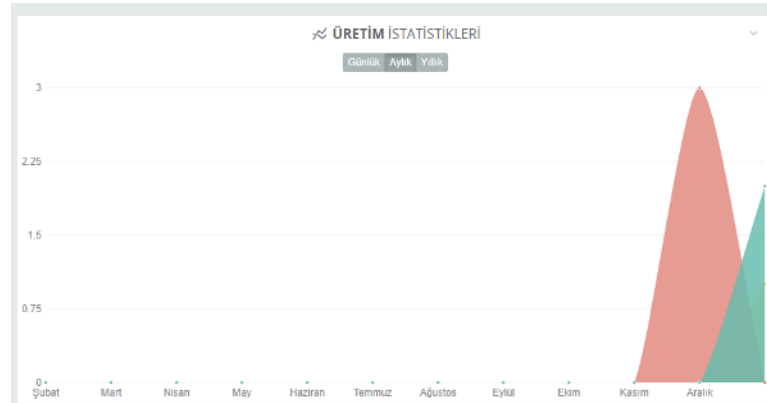
4.1.4. Tezgah Bazlı Performans Takibi

Şekil 4.4’de Tezgah bazlı personel performans takip ekranı görülmektedir. Makine parkurunda bir tezgahı aynı periyot içinde ya da farklı periyotlarda farklı personeller kullanabilir. Geliştirilen uygulama ile anlık olarak makine parkuru ve personel ilişkisi oluşturulmuştur. Bu sayede personele ait performans takibi, geçmiş dönemlerdeki üretim ile anlık karşılaştırmaların yapılabilmesi, aynı tezgahdaki farklı personellerin karşılaştırmaların yapılması ile performans ölçüm ve analizleri yapılabilir.



Şekil 4.4. Tezgah bazlı personelin performans takibi

İşletmenin üretim süreçleri, anlık, günlük, haftalık, aylık ve yıllık bazda takip edilebilmektedir. Bu da işletmenin üretim süreçlerinin hangi dönemlerde yoğun hangi dönemlerde ise boş olduğunu tahminlenmemizi sağlamaktadır. Şekil 4.5’de işletmenin üretim bandındaki üretim kapasitesine ait istatistikler görülmektedir.



Şekil 4.5. Üretim İstatistikleri

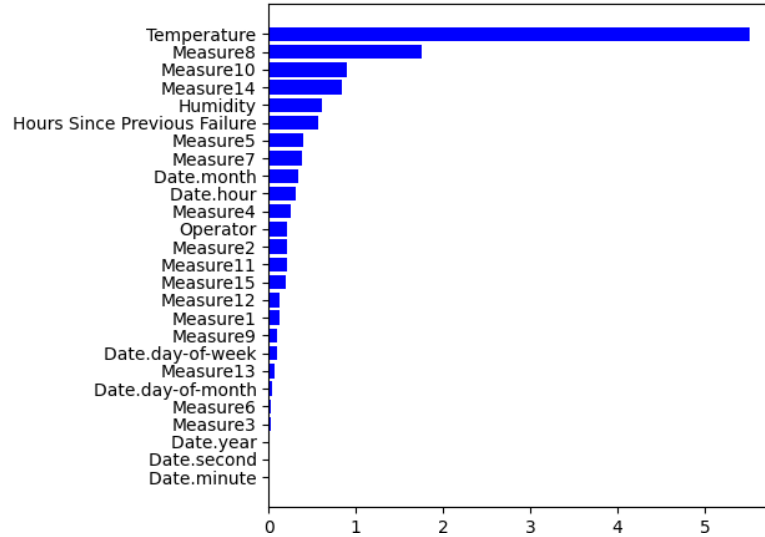
4.2. Makine Öğrenmesi Yöntemleri ile Kestirimci Arıza Tespit Modeli

Tez çalışmasında tezgah verileri üzerinden oluşturulan verisetine makine öğrenmesi yöntemleri uygulanarak kestirimci arıza tespit modeli oluşturulmuştur. Çalışmada Perceptron Öğrenme Algoritması, Ridge Regresyon Algoritması, Ensemble Öğrenme Algoritması, GBM Algoritması, SGD Algoritması, Navie Bayes Algoritması, XGBoost Algoritması, Random Forest

Algoritması, LGBM Algoritması, KNN Algoritması, Entropi ve Regresyon Temelli Karar Ağacı Algoritmaları, CatBoost Algoritması ve AdaBoost Algoritmaları kullanılarak başarımları karşılaştırılmıştır. Yapılan çalışmada öncelikle verisetindeki özniteliklerin seçimi ile ilgili çalışma yapılarak önemli olmayan öznitelikler çıkarılmıştır. Bu işlemten sonra makine öğrenmesi algoritmaları verisetine uygulanarak doğruluk oranları ve karmaşıklık matrisi üzerinden çeşitli performans metrikleri ortaya konularak bulgular tartışılmıştır.

