

**KOCAELİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM VE  
ANOMALİ TESPİTİ**

**EMRE AKSÖZ**

**KOCAELİ 2021**

**KOCAELİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM VE  
ANOMALİ TESPİTİ**

**EMRE AKSÖZ**

**Dr.Öğr.Üyesi Birol ARİFOĞLU**  
**Danışman, Kocaeli Üniv.**

.....

**Prof.Dr. Mehmet BAYRAK**  
**Jüri Üyesi, Sakarya Üniv.**

.....

**Prof.Dr. Ali Bekir YILDIZ**  
**Jüri Üyesi, Kocaeli Üniv.**

.....

**Tezin Savunulduğu Tarih: 07.04.2021**

## **ÖNSÖZ VE TEŞEKKÜR**

Tez çalışmamda desteğini esirgemeyen, çalışmalarına yön veren, bana güvenen danışmanım Dr. Öğr. Üyesi Birol ARİFOĞLU'na sonsuz teşekkürlerimi sunarım.

Nisan – 2021

Emre AKSÖZ



## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER DİZİNİ.....	iv
SİMGELER VE KISALTMALAR DİZİNİ .....	v
ÖZET.....	vi
ABSTRACT.....	vii
GİRİŞ .....	1
1. BAKIM TEMEL BİLGİLERİ .....	3
2. BAKIM TÜRLERİ .....	5
2.1. Düzeltici Bakım .....	5
2.1.1. Vadeli bakım .....	5
2.1.2. Acil bakım .....	6
2.2. Önleyici Bakım .....	6
2.2.1. Önbelirlemeli bakım.....	6
2.2.2. Kestirimci bakım .....	6
3. ENDÜSTRİ 4.0 .....	8
3.1. Nesnelerin İnterneti .....	9
3.2. Büyük Veri ve Analizi .....	10
3.3. Bulut Bilişim .....	11
3.4. Yapay Zeka .....	11
4. MAKİNE ÖĞRENMESİ TEMEL BİLGİLER.....	13
4.1. Veri Türleri.....	15
4.2. Veri Seti .....	16
4.3. Öznitelik Çıkarımı.....	17
4.4. Öznitelik Ölçekleme.....	17
4.5. Veri Bölme İşlemi .....	17
4.6. Öğrenme Nedir.....	18
4.7. Model Seçimi .....	20
4.8. Başarı Kriterleri.....	20
4.8.1. Regresyon başarı kriter metriği .....	20
4.8.2. Sınıflandırma başarı kriter metrikleri .....	21
4.9. Öğrenme Türleri.....	23
4.9.1. Denetimli öğrenme .....	23
4.9.2. Denetimsiz öğrenme.....	24
4.9.3. Takviyeli öğrenme.....	24
4.10. Makine Öğrenmesi Uygulama Geliştirme İş Akışı.....	25
5. MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM .....	26
5.1. Anomali Tespiti.....	26
5.2. Yoğunluk Tahmini (Density Estimation).....	27
5.3. Gauss Dağılımı (Normal Dağılım, Gaussian Distribution).....	28
5.4. Yoğunluk Tahmini Algoritması (Density Estimation Algorithm).....	32
6. UYGULAMA .....	33
6.1. Üretim Makinesi Hakkında Bilgilendirme.....	33

6.2. Anomali Durumu .....	36
6.3. Veri Toplama Uygulama.....	36
6.4. Merkezi Limit Teorimi Uygulama .....	38
6.5. Öznitelik Ölçekleme Uygulama .....	40
6.6. Test İçin Öznitelik Ölçekleme Uygulama.....	41
6.7. Makine Öğrenmesi Algoritması Uygulama .....	42
7. SONUÇLAR VE ÖNERİLER .....	44
KAYNAKLAR .....	46
EKLER.....	48
KİŞİSEL YAYIN VE ESERLER .....	59
ÖZGEÇMİŞ .....	60



## ŞEKİLLER DİZİNİ

Şekil 2.1. Bakım türleri Avrupa standardı EN 13306:2010 .....	5
Şekil 4.1. Makine öğrenmesi.....	13
Şekil 4.2. Veri türleri.....	15
Şekil 4.3. Ev fiyatını belirleyen evin özelliklerine ait veri kümesi .....	16
Şekil 4.4. Düşük öğrenme, tam öğrenme ve aşırı öğrenme .....	19
Şekil 4.5. Hata matrisi .....	21
Şekil 4.6. ROC eğrisi .....	22
Şekil 4.7. Makine öğrenmesi uygulama geliştirme iş akışı.....	25
Şekil 5.1. Uçak motoruna ait oluşturulan model.....	28
Şekil 5.2. Normal dağılım .....	28
Şekil 5.3. Normal dağılım güven aralıkları .....	29
Şekil 5.4. Normal dağılımda olasılık hesabı .....	30
Şekil 5.5. Standart normal dağılım.....	30
Şekil 5.6. Standart normal dağılımda olasılık hesabı.....	31
Şekil 6.1. Makine genel görünüm 1 .....	33
Şekil 6.2. Makine genel görünüm 2 .....	34
Şekil 6.3. Makine genel görünüm 3 .....	34
Şekil 6.4. Makine genel görünüm 4 .....	35
Şekil 6.5. Makine elektrik panosu.....	35
Şekil 6.6. Anomali durumu .....	36
Şekil 6.7. Veri toplama fonksiyon blok .....	37
Şekil 6.8. DATA değişkenin içeriği 1 .....	37
Şekil 6.9. DATA değişkenin içeriği 2 .....	38
Şekil 6.10. Merkezi limit teoremi fonksiyon blok .....	38
Şekil 6.11. NewData değişkenin içeriği 1 .....	39
Şekil 6.12. NewData değişkenin içeriği 2 .....	39
Şekil 6.13. Normalizasyon fonksiyon blok .....	40
Şekil 6.14. NormData değişkenin içeriği 1 .....	40
Şekil 6.15. NormData değişkenin içeriği 2 .....	41
Şekil 6.16. Motor parametrelerinin TEST değişkenine atanması .....	41
Şekil 6.17. Test için normalizasyon fonksiyon blok .....	42
Şekil 6.18. Yoğunluk tahmini algoritması fonksiyon blok .....	42
Şekil 6.19. Operatör ekranı uyarı mesajı.....	43

## SİMGELER VE KISALTMALAR DİZİNİ

$\sigma$	: Standart Sapma (Standard Deviation)
$\sigma^2$	: Varyans (Variance)
$\mu$	: Ortalama (Mean)
n	: Örneklem Sayısı (Sample Size)
z	: Standart Normal Değişken (Standard Normal Parameter)

### Kisaltmalar

AC	: Alternatif Akım (Alternative Current)
AUC	: Eğrinin Altında Kalan (Area Under Curve)
CPU	: Merkezi işlem birimi (Central Processing Unit)
FN	: Yanlış Negatif (False Negative)
FP	: Yanlış Pozitif (False Positive)
FPR	: Yanlış Pozitif Oranı (False Positive Rate)
IOT	: Nesnelerin İnterneti (Internet Of Things)
KFÖ	: Kalan Faydalı Ömür
PLC	: Programlanabilir Mantık Denetleyici (Programmable Logic Controller)
ROC	: Alıcı Çalışma Eğrisi (Receiver Operating Curve)
SSR	: Artık Kareler Toplamı (Sum Squares of Residual)
SST	: Toplam Kareler Toplamı (Sum Squares of Total)
TN	: Doğru Negatif (True Negative)
TP	: Doğru Pozitif (True Positive)
TPR	: Gerçek Pozitif Oranı (True Positive Rate)

## MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM VE ANOMALİ TESPİTİ

### ÖZET

Endüstriyel otomasyon ile üretim süreçleri büyük bir değişime uğradı. Günümüz konsepti olan Endüstri 4.0 ile otomasyon çok daha kapsamlı oldu ve yeni teknolojiler ile iyileştirilmesi gereken alanlar üzerinde çalışmalar hızlandı. Endüstri 4.0 konseptinin odaklandığı iyileştirme alanlarından birisi olan verimlilik esaslarını yerine getirmek için birçok teknoloji ve yöntem kullanılmaktadır. Verimlilik için çok sayıda ve farklı alanlarda çalışmalar yapılmaktadır. Bu tezde, kestirimci bakımın getirmiş olduğu faydalar ile verimlilik sağlanmak istenmiştir. Amaç, fabrikalarda üretim yapan makineler üzerindeki algılayıcılardan edinilen verileri analiz ederek, olası arızaları tespit ederek önceden operatöre bildirmek ve daha büyük arızalar oluşmadan müdahale ederek kestirimci bakım faaliyeti gerçekleştirmektir. Minimum duruş süresi ile üretim faaliyetinin sorunsuz devam etmesi, gereksiz bakım ve parça değişikliği gibi sorunların önüne geçmek için yapılan kestirimci bakım türünü, yine günümüzün modern tekniklerinden olan makine öğrenmesi kullanılarak uygulaması yapılmıştır.

**Anahtar kelimeler:** Anomali Tespiti, Endüstri 4.0, Kestirimci Bakım, Makine Öğrenmesi.



# **PREDICTIVE MAINTENANCE AND ANOMALY DETECTION WITH MACHINE LEARNING**

## **ABSTRACT**

Production processes have undergone a great change with industrial automation. With Industry 4.0, which is today's concept, automation has become much more comprehensive and studies on areas that need to be improved with new technologies have accelerated. Many technologies and methods are used to fulfill the principles of efficiency, which is one of the improvement areas that the Industry 4.0 concept focuses on. Numerous and different studies are carried out for efficiency. In this thesis, it is aimed to provide efficiency with the benefits of predictive maintenance. The aim is to analyze the data obtained from the sensors on the machines producing in the factories, to detect possible malfunctions and to inform the operator beforehand and to perform predictive maintenance activities by intervening before bigger malfunctions occur. The predictive maintenance type, which is carried out to prevent problems such as the continuation of the production activity without any problems with minimum downtime, unnecessary maintenance and part replacement, has been implemented using machine learning, which is also one of today's modern techniques.

**Key words:** Anomaly Detection, Industry 4.0, Predictive Maintenance, Machine Learning.

## GİRİŞ

Endüstri 4.0 akımı ile birlikte üretici işletmeler makinelerini ve hatlarını konseptte uygun bir şekilde yeniden dizayn etmeye başlamıştır. Üreticiler ve tedarikçiler bu talebi karşılamak ve akıma ayak uydurmak için yeni ürünleriyle birlikte tamamlayıcı rol oynamaktadırlar. Endüstriyel otomasyon sektörünü daha da ileriye taşıyacak olan bu akım ile birlikte en önemli konu verimlilik ve düşük maliyet olacaktır. Bunu gerçekleştirmek için işletmelerde yenilikler olacak ve yeni yatırımlar gündeme gelecektir. Otomasyon ile sadece makinelerin otomatikleştirilmesi değil kurulum sonrasında servis ve bakım maliyetlerini de en aza indirmek ve verimliliği artırmak çok önemli hale gelmektedir. İşleyen bir fabrikanın sorunsuz çalışması için bakım ve servis hizmetleri çok önemlidir. Eski sistem olan arıza sonrası bakım halen bir çok fabrikada devam etmektedir. Fakat işletmeler gelişime, teknolojiye ve Endüstri 4.0 konseptine ayak uydurmak için en verimli yöntemlerden birisi olan kestirimci bakımı uygulamak zorundadırlar. Kestirimci bakım yeni olmayan bir bakım türüdür. Tecrübeli ustaların deneyimlerini ve duyu organlarını kullanarak gerçekleştirdikleri tanımlar ile yapılmaktadır ve hala birçok işletmede bu devam etmektedir. Günümüzde Endüstri 4.0 konsepti ile birlikte yaşanan tüm gelişmeler ile birlikte fabrikalarda Nesnelerin İnterneti (IoT) ile algılayıcıların artması, makine ve ekipmana ait durum bilgilerini detaylı olarak öğrenmemize ve kayıt altına almamızı kolaylaştırmıştır.

Üretilen bu büyük miktarda veriyi yine Endüstri 4.0 konseptinde yer alan Makine öğrenmesi algoritmaları ile işleyerek kestirimci bakımın etkin ve verimli bir şekilde uygulanması sağlanmıştır. Kestirimci bakımın performans artışı ile beraber işletmeler de tasarruf ve verimlilik sağlamanın yanında kendilerini yeniliklere adapte etmiş olacaklardır.

Fabrikalarda karmaşık endüstriyel ürünlere rağmen sisteme eklenecek basit algılayıcılar ile elde edilen veriler analiz edildikten sonra, makine öğrenmesi algoritması ile geliştirilen yazılım uygulanarak en az maliyet ile kestirimci bakım faaliyeti gerçekleştirmiş olacaktır.

Bölüm 1’de, bakımın ne olduđu ve literatürde geçen kavramlar hakkında bilgiler verilmiştir. Bölüm 2’de, bakımın çeşitlerinden ve birbirleri arasındaki farklarından bahsedilmiştir. Bölüm 3’de Endüstri 4.0 ve içerdiği teknolojiler hakkında bilgiler verilmiş olup tez ile alakalı teknolojileri hakkında bilgiler verilmiştir. Bölüm 4’de Makine Öğrenmesi hakkında temel bilgiler, kullanıldığı yerler, neden ihtiyaç duyulduğu ve süreçleri hakkında bilgiler verilmiştir. Bölüm 5’de kestirimci bakımı makine öğrenmesi ile neden yapılması gerektiği ve bunu yapabilmek için hangi tekniklerin kullanıldığı matematiksel ifadeler kullanılarak formüllerle anlatılmıştır. Bölüm 6’da önceki bölümlerde bahsedilen teorik açıklamalar doğrultusunda kestirimci bakımın gerçek bir uygulama üzerinden nasıl yapılabileceği ilgili bir çalışma yapılmıştır.

## 1. BAKIM TEMEL BİLGİLERİ

Bakım, bir ögenin yaşam döngüsü boyunca gerekli işlevini yerine getirebileceği bir durumda tutması için tüm teknik, idari ve yönetsel eylemlerin birleşimidir [1]. Bakım planı, bakımı gerçekleştirmek için gereken faaliyetleri, prosedürleri, kaynakları ve zaman ölçeğini içeren yapılandırılmış ve belgelenmiş görevler kümesidir. Bakım stratejisi, kullanılabilirlik, maliyet azaltma, ürün kalitesi, çevrenin korunması, güvenlik, varlık değerinin korunması vb. bakım hedeflerine ulaşmak için kullanılan yöntemdir. Bakım yönetimi ise bakım hedeflerini, stratejilerini ve sorumluluklarını belirleyen yönetimin tüm faaliyetleri ve bunların bakım planlaması, bakım kontrolü, bakım faaliyetlerinin ve ekonomisinin iyileştirilmesi gibi yollarla uygulanmasıdır. Bakım, parça, bileşen, cihaz, alt sistem, işlevsel birim, ekipman veya sistem için olabilir. Bir sisteme ait bileşenler şu özelliklere sahip olmalıdır: Bulunabilirlik, güvenilirlik, sürdürülebilirlik, uygunluk, faydalı ömür, modası geçmemiş olması.

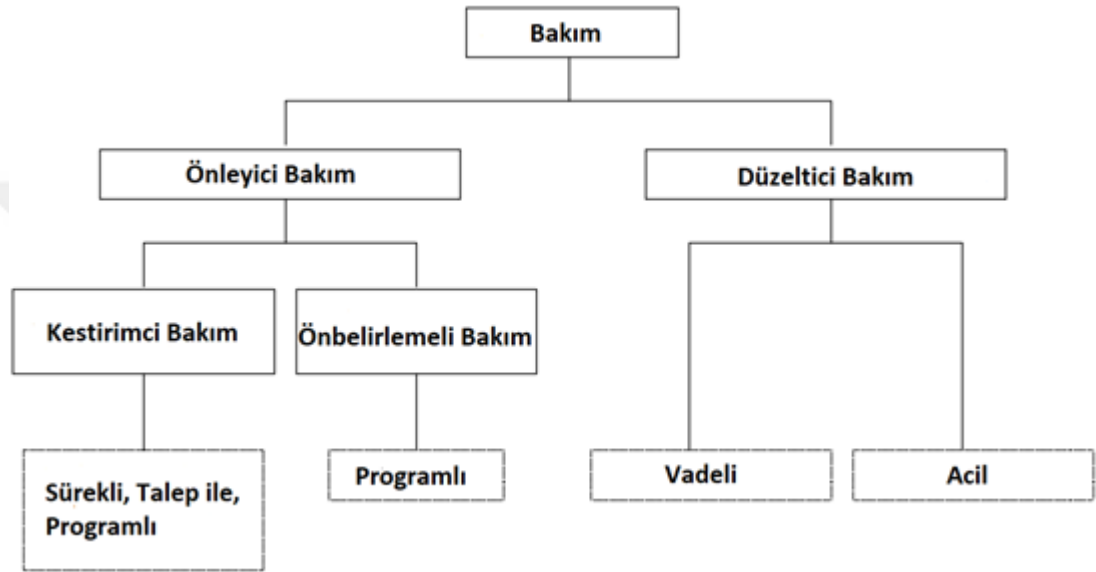
Arıza (Failure), bir ögenin gerekli bir işlevi yerine getirme yeteneğinin sona ermesidir. Bir arıza genellikle bir hatadan kaynaklanır. Hata (Fault), önleyici bakım veya planlanan diğer eylemler sırasındaki yetersizlik veya dış kaynak eksikliği nedeniyle gerekli bir işlevi yerine getirememeye ile karakterize edilen bir ögenin durumudur. Arıza bir olaydır, hata bir durumdur. Yıpranma arızası, ögenin çalışma süresi veya işlem sayısı ve ilgili uygulanan gerilmelerle artarak aşınmaya, malzeme kaybına, deformasyonuna veya değişikliğine neden olan fiziksel bir olaydır. Yaşlanma arızası, takvim zamanının geçmesiyle ortaya çıkma olasılığı artan arızadır. Bu süre ögenin çalışma süresinden bağımsızdır. Yaşlanma, malzemenin fiziksel veya kimyasal özelliklerinin değişmesini içeren fiziksel bir olaydır. Bozunma arızası, fiziksel durumunda zamanla, kullanımla veya dış nedenlerle oluşan zararlı değişikliklerdir [1].

