



**MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE DETAY ÜRETİM
ALANLARI İÇİN İŞ MERKEZİ KIRILIMINDA ÜRETİM SÜRESİ
TAHMİNLEME**

Tuğçe YÜCE

**YÜKSEK LİSANS TEZİ
YÖNETİM BİLİŞİM SİSTEMLERİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

HAZİRAN 2021

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Tuğçe YÜCE

30/06/2021

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE DETAY ÜRETİM ALANLARI İÇİN İŞ MERKEZİ KIRILIMINDA ÜRETİM SÜRESİ TAHMİNLEME

(Yüksek Lisans Tezi)

Tuğçe YÜCE

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Haziran 2021

ÖZET

Günümüz rekabet koşullarında, kısıtlı kaynakları verimli bir şekilde kullanabilmek, geleceğe dönük yatırımları belirleyebilmek için üretim süresi tahmini yapmak rekabet avantajı sağlamak isteyen firmalar için çok önemlidir. Fakat üretim süresi denildiğinde akla gelen zaman etüdü gibi geleneksel yöntemler ürün ve proses çeşitliliğinin çok fazla olduğu tesislerde büyük bir iş yükü gerektirerek zaman ve dolayısıyla maliyet artışına yol açmaktadır. Bu noktada makine öğrenmesi algoritmaları ile üretim süresi tahminleme zaman ve maliyet açısından büyük avantaj sağlamaktadır. Bu çalışmada bir üretim tesisine ait detay üretim alanlarında, iş merkezi kırılımında üretim sürelerinin makine öğrenmesi algoritmaları ile tahminlenmesi hedeflenmiştir. Tahminleme yöntemleri olarak başarılı sonuçlar elde edilen yapay sinir ağları, destek vektör regresyonu ve gradyan artırma makinesi algoritmalarının kullanılmasına karar verilmiştir. Çalışmanın devamında her bir iş merkezi için en başarılı sonucu veren algoritma ortalama mutlak yüzde hata değerine göre belirlenmiştir. Elde edilen sonuçlara göre, yapay sinir ağları ortalamada %28,22 hata oranı verirken destek vektör regresyonu %15,92, gradyan artırma makinesi ise %14,69 hata oranı vermiştir.

Bilim Kodu : 90619

Anahtar Kelimeler : Üretim, üretim süresi tahminleme, destek vektör regresyonu, gradyan artırma, yapay sinir ağı

Sayfa Adedi : 59

Danışman : Prof. Dr. Mehmet KABAK

PRODUCTION TIME ESTIMATION IN TASK CENTER REFRACTION FOR DETAIL
PRODUCTION AREAS WITH MACHINE LEARNING ALGORITHMS

(M. Sc. Thesis)

Tugce YUCE

GAZİ UNIVERSITY
INFORMATICS INSTITUTE

June 2021

ABSTRACT

In today's competitive conditions, it is very important for companies that want to have a competitive advantage to use limited resources efficiently, to estimate production time to identify future investments. But traditional methods, such as time study that comes to mind when it comes to production time, require a large workload in facilities where the range of products and processes is too high, resulting in a loss of time and therefore cost. At this point, production time estimation with machine learning algorithms provide a great advantage in terms of time and cost. In this study, it is aimed to estimate production times in task center breakdown in detail production areas of a production facility with machine learning algorithms. The use of artificial neural networks, support vector regression and gradient boosting machine algorithms, which have been successful results as prediction methods, has been decided. In the rest of the study, the algorithm that gave the most successful results for each task center was determined based on the average absolute percentage error value from performance metrics. According to the results obtained, the artificial neural networks gave an error rate of 28,22% on average, while the support vector regression gave an error rate of 15,92% and the gradient boosting machine gave an error rate of 14,69%.

Science Code : 90619

Key Words : Artificial neural network, gradient boosting machine, production, production time estimation, support vector regression

Page Number : 59

Supervisor : Prof. Dr. Mehmet KABAK

TEŞEKKÜR

Tez çalışmalarım boyunca kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam Sayın Prof. Dr. Mehmet KABAK'a sonsuz teşekkür ve saygılarımı sunarım.

Hayatımın her aşamasında beni destekleyen benimle gurur duyan canım annem Zehra ÇİFTÇİ'ye, nişanım Reşat TÜRE'ye ve bu süreçte bana destek olan tüm kıymetli dostlarıma çok teşekkür ederim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	3
3. METOTLAR	9
3.1. Yapay Sinir Ağları	9
3.1.1. Çok katmanlı algılayıcılar	10
3.1.2. Doğrusal vektör kuantizasyonu	12
3.1.3. Sayaçlı yayılım	12
3.1.4. Hopfield ağı	13
3.1.5. Boltzmann makinesi	14
3.1.6. Hamming ağı	15
3.1.7. Adaptif rezonans teorisi modelleri	16
3.1.8. Kendini düzenleyen harita ağı	17
3.1.9. Matematiksel model	18
3.2. Destek Vektör Regresyonu.....	20
3.2.1. Doğrusal destek vektör regresyonu	20
3.2.2. Dual problem	22
3.2.3. Doğrusal olmayan destek vektör regresyonu.....	24

Sayfa

3.2.4. Kayıp fonksiyonları	25
3.2.5. Çekirdek fonksiyonları	25
3.3. Gradyan Artırma Makinesi.....	26
3.4. Model Performans Değerlendirme Ölçütleri	28
3.4.1. Determinasyon katsayısı (R^2)	28
3.4.2. Hata kareleri ortalaması (MSE).....	28
3.4.3. Ortalama mutlak hata (MAE)	28
3.4.4. Ortalama mutlak yüzde hata (MAPE)	29
4. UYGULAMA.....	31
4.1. Öznitelik Seçimi	31
4.2. Veri Ön İşleme	33
4.3. Modellerin Uygulanması ve Hiperparametrelerin Belirlenmesi	33
4.3.1. Yapay sinir ağları uygulaması	34
4.3.2. Destek vektör regresyonu uygulaması.....	35
4.3.3. Gradyan artırma modeli uygulaması	35
4.4. Modellerin Karşılaştırılması.....	36
5. SONUÇ VE TARTIŞMA.....	41
KAYNAKLAR	45
EKLER.....	51
EK-1. Python ortamında kurulan yapay sinir ağları modeli kodları	52
EK-2. Python ortamında kurulan destek vektör regresyonu modeli kodları.....	54
EK-3. Python ortamında kurulan gradyan artırma makinesi modeli kodları	56
ÖZGEÇMİŞ	59

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Kullanım amacı ve YSA.....	10
Çizelge 4.1. Üretim süresinin aday parametreler ile korelasyonu	32
Çizelge 4.2. Veri ön işleme öncesi veri seti örneği.....	33
Çizelge 4.3. Veri ön işleme sonrası veri seti örneği	33
Çizelge 4.4. YSA modelleri için optimum nöron sayıları	34
Çizelge 4.5. DVR modelleri için optimum C ve Gamma değerleri.....	35
Çizelge 4.6. Gradyan artırma modelleri için optimum D, K, α ve H değerleri	36
Çizelge 4.7. Model tahmin başarıları.....	37

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Örnek çok katmanlı algılayıcı.....	11
Şekil 3.2. Doğrusal vektör kuantizasyonu mimarisi	12
Şekil 3.3. Sayaçlı yayılım ağının genel mimarisi	13
Şekil 3.4. Hopfield ağı mimarisi	14
Şekil 3.5. Boltzmann makinesi	15
Şekil 3.6. Hamming ağı	16
Şekil 3.7. ART ağının genel mimarisi	16
Şekil 3.8. Kendini düzenleyen harita ağı mimarisi	17
Şekil 3.9. Doğrusal ayrılabilen veriler	21
Şekil 3.10. Doğrusal destek vektör regresyonunda kayıp fonksiyonu.....	22
Şekil 3.11. Doğrusal ayrılamayan veriler	24
Şekil 3.12. Kayıp fonksiyonları	25
Şekil 3.13. Basit gradyan artırma modeli.....	26
Şekil 4.1. Akış Şeması	31
Şekil 4.2. Gerdirmeli şekil verme presi 1 gerçek süreler ve tahmini süreler	37
Şekil 4.3. Gerdirmeli şekil verme presi 2 gerçek süreler ve tahmini süreler	38
Şekil 4.4. 5 eksenli işleme tezgâhı 1 gerçek süreler ve tahmini süreler.....	38
Şekil 4.5. 5 eksenli işleme tezgâhı 2 gerçek süreler ve tahmini süreler.....	38
Şekil 4.6. Hassas frezeleme tezgâhı gerçek süreler ve tahmini süreler	39
Şekil 4.7. 5 eksenli frezeleme ve Su Jeti gerçek süreler ve tahmini süreler	39
Şekil 4.8. Ultrasonik kontrol tezgâhı 1 gerçek süreler ve tahmini süreler	40
Şekil 4.9. Ultrasonik kontrol tezgâhı 2 gerçek süreler ve tahmini süreler	40