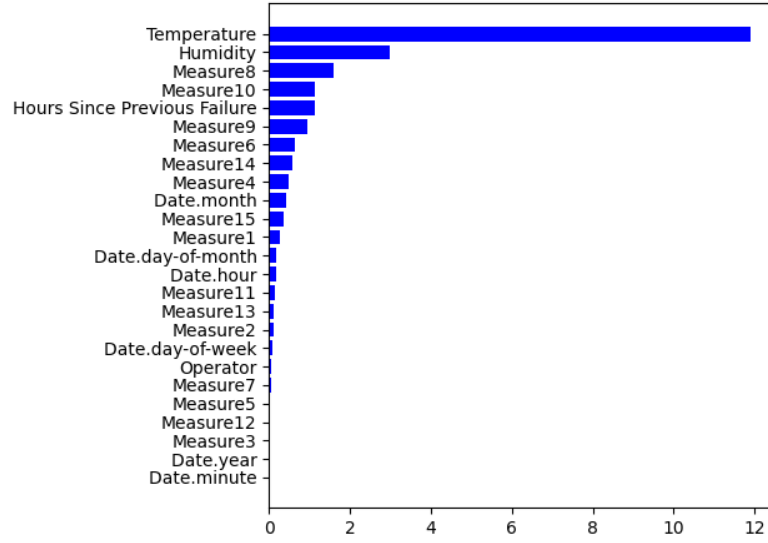
4.2.1. Öznitelik Seçimi

Çalışmada 28 öznitelik ve 8774 ölçüm verisinden oluşan ölçüm veriseti kullanılmıştır. Alınan ölçümler ve diğer öznitelik verilerinin etkinliğini anlamak ve modelin başarımını arttırabilmek için öncelikle model tabanlı öznitelik seçimi yapılmıştır. Öznitelik seçimi yapabilmek için model tabanlı öznitelik seçimi yapabilen SHAP (SHapley Additive Explanations) tekniği tercih edilmiştir [46]. Bu teknikte özniteliklerin verisetine ve dolayısı ile tahmine katkısı hiper parametrelerden bağımsız olarak belirlenebilmektedir. SHAP tekniği model tabanlı çalıştığı için özniteliklerin seçimi için kullanılacak olan modeller ile birlikte özellik seçimi yapılmıştır. Şekil 4.6’da CatBoost modeli üzerinde SHAP tekniği ile belirlenen öznitelik ağırlıkları görülmektedir. Şekilde görüldüğü gibi sıcaklık, nem, 8, 10 ve 14. sensör ölçümlerinin modele katkısının yüksek olduğu, dakika, saniye, yıl bilgisi, 3 ve 6. sensör ölçüm ağırlıklarının modele katkısının çok düşük olduğu tespit edilmiştir.



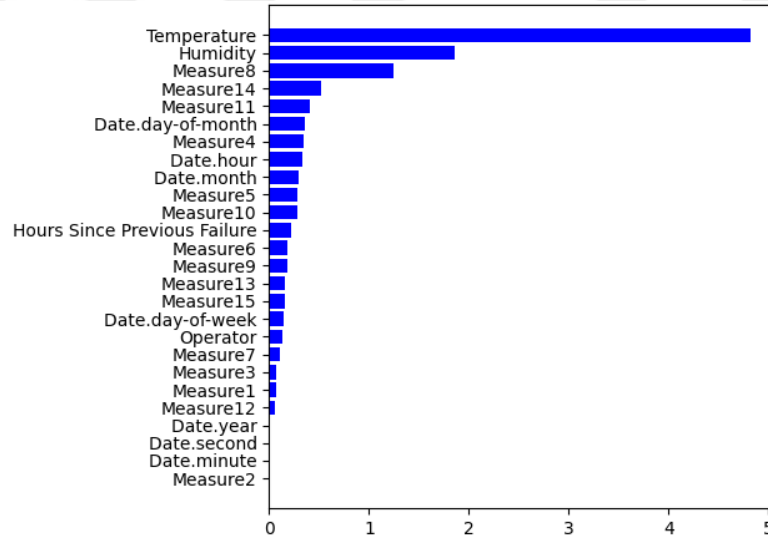
Şekil 4.6. CatBoost modeli ve SHAP tekniği ile özniteliklerin ağırlıkları

Şekil 4.7’de ise LightGBM modeli ve SHAP tekniği kullanılarak bulunan öznitelik ağırlıkları görülmektedir. Bu modelde de benzer şekilde sıcaklık ve nem bilgisi ile birlikte 8 ve 10 numaralı sensör bilgilerinin öznitelik seçimi bakımından en etkin olan öznitelikler olduğu, yıl, dakika, saniye ve 12 ile 13 numaralı sensör verilerinin öznitelik seçimi açısından etkisiz olduğu görülmüştür.