Muayene, bir sistemin ilgili bileşenlerini ölçerek, gözlemleyerek veya test ederek uygunluk incelemesi yapılmasıdır. Durum izleme, önceden belirlenmiş aralıklarla bir

öğenin gerçek durumunun özelliklerini ve parametrelerini ölçmeyi amaçlayan faaliyettir. Durum izleme, öğenin parametrelerindeki herhangi bir değişikliği zamanla değerlendirmek için kullanılmasıyla muayeneden farklıdır. İzleme sürekli olabilir, zaman aralığında veya belirli sayıda işlemten sonra olabilir. Uygunluk testi, bir öğenin bir özelliğinin belirtilen şartlara uygun olup olmadığını göstermek için yapılır. İşlev kontrolü, bakım işlemlerinden sonra öğenin gerektiği gibi çalışabildiğini doğrulamak için alınan eylemdir. Rutin bakım, düzenli veya tekrarlanan basit önleyici bakım faaliyetleridir. Örneğin, temizlik, bağlantıların sıkılması, konektörlerin değiştirilmesi, sıvı seviyesinin kontrol edilmesi, yağlama vb. Genel bakım, bir öğenin gerekli performans düzeyini korumak için gerçekleştirilen kapsamlı önleyici bakım eylemleridir. Genel bakım, belirlenen zaman aralıklarında veya işlem sayısında yapılabilir. Öğenin tamamen veya kısmen sökülmesini gerektirebilir. Onarım, arızalı bir öğenin gerekli işlevini yeniden sağlamak için gerçekleştirilen fiziksel eylemdir. Onarım ayrıca arıza tespitini ve işlev kontrolünü de içerir. Modifikasyon, bir öğenin bir veya daha fazla işlevini değiştirmeyi amaçlayan tüm teknik, idari ve yönetsel eylemlerin birleşimidir. Modifikasyon bir bakım eylemi değildir bir öğenin gerekli işlevinin yeni bir gerekli işlevle değiştirilmesiyle ilgilidir. Bakım görevi hazırlığı, bakım görevinin yerine getirilebilmesi için gerekli tüm bilgilerin sağlanması ve gerekli kaynakların belirlenmesidir. Hazırlık işin nasıl yapılacağına ilişkin açıklamayı, geçerli talimatları, gerekli izinleri, yedek parçayı, beceriyi, araçları vb. içerir. Bakım programı, belirli bir bakım görevinin ne zaman yapılması gerektiğini detaylandıran önceden hazırlanmış plandır [1].

## 2. BAKIM TÜRLERİ

EN 13306: 2010'a göre bakım Şekil 2.1'de gösterildiği gibi temel olarak, düzeltici bakım ve önleyici bakım olmak üzere iki tür bakım stratejisi vardır [1].



Şekil 2.1. Bakım türleri Avrupa standardı EN 13306:2010

### 2.1. Düzeltici Bakım

Arıza meydana geldikten sonra yapılan bakım türüdür. Arıza meydana geldikten hemen sonra bakım yapıldığı için arıza yapan ekipman ömrünü tamamladıktan sonra değişimi gerçekleşmiş olur. Fakat sistem durduğu için üretim kaynaklı oluşan kayıpları ve arıza yapan parçadan dolayı ani duran sistemde birbirleri ile bağlantılı diğer ekipmanların veya sistemin zarar görme ihtimali yüksek olduğundan dolayı, oluşan bu dolaylı kayıpları dikkate almayan bir bakım türüdür [2]. Pahalı parçaları olan, güvenliğin ön planda olduğu kompleks sistemler için uygun değildir.

#### 2.1.1. Vadeli bakım

Bir arıza tespitinden hemen sonra yapılmayan ancak verilen kurallara göre ertelenen bakım türüdür.

### **2.1.2. Acil bakım**

Kabul edilemez sonuçlardan kaçınmak için bir arıza tespit edildikten sonra gecikmeden gerçekleştirilen bakım türüdür.

## **2.2. Önleyici Bakım**

Önceden belirlenmiş aralıklarla veya önceden belirlenmiş kriterlere göre gerçekleştirilen ve bir öğenin arıza olasılığını veya işleyişindeki bozulmayı azaltmayı amaçlayan bakım türüdür. [1].

### **2.2.1. Önbelirlemeli bakım**

Belirlenen zaman aralıklarına veya kullanım sayısına uygun olarak, ancak önceden durum araştırması yapılmadan gerçekleştirilen bakım türüdür [1]. Belli bir zaman diliminde düzenli olarak veya belli bir kritere göre planlı bir şekilde bakım yapılır. Burada amaç, bir sistemin işleyişinin ani olarak veya beklenmedik şekilde bozulmasını engellemek için arıza meydana gelmeden önce planlı olarak sisteme ait ekipmanların düzenli olarak değiştirilmesidir. Bu bakımda dezavantaj olarak, kullanılabilir ömrü olmasına rağmen değiştirilen ekipmandır. Ancak tüm sistemin hasar görmesine karşı oluşan maliyete karşılık ekipman değiştirme maliyeti çok daha düşüktür [2].

### **2.2.2. Kestirimci bakım**

Kestirimci bakım, geçmişte tecrübeli ustaların deneyimlerini kullanarak gerçekleştirdikleri tanımlar ile gerçekleştirilmiştir ve söz konusu durumda da önemli verimlilik artışı sağlamıştır. Günümüzde bu ustaların tanı için kullandıkları görme, duyma, koklama ve dokunma duyularına ilave olarak algılayıcıların ve yapay zekanın katılması ile kestirimci bakımda başarı oranında önemli ölçüde artış sağlanmıştır [3].

Kestirimci bakım, sistemin plansız arızalanmasını önlemek ve hizmet ömrünü en yüksek seviyeye çıkarmak için ekipman durum verilerini değerlendirerek gerçekleştirilen bakım faaliyetidir [4]. Sistemi anlık ve sürekli olarak izleyerek geçmişte elde edilen normal çalışma koşullarına ait verilerle analiz edilerek, arızanın ortaya çıkma durumu tahmin edilir. Böylece ekipmanın arıza vermeden hemen önceki optimum durumu tespit edilir. Ekipmanın ömrü son anına kadar kullanılmış olur ve

gereksiz parça deęişikliği yapılmadan maliyet tasarrufu sağlanır. Makinenin bakımı ani arızalardan hemen önce yapıldığı için sisteme ait duruş ve üretim kayıpları minimum seviyeye indirilmiş olunur [2]. Kestirimci bakımın uygulanması, arıza süresinin azaltılmasına, yedek parça envanterinin optimize edilmesine ve ekipman ömrünün en üst düzeye çıkarılmasına yardımcı olur.

Belirli bir zaman çizelgesini takip eden önleyici bakımın aksine, kestirimci bakım analitik algoritmalar ve ekipman sensörlerinden gelen verilerle belirlenir. Kestirimci bakım sayesinde işletmeler sorunları ekipman arızalanmadan önce belirleyebilir, arızanın temel nedenini belirleyebilir ve ihtiyaç duyulduğunda bakımı planlayabilir [5].

Kestirimci bakımın temelinde iki parametresi vardır: Kalan Faydalı Ömür (KFÖ, Remaining Useful Life) ve anomali tespiti. Kalan faydalı ömür, bir makinenin onarım veya deęişim gerektirmeden önce çalıştırılmasının muhtemel olduğu süredir. Mühendisler, KFÖ'ü hesaba katarak bakımı planlayabilir, işletim verimliliğini optimize edebilir ve planlanmamış arıza sürelerini önleyebilir [6]. Arızaya kadar ne kadar süre olduğu ve bakımın ne zaman planlanması gerektiği tahmin edilebilir. KFÖ'ü tespit etmenin mevcut veri türüne bağlı olarak üç yöntemi vardır:

1. Benzer makinelerin arızaya ulaşmasının ne kadar sürdüğünü gösteren uzun süreli kayıt edilmiş verileri.
2. Teşhis etmek istediğimize benzer makinelerin normal çalışmadan arızaya geçişteki veriye ait geçmişleri.
3. Arızayı tespit eden bir durum göstergesinin bilinen eşik deęeri.

Endüstride üretim hatlarında ki makinelerde yüzlerce hareketli mekanizmalar mevcuttur. Bu mekanizmaların her birisinin KFÖ'nü hesaplamak için gereken yukarıda belirtilen veri türlerinden birine sahip olmak gerekir. Her parçanın normal ve arızalı çalışmasına ait çok detaylı ve büyük bir veri kaydı olması gerekir. Buna karşın, anomali tespiti ile sisteminizdeki deęişiklikleri anlık takip ederek çok büyük bir veri kaydı olmasına gerek olmadan arıza öncesi optimum durum tespit edilerek kestirimci bakım gerçekleştirilir. Veri setinde arızalı durum bilgilerinin olmasına gerek yoktur. Bu tezde anomali tespiti parametresi üzerine çalışma yapılmıştır.



### 3. ENDÜSTRİ 4.0

Endüstri 1.0, 18. yüzyılın sonlarında buhar makinesinden güç ve hareket üretilmesiyle başlayan endüstri devrimidir. Endüstri 2.0, 20. yüzyılın başlarında Henry Ford'un montaj bantları kullanarak ilk seri üretimi gerçekleştirilmesi ile başlamıştır. Endüstri 3.0, 1970'lerden itibaren endüstriyel otomasyon sistemlerinin doğması ve üretim süreçlerinde kullanılmaya başlaması ile olmuştur. Böylece ürünlerin daha hızlı, kaliteli ve düşük maliyet ile seri üretimlerin yapılması gerçekleşmiştir.

Akıllı fabrika, insan, makine ve ortam içinde bulunan nesnelerin yazılım ve donanımlar aracılığıyla, birbirleriyle etkileşim içinde işbirlikçi olarak çalışabildiği ve nesnelerin kendi kendine kararlar alabildiği sistemler bütünüdür. Hızla gelişen dünyada öğrenme hızını geliştirip değişime adapte olma yeteneğini artırmayan işletmeler rekabetçi olamayacaklardır.

Endüstri 4.0 toplamda 3 iyileştirme alanına odaklandığı söylenebilir:

- Hız: Pazara giriş süresinin kısaltılması.
- Esneklik: Aynı üretim hatlarının daha esnek, farklı ve kişiselleştirilmiş ürünlerin üretimi için kullanılması.
- Verimlilik: Daha az kas gücüne dayanarak, birim iş sonuçlarının artırılması. Hatayı yapan insan faktörü işin içinde olmadığı için sistem hatasız çalışmaktadır. Pahalı yatırım olan insan sistemde olmadığı için de sistem ucuzlamaktadır [7].

Endüstri 4.0'ın kullandığı teknolojiler:

- Siber Fiziksel Sistemler (Cyber Physical Systems)
- Yatay ve Dikey Entegrasyon (Vertical & Horizontal Integration)
- Akıllı Robotlar (Autonomous Robots - Co-robots, Collaborative robots)
- Nesnelerin İnterneti (IoT, Internet of Things)
- Büyük Veri ve Analizi (Big Data & Analytics)
- Bulut Bilişim (Cloud Computing)

- Arttırılmış Gerçeklik ve Sanal Gerçeklik (Augmented Reality & Virtual Reality)
- Eklemeli Üretim (Additive Manufacturing)
- Siber Güvenlik (Cyber Security)
- Yapay Zeka (Artificial intelligence)

Yukarıda maddeler halinde verilmiş olan Endüstri 4.0 teknolojilerinden bu tezin çalışma konusu ile ilgili olanlar aşağıda anlatılmıştır.

### **3.1. Nesnelerin İnterneti**

Eskiden sadece aynı platformda olan makineler kendi aralarında iletişim kurabilirken bugün endüstriyel otomasyon enstrümanları, sensörler, kablosuz iletişim teknolojileri ve internet sayesinde birbirleri ile her yerden iletişime geçebilmektedirler. Bunun gerçekleşebilmesi için tüm nesnelerin, Nesnelerin İnterneti (IoT, Internet Of Things) mimarisine uyumlu olarak tasarlanması ve standart iletişim protokollerine sahip olması gereklidir. IoT'nin temelinde bilgi yönetimi vardır. Birçok yerden farklı formatlarda gelen verilerin oluşturduğu büyük veri kümelerinden anlamlı bilgiler çıkarılıp, gerçek zamanlı kararlar alınması ile verimlilik, hız ve rekabet elde edilir [8].

İnternetin yaygınlaşması, sensör maliyetinde düşüş, sensör boyutunda küçülme gibi teknolojik gelişmeler ile sensör kullanımı epey yaygınlaşmıştır. Sensörler sayesinde nesnelere duyu yeteneği kazandırarak ölçülen veriler işlenmek için depolanır. IoT ile nesnelere kimlik kazandırarak ve varlık haline dönüştürülerek bir nesneye her yerden bilgi gönderip alınabilmektedir. Üretimin yeni standartları bu işleyiş üzerine kurulmaktadır. Dijital çağda birçok uygulama IoT ağları üzerinden gerçekleşecektir. IoT ile gerçek zamanlı veriler kullanılarak analizi yapılacak, doğru ve hızlı kararlar alınacaktır.

Nesnelerin interneti kavramına örnek olarak giyilebilir teknolojilerin gittikçe popüler hale gelmesi verilebilir. Bu sürecin hayatımızda yaygınlaşan akıllı saatler ile hız kazanacağı beklenmektedir. Akıllı saatler insanlara aktivite alışkanlıklarını takip etmesi, sağlıklı beslenmesi ve daha fazla egzersiz yapması için yardımcı olmaktadır. Bu cihazlar üzerinden toplanan veriler sadece bireylerin daha sağlıklı olmasına yardımcı olmanın yanı sıra, işverenler, sağlık sektörü çalışanları ve sigorta şirketleri için faaliyetlerin takip edilmesi açısından bir veri kaynağı oluşturmaktadır. Bu kişisel

verilerin paylaşılması ile karşılığında firmalardan ücret veya ödüller alınabilir, bu da insanların kendi özel verilerinin ticaretini yapma istekliliğinde bir artış olarak karşımıza çıkabilir.

### **3.2. Büyük Veri ve Analizi**

Veriye ulaşan, analiz eden ve bundan faydalı, anlamlı sonuçlar çıkarabilen işletmeler, her açıdan rekabet üstünlüğü elde etmektedirler. IoT ile veri miktarında ve hızında artış olacaktır. Veri analizi ile üretilen faydalı bilgiden işletmeler doğru kararlar alabilmektedirler.

Her şey hakkında artık daha fazla veriye sahip olmak isteği ve herhangi bir veriyi depolama ve analiz etme becerilerinin gelişmiş olması ile birlikte Büyük Veri hareketi hız kazanmıştır [9]. Gittikçe dijital hale gelen yaşamımızda yaptığımız her şey geride bir veri izi bırakır. Geçtiğimiz son yıllar içinde, insanoğlunun tüm tarihi boyunca ürettiğinden daha fazla veri üretildi. Sadece insana ait olan veriler sosyal medya, çekilen fotoğraflar ve videolar, cep telefonlarının sensörleri, akıllı saatler vb. yaşamımızda yer alan şeylerden üretilmektedir.

Üretilen verileri anlamlı ve işe yarar hale getirmediği hiçbir değeri yoktur, bunu yapmak için veriyi elde etmek ve analiz etmek gerekir. Eskiden veritabanlarında depolanabilecek veri miktarı sınırlıydı ve sistemler çok yavaştı. Gelişen bilgisayar donanımları ve analiz yöntemleri ile bu sorun artık aşılmış durumdadır.

Büyük veri ve analizi çok uzun zamandır var olan bir şeydi. Değişen teknoloji ve sürekli artan bağlanabilirlik sayesinde verinin boyutu, hızı ve analizlerin gelişmesi ile büyük veri değişkenlik göstermiştir.

Facebook, Google, Netflix, Microsoft, Apple, LinkedIn, Instagram, IBM, Twitter ve Amazon gibi devasa firmalar büyük veriyi etkili şekilde kullanmaktadır. Büyük verinin sadece Silikon Vadisi şirketlerinin yaptığı bir şey olduğu şeklinde yaygın bir yanlış fikir vardır [9]. Herhangi bir sanayi dalının veya en geleneksel şirketlerin bile faydalanabileceği bir şeydir. Piyasadaki ani yükseliş ve düşüşleri tahmin etmek için veriyi kullanan emlak şirketleri, müşterilerin akıllı telefonlarını kullanarak araçlarını

gerçekten ne kadar iyi sürdüklerini takip eden motorlu taşıt sigorta firmaları gibi diğer şirketler de bu şekilde çok sık büyük veriyi kullanmaya başladılar.

Büyük veri kullanımı gelecekte çok daha önemli ve vazgeçilmez olacaktır. Büyük veriyi görmezden gelen, kullanma planları bulunmayan işletmeler yarışın gerisinde kalacaklardır.

Veri saklama kapasitesindeki artış, veri saklamanın ucuzlaması, bilgisayar donanımlarının gelişmesi ile artan hesaplama gücü ve karmaşık veri kümelerinin analizinde kullanılan makine öğrenmesi yöntemlerinin gelişmesi ile birlikte veri miktarında üstel bir artış gerçekleşmektedir. Makine öğrenmesinin gelişmesini tetikleyen en önemli etken büyük veridir. Veri, makine öğrenmesinin temel taşıdır [10].

### **3.3. Bulut Bilişim**

Büyük veri, bulut bilişim veya veri merkezlerinde saklanır. Bulut, veriyi internette saklarken, veri merkezi, veriyi bir donanımda saklar.

Bulut (Cloud), verilerin internet ortamında saklanabilmesine, dış ortamlarda paylaşabilmesine olanak sağladığı gibi, kaynakların ortak kullanılması nedeniyle enerji ve yatırım maliyetlerinin indirgenmesine de olanak sağlamaktadır [8].