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ART	Adaptif rezonans teorisi
CNC	Bilgisayarlı sayısal kontrol (computer numerical control)
ÇKA	Çok katmanlı algılayıcılar
DVK	Doğrusal vektör kuantizasyonu
DVM	Destek vektör makinesi
DVR	Destek vektör regresyonu
FIFO	İlk giren ilk çıkar (first in first out)
GBM	Gradyan artırma makinesi (gradient boosting machine)
KDH	Kısa dönemli hafıza
MAE	Ortalama mutlak hata (mean absolute error)
MAPE	Ortalama mutlak yüzde hata (mean absolute percentage error)
MSE	Hata kareleri ortalaması (mean squared error)
RBF	Radyan tabanlı fonksiyon (radial basis function)
SNBC	Seçici naive Bayes sınıflandırıcısı (simple naive Bayesian classifier)
SQL	Yapılandırılmış sorgu dili (structured query language)
UDH	Uzun dönemli hafıza
YSA	Yapay sinir ağları

1. GİRİŞ

Üretim genel olarak bakıldığında; insan gereksinimlerinin karşılanabilmesi için yapılan her türlü ürün ya da hizmeti ortaya çıkarma eylemi olarak ifade edilebilir. Üretim sistemi ise işgücü, malzeme, bilgi, enerji, sermaye gibi girdilerin belirli bir dönüştürme sürecinden geçirilerek mal veya hizmetin üretildiği bir sistemdir. Bir üretim sisteminin elemanları; girdi, proses, çıktı, geri besleme ve çevresel faktörlerdir. Bu noktada üretim süresi doğrudan proses ile ilişkili olurken aynı zamanda dolaylı olarak diğer sistem elemanlarını da etkileyen önemli bir parametre olarak karşımıza çıkmaktadır.

Üretim süresi veya temin süresi, bir ürününün üretilmesi için gereken hammadde siparişinden ürünün paketlenmesine kadar geçen süre olarak ifade edilebilir. Günümüz rekabet koşullarında, doğru bir şekilde üretim süresinin belirlenmesi hem kaynakların etkin kullanılması hem de müşteriye doğru teslim süresi verebilmek açısından üretim ve yönetim kademeleri için önem taşıyan bir konudur. Ücretlendirme, maliyet kontrolü, üretim planlarının hazırlanması, kısa ve uzun vadeli tahminlerin yapılması konuları başta olmak üzere işletme içinde yürütülen tüm faaliyetlerde üretim süresi dikkate alınmadan yapılacak işlerin tutarlı ve yararlı olması mümkün değildir. Teslim süresini kısa vermek müşterileri kazanmayı sağlar ancak genellikle gecikmeyi beraberinde getirir. Teslim süresinin uzun tutulduğu durumlarda ise teslimatın tamamlanması kolay olur ancak bu sefer de müşteri daha kısa teslim süresi veren firmaya kayabilir. Yanlış tahmin edilen üretim süresi ve bunun üzerinden yapılan iş planlaması, firmaları zamanından önce işi bitirmelerine ve makinelerin boşa kalmalarına ya da zamanında yetiştirmemeleri sonuçlarını ortaya çıkarabilir. Firma her iki durumda da maddi açıdan zarara uğrayabilir [1].

Bir üretim sisteminin başarılı olup olmadığı siparişin belirtilen zamanda karşılanabilme oranı ile belli olur. Müşteriye iletilen teslim tarihini zamanında karşılamak için, ardışık birçok karar içeren üretim planlama yapmak gerekir. Bu nedenle üretim süresi temel parametre olmaktadır. Üretim kaynaklarının etkin kullanılması için gerekli olan çizelgeleme kısa ve orta vadede, kapasite analizleri ise uzun vadede üretim durumunun matematiksel ve somut olarak ortaya konulmasında kullanılan araçlardır. Bu açıdan üretim süresi; kapasite analizleri için temel parametre olmaktadır. Üretim süresinin doğru olmasına önem veren ve düzenli olarak güncel tutan şirketler, müşteri termini konularında daha az gecikmeler ile

rakiplerinin önüne geçmekte ve ekipman/tezgah ihtiyaçları daha düzenli takip edebilmektedirler.

Üretim süresi belirlemede doğrudan hesaplama yöntemleri olarak bilinen iş örnekleme ve zaman etüdü metotları daha doğru sonuçlar vermesine rağmen ürün çeşitliliğinin ve iş merkezlerinin çok fazla olduğu tesislerde büyük bir iş yükü gerektirerek maliyet ve zaman kaybına yol açmaktadır. Dolaysız ölçüm yöntemi olarak bilinen saptanmış hareket zamanları metodu ise doğrudan ölçüm yöntemlerine göre daha az analiz gerektirir fakat üretim çeşitliliğinin çok fazla olduğu üretim tesislerinde büyük iş yükü gerektirdiğinden maliyet ve zaman açısından beklenen kolaylığı sağlayamamaktadır. Bu noktada makine öğrenmesi algoritmaları ile üretim süresi tahminleme maliyet ve zaman açısından büyük avantaj sağlamaktadır.

Bu tezde, savunma ve havacılık sektöründe faaliyet gösteren bir üretim tesisine ait detay üretim alanlarında üretim süresi tahminlemek için makine öğrenmesi algoritmalarına başvurulmuştur. Her bir üretim tezgâhının iş merkezi olarak adlandırıldığı tesiste, iş merkezi detayında süre tahminlenmesi hedeflenmiş olup, makine öğrenmesi algoritmalarından yapay sinir ağları, destek vektör regresyonu ve gradyan artırma yöntemleri kullanılmıştır. Kullanılan yöntemler sonucu elde edilen hata oranları karşılaştırılıp analiz edilmiştir.

Çalışmanın ikinci bölümünde literatürde makine öğrenmesi algoritmaları kullanılarak yapılan tahminleme çalışmalarına yer verilmiş ve yapılan çalışmanın diğer çalışmalardan nasıl farklılaştığı açıklanmıştır. Süre tahminleme üzerine yapılan çalışmalar detaylı olarak incelenmiş olup, diğer tahminleme çalışmaları kısaca özetlenmiştir. Üçüncü bölüm, çalışmada üretim süresi tahminleme için kullanılan yapay sinir ağları, destek vektör regresyonu ve gradyan artırma yöntemlerini detaylı olarak açıklanmıştır. Uygulanan yöntemlerin karşılaştırılması için kullanılan model performans değerlendirme ölçütleri de detaylandırılmıştır. Dördüncü bölümde python programı kullanılarak yapılan uygulama adımları ve sonuçlar paylaşılmıştır. Son bölüm olan sonuç ve öneriler bölümünde ise uygulama bölümünde elde edilen sonuçlar karşılaştırılmıştır. Gelecek çalışmalarda yapılabileceklerle ilgili önerilerde bulunulmuştur.

2. LİTERATÜR TARAMASI

2000 yılı öncesinde üretim süresi tahminleme çalışmalarında istatistiksel analizler ön plana çıkmaktadır. İstatistiksel analizler sonucunda, istatistiksel modellerin sürecin özelliklerine göre güncellenmesi gerektiği [2] ve doğrusal modellerle sınırlı kaldığı [3] gözlenmiştir.

Makine öğrenmesinin ilk adımları 1950’lerde atılmıştır fakat “Yapay zeka kış uykusu” olarak da adlandırılan dönemde başta donanımsal kısıtlamalar ve diğer problemler sebebi ile bu alandaki çalışmalar durma noktasına gelmiştir. Duraklama dönemi sonrası 2000’lerin başında tekrar gözde olmasıyla beraber makine öğrenmesi alanında çalışmalar yoğunlaşmıştır [4]. Aynı şekilde üretim süresi tahminleme çalışmaları da makine öğrenmesi metotları üzerine yoğunlaşmıştır.

Literatür araştırmasının bundan sonraki kısmında makine öğrenimi ile süre tahminlemesi üzerine yapılan çalışmalar detaylı olarak incelenmiştir.

Janakiram ve arkadaşları, 2003 yılında çevrim süresi tahminleme için veri madenciliği yaklaşımı üzerinde çalışmışlardır. Veri madenciliğinin istatistiki verileri hızla yeniden analiz etme yeteneği ve ihtiyaç durumunda güncellenebilir olması açısından avantajlı bir yaklaşım olduğunu belirtmişlerdir [5]. Yine 2006 yılında yaptıkları çalışmada en yakın komşu, kümeleme ve regresyon ağacı metotları üzerinde veri madenciliği uygulamaları yaparak sonuçların performanslarını değerlendirmiş ve çalıştıkları veriler üzerinde regresyon ağacının daha başarılı tahminler yaptığını gözlemlemişlerdir [6]. Öztürk ve arkadaşları ise sipariş üzerine üretim yapan tesislerde teslim süresi tahmini için veri madenciliğinin kullanımını araştırmışlardır. Regresyon ağacı yaklaşımı, spesifik veri madenciliği yöntemi olarak seçilmiştir. Eğitim ve test verileri, bir atölye simülasyon modelinin varyasyonlarından oluşturulmuştur. Seçilen özniteliklerle veri madenciliği, doğrusal regresyon ve literatürdeki diğer üç ön zaman tahmin yöntemi ile karşılaştırılmıştır. Ampirik sonuçlar, veri madenciliği yaklaşımının öznitelik seçim şemasıyla birlikte bu yöntemlerden daha iyi performans gösterdiğini göstermiştir [7].