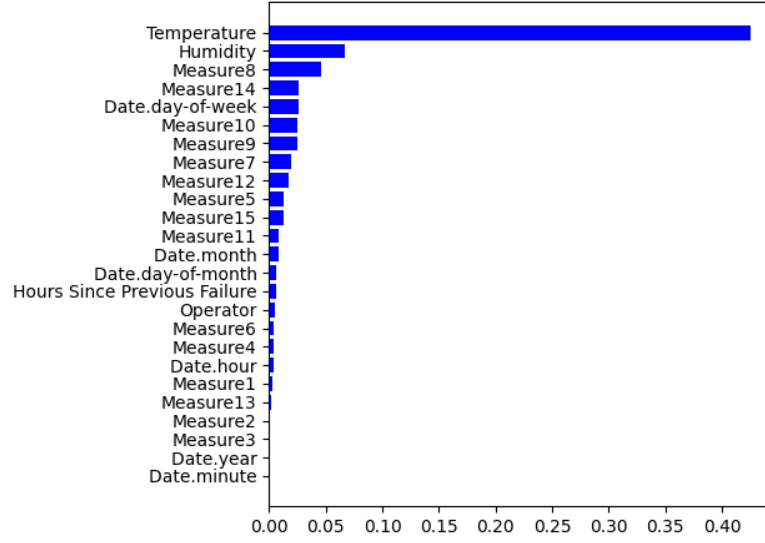
Şekil 4.7. LightGBM modeli ve SHAP tekniği ile özniteliklerin ağırlıkları

Şekil 4.8’de XGBoost modeli üzerinde SHAP tekniği ile öznitelik seçimi yapılmak istendiğinde ortaya çıkan öznitelik ağırlıkları görülmektedir. Bu modelde de SHAP tekniği ile gerçekleştirilen öznitelik seçiminde yaklaşık olarak benzer sonuçlar elde edilmiştir. Yine sıcaklık, nem gibi özellikler ile birlikte 8 ve 14 numaralı sensör verileri öznitelikler içindeki en önemli parametreler olduğu, saniye, dakika, yıl gibi bilgiler ile birlikte 2 ve 12 numaralı sensör verilerinin öznitelik seçiminde etkisiz kaldığı gözlemlenmiştir.



Şekil 4.8. XGBoost modeli ve SHAP tekniği ile özniteliklerin ağırlıkları

Son olarak öznitelik seçiminde SHAP tekniği ve Rastgele Orman (Random Forest) algoritması ile çalışma yapılmıştır. Çalışma Şekil 4.9’da görülmektedir. Bu çalışmada da benzer sonuçlar elde edildiği görülmüştür. Sıcaklık ve nem bilgileri ile birlikte önceki modellerde de olduğu gibi 8 ve 14 numaralı sensör özniteliklerinin etkin olduğu, yıl ve dakika gibi bilgilerin etkisiz olduğu gözlemlenmiştir.



Şekil 4.9. Rastgele Orman Algoritması ve SHAP tekniği ile özniteliklerin ağırlıkları

SHAP tekniği ve modeller üzerinde yapılan öznitelik seçimi işleminde özellikle nem ve sıcaklık verilerinin arıza tespiti için önemli bilgiler olduğu, yıl, saniye ve dakika bilgilerinin ise etkinliğinin son derece az olduğu görülmüştür.

4.2.2. Modellerin Uygulanması ve Sonuçlar

Tez çalışmasında öncelikle etkinliği az olan bu öznitelikler verisetinden temizlenmiştir. Temizlenen veriseti üzerinde Perceptron Öğrenme, Ridge Regresyon, Ensemble Öğrenme, GBM, SGD, Navie Bayes, XGBoost, Random Forest, LGBM, KNN, Entropi ve Regresyon Temelli Karar Ağacı, CatBoost ve AdaBoost algoritmaları kullanılarak modeller geliştirilmiş ve başarımları ortaya konmuştur. Çalışmada 28 öznitelik ve 8774 ölçüm verisinden oluşan verisetinin %80'i eğitim %20'si ise test verisi olarak belirlenerek arıza kestirimi için bir model ortaya konulmuştur. 8774 ölçüm verisinin %20'si yani 1757 test verisi üzerinde model başarımını ortaya koyabilmek için karmaşıklık matrisi ve Kesinlik, Duyarlılık, f1-score ve destek parametreleri analiz edilmiştir. Şekil 4.10'da modellere ait karmaşıklık matrisleri ve performans metrikleri ve Tablo 4.1'de çalışmada uygulanan modeller ve başarımları verilmiştir.