### **3.4. Yapay Zeka**

Yapay Zeka, bilginin saklanması ve işlenmesinde veri yapıları, algoritmalar, programlama dilleri ve teknikleri gibi bilgi işlem yöntemlerini kullanan, bilgisayar biliminin bir alt dalıdır. İnsan beyni, sayısal birkaç işlem yapabilmesinin yanında anlama, yorumlama, karar verme gibi olayları çok kısa sürede gerçekleştirebilir. Bilgisayarlar ise karmaşık sayısal işlemleri çok hızlı yapabilmesine rağmen olayı yorumlama, anlama, öğrenme, karar verme ve deneyim yoluyla elde edilmiş bilgileri kullanma gibi idrak yeteneklerinden yoksundurlar. Burada amaç insanın kavrayışına yönelik olan yeteneklerini, davranışlarını taklit ederek bilgisayar sistemlerinde bu yetenekleri modellemek ve insan zekasına gereksinim duyan işleri yapabilecek makineler üretmektir. Yapay zeka tekniklerinin her birinde amaç insan beynini modelleyerek problemlere çözüm üretebilmektir. Tüm bu teknikler, matematiksel formüller aracılığı ile insan beynini modelleyebilmektedir [11].

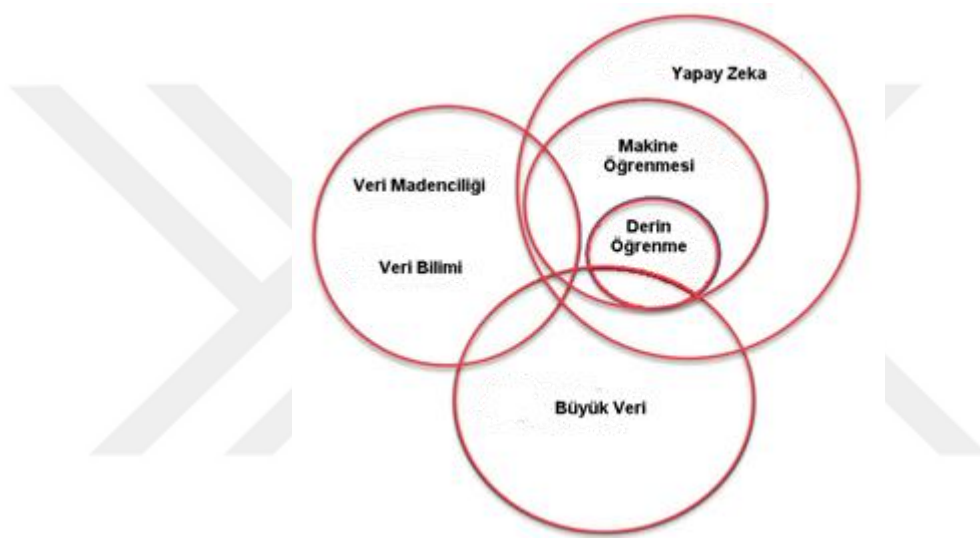
Yapay zekanın gelişiminde çok yüksek çalışma hızına sahip donanım sistemleri, sayısal ve görsel veriler ve öğrenmeyi gerçekleştirecek zeki algoritmalar sıralanabilir [12].

Kendilerine öğretilen tekrarlamalı işleri yapmaktan sorumlu olan robotlar, yapay zeka teknolojileriyle artık kendi kendine öğrenebilen, karar verebilen ve insanlarla bir arada etkileşimli olarak çalışabilir duruma gelmiştir.



#### 4. MAKİNE ÖĞRENMESİ TEMEL BİLGİLER

Şekil 4.1’de gösterildiği gibi Makine Öğrenmesi, Yapay Zeka’nın bir alt dalıdır [13]. Otonom sistemler, doğal dil işleme uygulamaları, borsa, finans, istenmeyen elektronik postaların tespiti, görüntü işleme, ses ve konuşma işleme, dolandırıcılık tespiti gibi birçok alanda yaygın olarak kullanılır.



Şekil 4.1. Makine öğrenmesi

Makine öğrenmesi, insanlar tarafından tasarlanıp yazılamayacak kadar karmaşık görevleri yerine getirebilmemizi sağlar. Dikkati dağılmayan, yorulmayan, psikolojisi bozulmayan, sendika kurmayan, doğum izni almayan, ücret artışı istemeyen makineler, insanların karşısında rakip olmaya başladılar. Makine öğrenmesi sadece veriden öğrenirler bu yüzden, verilerin sınırladığı çerçevenin dışına çıkarak yenilikler yapmaları, mantık yürütmeleri mümkün değildir. Makinede kullanılan algoritma, bir grubu diğerinden daha fazla temsil eden veri ile eğitilmişse bu bir yanlılığa yol açar [10].

Bir problemi çözmek için algoritmaya ihtiyaç vardır. Algoritma, girdiyi çıktıya dönüştüren talimatlar dizisidir. Klasik programlamada, program yazarak makineye talimatlar dizisi verirken makine öğrenmesinde, makineye veriler vererek bunlardan

öğrenmesini ve istenilen görevi yapması istenir [10]. Yani makine öğrenmesi, belirli bir amaç için programlama yapmak yerine örnekler içeren veri setlerinden öğrenir.

Makine öğrenmesi, bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek gelecekte oluşacak benzeri olaylar hakkında kararlar verebilmesi ve problemlere çözümler üretebilmesidir [14].

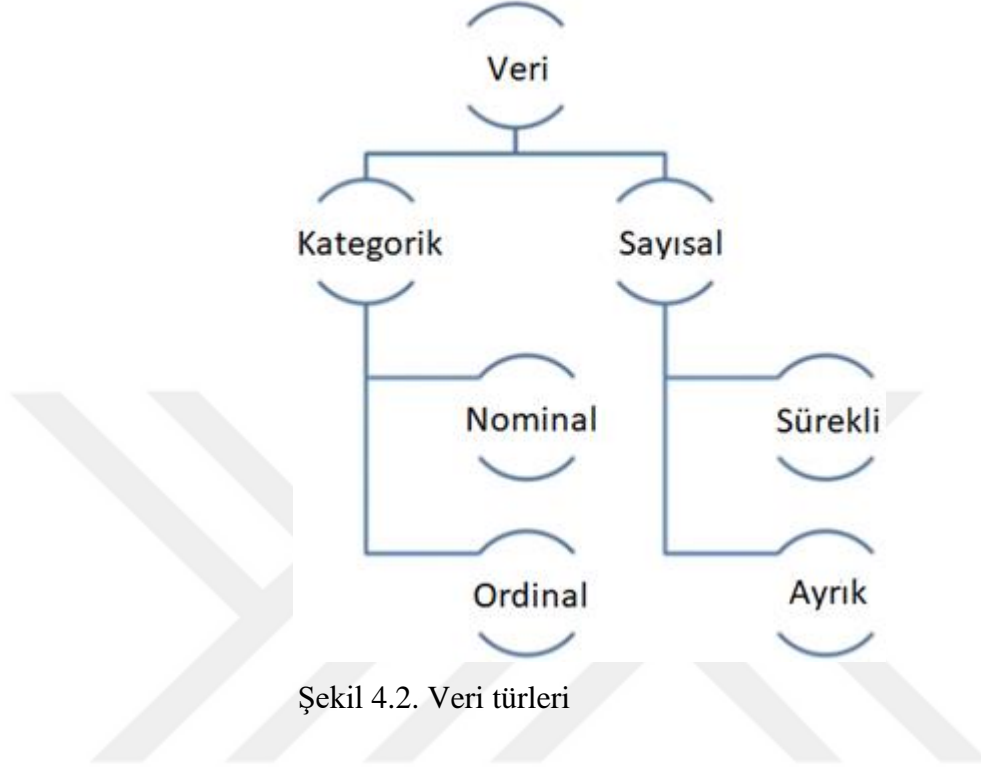
İnsanoğlunun öğrenme sistemi incelendiğinde tecrübe faktörü çok önemlidir. Ne kadar çok tecrübeye sahip olunursa o kadar az hata yapılır ve doğru kararlar verilir. Makine öğrenmesi, tecrübe kazanmış bir sistemin karşılaşacağı yeni bir durumu başarılı tahmin etmesi sürecidir. Buradaki tecrübe çok sayıda veri ile sağlanırken öğrenme işlemi ise her biri farklı bir çalışma prensibine dayanan makine öğrenmesi algoritmaları ile gerçekleşmektedir [12]. Bir makine öğrenmesi sisteminin başarısında veri sayısı ve algoritma tercihi önemli yer tutar.

Günümüzde e-postalar, videolar, ses dosyaları, resimler, tıklama akışları, mesajlar, arama sorguları, sosyal ağ etkileşimleri, sensörler ve cep telefonları ile çok büyük ham veri kümeleri oluşmaktadır. Makine öğrenmesi, veri bilimi ve veri madenciliği gibi yapay zekanın alanları ile verileri yakalamak, biçimlendirmek, depolamak, yönetmek, analiz etmek ve görselleştirmek için ham verinin bilgiye dönüşümü sürecine katkıda bulunurlar. Edinilen ham veri, veri ön işleme (data preprocessing) sürecinden geçtikten sonra bilgiye dönüşmüş olur. Veri ön işleme ile eksik, hatalı tutarsız veriler ayıklanır [12].

Geleneksel yöntemlerle geliştirilen yazılımlar, tanımlanan algoritmaya göre çalışırlar. Makineye gösterilen rakamların kaç olduğunu tahmin eden bir sistem yapılmak istenildiğinde standart programlama tekniği ile yapmak imkansızdır. Her rakamın görseline ait piksellerin tutulduğu bir matris düşünün ve bu her rakamlara ait piksel matrisinden n adet olsun. Çok fazla işlem yükü gerektiren bu çok boyutlu matris topluluğu makine öğrenmesi algoritmasına sokularak eğitilir ve eğitilen verinin dışında bir rakam makineye gösterildiğinde doğru tahminler yapması beklenir.

#### 4.1. Veri Türleri

Şekil 4.2’de görüldüğü üzere veri türleri sayısal ve kategorik olarak iki kısma ayrılır.



Şekil 4.2. Veri türleri

Sayısal (Nitel) veriler, yaş, kilo, boy gibi rakamlarla ifade edilen verilerdir. Sayısal veriler Sürekli (Continuous) ve Ayrık (Discrete) olarak ikiye ayrılır. Ayrık veriler, birbirinden sayılarak ayrılan verilerdir. Örneğin, 15 yaş, 10 kilo vb. Sürekli veriler, ölçüm ile kesintisiz elde edilen verilerdir. Örneğin, bir fırının sıcaklığını ölçerken elde edilen değer.

Kategorik (Nitel) veriler, cinsiyet, eğitim düzeyi, medeni hal, plaka kodları vb. günlük hayatımızda karşılaştığımız grup veya kategorilere ayrılan, dört işlem yapılamayan verilerdir. Kategorik veriler Nominal ve Ordinal olarak ikiye ayrılır. Ordinal veriler, aralarında büyüklük küçüklük ilişkisine girebilen fakat ölçü belirtmeyen verilerdir. Örneğin, bir kişiye kredi çıkma ihtimalinin düşük, orta veya yüksek olarak tanımlanması. Nominal veriler, sıralanamayan ve ölçülemeyen verilerdir. Örneğin, bir kanser hücresinin iyi huylu veya kötü huylu olarak tanımlanması.

Makine öğrenmesi uygulamalarında kategorik verilerin kullanılabilmesi için karşılık gelen sayısal verilere dönüştürülmelidir. Bu işlem terminolojide Label Encoding ve



One-Hot Encoding olarak geçmektedir. Bu tezin konusunda kategorik veri kullanılmayacağından detayına girilmemiştir.

## 4.2. Veri Seti

Bir Veri Seti (Data Set), satırlar ve sütunlardan oluşan iki boyutlu dizi olarak ifade edilir. Bir veri setindeki her satır bir örnek (sample), her sütun bir öznelik (özellik, feature) olarak adlandırılır. Veri setindeki son sütun sınıf (etiket, class, label) olarak adlandırılır. Öznelikler girdi (bağımsız değişken), sınıf çıktı (bağımlı değişken) olarak ifade edilir. Veri setleri makine öğrenmesinde kullanılacağı zaman Veri Kümesi (Data Frame) tipine çevrilir. Veri Kümesi'nin Dizi'den (Array) farkı, indeks ve sütun başlıklarının olmasıdır. Örneğin, bir ev fiyatını belirleyen özellikleri gösteren bir veri kümesi Şekil 4.3' de gösterilmiştir.

Index	bedrooms	bathrooms	sqft_living	floors	price
0	3	1	1180	1	221900
1	3	2.25	2570	2	538000
2	2	1	770	1	180000
3	4	3	1960	1	604000
4	3	2	1680	1	510000
5	4	4.5	5420	1	1.23e+06
6	3	2.25	1715	2	257500
7	3	1.5	1060	1	291850
8	3	1	1780	1	229500
9	3	2.5	1890	2	323000

Şekil 4.3. Ev fiyatını belirleyen evin özelliklerine ait veri kümesi

Bir veri setinde eksik veriler (missing values) olabilir. Makine öğrenmesi algoritmalarında veri sayısının yüksek olması performansı artırdığı gerçeği göz ardı edilerek kayıp verileri silmek doğru bir yaklaşım olmayacaktır. Kayıp verinin yerine aritmetik ortalama, mod veya medyan gibi istatistiksel ölçütler kullanılır. Kayıp verinin

bulunduđu sütunun ortalaması alınarak eksik veri hücrelerine yazılır [15]. Bu işleme terminolojide yaklaşık değeri verme (imputation) denir.

### **4.3. Öznitelik Çıkarımı**

Elde edilen ilk durum olan veri seti ham veridir (raw data). Bu ham veri setine ait özniteliklerin hepsi kullanılabilir. Ancak bazı öznitelikler makine öğrenmesi algoritmasına dahil edilmesi anlamlı olmayabilir yada algoritmanın verimini düşürebilir. Bu durumda bazı öznitelikler veri setinden silinerek çıkartılır. Bu işleme Öznitelik Çıkarımı (Feature Extraction) denir.

### **4.4. Öznitelik Ölçekleme**

Veri setinde özniteliklerin birimleri farklı olabilir ve aralarında bir ilişki bulunmayabilir. Örneğin, Şekil 4.3’de sqft\_living ile bedrooms öznitelikleri arasında birim ve miktar olarak farklılık bulunmaktadır. Birinci satırda 2570 değerine karşılık 2.25 değeri vardır. 2570 rakamı 2.25 değerini baskılar ve küçük olan değeri ihmal ettirerek etkisizleştirir. Tüm özniteliklerin aynı değeri aralığında ve koşullarda ifade edilmesine Öznitelik Ölçeklendirme (Feature Scaling) denir. Öznitelik ölçeklendirmek için Standardizasyon (Standardisation) ve Normalizasyon (Normalization) teknikleri uygulanabilir. Her makine öğrenmesi metodu öznitelik ölçeklendirme gerektirmeyebilir.

### **4.5. Veri Bölme İşlemi**

Makine öğrenmesinde öğretim işlemi yapılmadan önce veri seti, eğitim seti (train test) ve test seti (test set) olarak iki parçaya ayrılır. Eğitim setindeki veriler ile makine öğrenmesi algoritması eğitilir. Test setindeki yani sistemin daha önce görmediği veriler ile model test edilir. Makine öğrenmesinde oluşturulan bir modeli yeni veriler ile test etmek makine öğrenmesi modelini geçerliliği ve değerlendirilmesi için çok önemlidir. Bu değerlendirmeye Genelleme (Generalization) denir.

Genelleme, makine öğrenmesi ile eğitilen bir modelin daha önce görülmemiş yeni verilerle test edilerek modelin performansının değerlendirilmesi süreci olarak ifade edilir. Makine öğrenmesi çalışmalarındaki öncelikli ve en büyük amaç daha önce bilinmeyen veri öğelerini doğru bir şekilde tahmin eden bir öğrenme modeli

oluşturmaktır. Bu nedenle, oluşturulan öğrenme modeli, gelecekteki veri öğelerinin doğru bir şekilde tahmin edebilmesi için çok iyi genellenebilmelidir. Genelleme, Eğitim ve Test Seti (Train and Test Set) ve Çapraz Doğrulama (Cross Validation) olarak iki türlü yapılır.

Eğitim ve test seti, genelde %80 eğitim seti, %20 test seti olarak ayrılır [16]. Çapraz doğrulama, veri setinin eğitim ve test setinin ayrılması esnasında oluşabilecek düzensizliklere yönelik çözüm sunan bir yöntemdir.

#### 4.6. Öğrenme Nedir

Öğrenme, çevreye uyum sürecinde davranışta meydana gelen sürekli değişimlerdir. Bir problem çözümü için öğrenme durumu modellenmektedir. Problem için daha önceki tecrübeler, deneyimler veya olaylar sisteme aktararak sistem eğitilir. Bu sayede sistem bu deneyimler dışında yeni bir durum ile karşılaştığında öğrenme nedeniyle mantıklı bir davranış sergileyecektir.

Makine öğrenmesi uygulamalarında temel amaç elde edilen veriler üzerinden örüntüler (patterns) elde etmek ve yeni veriler için bu örüntüler üzerinden doğru tahminler yapabilmektir. Örneğin, amacımız en basit haliyle deneyime göre kişinin maaşını tahmin etmek olsun. Bu tahmin için oluşturulan matematiksel model Denklem (4.1)'de gösterilmiştir. Maaş değişkeni bağımlı (dependent) değişken, deneyim değişkeni bağımsız (independent) değişkendir.

$$\text{maas} = b_0 + b_1 * \text{deneyim} \quad (4.1)$$

$B_0$ , sabit (constant, bias) y eksenini kestiği noktadır.  $B_1$ , katsayı (coefficient) doğrunun eğimidir (slope). Burada makine öğrenmesi demek eğitim veri setine göre bu denklemde verilen  $b_0$  ve  $b_1$  katsayılarını hesaplamak demektir. Yani bu katsayıları bulduktan sonra oluşan değerler öğrenilmiş olan değerlerdir. Öğrenilen bu katsayılar denklemde yerine konulduktan sonra deneyim parametresine test için yeni değer verilir ve maaş tahmin edilir.