2007 yılında Alenezi ve arkadaşları çalışmalarında çok kaynaklı, çok ürünli sistemlerde gerçek zamanlı akış süresi tahmini için bir destek vektör regresyon modeli geliştirmişlerdir.

Kernel ve kayıp fonksiyonlarının çeşitli kombinasyonları incelenmiştir. Farklı karmaşıklığa sahip üç farklı çok kaynaklı sistem için destek vektörü regresyon modelinin tahmin hatası,

klasik zaman serisi modellerininkiyle (üstel yumuşatma ve hareketli ortalama) ve ileri beslemeli bir yapay sinir ağıyla karşılaştırılmıştır. Sonuçlar, destek vektörü regresyon modelinin daha düşük akış süresi tahmin hatasına sahip olduğunu ve daha sağlam olduğunu göstermiştir [8]. Destek vektör algoritması ile üretim süresi tahminlemesi yapılan bir diğer çalışma ise Lim ve arkadaşları tarafından 2019 yılında gerçekleştirilmiştir. Çalışmalarında, bir ürünün üretim süresini, iş emri ve üretim verilerini kullanarak destek vektör makinesi ile tahminlemeyi amaçlamışlardır. Algoritmanın performansı, yapay sinir ağı ve rassal orman algoritmaları ile karşılaştırılmıştır ve destek vektör makinesi algoritmasının yeterli sonucu verdiği görülmüştür. Model, %84,62 oranında doğru üretim süresi tahminlerken, fazla mesai tahminlemesini %65'in üzerinde doğrulukla tamamladığı görülmüştür [9].

Meidan ve arkadaşları, 2009 yılında bir yarı iletken üretim tesisinde, döngü süresinin değerini tahmin etmek ve temel faktörlerini tanımlamak için veriye dayalı bir metodoloji üzerine çalışma yapmışlardır. Veri kümesi simüle edilmiş bir fabrikadan elde edilmiştir. Elde edilen veri kümesine Bayes sınıflandırıcısı uygulamışlardır. Çalışma sonucunda %72,6 oranında tahmin doğruluğu elde edilmiştir. C5.0 karar ağacı ve sinir ağları algoritmaları ile de benzer sonuçları elde etmişlerdir [10]. 2011 yılına geldiklerinde yapmış oldukları çalışmaya yeni bir yaklaşım getirmişlerdir. Yeni yaklaşımlarında, ilk olarak koşullu karşılıklı bilgi maksimizasyonunu kullanarak en etkili faktörleri belirlemişlerdir. Sonrasında üretim süresi tahmini için minimal, en ayırt edici anahtar faktör setinin daha detaylı seçimi için seçici Naive Bayes sınıflandırıcısını (SNBC) uygulamışlardır. Simüle edilmiş bir üretimi temsil eden veri setine uygulanan Snbc tabanlı yaklaşım, üretim süresi tahmininin doğruluğunu yaklaşık %40 oranında artırırken, faktörlerin listesini 182'den 20'ye daraltmıştır [11].

2004 yılında Lio ve arkadaşları, otomatik malzeme taşıma sistemindeki hem öncelikli hem de normal partiler için teslimat sürelerini tahmin etmek amacıyla bir sinir ağı yaklaşımı üzerine çalışmışlardır. Her bir döngü için bir ağ modeli geliştirilmiştir. Çalışma sonucunda sinir ağı yaklaşımının ortalama teslimat sürelerinin tahmini için sağlam ve etkili olduğunu görülmüştür fakat sonuçlarda elde edilen güven aralığının daha az olması için daha fazla kural tanımlanması gerektiği belirtilmiştir [12]. Bilekdemir ise 2010 yılında veri madenciliği yaklaşım metodlarından karar ağacı C4.5 ile sınıflandırma tekniği kullanarak üretim süresi tahminlemeye çalışmıştır. Tahminleme performansının artırılması için çalışan işçiler ile ilgili veri toplanması gerektiğini vurgulamıştır. Çalışma sonucunda karar ağaçları ile sürekli tipteki değişkenlerin tahmini yapılamamasından dolayı çalışmanın yapay sinir ağları ile

geliştirilebileceğini belirtmiştir [13]. 2011 yılında Altın, otomotiv sektöründe her türlü ihtiyaca cevap verecek şekilde sektöre parça üreten döküm ve hammadde işleme firmasında, standart zaman tahminlemesi için çok katmanlı yapay sinir ağları kullanmıştır. CNC tezgâhları ele alan çalışmada 3 model üzerinden çalışmasını yapmıştır; ileri yayımlı öğrenme, dinamik geri yayımlı öğrenme ve moment geri yayımlı öğrenme. Çalışma sonucunda dinamik geri yayımlı öğrenme ve moment geri yayımlı öğrenme modelleri ile sonuç elde etmiştir [2]. Kurnaz ise 2019 yılında bir firmada üretimi yapılan kablo takımlarının ihale için öngörülen üretim sürelerinin tahmin edilebilmesi amacıyla makine öğrenmesi algoritmalarını kullanarak yapay sinir ağı modeli oluşturmuştur. Akabinde modelin tahmin gücünü ölçebilmek adına sonuçlar, çoklu regresyon modeli ile elde edilen sonuçlar ile karşılaştırılmıştır. Çalışma sonucunda yapay sinir ağları modelinin daha iyi sonuç getirdiğini görülmüştür [14].

Zhang ve arkadaşları 2015 yılında yaptıkları çalışmalarında, tahmin doğruluğunu ve model yorumlanabilirliğini artırmak amacıyla otoyol seyahat süresini analiz etmek ve modellemek için bir gradyan artırma makinesi (GBM) önermiştir. Önerilen yöntem, daha sonra, başka bir popüler topluluk yöntemi ve bir tezgâh işareti modeli ile karşılaştırılmıştır. Çalışma sonuçları, GBM modelinin otoyol seyahat süresi tahmininde önemli avantajlara sahip olduğunu göstermiştir [15]. Süre tahminleme amacıyla gradyan artırıcı regresyon ağacı yöntemini kullanan bir diğer çalışma ise 2020 yılında Kuo ve arkadaşları tarafından yapılmıştır. Çalışmalarında hastane acil servislerinde bekleme sürelerini tahminleme amacıyla gradyan artırma, destek vektör makineleri, çoklu doğrusal regresyon ve rassal orman algoritmalarını kullanmışlardır. Doğrusal regresyon modelini baz alarak diğer model sonuçları ile karşılaştırmışlardır. Sonuç olarak kullanılan dört modelde de hata kareleri ortalamasının doğrusal regresyon modeline göre %17-%22 arasında daha iyi sonuç verdiği görülmüştür [16].

Gyulai ve arkadaşları, 2018 yılında kapalı döngü bir üretim kontrolü uygulayarak kararlar almak için, üretim sürelerini proaktif olarak tahmin etmek amacıyla makine öğrenimi tekniklerini uygulayan yeni bir üretim veri analitiği aracı sunmuşlardır. Önerilen yöntem, regresyon tekniklerini uygular ve buna dayalı olarak, sevk kararlarında kullanılacak iş önceliklendirmesini destekler. Çalışmada, metotların performansını karşılaştırmak amaçlı ilk olarak, işler bir FIFO kuralına göre önceliklendirilmiştir, bu nedenle daha sonra gelen işler daha düşük öncelik kazanmıştır. Sonrasında daha gerçekçi bir temel olarak, Teslimat süreleri Little yasası ile tahmin edilmiştir. Little yasasına göre; bir sistemde kuyrukta olan

ortalama parça sayısı ile kuyrukta birim başına beklenen süre arasında bir ilişki vardır. Üçüncü durumda, önerilen, Makine öğrenimine dayalı teslim süresi önceliklendirmesi uygulanmıştır. Sonuç olarak FIFO kuralı en yüksek hatalı tahminleri verirken, makine öğrenimine dayanan tahminler en iyi sonuçları vermiştir. Makine öğreniminin dinamik ve ürün çeşitliliğinin fazla olduğu üretim alanlarında daha fazla parametreyi dikkate alabildiği için tercih edilmesi gerektiğini ve gelecek çalışmalarda makine öğrenmesi üzerine deneylerin artırılması gerektiğini vurgulamışlardır [17].