Gerçek Sonuçlar	0	1736	3
	1	7	11
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	0,79	0,61	0,69	18

a) Perceptron Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	18	0
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	0,99	1,0	0,99	1739
1	0,00	0,00	0,00	18

b) Ridge Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1738	1
	1	8	10
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	0,91	0,56	0,69	18

c) Gradient Boosting Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1736	3
	1	9	9
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	0,99	1,0	1,0	1739
1	0,75	0,50	0,60	18

d) Stochastic Gradient Descent Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	9	9
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	0,99	1,0	1,0	1739
1	1,0	0,50	0,67	18

e) NaiveBayes Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	6	12
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	1,0	0,67	0,8	18

f) XGBoost Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	6	12
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	1,0	0,67	0,8	18

g) Random Forest Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	12	6
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	0,99	1,0	1,0	1739
1	1,0	0,33	0,5	18

h) MLP Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	6	12
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	1,0	0,67	0,8	18

i) LightGBM Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	17	1
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	1,0	0,72	0,84	18

j) KNN Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1731	8
	1	6	12
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	0,6	0,67	0,63	18

j) Decision Tree (J45) Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1734	5
	1	5	13
		0	1

Tahmin Sonuçları

Kesinlik Duyarlılık F1-Skor Destek

0	1,0	1,0	1,0	1739
1	0,72	0,72	0,72	18

k) Decision Tree (Gini) Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	6	12
		0	1

Tahmin Sonuçları

	Kesinlik	Duyarlılık	F1-Skor	Destek
0	1,0	1,0	1,0	1739
1	1,0	0,67	0,8	18

1) CatBoost Karmaşıklık Matrisi ve Performans Metrikleri

Gerçek Sonuçlar	0	1739	0
	1	6	12
		0	1

Tahmin Sonuçları

	Kesinlik	Duyarlılık	F1-Skor	Destek
0	1,0	1,0	1,0	1739
1	1,0	0,67	0,8	18

m) AdaBoost Karmaşıklık Matrisi ve Performans Metrikleri

Şekil 4.10. Modellere ait karmaşıklık matrisleri ve performans metrikleri

Tablo 4.1. Model başarımları

Model Adı	Model Parametreleri	Model Doğruluğu (Accuracy)
Perceptron	{}	0.99430
Ridge Sınıflandırıcısı	{ 'alpha': 0.187381 }	0.98975
Gradient Boosting	{ "n_estimators": 400, "learning_rate": 0.1, "max_depth": 5, "min_samples_leaf": 5, "random_state": 0, "verbose": true }	0.99487
Stochastic Gradient Descent	{ 'alpha': 0.0001 }	0.99317
Navie Bayes	{}	0.99487
XGBoost Sınıflandırıcısı	{}	0.99688
Random Forest	{ "n_estimators": 1000, "n_jobs": -1 }	0.99648
MLP Sınıflandırıcısı	{ 'max_iter': 500, 'alpha': 0.0001, 'solver': 'sgd' }	0.99317
LightGBM Sınıflandırıcısı	{ "boosting_type": "gbdt", "class_weight": null, "colsample_bytree": 1.0, "importance_type": "split", "learning_rate": 0.1, "max_depth": -1, "min_child_samples": 20, "min_child_weight": 0.001, "min_split_gain": 0.0, "n_estimators": 100, "n_jobs": -1, "num..." }	0.99715
KNN Sınıflandırıcısı	{ 'n_neighbors': 3 }	0.99032
Karar Ağacı – J45	{ "criterion": "entropy", "max_depth": 20, "random_state": 0 }	0.99203
Karar Ağacı – Gini	{ "criterion": "gini", "max_depth": 50, "random_state": 0 }	0.99430
CatBoost Sınıflandırıcısı	{ "iterations": 1000, "learning_rate": 0.1, "random_seed": 123, "allow_writing_files": false }	0.99674
Adaboost Sınıflandırıcısı	{ "n_estimators": 200, "learning_rate": 1, "random_state": 0 }	0.99658

Şekil 4.10 ve Tablo 4.1’de de açıkça görüldüğü gibi arıza kestirim için oluşturulan model doğruluğu %99715 ile LightGBM algoritması ile gerçekleşmiştir. Çalışmada geliştirilen modelde öncelikle 28 öznitelik ve 8774 ölçüm verisinden oluşan veriseti üzerinde öznitelik seçimi yapılarak etkinliği olmayan saniye, dakika ve yıl öznitelikleri çıkarılarak öznitelik sayısı 25’e düşürülmüştür. Modelin aşırı öğrenmesini engellemek için veri kümesindeki veriler üzerinde çoğaltma işlemi yapılarak arıza verileri 4 katına çıkarılmıştır. Bu işlemden sonra oluşan nihai veriseti üzerinde makine öğrenmesi algoritmaları uygulanmıştır. Çalışma sonrası çıkan sonuçlar incelendiğinde modelin başarılı olduğu gözlemlenmiştir. Çalışmada, bir fabrikadaki tezgahlardan alınacak olan düzenli veriler ile tezgahta oluşabilecek olan arızaların önceden tahmin edilmesine olanak sağlayacak bir model ortaya konmuştur.



5. SONUÇ

Bu tez çalışmasında özellikle endüstri 4.0 ile birlikte oluşan yeni teknolojiler ve bu teknolojilerin oluşturmuş olduğu büyük veriye ilişkin çözümler incelenerek endüstri 4.0 entegrasyonu için bulut tabanlı üretim yönetim sistemi önerilmiştir.