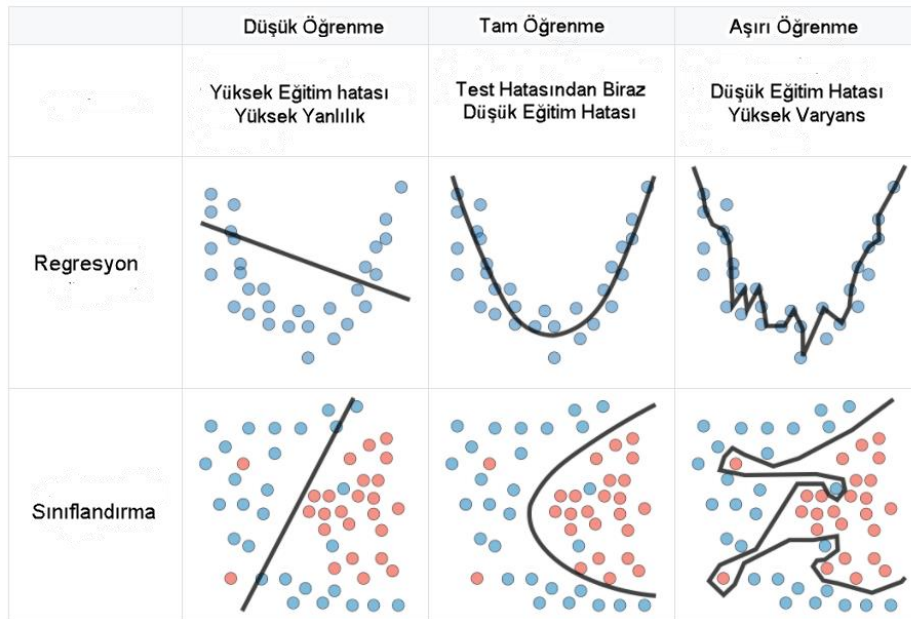
Düşük öğrenme (Underfitting), model eğitim verilerine uymayarak yetersiz öğrenmesidir. Modelin problemin çözümünü içermediğini ve zayıf kaldığını gösterir. Yeni verilerde çok düşük tahminler yapar. Düşük öğrenmenin önüne geçmek için

yapılacak olan bazı yöntemler: Daha fazla öznitelik ekleyerek model karmaşıklığını artırmak. Düşük öğrenme, düşük varyans ve yüksek yanlılığa sahiptir.

Aşırı öğrenme (Overfitting), modeli eğitirken ezber yapmasına denir. Modelin tahmin ettiği verilerin gerçek veriler etrafında aşırı dağılmasıdır. Eğitim verilerinde yüksek tahmin doğruluğu yapar ancak yeni verilerde çok düşük tahminler yapar. Aşırı öğrenmenin önüne geçmek için yapılacak olan bazı yöntemler: Düzenleştirme (Regularization), K Katlamalı Çapraz Doğrulama (K-Fold Cross Validation), daha fazla veri eklemek ve erken durdurma. En çok kullanılan düzenleştirme yöntemleri: Lasso Regresyon, Ridge Regresyon ve Elastic Net. Lasso, değişken seçme yöntemi olarak da kullanılır. Aşırı öğrenme, düşük yanlılık ve yüksek varyansa sahiptir.

Şekil 4.4’de düşük öğrenme, normal öğrenme ve aşırı öğrenmeye ait görsel mevcuttur [17]. Yanlılık (Bias), modelin ne kadar yanlış olduğunu ölçer. Gerçek değerlerden tahmin edilen değerlerin ne kadar uzak olduğudur. Tahmin edilen değerler gerçek değerlerden uzaksa, yanlılık yüksektir. Varyans (Variance), modelin tahmin ettiği verilerin, gerçek verilerin etrafında nasıl saçıldığını ölçer. Tahminlerdeki değişkenlik miktarıdır. Varyans yüksek ise model fazla geneldir.

İyi bir modele sahip olmak için düşük yanlılık ve düşük varyansa sahip olmak gerekir. Çapraz doğrulama sayesinde yanlılık-varyans ikilemi çözülmüş olur.



Şekil 4.4. Düşük öğrenme, tam öğrenme ve aşırı öğrenme

## 4.7. Model Seçimi

En iyi parametreleri seçmek (choosing best parameter) anlamına gelmektedir. Kullanıcı tarafından belirlenecek olan bu parametreler hiperparametreler adını alır. Hiperparametre (Hyperparameters), makine öğrenmesi modelini tasarlayan kişiye bırakılmış, probleme, veri setine göre değişiklik gösteren, en yüksek doğruluk oranına erişilmeye çalışmak için ayarlanan ince ayar (tune) parametreleridir. Model seçimi yaparken Çapraz Doğrulama (Cross Validation) kullanılarak yapılır. Çapraz doğrulama için K-Katlı Çapraz Doğrulama (K-Fold Cross Validation) yöntemi kullanılır. Veri seti K adet eşit parçaya bölünür. Her bir parçayı rastgele bir şekilde eğitim ve test verisi olarak ikiye ayırır. Her parçada modelin doğrulukları tahmin edilir ve K adet parçanın ortalaması alınarak en büyük doğruluğu veren modele ait hiperparametre değerleri seçilir.

Model seçimi yaparken modelin sadece parametreleri değil makine öğrenmesi algoritmalarının nasıl değerlendireceği konusunda da bir seçim yapılmalıdır. Başarı değerlendirme seçenekleri Bölüm 4.8’de Başarı Kriterleri başlığı altında anlatılmıştır.

## 4.8. Başarı Kriterleri

Bir makine öğrenmesi modelinin tahmin sonucunun ne kadar başarılı ve doğru olduğunun değerlendirilmesi işlemidir. Sınıflandırma ve regresyon problemleri için ayrı başarı değerlendirme metrikleri kullanılmaktadır.

### 4.8.1. Regresyon başarı kriter metriği

R-Kare (R-Square) değeri bire ne kadar yakın ise tahmin başarısı o kadar iyi demektir. Başarı kriterinin formülü Denklem (4.2)’de gösterilmiştir. SSR (Sum Square of Residual), artık kareler toplamıdır. SST (Sum Square of Total), toplam kareler toplamıdır.

$$(RKare)=1-\left(\frac{SSR}{SST}\right) \quad (4.2)$$

#### 4.8.2. Sınıflandırma başarı kriter metrikleri

Sınıflandırma algoritmalarını kullanarak yapılan çalışmalarda en büyük yanılgılardan biri başarı kriteri olarak sadece doğruluk oranına bakmaktır. Özellikle dengesiz veri setlerinde (imbalanced data sets) doğruluk oranı bize pek bilgi vermez.

Hata matrisi (Confusion matrix), veride ki var olan durum ile sınıflandırma modelinin doğru ve yanlış tahminlerinin sayısını gösterir. Matrisin diyagonal hizadaki değerleri ne kadar yüksek olursa tahmin başarısı o kadar yüksek demektir. Bir veri setinde ki sınıf sayısı N adet ise hata matrisi NxN'lik bir matris olur. Şekil 4.5'de 2x2'lik bir hata matris örneği gösterilmiştir.

- TP: Hasta olana hasta demek. (Doğruya doğru demek).
- FP: Hasta olana hasta değil demek. (Doğruya yanlış demek).
- TN: Hasta olmayana hasta değil demek. (Yanılsa yanlış demek).
- FN: Hasta olmayana hasta demek. (Yanılsa doğru demek).

		Var olan Durum	
		Pozitif Durumlar	Negatif Durumlar
Tahmin	Pozitif	TP	FN
	Negatif	FP	TN

Şekil 4.5. Hata matrisi

Doğruluk (Accuracy), bir sınıflandırma modelinin doğru olduğu tahminlerinin oranıdır. Formülü Denklem (4.3)'de gösterilmiştir.

$$\text{Doğruluk} = (TP+TN) / (TP+TN+FN+FP) \quad (4.3)$$

Duyarlılık (Sensitivity, Recall), gerçekte pozitif bir sonuç elde edildiğinde, test sonucunun da pozitif olması olasılığıdır. Formülü Denklem (4.4)'de gösterilmiştir.

$$\text{Duyarlılık} = TP / (TP + FN) \quad (4.4)$$

Kesinlik (Precision), pozitif bir test sonucu elde edildiğinde, gerçek sonucun da pozitif olması olasılığıdır. Formülü Denklem (4.5)'de gösterilmiştir.

$$\text{Kesinlik} = \text{TP} / (\text{TP} + \text{FP}) \quad (4.5)$$

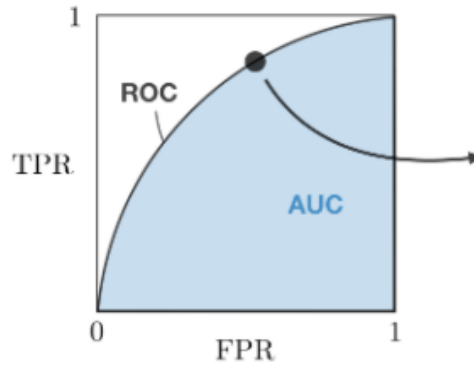
Belirginlik (Özgüllük, Specificity), gerçekte negatif bir sonuç elde edildiğinde test sonucunun da negatif olması olasılığıdır. Modelimiz tarafından olumsuz olarak tahmin edilen gerçek olumsuz vakaların oranı hakkında bizi bilgilendirir. Formülü Denklem (4.6)'da gösterilmiştir.

$$\text{Belirginlik} = \text{TN} / (\text{TN} + \text{FP}) \quad (4.6)$$

F1-Skor (F1-Score), duyarlılık ve kesinliğin harmonik ortalamasıdır. Formülü Denklem (4.7)'de gösterilmiştir.

$$\text{F1 Ölçüm} = 2 * \text{duyarlılık} * \text{kesinlik} / (\text{duyarlılık} + \text{kesinlik}) \quad (4.7)$$

Alıcı Çalışma Eğrisi (Receiver Operating Characteristic, ROC), bir olasılık eğrisidir. Modelleri sınıflar arasında ne kadar iyi ayırt edebildiğini gösterir. Eğrinin Altında Kalan (Area Under Curve, AUC) alan ne kadar büyükse modelin başarı oranı o kadar yüksektir. Şekil 4.6'da ROC eğrisinde x ekseninde Yanlış Pozitif Oranı (False Positive Rate, FPR) ve y ekseninde Gerçek Pozitif Oranı (True Positive Rate, TPR) vardır. Denklem (4.8) ve Denklem (4.9)'da TPR ve FPR'nin formülü gösterilmiştir.



Şekil 4.6. ROC eğrisi

$$\text{TPR} = \text{TP} / (\text{TP} + \text{FN}) \quad (4.8)$$

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) \quad (4.9)$$

Matthews Korelasyon Katsayısı, değişkenler arasındaki ilişkiyi göstermek için kullanılan bir değerdir. Yani değişkelerde meydana gelen değişikliklerin birbirini etkileyip etkilemediklerini gösterir. Hesaplandığında -1 ile 1 arasında bir değer alır.

Negatif deęerler negatif iliřkiyi, pozitif deęerler ise pozitif iliřki olduęunu gsterir. Deęerler 1 veya -1 olduęunda mükemmel bir iliřki vardır. Deęerler 0'a yaklařtıklarında deęiřkenlerin aralarındaki iliřki de azalır. Korelasyon baęıntısını kullanarak iki deęiřken arasında doęrusal bir iliřki olup olmadıęı bulunur. Örneęin, sıcaklık ile tüketim, soru sayısındaki deęiřikliklerle öęrenci başarısı arasındaki iliřki, üretim ile ücret artışı arasındaki iliřki gibi.

#### **4.9. Öęrenme Türleri**

Makine öęrenmesi problemlerinin çözümleri için temel olarak üç çeřit öęrenme sistemi vardır:

1. Denetimli Öęrenme (Danıřmanlı Öęrenme, Supervised Learning)
2. Denetimsiz Öęrenme (Danıřmansız Öęrenme, Unsupervised Learning)
3. Takviyeli Öęrenme (Destekleyici Öęrenme, Reinforcement Learning)

##### **4.9.1. Denetimli öęrenme**

Veri kümesinde, girdi deęerlerine karřılık hangi çıktıların olduęu bilinen makine öęrenmesi sistemidir. Örneęin Şekil 4.3'de veri kümesinde olan evin banyo sayısı, kat sayısı, yatak odası sayısı ve salonun metre karesi özellikleri girdilerdir. Bu girdilere karřılık gelen evin fiyatı ise çıktıdır. Denetimli öęrenme problemleri çözümlenirken Sınıflandırma (Classification) ve Regresyon (Regression) olarak ikiye ayrılır.

Sınıflandırma, girdi deęiřkenlerine karřılık çıktının kategorik olarak tahmin edilmesidir. Örneęin, yař, boy ve kilo özelliklerine karřılık kiřinin, erkek veya kız olduęunu tahmin etmek. Gelen mailleri spam veya spam deęil olarak ayırmak. Tümörlü bir hastanın, tümörünün iyi huylu veya kötü huylu olduęunu ayırmak. Resmi gösterilen bir hayvanın kedi veya köpek olduęunu tahmin etmek vb. En sık kullanılan sınıflandırma metodları: Lojistik Regresyon (Logistic Regression), K-En Yakın Komřu Algoritması (K-Nearest Neighbour Algorithm), Destek Vektör Makineleri (Support Vector Machines), Karar Aęaçları (Decision Trees), Rastgele Ormanlar (Random Forest), Naif Bayes (Naiv Bayes).

Regresyon, girdi deęiřkenlerine karřılık çıktının sayısal (sürekli) bir deęer olarak tahmin edilmesidir. Regresyon analizi gerçekteřtirilirken verilerin temsil edileceęi en



uygun eğrinin çizilmesi hedeflenir [12]. Çizilen eğrinin denklemi regresyon modelini temsil etmektedir. Örneğin, hava durumu tahmini, bir evin büyüklüğüne ve lokasyonuna göre fiyat tahmini, borsa tahmini vb. En sık kullanılan regresyon metodları: Doğrusal Regresyon (Linear Regression), Çoklu Doğrusal Regresyon (Multiple Linear Regression), Polinom Regresyon (Polynomial Regression), Karar Ağaçları (Decision Trees), Rastgele Ormanlar (Random Forest).

#### **4.9.2. Denetimsiz öğrenme**

Veri setinde sınıflara ayrılmamış veriler ile çalışılır, girdi değerlerine karşılık hangi çıktıların oluştuğu bilinmemektedir. Denetimsiz öğrenme ile veri kümesinde etiketlenmemiş veriler arasındaki ilişkileri ortaya çıkarmak amaçlanır, verinin ne tür bir desene sahip olduğunu bulmaya çalışır. Denetimsiz öğrenme problemleri: Kümeleme (Clustering) ve Boyut Azaltma (Dimensionality Reduction) olarak ikiye ayrılır.

Kümeleme, veri setindeki örnekler arasındaki benzerliklerin hesaplanması ile kümelerin (sınıfların) oluşturulmasıdır. En sık kullanılan kümeleme metodları: K-Ortalamalar Kümeleme (K-Means Clustering), Hiyerarşik Kümeleme (Hierarchical Clustering).

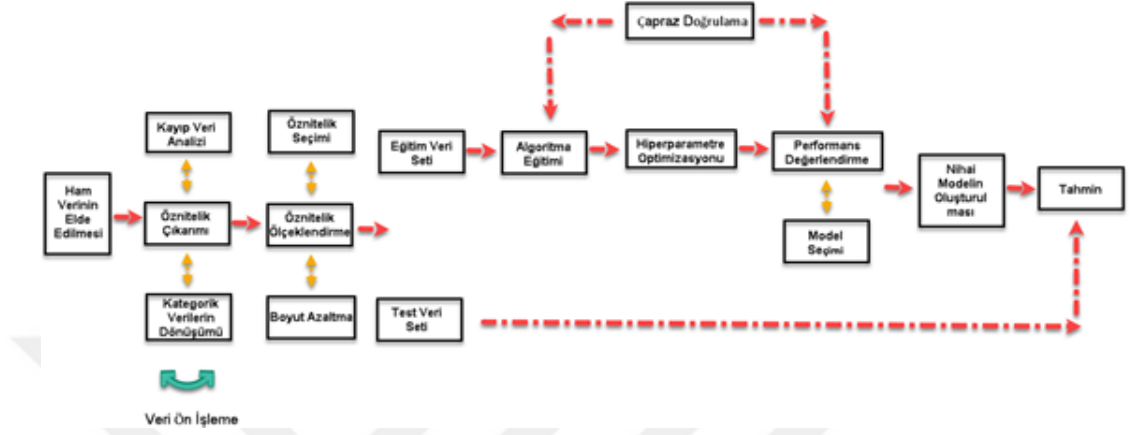
Boyut azaltma, bilgi kaybını en az düzeyde tutarak öznitelik sayısının düşürülmesi yöntemidir. Veri setleri çok sayıda öznitelik içerebilir ve bu veri setleriyle çalışmak yüksek hesaplama gücü ve çok zaman gerektirir. Çok fazla öznitelik barındıran veri setlerinde veri analizi ile alakalı sorunlar da oluşabilir. Ayrıca çok boyutlu verileri görselleştirmek zor olacaktır. Tüm bu sebeplerden ötürü boyut indirgeme yöntemi kullanılır. Kullanılan yöntemin adı: Temel Bileşen Analizi (Principal Component Analysis).