Makine öğrenmesi algoritmaları yalnızca süre tahminleme problemlerinde değil farklı sektörlerde ve farklı tahminleme konularında da ön plana çıkmaktadır. Bunlara örnek verecek olursak; Var ve arkadaşları, yapay sinir ağları yöntemini kullanarak İstanbul'a ait bir bölgenin elektrik puant yükünü tahmin etmişlerdir. Çalışma sonucunda %10 hata oranının altına inilmiş ve arzu edilen amaca ulaşılmıştır [18]. Kartal, kalp ameliyatı sırasında ya da kalp ameliyatı geçirdikten kısa bir süre sonra hastaya ait hayati riski sınıflandırmaya dayalı makine öğrenmesi teknikleri ile belirleme üzerine çalışmıştır. Sınıflandırmaya dayalı makine öğrenmesi algoritmalarından Naive Bayes sınıflandırıcı, k-en yakın komşu algoritması, lojistik regresyon analizi, ID3 ve C4.5 karar ağacı algoritmaları kullanılmış olup en yüksek performans değerinin makine öğrenmesi tekniklerinden Naive Bayes algoritmasının sahip olduğu sonucuna varılmıştır [19]. Sarı, yapay sinir ağları yöntemi kullanarak traktör firmalarına satılan motor yataklarının yıllık satışlarını tahminlemeyi ve satışları etkileyen faktörleri belirlemeyi amaçlamıştır. Çalışma sonuçlarını üssel düzeltme, hareketli ortalama ve çoklu regresyon yöntemlerinden elde edilen sonuçlar ile karşılaştırmış ve yapay sinir ağlarının en iyi sonucu verdiğini vurgulamıştır [20]. İmamoğlu ve arkadaşları yapmış oldukları çalışmada destek vektör regresyon ve ikiz destek vektör regresyon metotları ile tedarikçilerin kredi endeksini tahmin etmeye çalışmışlardır. Çalışma sonucunda ikiz destek vektör regresyon modelinin daha iyi bir genel performansa sahip olduğu ve daha düşük tahmin hatası verdiği görülmüştür [21]. Yang ve arkadaşları, kısa vadeli trafik yoğunluğu tahmini yapmak için gradyan artırma, destek vektör makinesi ve yapay sinir ağları yöntemlerini kullanmışlardır. Çalışma sonucunda gradyan artırma yönteminin daha düşük hata oranı verdiği görülmüştür [22]. Bardak ve arkadaşları, yapay sinir ağları ve derin öğrenme algoritmalarını farklı nanokompozitlerin çekme testleri sırasında oluşan deformasyonunu tahmin etmek için kullanmıştır. Çalışma derin öğrenme modelinin daha yüksek doğruluk oranına sahip olduğunu göstermiştir [23]. Kurt ve arkadaşları, yapay sinir ağları kullanarak Türkiye'de bulunan kâğıt-karton sanayi firmalarının ihracat rakamlarını

tahmin etmeyi amaçlamışlardır. Çalışma sonuçlarının başka tahmin modelleri ile de karşılaştırılması gerektiği ileri çalışmalar için önerilmiştir [24]. Yiğit ve arkadaşları, Türkiye’de faaliyet gösteren üç mobil operatöre ait 2017 yılı hız testi verileri ile kullanıcıların deneyimleyebileceği servis tiplerini tahminlemek üzere makine öğrenmesi yöntemlerinden faydalanarak sınıflandırma modelleri oluşturmuştur. Sınıflandırma algoritmalarının karşılaştırılması aşamasında, tüm mobil servis tipleri lojistik regresyon, doğrusal ayırmacılık analizi, k- en yakın komşu, karar ağacı, Naive Bayes, destek vektör makineleri, rassal orman ve gradyan artırma algoritmaları ile karşılaştırılmıştır. Sonuç olarak; 4K video izleme ve çevrimiçi video oyun oynama mobil servislerinde gradyan artırma algoritması, FullHD video izleme ve mesajlaşma mobil servislerinde lojistik regresyon algoritmasının en iyi sonuçları verdiği tespit edilmiştir [25]. Touzani ve arkadaşları, çalışmalarında gradyan artırma makinesine dayanan bir enerji tüketimi tahminleme modeli önermiştir. Sonuçlar, gradyan artırma algoritmasının doğrusal regresyon ve rassal orman algoritmalarına kıyasla daha başarılı sonuçlar verdiği görülmüştür [26]. Zhou ve arkadaşları, madencilik ve jeoteknik mühendislik projelerinde en önemli görevlerden biri olan eğim stabilitesinin tahmin edilmesi için gradyan artırma makinesi (GBM) yöntemini kullanan yeni bir tahmin yöntemi sunmuştur. Önerilen modelin performansını ölçmek için eğri altındaki alan (AUC), sınıflandırma doğruluk oranı ve Cohen'in Kappa katsayısı kullanılmıştır. Gradyan artırma makinesi ile elde edilen doğruluk oranı karşılaştırılan AUC, sınıflandırma doğruluk oranı ve Cohen'in kappa değerlerine ulaştığı için yüksek güvenilirliğe sahip olduğunu göstermiştir [27].

Bu çalışmada ise, tahminleme üzerine yapılan çalışmalarda yüksek performans gösteren destek vektör regresyonu, yapay sinir ağları ve gradyan artırma algoritmaları üretim süresi tahminlemek üzere kullanılmıştır. Bu çalışmanın özgün yanı üretim süresi veri setinin oldukça büyük olması ve veri seti oluşturulurken parçaların toplam süresi değil, proses bazlı sürelerinin tahmin edilmesi üzerine modellerin kurulmasıdır. Proses bazlı üretim süresi tahminlenmesindeki asıl amaç, farklı proseslerin gerekliliklerini kendi içinde ayrıştırabilmektir. Teslim süresi bazlı modellerde proses farklılıklarının oluşturabileceği kısıtlar göz ardı ediliyor olacağından, proses bazlı üretim süresi tahminlemede daha gerçekçi ve net tahmin modellerin kurulması öngörülmektedir.



3. METOTLAR

Üretim süresi tahminleme modeli oluşturulurken çalışmanın yapıldığı üretim tesisinde yapılmakta olan detay üretim alanları için 4 temel proses olarak kompozit imalat, sac metal imalat, kalite ve talaşlı imalat alanları seçilmiştir. İş merkezleri belirlenirken, uzman görüşlerine ve üretim sürecinde iş yükü dağılımının yüksek olmasına dikkat edilmiştir.

Seçilen alanlarda uygulanacak metotlar ise yapılan literatür taraması sonucunda tahminleme problemlerinde yüksek başarı sağlayan yapay sinir ağları, destek vektör regresyonu ve gradyan artırma modelleri olarak belirlenmiştir.

3.1. Yapay Sinir Ağları

Yapay sinir ağları (YSA), beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini insan beyninin çalışma mekanizmasını taklit ederek gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA, özellikle arka planda olan veri ilişkisi bilinmediği durumlarda modelleme için güçlü bir yaklaşımdır. Bu yapının temel unsuru bilgi işlem sisteminin yapısıdır. İnsan beyninde olduğu gibi, belirli sorunları çözmek için bir arada çalışan çok sayıda birbirine bağlı işlem elementinden oluşur. YSA tıpkı insanlar gibi öğrenir. Bir YSA öğrenme süreci boyunca örüntü tanıma veya veri sınıflandırması gibi belirli bir uygulama için yapılandırılmıştır. Biyolojik sistemlerde öğrenme işlemi nöronlar arasında var olan sinaptik bağlantılar üzerinde ayarlamalar ile olur. Bu durumda YSA için de geçerlidir [28].

Genellikle sinir ağları olarak adlandırılan YSA, beynin geleneksel dijital bilgisayarlardan tamamen farklı bir şekilde hesap yaptığı kabulü üzerine motive olmuştur [29]. YSA, beyin gibi biyolojik sinir sistemlerinin süreç bilgisinden esinlenen bir bilgi işlem paradigmasıdır. Bu paradigmanın ana unsuru, bilgi işlem sisteminin yeni yapısıdır [30]. Bir YSA, beyindeki biyolojik nöronların doğal ağlarında gözlenen süreçten esinlenen hesaplamalı bir yapıdır. Bağlantılı olan basit hesaplamalı birimlerden, nöronlardan oluşur. YSA, geniş uygulama alanları ve karmaşık sorunları tedavi etme kolaylıkları nedeniyle, büyük ilgi odağı haline gelmiştir.

YSA, insanlar tarafından gerçekleştirilmiş örnekleri (gerçek beyin fonksiyonlarının ürünü olan örnekleri) kullanarak olayları öğrenebilen, çevreden gelen olaylara karşı nasıl tepkiler

üretileceğini belirleyebilen bilgisayar sistemleridir. İnsan beyninin fonksiyonel özelliklerine benzer şekilde öğrenme, ilişkilendirme, sınıflandırma, genelleme, özellik belirleme ve optimizasyon gibi konularda başarılı bir şekilde uygulanmaktadır. Örneklerden elde ettiği bilgiler ile kendi deneyimlerini oluşturur ve daha sonra, benzer konularda benzer kararları verir [28].

YSA'lar yoğun şekilde birbirine bağlı adaptif işlem birimlerinden oluşan paralel hesaplama modelleridir. Bu ağlar doğrusal olmayan statik veya dinamik sistemlerin ince taneli paralel uygulamalarıdır. Bu ağların önemli bir özelliği, “örneklemeli öğrenmenin” problem çözmedeki “programlamanın” yerini aldığı uyarlanabilir doğalarıdır. Bu, hesaplama modellerinin, çözülecek problemin çok az veya eksik bir şekilde anlaşıldığı ancak eğitim verilerinin hazır olduğu yerlerde uygulama alanlarında çok çekici olmasını sağlar. YSA'lar, regresyon modelinin ve diğer ilgili istatistiksel tekniklerin geleneksel olarak kullanıldığı, sınıflandırma ve tahmin alanında tanınmaktadır [31].

Genel olarak YSA ile tahmin, veri filtreleme, veri kavramlaştırma, veri ilişkilendirme ve sınıflandırma fonksiyonlarını gerçekleştirmek mümkündür. Herhangi bir problemin çözümü için uygulanacak olan YSA modeli öncelikle problemin türüne bağlıdır. Amaçlarına göre yaygın olarak kullanılan YSA modelleri bulunmaktadır. Bu yaygın kullanılan modeller Çizelge 3.1’de gösterilmektedir [32].