Teknolojinin gelişmesi ile birlikte hemen hemen hayatımızın her alanında kullanılan teknolojiler endüstriyel alanlarda da karşılık bulmaya başlamıştır. Büyük veri çözümlerine olan ihtiyaç her geçen gün artmaya başlamıştır. Yapılan çalışma özellikle büyük veri altyapısı kurgulamak isteyen kurum ve kuruluşlara rehber olacak şekilde ortaya konulmuştur. Bununla beraber sektörün ihtiyaçları doğrultusunda yapılan çalışmalar ve literatür incelenerek gelecek çalışmalar hakkında bir yol haritası çıkarılmıştır. Yapılan çalışmada özellikle endüstri 4.0'ı oluşturan bileşenlerin tüm endüstriyel alanlarda tek düze olmadığı, uygulamalarda farklılıklar olduğu görülmüştür. Bu da verinin ölçeklenmesinden güvenliğine, ihtiyaç analizinden karar destek sistemlerine kadar farklılıklar oluşmasına ve nihai ürün için oluşturulan iş akışlarında farklılıklar oluşmasına neden olmuştur. Dördüncü sanayii devriminin henüz başlarında olan ülkemiz işletmelerinin ivme kazanabilmesi için yapay zeka destekli çözümlerin ortaya konulması gerekliliği görülmüştür. Bu bağlamda yapılan tez çalışmasında endüstri 4.0 tabanlı imalat gerçekleştiren ya da gerçekleştirme hedefinde olan işletmelerin doğrudan kullanabileceği yapay zeka ve makine öğrenme algoritmaları kullanan bir karar destek sistemi ortaya konmuştur. Gerçekleştirilen tez çalışması mevcut üretim yönetim sistemlerine birçok özgün değeri katacak niteliktedir. Çalışmada gerçekleştirilen çözümler ile hibrit yazılım altyapısına uyumlu olabilmek bilgisayar kontrollü olmayan üretim tezgahlarındaki verilerinde sayısallaştırılması, toplanması ve analizinin yapılabilmesini sağlayabilmek, makine parkuru ve personel performanslarının takibini üretim reçetesinden bağımsız bir şekilde gerçekleştirebilmek, yapay zeka altyapısı ile iş zekası çözümü ve sayısallaştırılmış veriler üzerinden üretim yönetimi ve risk / fırsat analizi gerçekleştirebilmek gibi birçok kabiliyet kazandırılmıştır.

Tez çalışması ile gerçekleştirilen model fabrikadaki tezgahlardan alınan IoT verileri, üretim bandı verileri, tezgah iş yükü bilgisi, personel görev ve iş yükü bilgileri, üretim verileri gibi anlık olarak akan ham veriler üzerinden kural tabanlı çözümler ortaya ve tezgah üzerinden alınmış sıcaklık, çalışma süreleri vb. gibi sensör verilerden oluşan veriseti üzerinde makine öğrenmesi teknikleri ile arıza kestirimi yapabilecek bir karar destek sistemi oluşturularak başarıyı ortaya konmuştur.

KAYNAKLAR

- [1] Dokuz, A. Ş., Çelik, M. (2017). "Bulut Bilişim Sistemlerinde Verinin Farklı Boyutları Üzerine Derleme" Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi 6, 316-338.
- [2] Pişirir, E., Sevgi, C., Chouseinoglou, O., Uçar, E. (2018). Türkiye’de Yazılım Geliştirme Takımlarının Bulut Kullanımı, Kullanım İsteği ve Uygunluk Algısı
- [3] Alam, T., (2021). Cloud Computing and its role in the Information Technology. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 1, 108-115.
- [4] Arwa M., Mosab H., Suleman K., Ahmed A., Sharief F. B., Muhammad I., Marsono M.N., (2021). Software-defined networks for resource allocation in cloud computing: A survey, Computer Networks, Volume 195.
- [5] Jaswinder S., Gaurav D., (2014). A survey on cloud computing approaches, Materials Today, 2021
- [6] Çetiner Y., Bulut Bilişim ve Örnek Bir SaaS Uygulaması (Yüksek Lisans Tezi), Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü.
- [7] Bulut, E., Akçacı, T., (2017). Endüstri 4.0 ve inovasyon göstergeleri kapsamında Türkiye analizi. ASSAM Uluslararası Hakemli Dergi, cilt 4(7), ss. 55-77.
- [8] Özsoylu, A. F., (2017). Endüstri 4.0. Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21(1), 41-64.
- [9] Yıldız, A., (2018). Endüstri 4.0 ve Akıllı Fabrikalar. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22(2), 546-556.
- [10] Pamuk, N. S., Soysal, M., (2018). Yeni Sanayi Devrimi Endüstri 4.0 Üzerine Bir İnceleme. Verimlilik Dergisi, (1), 41-66.
- [11] Kayar, A., Ayvaz, B., Öztürk, F., (2018). Akıllı Fabrikalar, Akıllı Üretim: Endüstri 4.0’a Genel Bakış. In International Eurasian Conference on Science, Engineering and Technology (EurasianSciEnTech) (pp. 22-23).
- [12] Calp, M. H., Bahçekapılı, E., Berigel, M., (2019). Endüstri 4.0 Kapsamında Akıllı Fabrikaların İncelenmesi.
- [13] Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., ... & Zdeborová, L. (2019). Machine learning and the physical sciences. Reviews of Modern Physics, 91(4).
- [14] Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.
- [15] Wei, J., Chu, X., Sun, X. Y., Xu, K., Deng, H. X., Chen, J., ... & Lei, M. (2019). Machine learning in materials science. InfoMat, 1(3), 338-358.
- [16] Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2016). Machine learning basics. Deep learning, 98-164.
- [17] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195-202.
- [18] Sra, S., Nowozin, S., & Wright, S. J. (Eds.). (2012). Optimization for machine learning. Mit Press.