#### **4.9.3. Takviyeli öğrenme**

Amaç, çevresi ile sahip olduğu etkileşimlere dayanarak performansını artıran bir sistem geliştirmektir. Bu performans artışı bir ödül-ceza sistemine dayanır. Örneğin yüksek sözlü notu ile bir öğrencinin teşvik edilmesi gibi doğru karar verildiğinde ödüllendirme, yanlış karar verildiğinde cezalandırma prensibine dayanır [12].

#### 4.10. Makine Öğrenmesi Uygulama Geliştirme İş Akışı

Şekil 4.7’de bir denetimli makine öğrenmesi problemi çözümü için temel iş akışı gösterilmiştir.



Şekil 4.7. Makine öğrenmesi uygulama geliştirme iş akışı

## 5. MAKİNE ÖĞRENMESİ İLE KESTİRİMCİ BAKIM

Pek çok kestirimci bakım yaklaşımı makine öğrenmesi algoritmalarına dayanır, bu nedenle doğru bir model oluşturmak için yeterli veri olmalıdır. Kestirimci bakım için bu veriler genellikle makinelerdeki sensörlerden gelir. Sensörlerden alınan normal çalışma koşulları verileri kaydedilir. Yeterli sayıda ve aralıklarda kayıt edilen veri makine öğrenmesi algoritması ile anlık alınan yeni veriler üzerinden sürekli ve anlık olarak test edilir. Oluşabilecek anomali durumlarını tespit etmek için Anomali Tespiti (Anomaly Detection) yöntemi kullanılır.

### 5.1. Anomali Tespiti

Anomali tespiti, verilerin çoğunluğundan önemli ölçüde farklılık gösteren olayların belirlenmesidir.

Makine öğrenmesi ile kestirimci bakımda kullanılan anomali tespiti yöntemine gerçek hayattan bir örnek Rolls-Royce firması gösterilebilir. Rolls-Royce, 500 havayolu ve 150'den fazla askeri kuvvet tarafından kullanılan çok büyük motorlar imal eden bir firmadır [9]. Bu tarz hayati önem taşıyan ürünlerin sağlık durumunu takip etmesi ve potansiyel sorunları daha gerçekleşmeden saptanması çok önemlidir. Motor üzerine yerleştirilen sensörlerden alınan veriler analiz edilerek anormallikler araştırılır ve motorun bakıma ihtiyacı olup olmadığı durum tespit edilir. Firmanın topladığı veriler daha sağlam ürünler tasarlamasına, ürünlerini etkili bir şekilde korumasına ve tüm müşterilerine daha iyi bir hizmet sunmasına yardımcı olmaktadır.

Başka bir örnek GE firması yaklaşık 150 milyar dolar gelir elde eden ve pek çok alt firmasında 300.000'den fazla çalışanı olan dev şirkettir [9]. Enerji santralleri düzeyinde bir operasyon söz konusu olduğunda, çalışma verimliliğindeki ufak bir değişiklik, işletim maliyetleri ve güvenlik uygulamaları üzerinde çok büyük etkilere sahip olabilir. Önemli makinelerin arızalanması doğrudan gelir kaybına neden olabilir ve sistemlerin bakımı ve onarımı için yüksek maliyetli işgücü gereklidir. GE'nin çalıştığı tüm sektörlerdeki makinelere yerleştirilmiş sensörler tarafından toplanan ver

iler, makinenin nasıl çalıştığı hakkında bilgi vermesi için ölçülür ve analiz edilir. Böylece küçük değişikliklerin yarattığı etki yakından takip edilir. Örneğin, havacılık sektöründe, bakım işlemleri otomatik olarak programlanır ve böylece beklenmedik bakım işlerinin yol açacağı ertelemeler veya atölye sıklığı azaltılır. Sistem, parçaların ne zaman arızalanma ihtimali olduğu tahmin eder ve motorların onarım işlemi için randevu almanın yanı sıra, gerekli olan yedek parçaların bulunmasını sağlar. GE sadece makinelerin arızalarının azaltılması sonucunda yılda ortalama 8 milyon dolar tasarruf sağlayabileceğini söylemiştir [9].

Arızaları tahmin etmek ve minimumda tutmak mümkünken makinelerin arızalanmasını beklememek gerekir. Bilgisayarlar bu tür tahminleri insanlardan çok daha güvenilir şekilde yaparlar.

Anomali tespiti problemi esas olarak denetimsiz öğrenme problemi olsa da, denetimli öğrenme problemine benzeyen bazı yönleri de vardır. Anomali tespiti için çok sayıda makine öğrenmesi yöntemi bulunmaktadır. Bunlar: Isolation Forest (IF), Orca (KNN), Local Outlier Factor (LOF), Random Forest (RF) ve Yoğunluk Tahmini (Density Estimation). Bu tezde Yoğunluk Tahmini yöntemi kullanılmış ve anlatılmıştır.

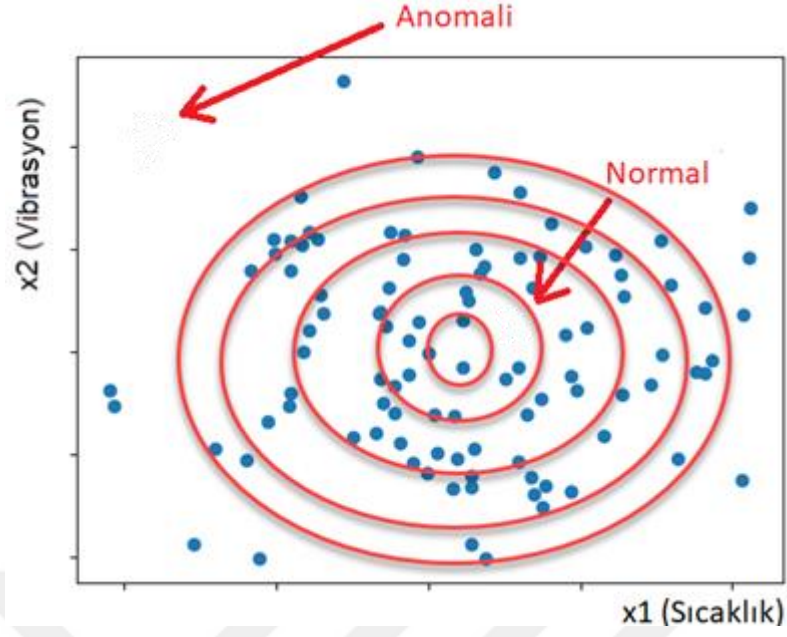
## 5.2. Yoğunluk Tahmini (Density Estimation)

Anomali içermeyen ve normal dağılmış veri seti ile eğitim yapılır, verinin içerisinde anomali içeren örnekler olmasına gerek yoktur. Eğitim seti etiketsiz yani çıktı değerlerinin olmadığı şekildedir. Oluşturulan model bir olasılık modelidir ve  $p(x)$  ile ifade edilir.

Örneğin bir uçak motoruna ait veri seti, denklem (5.1)'deki gibi ifade edilmiştir.

$$\text{Veriset: } \{x^1, x^2, \dots, x^m\} \quad (5.1)$$

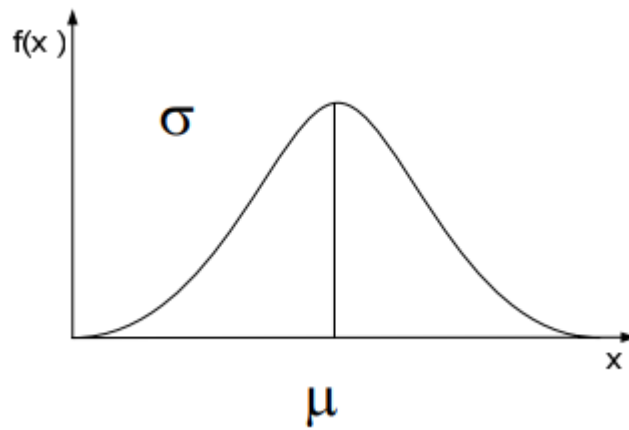
Şekil 5.1'de uçak motoruna ait iki tane özneliği olan modelin grafiği çizilmiştir. Bunlar: sıcaklık ve vibrasyon. Bu iki öznelik kullanılarak makine öğrenmesi modeli oluşturulur. Oluşturulan model yeni test verileriyle anlık olarak test edilir. Şekil 5.1'de gösterildiği gibi hesaplanan yoğunluk tahmini ile çemberlerin olduğu düşünülürse, çemberin içinde kalan noktalar normal noktalar, dışında kalan noktalar anomali noktalarıdır.



Şekil 5.1. Uçak motoruna ait oluşturulan model

### 5.3. Gauss Dağılımı (Normal Dağılım, Gaussian Distribution)

Yoğunluk tahmini algoritmasında kullanılacak olan verinin olması gereken dağılımıdır. Normal dağılım, günlük yaşama uygunluğu en yüksek dağılımdır. Şekil 5.2’de bir çan eğrisine benzemektedir ve denklem (5.2)’deki gibi hesaplanır. Bu denklem gauss dağılımının, olasılık yoğunluk fonksiyonu (probability dense function) olarak geçmektedir [18].



Şekil 5.2. Normal dağılım

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.2)$$

$$\mu = \frac{1}{n} \sum_{i=1}^n x(i) \quad (5.3)$$

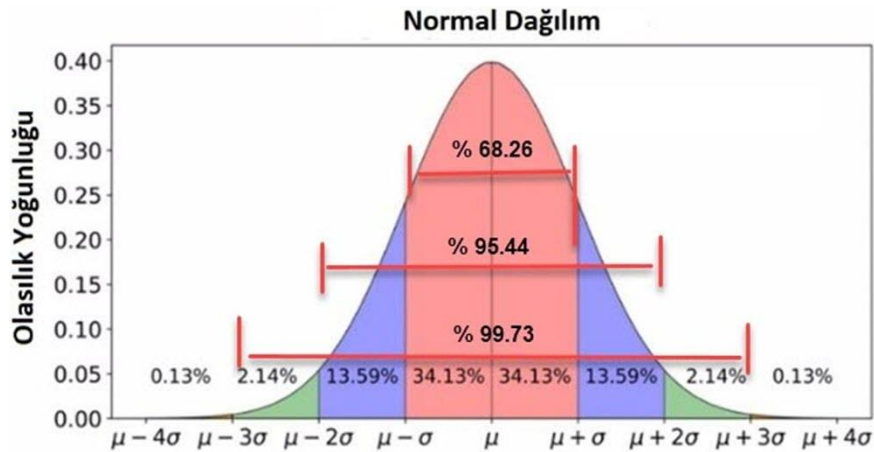
$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x(i) - \mu)^2} \quad (5.4)$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x(i) - \mu)^2 \quad (5.5)$$

N, örneklem sayısı (sample size), i, indis olmak üzere Denklem (5.3)'de  $\mu$  ifadesi ortalama değerdir ve çan eğrisinin tepe noktasının hangi x değeri için olduğunu belirler. Denklem (5.4)'de  $\sigma$  ifadesi dağılımın üzerinde uygulandığı topluluğun standart sapmasıdır ve çan eğrisinin şeklinin genişliğini belirtir. Standart sapma, verilerin ne kadarının ortalamaya yakın olduğunu gösterir. Standart sapma küçükse veriler ortalamaya yakın yerlerde dağılmıştır denir. Standart sapma küçüldükçe eğri dikleşmeye, standart sapma büyüdükçe eğri düzleşmeye başlamaktadır. Bunun temel sebebi ise normal dağılım bir yoğunluk eğrisidir ve toplam alanı 1'e eşittir. Standart sapma arttıkça yükseklik azalmakta ve genişlik artmaktadır ancak toplam alan değişmemektedir [19]. Denklem (5.5)'de  $\sigma^2$  ifadesi varyansdır, standart sapmanın karesidir. Çan eğrisinin altında kalan alan 1'dir. Maksimum nokta  $\mu$  ile eşleşir. Eğri simetriktir.

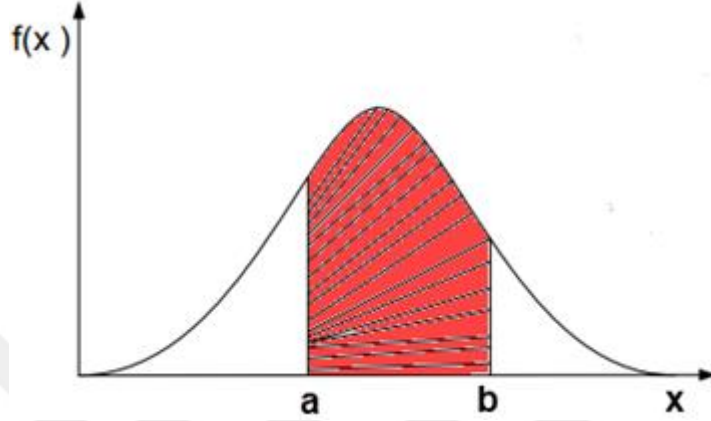
Normal dağılımda Şekil 5.3'de gösterildiği gibi güven aralıkları vardır. Bunlar;

- $1\sigma$  ve  $-1\sigma$  arasındaki değerler dağılımın %68.26'sının oluşturur.
- $2\sigma$  ve  $-2\sigma$  arasındaki değerler dağılımın %95.44'ünün oluşturur.
- $3\sigma$  ve  $-3\sigma$  arasındaki değerler dağılımın %99.73'ünün oluşturur.



Şekil 5.3. Normal dağılım güven aralıkları

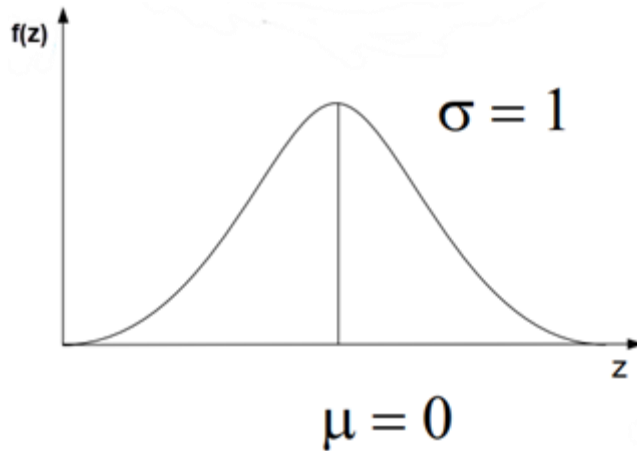
Bütün dağılım fonksiyonlarının ortak özelliği, çizilen eğrinin altında kalan alanların toplamı yani, olasılıkların toplamı 1'dir. Bir olasılık dağılımında tek bir noktanın olasılığı yoktur, bir alan oluşturması gerekir. Şekil 5.4'de eğri altında kalan taralı alanın olasılığını hesaplamak için Denklem (5.6) kullanılır.



Şekil 5.4. Normal dağılımda olasılık hesabı

$$P(a < x < b) = \int_a^b f(x) dx \quad (5.6)$$

Denklem (5.6)'da verilen integrali almak her seferinde zor olacağından ve işlem yükü getireceğinden dolayı, bu integraller hesaplanmış ve bir Z tablosunda oluşturulmuştur. Z tablosu, normal dağılıma ait olasılıkları veren tablodur. Ortalaması sıfır ve varyansı 1 haline getirilen normal dağılıma standart normal dağılım denir. Şekil 5.5'de verilen standart normal dağılım grafiğinin altında kalan alanlar (olasılıklar) Z tablosu ile hesaplanır.

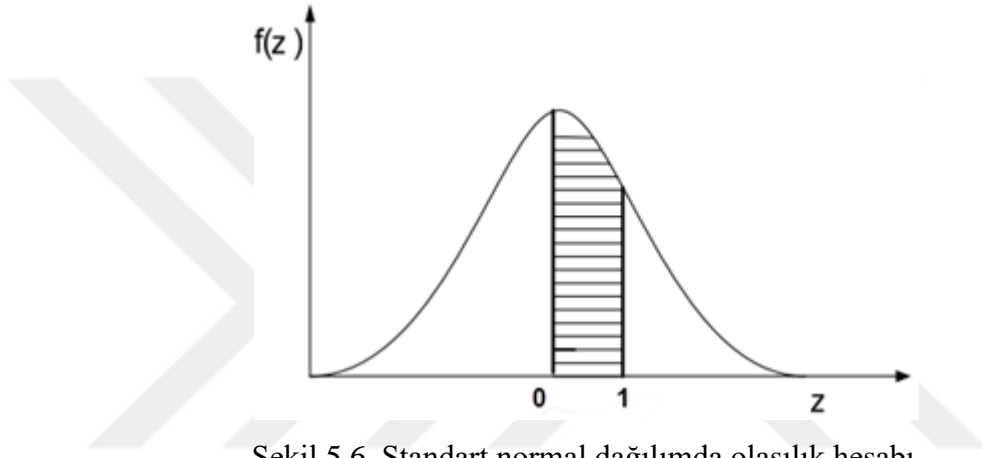


Şekil 5.5. Standart normal dağılım

Bir deęişken normal daęılımdan, standart normal daęılım deęişkenine dönüştürülürken denklem (5.7) kullanılır. Bu işleme standartlaştırma denir.

$$z = \frac{x - \mu}{\sigma} \quad (5.7)$$

Örneğin şekil 5.6'de 0 ile 1 aralığındaki olasılık hesaplanmak istendiğinde Z tablosunda karşılık gelen deęer bulunur. Bir veri setinin ortalamasını ve varyansı biliniyorsa Denklem (5.7) kullanılarak dönüştürme yapılır ve bulunan z deęerlerine karşılık, Z tablosuna bakılarak olasılık hesapları yapılabilir.



Şekil 5.6. Standart normal daęılımda olasılık hesabı

Bir veri seti normal daęılıma sahipse, ortalama ve varyansı biliniyorsa herhangi bir aralıktaki olasılık deęeri bulunabilir. İstatistikte küçük örneklere bakarak bütün hakkında fikir edinmek vardır. Veri seti normal daęılıma sahip deęilse Merkezi Limit Teorimi (Central Limit Theorem) ile normal daęılıma yakınsatılabilir.