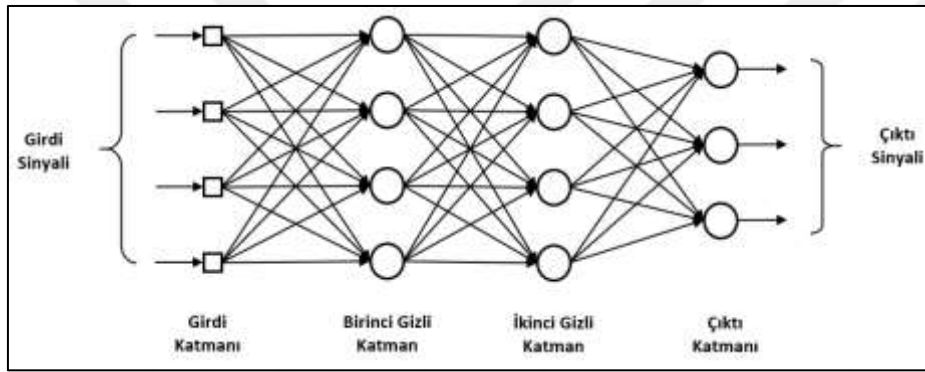
Çizelge 3.1. Kullanım amacı ve YSA

Kullanım Amacı	Ağlar
Tahmin	Çok Katmanlı Algılayıcılar (Multi Layered Perceptrons)
Sınıflandırma	Doğrusal Vektör Kuantizasyonu (Learning Vector Quantization)
	Sayaçlı Yayılım (Counter-Propagation)
	Olasılık Sinir Ağı (Probabilistic Neural Network)
Veri İlişkilendirme	Hopfield Ağı (Hopfield)
	Boltzman Makinesi (Boltzman Machine)
	Hamming Ağı (Hamming Network)
	Çift Yönlü Çağrışımlı Bellek (Bidirectional Associative Memory)
	Uzaysal- Mekansal Geçici Örüntü Tanıma (Spatio-Temporal Pattern Recognition)
Veri Kavramlaştırma	Adaptif Rezonans Teorisi (Adaptive Resonance Theory)
	Kendisini Düzenleyen Harita Ağı (Self Organizing Map)
Veri Filtreleme	Devridaim YSA (Recirculation)

3.1.1. Çok katmanlı algılayıcılar

Çok katmanlı algılayıcılar (ÇKA), Rumelhart ve arkadaşları tarafından geliştirilmiştir. Geri

yayılım algoritmasının kullanıldığı çok katmanlı algılayıcıların öğrenme kuralı, en küçük kareler yöntemine dayalı delta öğrenme kuralının genelleştirilmiş halidir. Bu nedenle öğrenme kuralı “Genelleştirilmiş Delta Kuralı” olarak da bilinmektedir. Delta kuralına göre bir nöronun gerçek çıktı değeri ile istenilen çıktı değeri arasındaki farkı azaltmak için ağırlıkların sürekli olarak ayarlanması ve geliştirilmesi gerekmektedir [33]. Bu kurala göre öğrenme sırasındaki bağlantı ağırlıkları sürekli olarak değiştirilerek, ağırlıklar için optimum değerler bulunur. “Öğretmenli Öğrenme” yöntemi bu ağ yapısında kullanılır. Bu yöntemde hedef çıktı değerleriyle ilişkilendirilmiş olan her bir vektör, öğrenmesi için ağa sunulur. Ağa sunulan ağırlıklar, belirtilen öğrenme kuralına göre düzeltilir [34]. Şekil 3.1’de örnek birçok katmanlı algılayıcı gösterilmiştir [35].



Şekil 3.1. Örnek çok katmanlı algılayıcı

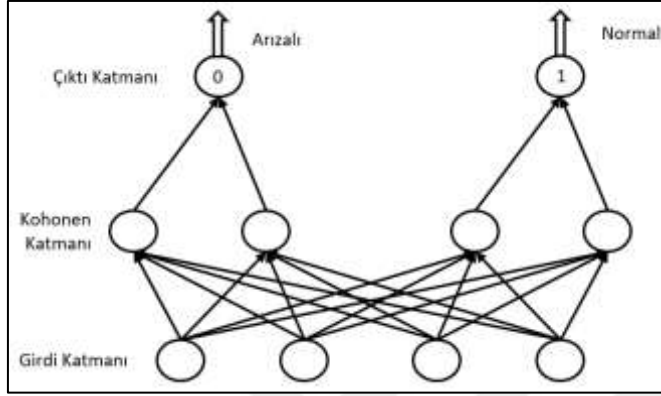
Girdi Katmanı: Dış dünyadan gelen girdileri alarak ara katmana gönderir. Bu katmanda bilgi işleme olmaz. Gelen her bilgi geldiği gibi bir sonraki katmana gider. Birden fazla girdi gelebilir. Her proses elemanının sadece bir tane girdisi ve bir tane çıktısı vardır.

Ara Katmanlar: Girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Bir ÇKA ağında birden fazla ara katman ve her katmanda birden fazla proses elemanı olabilir.

Çıktı Katmanı: Ara katmandan gelen bilgileri işleyerek ağı girdi katmanından verilen girdilere karşılık ağı ürettiği çıktıları belirleyerek dış dünyaya gönderir. Bir çıktı katmanında birden fazla proses elemanı olabilir [28].

3.1.2. Doğrusal vektör kuantizasyonu

Doğrusal vektör kuantizasyonu (DVK) ağları destekli öğrenme stratejisini kullanırlar. Eğitim sırasında ağı sadece öğrenilmesi istenen girdiler verilmekte ve ağın çıktısı kendisi üretmesi istenmektedir. Ağ çıktısını ürettikten sonra, ağı sunulan girdi vektörüne karşılık gelen ağın ürettiği ağı doğru veya yanlış olup olmadığı söylenir [36].



Şekil 3.2. Doğrusal vektör kuantizasyonu mimarisi [28]

Girdi Katmanı: Bu katmanda bilgi işleme olmaz. Dış dünyadan alınan örneklerin ağı gösterilmesi bu katmanda sağlanmaktadır. Gelen bilgiler girdi vektörünü oluşturur.

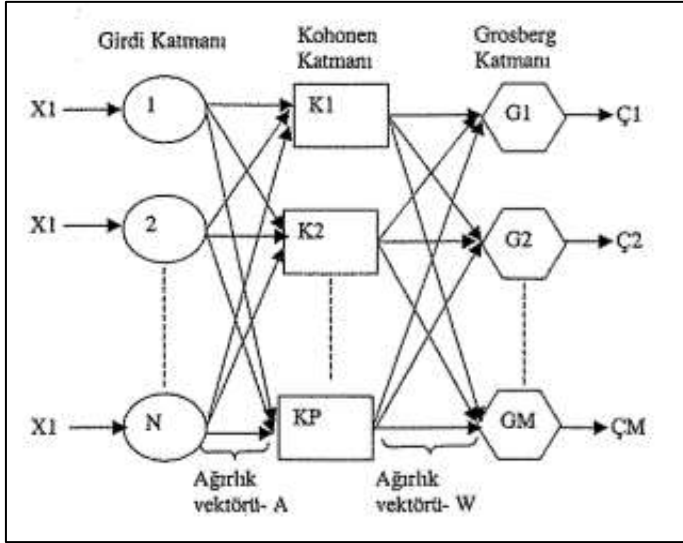
Kohonen Katmanı: Bu katmanda girdi setine en yakın olan ağırlık vektörü belirlenmektedir. Bu katmandaki her eleman bir referans vektörünü göstermektedir. Girdi vektörü, girdi katmanı ile kohonen katmanı arasındaki ağırlıkların oluşturduğu referans vektörüne haritalanmaktadır.

Çıktı Katmanı: Bu katmanda ise girdinin ait olduğu sınıf belirlenir.

3.1.3. Sayaçlı yayılım

İki veya daha fazla temel ağ tasarımının özelliklerini birleştiren bir karma ağ örneğidir. Kohonen ve Grossberg ağlarının olumlu yanlarını bir araya getirmiştir.

Gizli katman, denetlenmemiş öğrenmeye sahip bir Kohonen ağıdır ve çıktı katmanı, gizli katmana tamamen bağlı bir Grossberg (dış mekân) katmanıdır. Çıkış katmanı Widrow-Hoff kuralı tarafından eğitilmiştir. Basit bir kategori numarası yerine bir desenin çıktısını verir. İki yönlü bir ilişkisel bellek olarak da görüntülenebilir [37].



Şekil 3.3. Sayaçlı yayılım ağına genel mimarisi [38]

Girdi Katmanı: ÇKA ağlarında olduğu gibi sadece bilginin transfer edilmesini sağlar. Herhangi bir bilgi işleme olmaz.

Kohonen Katmanı: Bu katman girdi vektörünün sınıflandırılmasını sağlamaktadır.

Grosberg Katmanı: Bu katman ağına çıkışını belirler [28].