- [19] Sammut, C., & Webb, G. I. (Eds.). (2011). Encyclopedia of machine learning. Springer Science & Business Media.
- [20] Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Information Fusion*, 50, 92-111.
- [21] Schmitta, A., Carliera, F., Renault, V., (2018). Dynamic bridge generation for IoT data exchange via the MQTT protocol, The 9th International Conference on Ambient Systems, Networks and Technologies.
- [22] Cucu, P. (2018). All You Need to Know and Apply, Nedir?. Alındığı yer: <https://www.kocdigital.com/uzmanliklarimiz/iot-nedir>.
- [23] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018, July). Machine learning approach for predictive maintenance in industry 4.0. In 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA) (pp. 1-6). IEEE.
- [24] Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, 59(16), 4773-4778.
- [25] Penumuru, D. P., Muthuswamy, S., & Karumbu, P. (2020). Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing*, 31(5), 1229-1241.
- [26] Brik, B., Bettayeb, B., Sahnoun, M. H., & Duval, F. (2019). Towards predicting system disruption in industry 4.0: machine learning-based approach. *Procedia computer science*, 151, 667-674.
- [27] Lee, C., & Lim, C. (2021). From technological development to social advance: A review of Industry 4.0 through machine learning. *Technological Forecasting and Social Change*, 167.
- [28] Khayyam, H., Jamali, A., Bab-Hadiashar, A., Esch, T., Ramakrishna, S., Jalili, M., & Naebe, M. (2020). A novel hybrid machine learning algorithm for limited and big data modeling with application in industry 4.0. *IEEE Access*, 8, 111381-111393.
- [29] Penumuru, D. P., Muthuswamy, S., & Karumbu, P. (2020). Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing*, 31(5), 1229-1241.
- [30] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018, July). Machine learning approach for predictive maintenance in industry 4.0. In 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA) (pp. 1-6). IEEE.
- [31] Romeo, L., Loncarski, J., Paolanti, M., Bocchini, G., Mancini, A., & Frontoni, E. (2020). Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. *Expert Systems with Applications*, 140, 112869.

- [32] Tran, M. Q., Elsis, M., Mahmoud, K., Liu, M. K., Lehtonen, M., & Darwish, M. M. (2021). Experimental setup for online fault diagnosis of induction machines via promising IoT and machine learning: Towards industry 4.0 empowerment. *IEEE Access*, 9, 115429-115441.
- [33] Ansari, F., Erol, S., & Sihn, W. (2018). Rethinking human-machine learning in industry 4.0: how does the paradigm shift treat the role of human learning?. *Procedia manufacturing*, 23, 117-122.
- [34] Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. *Computers in Industry*, 123, 103298.
- [35] Li, Y., Carabelli, S., Fadda, E., Manerba, D., Tadei, R., & Terzo, O. (2020). Machine learning and optimization for production rescheduling in Industry 4.0. *The International Journal of Advanced Manufacturing Technology*, 110(9), 2445-2463.
- [36] Carvajal Soto, J. A., Tavakolizadeh, F., & Gyulai, D. (2019). An online machine learning framework for early detection of product failures in an Industry 4.0 context. *International Journal of Computer Integrated Manufacturing*, 32(4-5), 452-465.
- [37] Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19), 8211.
- [38] Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Information Fusion*, 50, 92-111.
- [39] Calabrese, M., Cimmino, M., Fiume, F., Manfrin, M., Romeo, L., Ceccacci, S., ... & Kapetis, D. (2020). SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. *Information*, 11(4), 202.
- [40] Carletti, M., Masiero, C., Beghi, A., & Susto, G. A. (2019, October). Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis. In *2019 IEEE international conference on systems, man and cybernetics (SMC)* (pp. 21-26). IEEE.
- [41] Jafery, W. A. Z. W. C., Omar, M. S. S., Ahmad, N. A., & Ithnin, H. (2019, December). Classification of Patents according to Industry 4.0 Pillars using Machine Learning Algorithms. In *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)* (pp. 1-6). IEEE.
- [42] Bortolini, M., Ferrari, E., Gamberi, M., Pilati, F., & Faccio, M. (2017). Assembly system design in the Industry 4.0 era: a general framework. *IFAC-PapersOnLine*, 50(1), 5700-5705.
- [43] Dogan, O., & Gurcan, O. F. (2018, July). Data perspective of Lean Six Sigma in industry 4.0 Era: a guide to improve quality. In *Proceedings of the international conference on industrial engineering and operations management Paris*.
- [44] Bagheri, B., Yang, S., Kao, H. A., & Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine*, 48(3), 1622-1627.
- [45] Machine Failures Dataset, <https://bigml.com/user/czuriaga>, Erişim: 1 Şubat 2022