Merkezi limit teorimi,  $\mu$  ortalamasına ve  $\sigma$  standart sapmasına sahip bir ana kütleden  $n$  büyüklüğünde rastgele tekrarlamalı örnekler alınırsa daęılım, normal daęılıma yakınsar. Bu daęılımın ortalama deęeri ana popülasyon ortalamasına yakın olacaktır. Aynı şekilde ana popülasyonun standart sapması, örneklem standart sapmasına yakın olacaktır. Örneklem ortalamasının daęılımı, örnekleme alınan daęılımdan bağımsız olarak, örneklem büyüklüğü arttıkça normal daęılıma doğru eęilim gösterir. Pratikte 30 adet örneklem veya daha fazlası normal daęılıma yakınsatmak için yeterlidir.



Bir veri setinin normal dağılım olup olmadığı bilinmiyorsa Denklem (5.7)'e göre z hesaplanmaz. Veri setine merkezi limit teorimi uygulandıktan sonra Denklem (5.8) kullanılır.

$$z = \frac{x - \mu}{\sigma/\sqrt{n}} \quad (5.8)$$

#### 5.4. Yoğunluk Tahmini Algoritması (Density Estimation Algorithm)

Normal dağılıma sahip olan bir veri setinden kullanılacak olan öznitelikler belirlenir. Bu tezin uygulamasında daima iki öznitelik kullanılacaktır. Seçilen öznitelikler kullanılarak Denklem (5.10) ve (5.11)'e göre model eğitme işlemi yapılır. Bu eğitme işlemine makine öğrenmesi terminolojisinde fit etmek denir. Bu makine öğrenmesi yönteminde fit edilen parametreler:  $\mu$  ve  $\sigma$ . Model fit edildikten sonra yeni test verileriyle Denklem (5.12) kullanılarak anlık olarak yoğunluk tahmini hesaplaması yapılır .

$$\text{Eğitim Seti} = \{x(1), \dots, x(m)\} \quad x \in \mathbb{R}^n \quad (5.9)$$

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad (5.10)$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2 \quad (5.11)$$

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_j}} e^{-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}} \quad (5.12)$$

Anlık yoğunluk tahmini hesabının sonucu, seçilen bir  $\mathcal{E}$  (*epsilon*) değerinden küçükse anomali vardır. Sonucun değerlendirilmesi anlık olarak Denklem (5.13)'e göre yapılır [20].

$$\begin{cases} \text{Eğer } p(x) < \mathcal{E}, & \text{Anomali} \\ \text{Eğer } p(x) > \mathcal{E}, & \text{Normal} \end{cases} \quad (5.13)$$

## 6. UYGULAMA

Makine öğrenmesi ile kestirimci bakım için gerçek hayattan bir örnek uygulama üzerinden anlatımı yapılmıştır. Bu uygulamada bir üretim makinesinde proses kaynaklı anlık bir anomali durumundan dolayı, üretilen ürünün makinenin bir bölgesine sıkışması durumunu algılayıp operatöre uyarı veren bir sistem oluşturulmuştur. Bu algılama yöntemi tezin üst kısımlarında detaylı olarak anlatılmıştır. Bu yöntem kodlanarak sistem üzerinden devreye alınması yapılmıştır. İlgili kodların detayları Ek’de belirtilmiştir.

### 6.1. Üretim Makinesi Hakkında Bilgilendirme

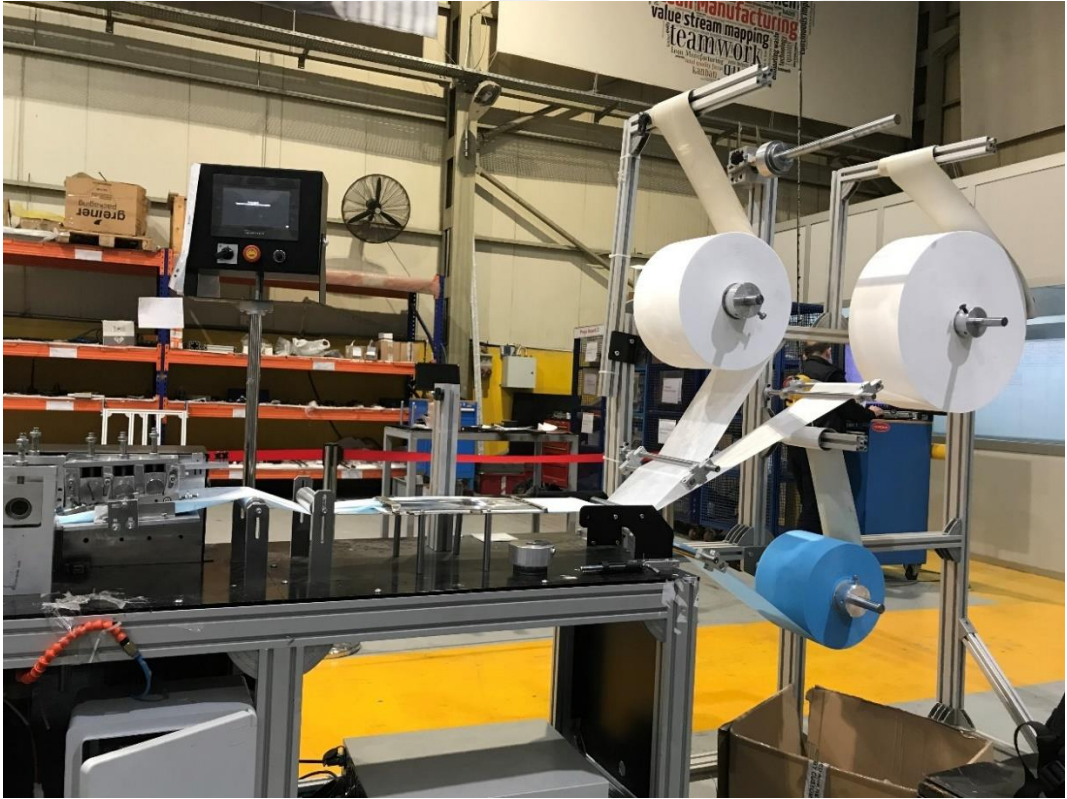
Bu tezde uygulaması yapılan örnek uygulama bir maske üretim makinesidir. Makine ile ilgili genel görünüm Şekil 6.1, Şekil 6.2, Şekil 6.3, Şekil 6.4’de ve Şekil 6.5’de gösterilmiştir.



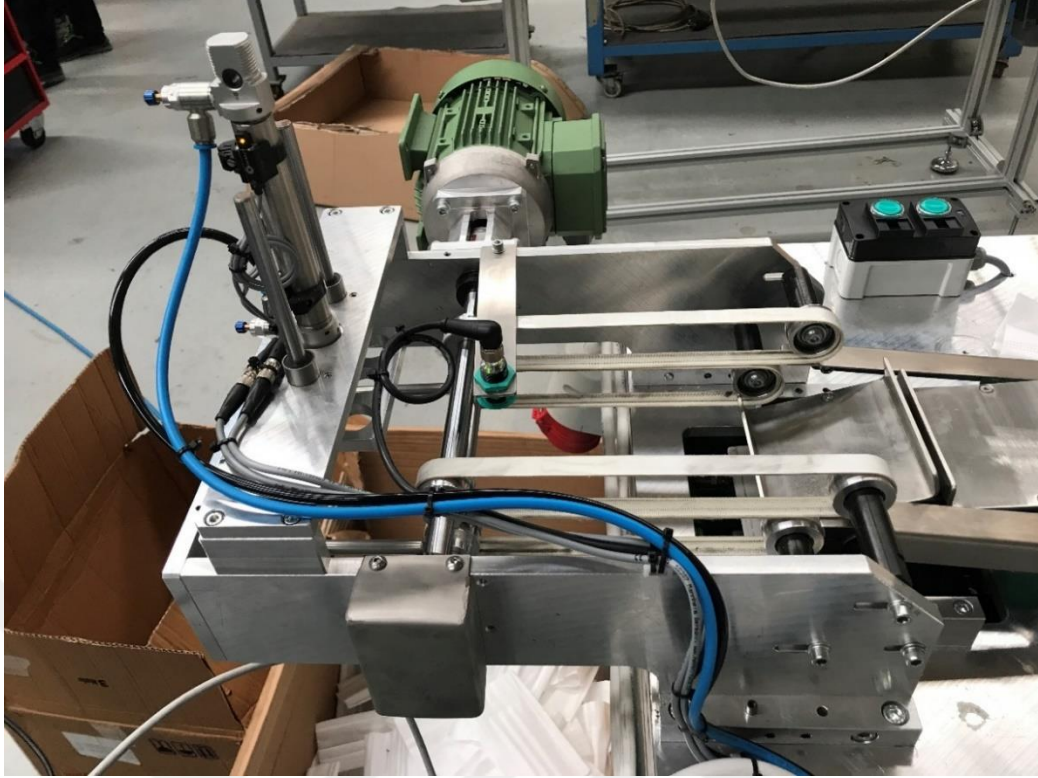
Şekil 6.1. Makine genel görünüm 1



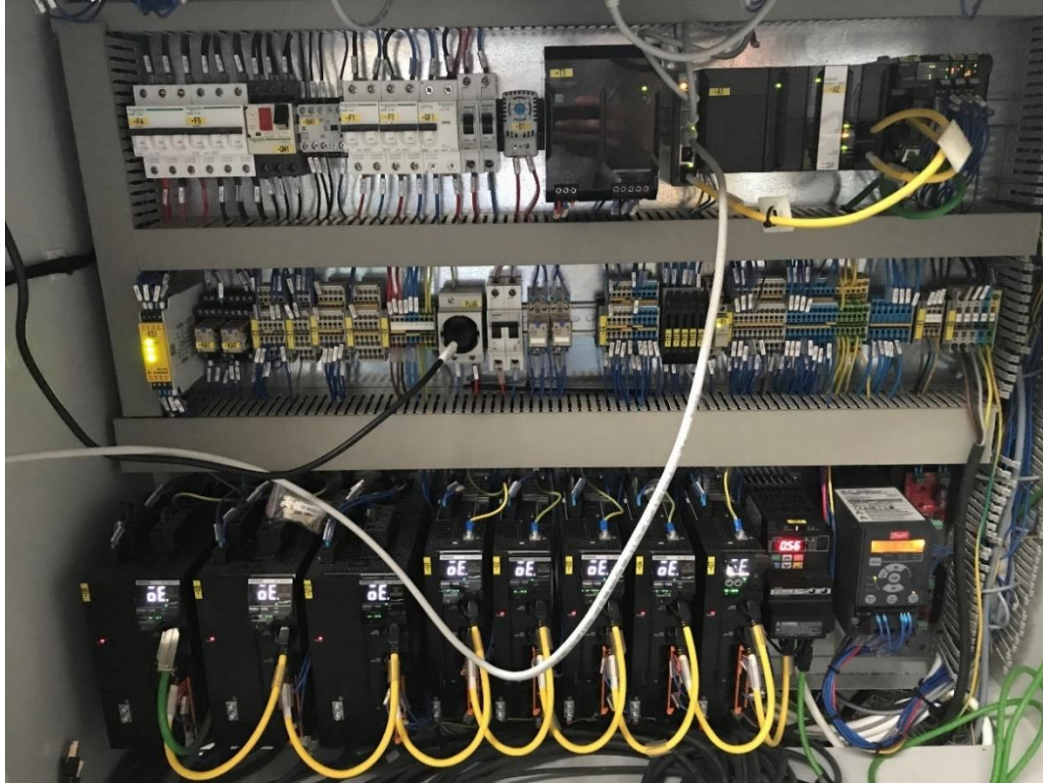
Şekil 6.2. Makine genel görünüm 2



Şekil 6.3. Makine genel görünüm 3



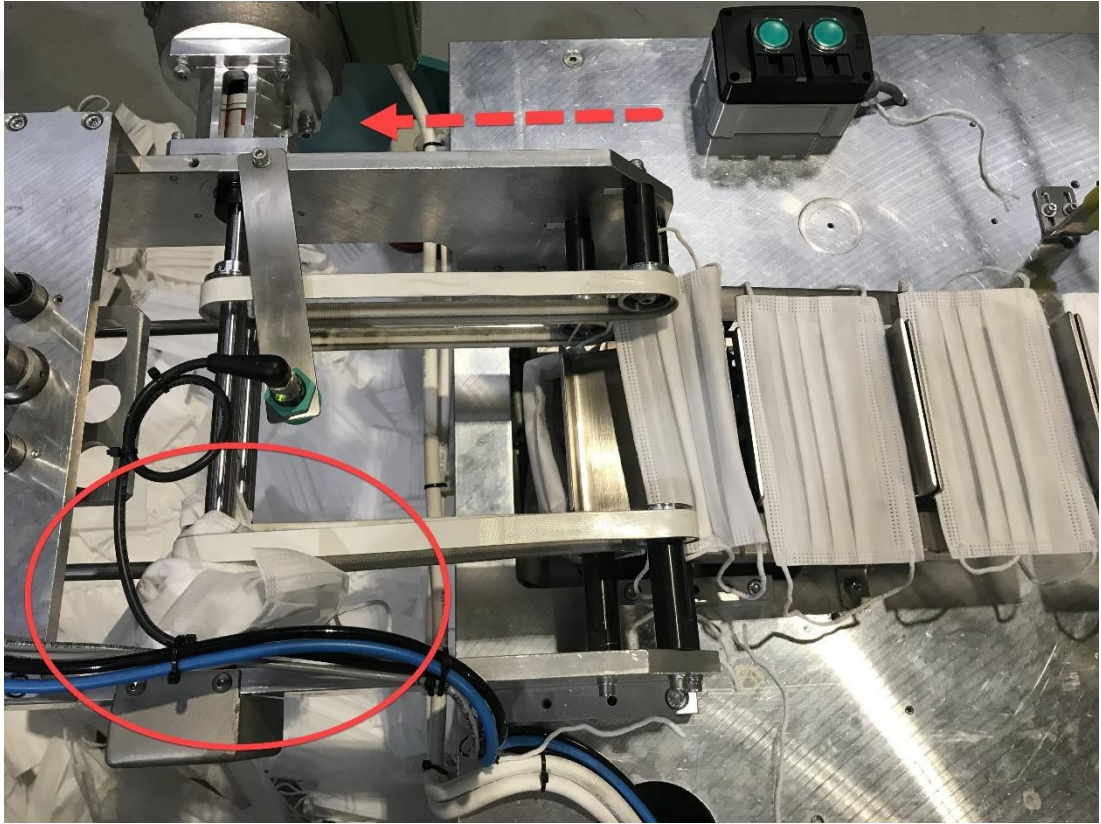
Şekil 6.4. Makine genel görünüm 4



Şekil 6.5. Makine elektrik panosu

## 6.2. Anomali Durumu

Üretilen ürün taşıyıcı konveyör üzerinde çıkışa doğru ilerledikten sonra, çıkışta asenkron bir motorun miline bağlı bir kayış sistemi ile ürünü hızlandırarak dışarıya çıkarılması sağlanır. Hızlanan ürün kayışın sonunda piston ile aşağı itilerek başka bir konveyör üzerine boşaltılır. Bazı durumlarda Şekil 6.6'da kırmızı daire içine alınmış olan durum meydana gelmektedir. Ürün kayışın arasına dolanarak motora durdurucu bir direnç göstererek motoru yavaşlatmakta ve ürün tahliyesi yapılamamaktadır. Amaç, bu problemi bir anomali olarak algılayıp operatöre bilgilendirmesi yapılmasıdır.

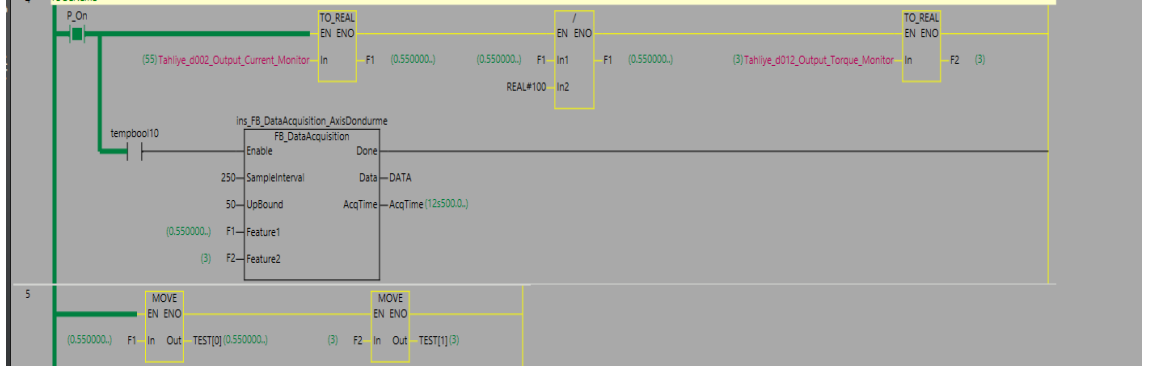


Şekil 6.6. Anomali durumu

## 6.3. Veri Toplama Uygulama

Kayış sistemi bir asenkron motora bağlıdır. Asenkron motor bir AC sürücü üzerinden çalışmaktadır. AC sürücü ile kontrolör (CPU) haberleşme ile birbirlerine bağlıdır. AC sürücünden motora ait veriler elde edilebilir. Asenkron motor için akım ve tork bilgileri en önemli ve herşeyi anlatan veriler olduğu için bu bilgileri sürücü üzerinden almak yeterli olacaktır. Kurulacak olan makine öğrenmesi sistemi motorun akım ve tork parametreleri kullanılarak iki öznitelikli olacaktır. Bu iki özniteliği kontrolör ile

sürücü üzerinden okunarak kayıt altına alınacaktır. Burada CPU olarak bir PLC kullanılmıştır.