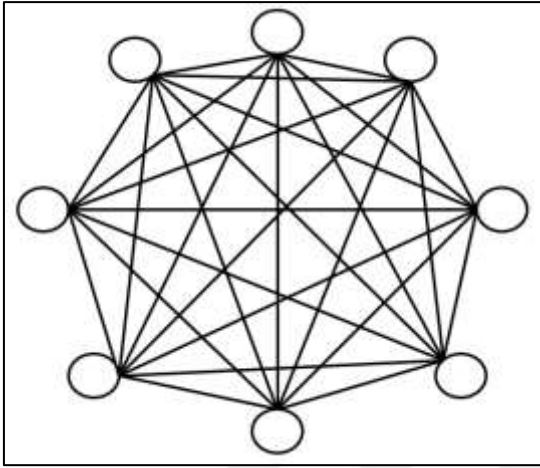
Eğitim iki aşamalı bir prosedürdür. İlk olarak, Kohonen katmanı giriş kalıpları üzerinde eğitilmiştir. Bu adımda çıktı katmanında değişiklik yapılmaz. Orta katman, tüm giriş modellerini doğru şekilde sınıflandırmak için eğitildiğinde, giriş ve orta katmanlar arasındaki ağırlıklar sabit tutulur ve çıktı katmanı, orta ve çıkış katmanları arasındaki ağırlıkları ayarlayarak doğru çıktı desenleri üretmek üzere eğitilir [38].

3.1.4. Hopfield ağı

Hopfield ağı 1982’de John Hopfield tarafından geliştirilmiştir. İkili değerler kullanılan Hopfield ağı genel olarak diğer ağ örneklerini kapsamaktadır. Bu özgün ağ ile her bir eleman işlenerek bir ikili sayı biçimi oluşturulur. Bu elemanlar girişlerin toplamı ve çıkış sayıları bir veya sıfır olarak hesaplanarak yüklenir. Hopfield ağ topolojisi diğer ağ yapılarından farklıdır. Hopfield ağında, giriş süzme katmanı, Hopfield katmanı ve çıkış katmanı olmak üzere 3 katman olmasına rağmen farklı katmanlar yoktur. Bu ağda her düğüm diğer tüm düğümlerle birbirine bağlıdır. Ayrıca, bağlantılar iki yönlüdür (bilgi her iki yönde akar) ve simetriktir. Her bağlantı için hesaplanan ve her iki yönde akan veriye uygulanan bir ağırlık

değeri vardır.

Hopfield ağı, geri yayımlı ağların eğitildiği gibi eğitilmez. Bunun yerine, örnek desen grupları seçilir ve bu gruplar ağın ağırlıklarının başlangıç değerlerini saptamak için kullanılır. Bu adım bir sefer yapıldıktan sonra herhangi bir desen ağa sunulur ve bu da giriş desenine en çok benzeyen örnek desenlerden biriyle sonuçlandırılır. Çıkış deseni ise birimlerin durumlarına bakılarak ağdan okunabilir [39].



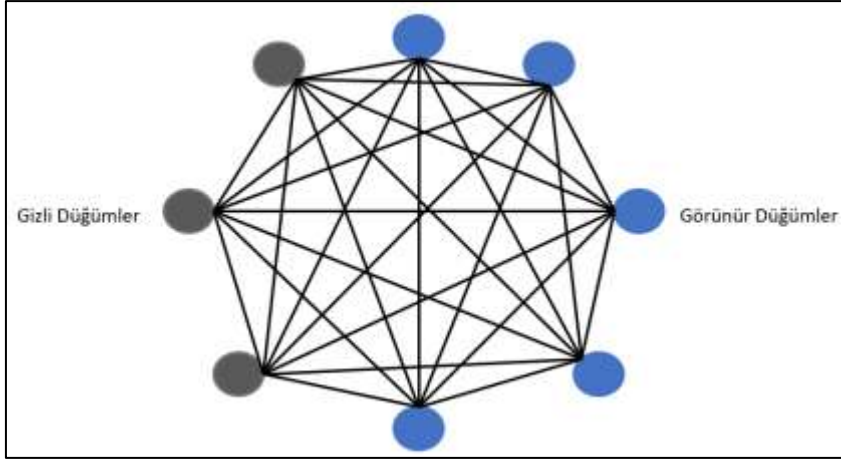
Şekil 3.4. Hopfield ağı mimarisi [39]

3.1.5. Boltzmann makinesi

Boltzmann makinesi, açık mı kapalı mı olacağına dair stokastik kararlar veren simetrik olarak bağlı, nöron benzeri birimlerden oluşan bir ağıdır. Boltzmann makineleri, eğitim verilerindeki karmaşık düzenlilikleri temsil eden ilginç özellikleri keşfetmelerine olanak tanıyan basit bir öğrenme algoritması olarak tanımlanabilir [40]. Öğrenme algoritması, birçok özellik detektörü katmanına sahip ağlarda çok yavaştır, ancak tek katmanlı özellik detektörlerine sahip "sınırlı Boltzmann makinelerinde" hızlıdır. Sınırlı Boltzmann makineleri oluşturarak, birinin özellik etkinleştirmelerini bir sonrakinin eğitim verisi olarak kullanarak birçok gizli katman verimli bir şekilde öğrenilebilir.

Boltzmann makineleri, oldukça farklı iki hesaplama problemini çözmek için kullanılır. Bir arama problemi için, bağlantılar üzerindeki ağırlıklar sabittir ve bir maliyet fonksiyonunu temsil etmek için kullanılır. Bir Boltzmann makinesinin stokastik dinamikleri daha sonra maliyet fonksiyonunun düşük değerlerine sahip ikili durum vektörlerini örneklemesine izin verir.

Bir öğrenme problemi için, Boltzmann makinesine bir dizi ikili veri vektörü gösterilir ve bu vektörleri yüksek olasılıkla oluşturmayı öğrenmesi gerekir. Bunu yapmak için, diğer olası ikili vektörlere göre veri vektörlerinin maliyet fonksiyonunun düşük değerlerine sahip olması için bağlantılarda ağırlık bulması gerekir. Bir öğrenme problemini çözmek için Boltzmann makineleri ağırlıklarında çok sayıda küçük güncelleme yapar ve her güncelleme birçok farklı arama problemini çözmelerini gerektirir. [41].



Şekil 3.5. Boltzmann makinesi [42]

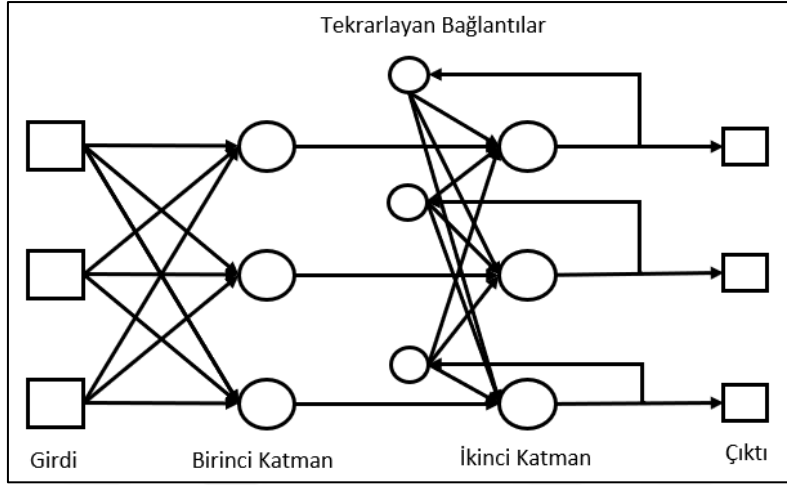
3.1.6. Hamming ağı

Hopfield ağıının genişletilmesi ile ortaya çıkan Hamming ağı Richard Lippman tarafından 1980'lerin ortalarında geliştirilmiştir. Giriş vektörleri için en az iki sayı hatasının temel sınıflandırmasını yerine getiren Hamming ağı tarafından hata aralığı tanımlanmaktadır.

Hamming ağıının öğrenmesi Hopfield ağıının öğrenme yöntemlerine benzemektedir. Burada çıkış sabitleri hem çıkış kategorisine hem de giriş vektörüne ait olurlar. Hopfield seviyesinde tekrar edilen yapı tüm bağlantı yüklerinde ortalama bir düzelme sağlar. Bağlantı yükleri, birinci sabit giriş kategori seviyesinde karşılaştırma hesapları doğurur. Çıkış kategorisi işlem elemanları ile giriş döğüm numaraları farkı Hamming örnek giriş vektörleri aralığında eşittir.

Bağlantı yükleri tıpkı kategori seviyelerinde olduğu gibi tekrarlar ve Hopfield ağıındaki gibi öğrenirler. Giriş vektör uygulamasıyla, normal ileri besleme işlemi olur. Giriş seviyesi ve çıkışları yeteri kadar uzun ve daha alçak giriş kategorileriyle karşılaştırılarak hesaplanır. Hamming ağı Hopfield ağıına göre daha üstündür. Hamming ağı en uygun durumda en az hata sınıfında çalışır. Hamming ağı, Hopfield ağıına göre daha çabuk ve doğru sonuç verir.