- [46] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.



EKLER

EK- 1: SHAP VE RASTGELE ORMAN ALGORİTMASI KULLANARAK ÖZNİTELİK SEÇİMİ KAYNAK KODLARI

```
def plot_feature_importance(self, _dataset, feature_names, shap_values):
    feature_selection_result_folder = _dataset.values.feature_selection_result_folder

    # Get the mean absolute contribution for each feature
    aggregate = np.mean(np.abs(shap_values[:, 0:-1]), axis=0)

    # sort by magnitude
    z = [(x, y) for y, x in sorted(zip(aggregate, feature_names), reverse=False)]
    z = list(zip(*z))
    # plt.figure(figsize=(14, 14))
    plt.title('Feature Importance')
    plt.barh(z[0], z[1], color='b', align='center')
    # plt.xticks(rotation=90)
    plt.tight_layout()
    time_string = time.strftime("%d%m%Y_%H%M%S")
    file_name_image = "{0}/{1}_{2}.png".format(feature_selection_result_folder, "shap_random_forest", time_string)
    plt.savefig(file_name_image)
    # plt.show()

def shap_random_forest(self, dataset):
    _dataset = DataProcess(dataset)
    all_data_df = _dataset.values.all_data_df
    temp_df = all_data_df
    X = _dataset.values.X
    y = _dataset.values.y

    split_dict = {"test_size": 0.2,
                  "random_state": 7}

    # create a train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, **split_dict)

    scaler = StandardScaler()

    scaler.fit(X_train)

    X_train = scaler.transform(X_train)

    random_forest = RandomForestClassifier(n_estimators=10, n_jobs=-1)

    random_forest.fit(X_train, y_train)

    explainer = shap.TreeExplainer(random_forest)

    shap.initjs()

    #shap_values = explainer.shap_values(X.loc[[100]])

    #shap.summary_plot(shap_values, X)

    shap_values = explainer.shap_values(X)

    features = _dataset.values.features

    feature_names = list(X.columns.values)
    vals = np.abs(shap_values).mean(0)
    self.plot_feature_importance(_dataset, feature_names, vals)
```

EK- 2: SHAP VE CATBOOST ALGORİTMASI KULLANARAK ÖZNİTELİK SEÇİMİ KAYNAK KODLARI

```
def plot_feature_importance(self, _dataset, feature_names, shap_values):
    feature_selection_result_folder = _dataset.values.feature_selection_result_folder

    # Get the mean absolute contribution for each feature
    aggregate = np.abs(shap_values[:,])

    # sort by magnitude
    z = [(x, y) for y, x in sorted(zip(aggregate, feature_names), reverse=False)]
    z = list(zip(*z))
    #plt.figure(figsize=(14, 14))
    plt.title('Feature Importance')
    plt.barh(z[0], z[1], color='b', align='center')
    # plt.xticks(rotation=90)
    plt.tight_layout()
    time_string = time.strftime("%d%m%Y_%H%M%S")
    file_name_image = "%d/%d/%d.png" % (feature_selection_result_folder, "shap_catboost", time_string)
    plt.savefig(file_name_image)
    # plt.show()

def shap_catboost(self, dataset):
    _dataset = DataProcess(dataset)
    all_data_df = _dataset.values.all_data_df
    temp_df = all_data_df
    X = _dataset.values.X
    y = _dataset.values.y

    split_dict = {"test_size": 0.2,
                  "random_state": 7}

    # create a train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, **split_dict)

    scaler = StandardScaler()

    scaler.fit(X_train)

    X_train = scaler.transform(X_train)

    #allow_writing_files=False catboost_info klasorunu olusturmamak icin
    model = CatBoostClassifier(iterations=1000, learning_rate=0.1, random_seed=123, allow_writing_files=False)
    model.fit(X_train, y_train, verbose=False, plot=False)
    explainer = shap.TreeExplainer(model)
    shap_values = explainer.shap_values(X)

    #shap.summary_plot(shap_values, X)

    features = _dataset.values.features

    feature_names = list(X.columns.values)
    vals = np.abs(shap_values).mean(0)
    # feature_importance = pd.DataFrame(list(zip(feature_names, vals)), columns=['col_name', 'feature_importance_vals'])
    self.plot_feature_importance(_dataset, feature_names, vals)

    #self.plot_feature_importance(features, shap_values[0])
```