Şekil 6.7. Veri toplama fonksiyon blok

Şekil 6.7’de motora ait akım ve tork bilgileri sürücüden okunarak F1 ve F2 parametrelerine atanmıştır. Akım değişkeninin birimi Amper (A), tork değişkenin birimi Newton metredir (Nm). Normal çalışma koşullarına ait verileri kayıt altına almak için bir Veri Toplama (Data Acquisition) fonksiyon bloğu kod parçacığı yazılmıştır. Bu fonksiyon bloğa ait kod parçacığı Ek-A’da gösterilmiştir. 250 ms aralıklarla 50 adet örnek toplanmıştır ve DATA değişkenine kayıt edilmiştir. Kayıt edilen verinin içeriği Şekil 6.8 ve Şekil 6.9’da gösterilmiştir.

Watch (Table)1			
Device name		Program0.DAT	
new_Controller_0			
	Feature1 (REAL)	Feature2 (REAL)	Class (REAL)
[0]	0.55000001	3	0
[1]	0.55000001	3	0
[2]	0.55000001	3	0
[3]	0.55000001	3	0
[4]	0.55000001	3	0
[5]	0.55000001	3	0
[6]	0.55000001	3	0
[7]	0.55000001	3	0
[8]	0.55000001	3	0
[9]	0.55000001	3	0
[10]	0.55000001	3	0
[11]	0.55000001	3	0
[12]	0.55000001	3	0
[13]	0.55000001	3	0
[14]	0.55000001	3	0
[15]	0.55000001	3	0
[16]	0.56	3	0
[17]	0.55000001	3	0
[18]	0.55000001	3	0
[19]	0.56	3	0
[20]	0.56	3	0
[21]	0.55000001	3	0
[22]	0.56	3	0
[23]	0.56	3	0
[24]	0.56	3	0
[25]	0.56	4	0

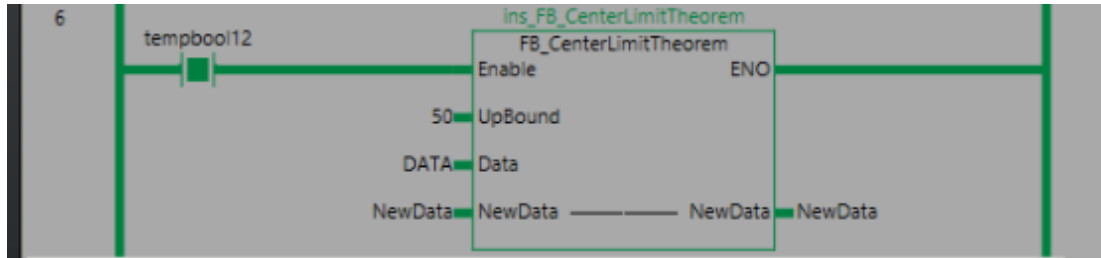
Şekil 6.8. DATA değişkenin içeriği 1

Watch (Table)1			
Device name		Program0.DATA	
Feature1 (REAL)	Feature2 (REAL)	Class (REAL)	
[26]	0.56	4	0
[27]	0.56	4	0
[28]	0.55000001	3	0
[29]	0.55000001	3	0
[30]	0.55000001	3	0
[31]	0.55000001	3	0
[32]	0.55000001	3	0
[33]	0.55000001	3	0
[34]	0.55000001	3	0
[35]	0.55000001	3	0
[36]	0.55000001	3	0
[37]	0.55000001	3	0
[38]	0.55000001	3	0
[39]	0.55000001	3	0
[40]	0.56	4	0
[41]	0.55000001	3	0
[42]	0.55000001	3	0
[43]	0.55000001	3	0
[44]	0.55000001	3	0
[45]	0.55000001	3	0
[46]	0.56	4	0
[47]	0.56	4	0
[48]	0.56	4	0
[49]	0.56	3	0
[50]	0.55000001	3	0
[51]	0	0	0

Şekil 6.9. DATA değişkeninin içeriği 2

#### 6.4. Merkezi Limit Teorimi Uygulama

Normal çalışma koşullarına ait veri kayıt edildikten sonra, veriyi normal dağılıma yakınsatmak için merkezi limit teorimi uygulanmıştır. Bunun için bir Merkezi Limit Teoremi (Center Limit Theorem) fonksiyon bloğu kod parçacığı yazılmıştır. Bu fonksiyon bloğa ait kod parçacığı Ek-B’de gösterilmiştir. Şekil 6.10’da gösterildiği gibi daha önce kayıt edilen ve DATA değişkenine atılan verilere, merkezi limit teorimi uygulanarak NewData değişkenine atanmıştır.



Şekil 6.10. Merkezi limit teoremi fonksiyon blok

NewData değişkenine ait veriler Şekil 6.11 ve Şekil 6.12’de gösterilmiştir.

Watch (Table)1			
Device name		Program0.NewData	
	new_Controller_0		
	Feature1 (REAL)	Feature2 (REAL)	Class (REAL)
[0]	0.55103451	3.1034484	0
[1]	0.55241382	3.0344827	0
[2]	0.55310345	3.0689654	0
[3]	0.55241382	3.1034484	0
[4]	0.55206895	3.1034484	0
[5]	0.55172414	3.0689654	0
[6]	0.55206895	3.0689654	0
[7]	0.55275863	3.1379311	0
[8]	0.55206895	3.1379311	0
[9]	0.55344826	3.1379311	0
[10]	0.55310345	3.1724138	0
[11]	0.55241382	3.0689654	0
[12]	0.55344826	3.1724138	0
[13]	0.55241382	3.1379311	0
[14]	0.55379313	3.2068965	0
[15]	0.55344826	3.1379311	0
[16]	0.55344826	3.1724138	0
[17]	0.55344826	3.1379311	0
[18]	0.55206895	3.0344827	0
[19]	0.55206895	3.1379311	0
[20]	0.55275863	3.0689654	0
[21]	0.55241382	3.1724138	0
[22]	0.55310345	3.1034484	0
[23]	0.55310345	3.1034484	0
[24]	0.55310345	3.1724138	0
[25]	0.55275863	3.1034484	0

Şekil 6.11. NewData değişkenin içeriği 1

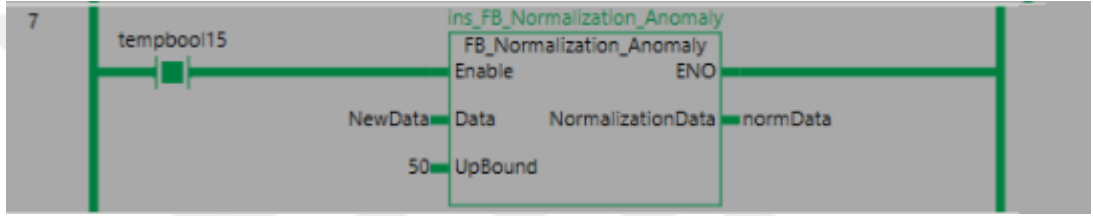
Watch (Table)1			
Device name		Program0.NewData	
	new_Controller_0		
	Feature1 (REAL)	Feature2 (REAL)	Class (REAL)
[26]	0.55448276	3.275862	0
[27]	0.55344826	3.2413793	0
[28]	0.55310345	3.2413793	0
[29]	0.55310345	3.1379311	0
[30]	0.55241382	3.1034484	0
[31]	0.55206895	3.1034484	0
[32]	0.55344826	3.1724138	0
[33]	0.55137932	3.0689654	0
[34]	0.55379313	3.1724138	0
[35]	0.55241382	3.0689654	0
[36]	0.55241382	3.1379311	0
[37]	0.55310345	3.2068965	0
[38]	0.55206895	3.1724138	0
[39]	0.55241382	3.1724138	0
[40]	0.55241382	3.1724138	0
[41]	0.55241382	3.1034484	0
[42]	0.55206895	3.1379311	0
[43]	0.55517244	3.3793104	0
[44]	0.55310345	3.0344827	0
[45]	0.55206895	3.0689654	0
[46]	0.55344826	3.2413793	0
[47]	0.55275863	3.1379311	0
[48]	0.55275863	3.1724138	0
[49]	0.55310345	3.2068965	0
[50]	0.55275863	3.1379311	0
[51]	0	0	0

Şekil 6.12. NewData değişkenin içeriği 2



## 6.5. Öznitelik Ölçekleme Uygulama

Makine öğrenmesi algoritmasında kullanılacak olan özniteliklerin birimleri ve büyüklükleri birbirinden farklı olduğu için bu değişkenlerin aynı değer aralığı ve koşullarına indirgenmesi gerekmektedir. Öznitelik Ölçekleme (Feature Scaling) algoritmalarından birisi olan Normalizasyon (Normalization) metodu kullanılmıştır ve bunun için bir Normalizasyon fonksiyon bloğu kod parçasığı yazılmıştır. Bu fonksiyon bloğa ait kod parçasığı Ek-C’de gösterilmiştir. Şekil 6.13’de gösterildiği gibi daha önce normal dağılım yakınsatması yapılan NewData değişkeni ölçeklemesi yapılarak NormData değişkenine atanmıştır.



Şekil 6.13. Normalizasyon fonksiyon blok

NormData değişkenine ait veriler Şekil 6.14 ve Şekil 6.15’de gösterilmiştir.

Watch (Table)1			
Device name		Program0.normData	
new_Controller_0	Feature1 (REAL)	Feature2 (REAL)	Class (REAL)
[0]	0	0.20000042	0
[1]	0.33333334	0	0
[2]	0.49999279	0.09999986	0
[3]	0.33333334	0.20000042	0
[4]	0.2499892	0.20000042	0
[5]	0.16665946	0.09999986	0
[6]	0.2499892	0.09999986	0
[7]	0.41666308	0.30000028	0
[8]	0.2499892	0.30000028	0
[9]	0.58332253	0.30000028	0
[10]	0.49999279	0.40000013	0
[11]	0.33333334	0.09999986	0
[12]	0.58332253	0.40000013	0
[13]	0.33333334	0.30000028	0
[14]	0.66666669	0.5	0
[15]	0.58332253	0.30000028	0
[16]	0.58332253	0.40000013	0
[17]	0.58332253	0.30000028	0
[18]	0.2499892	0	0
[19]	0.2499892	0.30000028	0
[20]	0.41666308	0.09999986	0
[21]	0.33333334	0.40000013	0
[22]	0.49999279	0.20000042	0
[23]	0.49999279	0.20000042	0
[24]	0.49999279	0.40000013	0
[25]	0.41666308	0.20000042	0

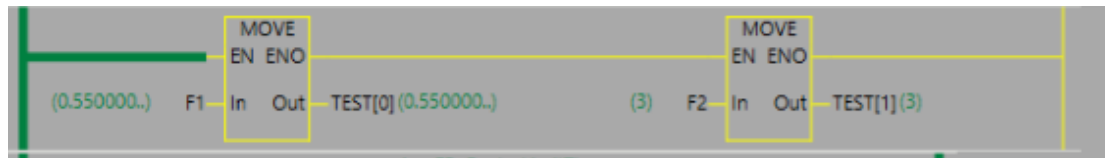
Şekil 6.14. NormData değişkeninin içeriği 1

Watch (Table1)			
Device name		Program0.normData	
new_Controller_0		Program0.normData	
	Feature1 (REAL)	Feature2 (REAL)	Class (REAL)
[26]	0.83332614	0.69999975	0
[27]	0.58332253	0.59999985	0
[28]	0.49999279	0.59999985	0
[29]	0.49999279	0.30000028	0
[30]	0.33333334	0.20000042	0
[31]	0.2499892	0.20000042	0
[32]	0.58332253	0.40000013	0
[33]	0.08332973	0.09999986	0
[34]	0.66666669	0.40000013	0
[35]	0.33333334	0.09999986	0
[36]	0.33333334	0.30000028	0
[37]	0.49999279	0.5	0
[38]	0.2499892	0.40000013	0
[39]	0.33333334	0.40000013	0
[40]	0.33333334	0.40000013	0
[41]	0.33333334	0.20000042	0
[42]	0.2499892	0.30000028	0
[43]	1	1	0
[44]	0.49999279	0	0
[45]	0.2499892	0.09999986	0
[46]	0.58332253	0.59999985	0
[47]	0.41666308	0.30000028	0
[48]	0.41666308	0.40000013	0
[49]	0.49999279	0.5	0
[50]	0.41666308	0.30000028	0
[51]	0	0	0

Şekil 6.15. NormData değişkenin içeriği 2

## 6.6. Test İçin Öznitelik Ölçekleme Uygulama

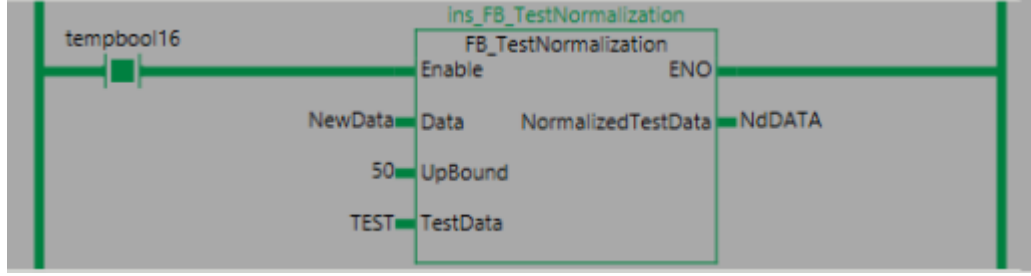
Buraya kadar makine öğrenmesi algoritması için eğitim verisi olan NormData oluşturuldu. Algoritmayı çalıştırmak için sistemden anlık olarak alınan test verisi algoritmaya gönderilecek ancak eğitim verisi ölçeklendirilmiş veri olduğu için test verisinde ölçeklendirilmiş şekilde olmalıdır. Bunun için Test verisine ait ölçekleme fonksiyon bloğu kod parçacığı yazılmıştır. Bu fonksiyon bloğa ait kod parçacığı Ek-D'de gösterilmiştir.



Şekil 6.16. Motor parametrelerinin TEST değişkenine atanması

Eğitim verisi oluşturduktan sonra test aşamasına geçilecektir. Makine çalıştırdıktan sonra motora ait akım ve tork bilgileri Şekil 6.16'da gösterildiği gibi TEST değişkeni

içerisine sürekli olarak gönderilmektedir. TEST değişkeni bir dizidir. 0. elemanı akım 1. elemanı tork bilgisidir.

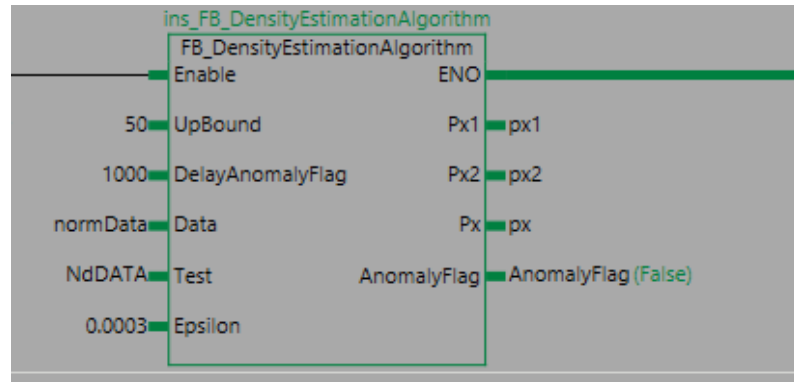


Şekil 6.17. Test için normalizasyon fonksiyon blok

TEST değişkeni içerisine anlık olarak gönderilen motor bilgileri Şekil 6.17’de gösterildiği gibi normal dağılımı yapılmış NewData verileri kullanılarak ölçeklendirmesi yapılmıştır ve NdDATA değişkenine atanmıştır.

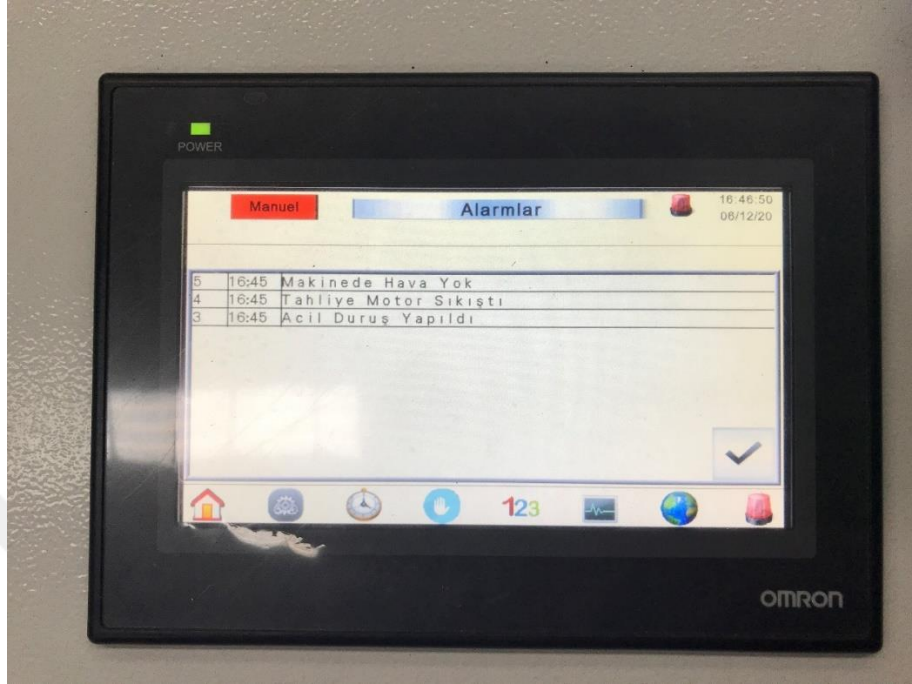
### 6.7. Makine Öğrenmesi Algoritması Uygulama

Şekil 6.18’de buraya kadar hazırlığı yapılan tüm süreçlerin uygulandığı Yoğunluk Tahmini Algoritması’na ait bir fonksiyon blok kod parçacığı yazılmıştır. Bu fonksiyon bloğa ait kod parçacığı Ek-E’de gösterilmiştir. Eğitimi yapılan ve ölçeklendirilmiş NormData, test için motordan alınan anlık olarak ölçeklendirilmiş NdData, fonksiyon blok girişine bağlanmıştır. Hesaplama sonucunu, uygulamaya göre belirlenen bir eşik değeri olan  $\mathcal{E}$  değeri ile karşılaştırarak bir anomali bit durumu değişkeni bloğun çıkışına gönderilir. Fonksiyon bloğun içinde normal çalışma koşullarına ait hesaplanan olasılık fonksiyonu Ppx değişkenin, istenilen bir yüzde değeri alınarak da  $\mathcal{E}$  değerine atama yapılarak eşik değeri belirlenebilir. Hesaplama sonucu epsilon değerinden küçükse ve bir filtre süre gecikme uygulanırsa AnomalyFlag biti TRUE olur.