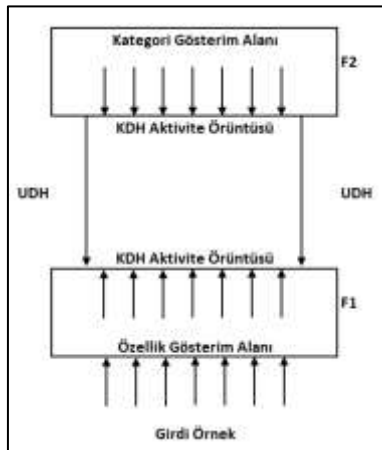
[43].



Şekil 3.6. Hamming ağı [44]

3.1.7. Adaptif rezonans teorisi modelleri

Online öğrenebilen ve gerçek zamanlı çalışabilen ağlardır. ART ağları, yeni durumlara adapte olabilmekte oldukça güçlüdürler. Bunu sağlayabilmek için bir taraftan öğrenirken diğer bir taraftan da unutabilmektedirler. Bir ART ağı, gösterim katmanı (F1) ve kategori katmanı (F2) olmak üzere iki katmandan oluşur. F1 ve F2 katmanları arasında hem aşağıdan yukarı hem de yukarıdan aşağı ağırlık vektörü vardır. Eğitim sırasında bu ağırlıklar örneklerle bakılarak değiştirilirler.



Şekil 3.7. ART ağının genel mimarisi [28]

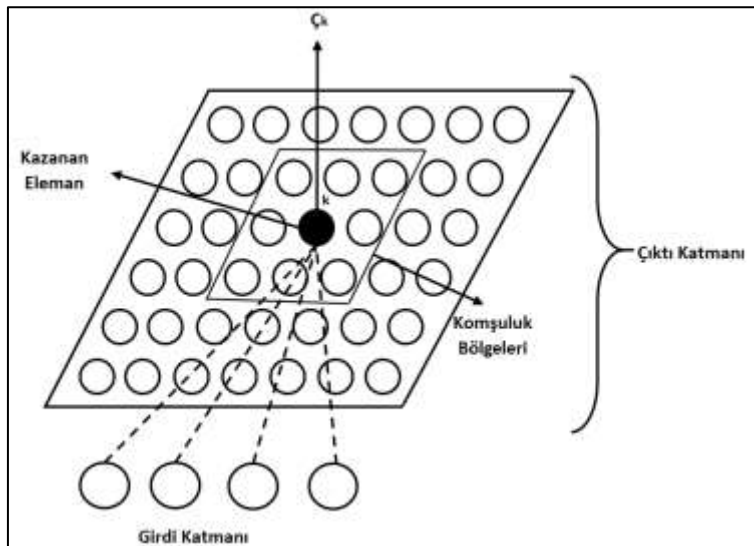
ART ağlarında girdiler direkt olarak sınıflandırılmazlar. Öncelikle girdilerin özellikleri incelenerek F1 aktivasyonu belirlenir. UDH'daki bağlantı değerlerini ile gelen bilgiler

kategorilere ayrılarak F2 katmanına gönderilir. F2 katmanındaki sınıflandırma ile F1 katmanından gelen sınıflandırma birbirleri ile eşleştirilerek, eğer örnek belirlenmiş bir sınıfa uyuyorsa o kategoride gösterilir. Aksi takdirde, ya yeni bir sınıf oluşturulur ya da girdinin sınıflandırılması yapılmaz [28].

3.1.8. Kendini düzenleyen harita ağı

Genel olarak sınıflandırma yapmak için kullanılmaktadırlar. Bu ağların girdi vektörlerini sınıflandırmak ve girdi vektörlerinin dağılımını öğrenebilme yetenekleri çok yüksektir. Bu ağların en temel özelliği olayları öğrenmek için bir öğretmen veya ağı üretmesi gereken çıktıların ağı söylenme zorunluluğunun olmamasıdır. Özellikle beklenen çıktıların belirlenemediği problemler için kullanılmaktadır.

Yapısal olarak da bu ağlar diğerlerinden farklıdır. Ağ girdi ve çıktı katmanından oluşmaktadır. Çıktı katmanı 2 boyutlu bir düzlemi göstermektedir. Proses elemanları bu düzlem üzerine dağılmış vektörleri gösterirler. Bu ağlar yarışmayı kazanma ve kazanan elemanı 1 diğerlerinin 0 değerini alması ilkesine dayanmaktadır. Bir girdi verildiğinde çıktı uzayında yarışmayı kazanan ve onun etrafındaki komşuları eğitim sırasında ağırlıklarını değiştirmektedir [28].



Şekil 3.8. Kendini düzenleyen harita ağı mimarisi [28]

3.1.9. Matematiksel model

Geri yayılım algoritması çok katmanlı ileriye dönük YSA'larda kullanılır. Geri yayılım algoritması, yapay nöronların katmanlar halinde organize edildiği ve sinyallerini “ileri” gönderdiği ve ardından hataların geriye doğru yayıldığı anlamına gelir. Ağ, giriş katmanındaki nöronlar tarafından iletilen girdileri alır ve çıktı katmanındaki nöronlara ağırlık çıkırtısını iletir. Bir veya daha fazla ara gizli katman olabilir. Geri yayılım algoritması denetimli öğrenmeyi kullanır, bu da algoritmaya ağırlık hesaplaması istenen girdi ve çıktı örneklerinin sağlandığı ve daha sonra hatanın (gerçek ve beklenen sonuçlar arasındaki fark) hesaplandığı anlamına gelir. Geri yayılım algoritması mantığı, YSA eğitim verilerini öğrenene kadar bu hatayı azaltmak üzerinedir. Yapay sinir hücrelerinin geri yayılım algoritmasını uygulayan YSA'lardaki aktivasyon işlevi, toplama işlevi seçildiğinde algılayıcının girdileri (x) ile bunlara denk gelen ağırlıklar (w) çarpılarak bir sonuç elde edilir. Eş. 3.1’de ilgili eşitlik gösterilmektedir.

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^n x_{ji} w_{ji} \quad (3.1)$$

Aktivasyon fonksiyonundan elde edilen değer, çıktı üretmek için aktivasyon fonksiyonuna ihtiyaç duyar. Farklı eğitim kümeleri ya da ihtiyaçlar için farklı aktivasyon fonksiyonları kullanılabilir. Sigmoid fonksiyonu, doğrusal olmayan çıktıların hesaplanması ve türevi alınabilen sürekli bir fonksiyon olmasından dolayı tercih edilir. Sigmoid fonksiyonu eşitliği Eş. 3.2’de gösterilmektedir.

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{-A_j(\bar{x}, \bar{w})}} \quad (3.2)$$

Sigmoid fonksiyonu büyük pozitif sayılar için bire çok yakın, büyük negatif sayılar için sıfıra çok yakındır. Bu, nöronun düşük ve yüksek çıkışı arasında yumuşak bir geçişe izin verir.

Eğitim sürecinin amacı, belirli girdiler verildiğinde istenen çıktıyı elde etmektir. Hata, gerçek ve istenen çıktı (d) arasındaki fark olduğundan, hata ağırlıklara bağlıdır ve hatayı en aza indirmek için ağırlıkların ayarlanması gerekmektedir. Her bir nöronun j . çıktı birimi için hata fonksiyonu Eş. 3.3’te paylaşılmıştır.

$$e_j(\bar{x}, \bar{w}, d) = (O_j(\bar{x}, \bar{w}) - d_j)^2 \quad (3.3)$$

Elde edilen hatanın sıfır olmasını engellemek için çıktı ile istenen hedef arasındaki farkın karesi alınmaktadır. Ağın hatası, çıktı katmanındaki tüm nöronların hatalarının toplamı olacaktır (Eş. 3.4):

$$E_J(\bar{x}, \bar{w}, d) = \sum (O_J(\bar{x}, \bar{w}) - d_J)^2 \quad (3.4)$$

Geri yayılım algoritması ile hatanın çıktıya, girdilere ve ağırlıklara nasıl bağlı olduğu hesaplanır. Hesaplandıktan sonra, gradyan iniş yöntemi kullanılarak ağırlıklar ayarlanır (Eş. 3.5):

$$\Delta w_{ji} = -n \frac{\partial E}{\partial w_{ji}} \quad (3.5)$$

Yukarıda gösterilen formül, çıktı katmanı ile bir önceki gizli katman arasındaki ağırlık değerlerinin her biri için değişim miktarını göstermektedir. n öğrenme oranını temsil etmektedir. Bu formül şu şekilde yorumlanabilir: her bir ağırlık w_{ji} 'nin ayarlanması, önceki ağırlığın w_i 'ye göre E 'nin türevi olan ağın hatasına bağımlılığı ile çarpılan sabit negatif öğrenme oranı olacaktır.