EK- 3: ADABOOST MODELİ KAYNAK KODLARI

```
def create_model_adaboost(self, dataset, defined_parameters=False, sample_size=None):

    _dataset = DataProcess(dataset)

    all_data_df = _dataset.values.all_data_df

    X = []

    y = []

    if sample_size is not None:
        X = _dataset.values.X.head(sample_size)
        y = _dataset.values.y.head(sample_size)
    else:
        X = _dataset.values.X
        y = _dataset.values.y

    split_dict = {"test_size": 0.2,
                  "random_state": 7}

    # create a train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, **split_dict)

    param_dict = {"n_estimators": 200,
                  "learning_rate": 1,
                  "random_state": 0}

    scaler_type = 'StandardScaler'

    scaler = StandardScaler()

    scaler.fit(X_train)

    X_train = scaler.transform(X_train)

    X_test = scaler.transform(X_test)

    model = AdaBoostClassifier(**param_dict)

    #Cross Validation

    cv_kfold = KFold(n_splits=5, shuffle=True, random_state=42)

    cv_result = np.array2string(cross_val_score(model, X_train, y_train, cv=cv_kfold), separator=",")

    print(cv_result)

    model.fit(X_train, y_train)

    # prediction
    y_pred = model.predict(X_test)

    db.add_log(process_id=self.process_id, process=self.process, message='prediction with X_test is done')

    # view accuracy
    print('AdaBoost Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))

    accuracy_result = 'AdaBoost Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred))

    accuracy = accuracy_score(y_test, y_pred)
```

EK- 4: CATBOOST MODELİ KAYNAK KODLARI

```
def create_model_catboost(self, dataset, defined_parameters=False, sample_size=None):

    _dataset = DataProcess(dataset)

    all_data_df = _dataset.values.all_data_df

    X = []

    y = []

    if sample_size is not None:
        X = _dataset.values.X.head(sample_size)
        y = _dataset.values.y.head(sample_size)
    else:
        X = _dataset.values.X
        y = _dataset.values.y

    split_dict = {"test_size": 0.2,
                  "random_state": 7}

    # create a train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7)

    param_dict = {"iterations": 1000,
                  "learning_rate": 0.1,
                  "random_seed": 123,
                  "allow_writing_files": False}

    scaler_type = 'StandardScaler'

    scaler = StandardScaler()

    scaler.fit(X_train)

    X_train = scaler.transform(X_train)

    X_test = scaler.transform(X_test)

    #allow_writing_files=False catboost_info klasorunu olusturmamak icin
    model = CatBoostClassifier(**param_dict)

    # Cross Validation

    cv_kfold = KFold(n_splits=5, shuffle=True, random_state=42)

    cv_result = np.array2string(cross_val_score(model, X_train, y_train, cv=cv_kfold), separator=",")

    print(cv_result)

    model.fit(X_train, y_train, verbose=False, plot=False)

    # prediction
    y_pred = model.predict(X_test)

    # view accuracy
    print('CatBoost Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))

    accuracy_result = 'CatBoost Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test, y_pred))

    accuracy = accuracy_score(y_test, y_pred)
```

EK- 5: RASTGELE ORMAN MODELİ KAYNAK KODLARI

```
def create_model_random_forest(self, dataset, defined_parameters=False, sample_size=None):

    _dataset = DataProcess(dataset)

    all_data_df = _dataset.values.all_data_df

    X = []

    y = []

    if sample_size is not None:
        X = _dataset.values.X.head(sample_size)
        y = _dataset.values.y.head(sample_size)
    else:
        X = _dataset.values.X
        y = _dataset.values.y

    split_dict = {"test_size": 0.2,
                  "random_state": 7}

    # create a train/test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7)

    param_dict = {"n_estimators": 1000,
                  "n_jobs": -1}

    scaler_type = "StandardScaler"

    scaler = StandardScaler()

    scaler.fit(X_train)

    X_train = scaler.transform(X_train)

    X_test = scaler.transform(X_test)

    random_forest = RandomForestClassifier(**param_dict)

    # Cross Validation

    cv_kfold = KFold(n_splits=5, shuffle=True, random_state=42)

    cv_result = np.array2string(cross_val_score(random_forest, X_train, y_train, cv=cv_kfold), separator=",")

    print(cv_result)

    random_forest.fit(X_train, y_train)

    random_forest_preds = random_forest.predict(X_test)

    print("Random Forest Model accuracy score:\t", metrics.accuracy_score(random_forest_preds, y_test))

    accuracy_result = 'Random Forest Model accuracy score: {0:0.4f}'.format(metrics.accuracy_score(random_forest_preds, y_test))

    accuracy = accuracy_score(y_test, random_forest_preds)
```