Şekil 6.18. Yoğunluk tahmini algoritması fonksiyon blok

Anomali durum deęiřkeni TRUE olursa operatöre bir operatör ekranı üzerinden Şekil 6.19'daki gibi tahliye motor sıkıřtı gibi bir ifade ile bilgilendirme mesajı oluřturulur.



Şekil 6.19. Operatör ekranı uyarı mesajı

## 7. SONUÇLAR VE ÖNERİLER

Her işletme için Endüstri 4.0 teknolojilerinden, Nesnelerin İnterneti, Büyük Veri ve Analizi ile Yapay Zeka'yı kullanarak, hız, esneklik ve verimlilik alanlarında iyileştirmeler yapılması gereği nedenleriyle birlikte önceki bölümlerde anlatılmıştır.

Yapılan çalışma hiyerarşik olarak ifade edilirse; kestirimci bakım, makine öğrenmesi, anomali tespiti ve yoğunluk tahmini algoritması şeklindedir. Bir üretim makinesine ait bir motorun ilgili parametreleri kayıt altına alınarak, proses kaynaklı bir sorun oluştuğunda, gerçek zamanlı durum izlemesi ile operatörü anlık bilgilendirerek oluşabilecek büyük hasarların önüne geçilmiş olacaktır.

Makine öğrenmesi ile kestirimci bakımda hem fiziksel bir hasarın önüne geçmek hem de makinenin duruş süresinden kaynaklı üretim kayıplarını azaltmak mümkün olur. Örneğin, makinenin dakikada 70 adet üretim yaptığı düşünülüğünde söz konusu makine saatte 4200 adet, günde 100800 adet ürün üretmiş olur. Kestirimci bakım kullanılmayıp gün içerisinde bahsedilen türde sadece bir anomali olduğu varsayımı ile günde 2 saatlik ayar ve düzenleme duruşu olacaktır. Bu duruş gün içinde 8400 adetlik kayıp anlamına gelmektedir. Makine öğrenmesi ile kestirimci bakım uygulanması durumunda, makinede herhangi bir fiziksel arıza olmadan önce sistem durdurulup operator bilgilendirilebilecektir. Böylece hem zaman hem de maddi kayıp olmadan sistem düzeltilebilecektir.

Tezde yapılan çalışmada makine öğrenmesi ile kestirimci bakım ve anomali tespitinin nasıl gerçekleşebileceği ve gerçek hayatta işletmelerde nasıl kullanılabileceği ile ilgili yapılan gerçek bir uygulama gösterilmiştir. İşletme ihtiyaçlarına dayalı olarak ve hayal gücü ile sınırsız bir şekilde bu uygulamalar çeşitlendirilebilir ve geliştirilebilir.

Makine öğrenmesi ile kestirimci bakım ve anomali tespitinde, farklı algoritmalar kullanılarak bu tezde uygulanan algoritma ile karşılaştırılması yapılabilir. Bu tezde kontrolcü olarak bir PLC kullanılmıştır ancak istenilirse tüm bu algoritmalar pyton dili

ile yazılarak bilgisayarda kořturulup üretilecek olan anomali sinyalinini, PLC'ye haberleşme ile gönderdikten sonra anomali şartları oluşturulabilir.



## KAYNAKLAR

- [1] BS EN 13306, Maintenance - Maintenance terminology, *British Standards Institution (BSI)*, London, 2010.
- [2] Aksöz E., Arifoğlu B., Makine Öğrenmesi İle Kestirimci Bakım Ve Anomali Tespiti Analizi, *4. Uluslararası Uygulamalı Bilimler Kongresi*, Artvin, Türkiye, 13 Eylül 2020.
- [3] Öztanır O., Makine Öğrenmesi Kullanılarak Kestirimci Bakım, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2018.
- [4] Mathworks, Introduction to Predictive Maintenance with MATLAB, 2019.
- [5] Mathworks, Predictive Maintenance with MATLAB, 2016.
- [6] Mathworks, Three Ways to Estimate Remaining Useful Life, 2018.
- [7] Ersoy A. R., Dijital Enterprise Industry 4.0, 2016.
- [8] Apilioğulları L., *Dijital Dönüşümün Yol Haritası*, 2. Baskı, İstanbul, 2019.
- [9] Marr B., *Büyük Veri İş Başında*, 1. Baskı, MediaCat, İstanbul, 2020.
- [10] Gürsakal N., *Makine Öğrenmesi*, 1. Baskı, Dora, Bursa, 2018.
- [11] Yılmaz A., *Yapay Zeka*, 5. Baskı, Kodlab, İstanbul, 2019.
- [12] Uğuz S., *Makine Öğrenmesi*, 1. Baskı, Nobel, Ankara, 2019.
- [13] Singh K., Understanding Different Components & Roles in Data Science. [https://dimensionless.in/understanding-different-components-roles-in-data-science/?utm\\_source=dlvr.it&utm\\_medium=twitter](https://dimensionless.in/understanding-different-components-roles-in-data-science/?utm_source=dlvr.it&utm_medium=twitter) (Ziyaret tarihi: 10 Kasım 2020).
- [14] Öztemel E., *Yapay Sinir Ağları*, 4. Baskı, PapatyaBilim, İstanbul, 2016.
- [15] Şeker Ş. E., Python ile Makine Öğrenmesi, Udemy, <https://www.udemy.com/course/makine-ogrenmesi/> (Ziyaret tarihi: 28 Ocak 2019).
- [16] Team D., A'dan Z'ye Makine Öğrenmesi, Udemy, <https://www.udemy.com/course/machine-learning-ve-python-adan-zye-makine-ogrenmesi-4/> (Ziyaret tarihi: 3 Kasım 2020).

- [17] Amidi S., Machine Learning tips and tricks cheatsheet, Stanford University, <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks#model-selection> (Ziyaret tarihi: 5 Kasım 2020).
- [18] Şeker S. E., Normal Dağılım (Normal Distribution, Gauss Distribution). <http://bilgisayarkavramlari.sadievrenseker.com/2011/06/08/normal-dagilim-normal-distribution-gauss-distribution/> (Ziyaret tarihi: 22 Ekim 2020).
- [19] Öğündür G., Normal Dağılım , Z-Score Standardizasyon ve Normalizasyon. <https://medium.com/@gulcanogundur/normal-dagilim-z-score-ve-standardizasyon-782963bc123e> (Ziyaret tarihi: 22 Ekim 2020).
- [20] Ritchie N., Anomaly Detection. <https://www.ritchieng.com/machine-learning-anomaly-detection/> (Ziyaret tarihi: 22 Ekim 2020).







**EKLER**

## Ek-A

```
IF Enable=TRUE THEN
IF LastTrigger[0]=FALSE THEN
Clear(Data);
END_IF;
// get Clock
TSampleStart:=TRUE;
IF GetClock=TRUE THEN
TSampleStart:=FALSE;
END_IF;
TSample(In:=TSampleStart, PT:=NanoSecToTime(In:=SampleInterval*1000000),
Q=>GetClock);
//
IF i < UpBound+1 THEN
IF GetClock=TRUE THEN
Data[i].Feature1:=Feature1;
Data[i].Feature2:=Feature2;
INC(i);
END_IF;
ELSE
Done:=TRUE;
END_IF;
// n adet veri ne kadar zaman sürecektir
Tempms:= SampleInterval*UpBound;
AcqTime:=NanoSecToTime(In:=Tempms*1000000);
LastTrigger[0]:=TRUE;
ELSE
i:=0;
TSample(In:=FALSE);
Done:=FALSE;
LastTrigger[0]:=FALSE;
END_IF;
```

## Ek-B

```
LowBound:=0;
ArySize:=(UpBound-LowBound)+1; // dizi uzunluğu
IF Enable=TRUE THEN
//merkezi limt teorimine göre dataları normal dağılıma yakınsat
IF LastTrigger[0]=FALSE THEN
i[0]:=0;
TempRnd1:=0;
TempRnd2:=0;
SelectIndex1:=0;
SelectIndex2:=0;
LastTrigger[0]:=TRUE;
FOR i[0]:=LowBound TO UpBound BY 1 DO
FOR i[1]:=0 TO 29 BY 1 DO // pratikte 30 örnek seçmek yeterli
ins_Rand(Execute:=TRUE, Seed:=UINT#42);
TempRnd1:=TO_REAL( ins_Rand.Rnd*(ArySize-1));
SelectIndex1:=TO_INT( RoundUp(TempRnd1));
SelectionAry1[i[1]]:=Data[SelectIndex1].Feature1;
ins_Rand(Execute:=FALSE);
//
ins_Rand2(Execute:=TRUE, Seed:=UINT#42);
TempRnd2:=TO_REAL( ins_Rand2.Rnd*(ArySize-1));
SelectIndex2:=TO_INT( RoundUp(TempRnd2));
SelectionAry2[i[1]]:=Data[SelectIndex2].Feature2;
ins_Rand2(Execute:=FALSE);
END_FOR;
NewData[i[0]].Feature1:=AryMean(In:=SelectionAry1[0], Size:=29);
NewData[i[0]].Feature2:=AryMean(In:=SelectionAry2[0], Size:=29);
END_FOR;
END_IF;
ELSE
Clear(LastTrigger);
```

```
Clear(i);  
Clear(NewData);  
Clear(SelectionAry1);  
Clear(SelectionAry2);  
ins_Rand(Execute:=FALSE);  
ins_Rand2(Execute:=FALSE);  
END_IF;
```



## Ek-C

```
LowBound:=0;
ArySize:=(UpBound-LowBound)+1; // dizi uzunluğu
IF Enable=TRUE THEN
TempAryMin1:=Data[LowBound].Feature1;
TempAryMax1:=Data[LowBound].Feature1;
TempAryMin2:=Data[LowBound].Feature2;
TempAryMax2:=Data[LowBound].Feature2;
FOR i[0]:=0 TO UpBound BY 1 DO
IF Data[i[0]].Feature1<TempAryMin1 THEN
TempAryMin1:=Data[i[0]].Feature1;
END_IF;
IF Data[i[0]].Feature1>TempAryMax1 THEN
TempAryMax1:=Data[i[0]].Feature1;
END_IF;
IF Data[i[0]].Feature2<TempAryMin2 THEN
TempAryMin2:=Data[i[0]].Feature2;
END_IF;
IF Data[i[0]].Feature2>TempAryMax2 THEN
TempAryMax2:=Data[i[0]].Feature2;
END_IF;
END_FOR;
FOR i[1]:=LowBound TO UpBound BY 1 DO
IF TempAryMax1-TempAryMin1=0 THEN
NormalizationData[i[1]].Feature1:=1;
ELSE
NormalizationData[i[1]].Feature1:= (Data[i[1]].Feature1-
TempAryMin1)/(TempAryMax1-TempAryMin1);
END_IF;
IF TempAryMax2-TempAryMin2=0 THEN
NormalizationData[i[1]].Feature2:= 1;
ELSE
```

```
NormalizationData[i[1]].Feature2:= (Data[i[1]].Feature2-  
TempAryMin2)/(TempAryMax2-TempAryMin2);  
END_IF;  
NormalizationData[i[1]].Class:=Data[i[1]].Class;  
END_FOR;  
ELSE  
Clear(NormalizationData);  
END_IF;
```



## Ek-D

```
LowBound:=0;
ArySize:=(UpBound-LowBound)+1; // dizi uzunluğu
IF Enable=TRUE THEN
TempAryMin1:=Data[LowBound].Feature1;
TempAryMax1:=Data[LowBound].Feature1;
TempAryMin2:=Data[LowBound].Feature2;
TempAryMax2:=Data[LowBound].Feature2;
FOR i[0]:=0 TO UpBound BY 1 DO
IF Data[i[0]].Feature1<TempAryMin1 THEN
TempAryMin1:=Data[i[0]].Feature1;
END_IF;
IF Data[i[0]].Feature1>TempAryMax1 THEN
TempAryMax1:=Data[i[0]].Feature1;
END_IF;
IF Data[i[0]].Feature2<TempAryMin2 THEN
TempAryMin2:=Data[i[0]].Feature2;
END_IF;
IF Data[i[0]].Feature2>TempAryMax2 THEN
TempAryMax2:=Data[i[0]].Feature2;
END_IF;
END_FOR;
IF TempAryMax1-TempAryMin1=0 THEN
NormalizedTestData[0]:=1;
ELSE
NormalizedTestData[0]:= (TestData[0]-TempAryMin1)/(TempAryMax1-
TempAryMin1);
END_IF;
IF TempAryMax2-TempAryMin2=0 THEN
NormalizedTestData[1]:= 1;
ELSE
```

```
NormalizedTestData[1]:= (TestData[1]-TempAryMin2)/(TempAryMax2-  
TempAryMin2);  
END_IF;  
ELSE  
Clear(NormalizedTestData);  
END_IF;
```





## Ek-E

```
LowBound:=0;
ArySize:=(UpBound-LowBound)+1; // dizi uzunluğu
pi:=3.14159;
IF Enable=TRUE THEN
// mean fit parameter
IF LastTrigger[6]=FALSE THEN
i[6]:=0;
TempSum1:=0;
TempSum2:=0;
LastTrigger[6]:=TRUE;
FOR i[6]:=LowBound TO UpBound BY 1 DO
TempSum1:=Data[i[6]].Feature1 + TempSum1;
TempSum2:=Data[i[6]].Feature2 + TempSum2;
END_FOR;
MeanF1:=TempSum1/ArySize;
MeanF2:=TempSum2/ArySize;
END_IF;
//Standart sapma fit parameter
IF LastTrigger[7]=FALSE THEN
i[7]:=0;
TempSum3:=0;
TempSum4:=0;
Temp2Sum3:=0;
Temp2Sum4:=0;
LastTrigger[7]:=TRUE;
FOR i[7]:=LowBound TO UpBound BY 1 DO
TempSum3:=(Data[i[7]].Feature1-MeanF1)*(Data[i[7]].Feature1-MeanF1);
TempSum4:=TempSum3+TempSum4;
Temp2Sum3:=(Data[i[7]].Feature2-MeanF2)*(Data[i[7]].Feature2-MeanF2);
Temp2Sum4:=Temp2Sum3+Temp2Sum4;
END_FOR;
```

```

TempSum5:=TempSum4/(ArySize-1);
SdF1:=SQRT(TempSum5);
Temp2Sum5:=Temp2Sum4/(ArySize-1);
SdF2:=SQRT(Temp2Sum5);
END_IF;
//compute p(x)
//Anomaly epsilon evaluation
Temp4:=2*pi;
IF SdF1=0 THEN
Temp5:=0;
Ppx1:=1;
ELSE
Temp5:=-(Test[0]-MeanF1)**2/(2*(SdF1)**2);
Ppx1:=(1/(SQRT(Temp4)*SdF1))*EXP(Temp5); //compute
END_IF;
IF SdF2=0 THEN
Temp6:=0;
Ppx2:=1;
ELSE
Temp6:=-(Test[1]-MeanF2)**2/(2*(SdF2)**2);
Ppx2:=(1/(SQRT(Temp4)*SdF2))*EXP(Temp6); //compute
END_IF;
Ppx:=Ppx1*Ppx2;
IF Ppx<Epsilon THEN
DelayAnomaly(In:=TRUE,
PT:=NanoSecToTime(In:=DelayAnomalyFlag*1000000));
IF DelayAnomaly.Q=TRUE THEN
AnomalyFlag:=TRUE;
END_IF;
ELSE
DelayAnomaly(In:=FALSE);
AnomalyFlag:=FALSE;
END_IF;

```

```
ELSE
Clear>LastTrigger);
Clear(i);
Clear(MeanF1);
Clear(MeanF2);
Clear(SdF1);
Clear(SdF2);
Clear(Px1);
Clear(Px2);
Clear(Px);
Clear(Ppx1);
Clear(Ppx2);
Clear(Ppx);
AnomalyFlag:=FALSE;
DelayAnomaly(In:=FALSE);
Clear(SelectionAry1);
Clear(SelectionAry2);
Clear(RandomizedIndex1);
Clear(RandomizedIndex2);
ins_Rand(Execute:=FALSE);
ins_Rand2(Execute:=FALSE);
END_IF;
```

## KİŞİSEL YAYIN VE ESERLER

**Aksöz E.**, ARİFOĞLU, B., Makine Öğrenmesi ile Kestirimci Bakım Ve Anomali Tespiti Analizi, *4. Uluslararası Uygulamalı Bilimler Kongresi*, Artvin, Türkiye, 13 Eylül 2020.

**Aksöz E.**, ARİFOĞLU, B., Makine Öğrenmesi ile Dolandırıcılık Tespiti, *V. Uluslararası Bilimsel ve Mesleki Çalışmalar Kongresi – Mühendislik*, Online, Türkiye, 12-15 Aralık 2020.



## ÖZGEÇMİŞ

Emre Aksöz ilk, orta ve lise öğrenimini İstanbul'da tamamladı. 2010 yılında girdiği Kocaeli Üniversitesi Elektrik Mühendisliği Bölümü'nden 2014 yılında mezun oldu. 2018 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektrik Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimine başladı. 2014 yılında BERTA OTOMASYON firması ile başlayan otomasyon sektörü serüveni başka firmalarla da devam etti. 2018 yılından itibaren OMRON firmasında Saha Uygulama Mühendisi olarak çalışmaktadır.