Ayarlamanın boyutu, ağırlığın işlev hatasına yaptığı katkıya bağlı olacaktır. Burada, eğer ağırlık hataya çok katkıda bulunuyorsa, ayarlama daha büyük olacaktır. Uygun ağırlıklar bulunana kadar Eş. 3.5 kullanılır. Geri yayılım algoritmasının amacı budur. İlk olarak, hatanın çıktıya ne kadar bağlı olduğunun hesaplaması gerekir (Eş. 3.6).

$$\frac{\partial E}{\partial o_j} = 2(o_j - d_j) \quad (3.6)$$

Ve sonra, çıktının aktivasyona ne kadar bağlı olduğu ağırlıklara (Eş.3.1 ve Eş.3.2'den) bağlıdır (Eş. 3.7):

$$\frac{\partial o_j}{\partial w_{ji}} = \frac{\partial o_j}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} = o_j(1 - o_j)x_i \quad (3.7)$$

Bu Eş 3.6'dan ve Eş. 3.7'den görülebilir (Eş. 3.8):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ji}} = 2(o_j - d_j)o_j(1 - o_j)x_i \quad (3.8)$$

Böylece, her ağırlığın (Eş 3.5 ve Eş.3.8'den) ayarlanması Eş.3.9'daki gibi olacaktır:

$$\Delta w_{ji} = -2n(O_j - d_j)O_j(1 - O_j)x_i \quad (3.9)$$

Bu formül iki katmanlı bir YSA eğitimi için olduğu gibi kullanılabilir. Ağı bir kat daha eğitmek için; eğer önceki bir katmanın ağırlığı (V_{ik}) ayarlanmak isteniyor ise, öncelikle hatanın ağırlığı değil, önceki katmandan gelen girdilere nasıl bağlı olduğunun hesaplaması gerekir (Eş 3.10- Eş.3.11).

$$\Delta v_{ik} = -n \frac{\partial E}{\partial v_{ik}} = -n \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial v_{ik}} \quad (3.10)$$

$$\frac{\partial E}{\partial w_{ji}} = 2(O_j - d_j)O_j(1 - O_j)w_{ji} \quad (3.11)$$

Başka bir katman daha eklemek istendiğinde aynı adımlar uygulanabilir. Hatanın ilk katmanın girişlerine ve ağırlıklarına nasıl bağlı olduğunu hesaplanabilir. Pratik nedenlerden dolayı, geri yayılım algoritmasını uygulayan YSA'ların çok fazla katmanı yoktur, çünkü ağların eğitimi için zaman katlanarak büyür. Ayrıca, daha hızlı öğrenmeye olanak sağlayan geri yayılım algoritmasına yönelik iyileştirmeler vardır [45].

3.2. Destek Vektör Regresyonu

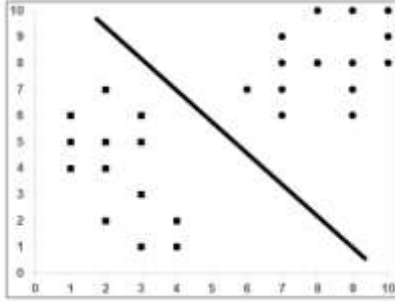
Destek Vektör Makineleri (DVM), regresyon ve sınıflandırma problemleri için kullanılmak üzere Vapnik tarafından geliştirilen bir öğrenme yöntemidir [46]. Bu yöntem, optimizasyon temelli öğrenme algoritması olup girdi vektör uzayını kernel fonksiyonları ile çok boyutlu özellik uzayını eşler. DVM, geleneksel öğrenme yöntemleriyle karşılaştırıldığında doğrusal olmayan problemleri çözmedeki performansı ile ön plana çıkmaktadır.

Genellikle sınıflandırma problemleri için kullanılmakta olan DVM, Smola ve arkadaşları tarafından regresyon uyarlaması olarak destek vektör regresyonu (DVR) adı ile ileri sürülmüştür [47]. Destek vektör regresyonunda verilerin doğrusal olarak ayrılabilir yapıda olması veya doğrusal olarak ayrılamayan yapıda olması olarak iki durum ile karşılaşılır.

3.2.1. Doğrusal destek vektör regresyonu

Smola ve arkadaşları, 2002 yılında destek vektör regresyonu terimini ileri sürdükten sonra 2004 yılında matematiksel modelleriyle daha detaylı bir çalışma yapmışlardır [48]. Çalışmalarında farklı bir kernel kullanmamışlardır.

N elemandan oluşan eğitim verilerinin $\theta = \{x_i, y_i\}, i = 1, 2, \dots, N$ olduğu varsayılırsa, regresyon problemlerinde yanıt değişkeni $y_i \in \{-1, 1\}$ yerine $y_i \in R$ olarak tanımlanmaktadır. Doğrusal olarak ayrılabilirlik durumunda, bu iki değerli veriler direkt olarak bir aşırı düzlem ile ayrılabilir (şekil 3.9).



Şekil 3.9. Doğrusal ayrılabilir veriler [49]

$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ şeklinde eğitim örnekleri tanımlanmış olsun, ϵ -duyarsız destek vektör regresyonunda (ϵ -DVR) tüm eğitim verisi için bağımlı değişken y_i 'lerden maksimum ϵ sapmaya sahip olan en uygun $f(x)$ fonksiyonunun bulunması amaçlanır. Başka bir deyişle, ϵ 'dan küçük olduğu sürece hatalar önemsizlenecektir, fakat bu değerden daha büyük bir sapma değeri de kabul edilmeyecektir. Doğrusal durumlar için Eş. 3.12 f fonksiyonunu ifade etmektedir:

$$f(x) = wx + b \quad w \in R^D, b \in R \quad (3.12)$$

Eşitlikte verilen fonksiyonun düz olması w değerinin küçük olduğu anlamına gelmektedir. $\|w\|^2$ normunu minimize etmek bu durumu sağlamanın bir yoludur. Eş. (3.13)' te konveks optimizasyon problemi ifade edilmiştir:

$$\min \frac{1}{2} \|w\|^2 \quad (3.13)$$

Kısıtlar;

$$y_i - (wx_i + b) \leq \epsilon \quad (3.14)$$

$$(wx_i + b) - y_i \leq \epsilon$$

Bazı durumlarda Eş. (3.13)'te ele alındığı gibi, (x_i, y_i) çiftlerini ϵ kesinliğinde yakınsayan bir f fonksiyonu mevcut olmayabilir ya da bazen bazı hatalar göz ardı edilmek istenebilir. Bu gibi durumlarda, bunu mümkün kılmak için esnek (yumuşak) marjin kayıp fonksiyonu

kullanılır. Modele ξ^+ ve ξ^- gevşek değişkenleri dâhil edilir [50]. Ekleme sonrası optimizasyon problemi modeli Eş. (3.15)'e dönüşmektedir [48]:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) \quad (3.15)$$

Kısıtlar;

$$y_i - (wx_i + b) \leq \varepsilon + \xi_i^+ \quad (3.16)$$

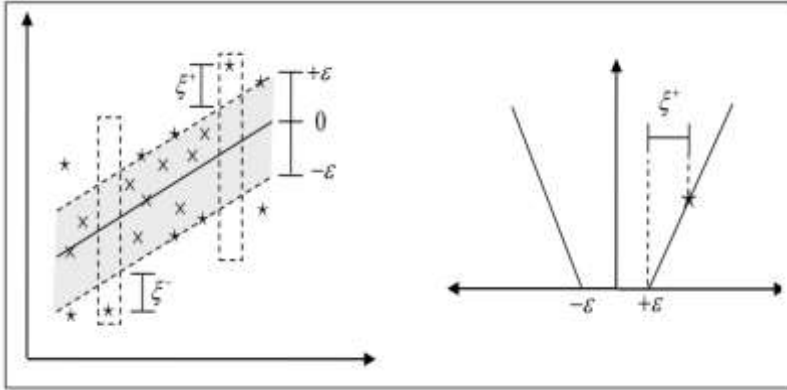
$$(wx_i + b) - y_i \leq \varepsilon + \xi_i^-$$

$$\xi_i^+, \xi_i^- \geq 0$$

$C > 0$ ceza katsayısı f fonksiyonunun düzlüğü ve tolere edilecek hata miktarı arasında dengeyi sağlamaktadır. ε -duyarsız kayıp fonksiyonu Eş (3.17)'de tanımlanmaktadır:

$$|\xi|_\varepsilon = \begin{cases} 0, & |\xi| \leq \varepsilon \\ |\xi| - \varepsilon, & d.d \end{cases} \quad (3.17)$$

Yalnızca ε değerinden büyük olan değerler doğrusal olarak cezalandırılmaktadır (Şekil 3.10).



Şekil 3.10. Doğrusal destek vektör regresyonunda kayıp fonksiyonu [48]

3.2.2. Dual problem

Lagrange çarpanları kullanılarak optimizasyon probleminin dual formu elde edilmektedir (Eş. 3.18).

$$MaxL = \begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) - \sum_{i=1}^n (\beta_i^+ \xi_i^+ + \beta_i^- \xi_i^-) - \sum_i \alpha_i^- (\varepsilon + \xi_i^- - y_i + (wx_i + b)) \\ - \sum_i \alpha_i^+ (\varepsilon + \xi_i^+ + y_i - (wx_i + b)) \end{cases} \quad (3.18)$$

Eş. (3.18)'de görülen $\beta_i^+, \beta_i^-, \alpha_i^+, \alpha_i^-$ ifadeleri Lagrange çarpanlarıdır. L'nin w, b, ξ^+, ξ^- 'ye göre kısmi türevi alınır Eş. (3.19) - (3.22) elde edilir:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n x_i (\alpha_i^- - \alpha_i^+) \quad (3.19)$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^n (\alpha_i^+ - \alpha_i^-) = 0 \quad (3.20)$$

$$\frac{\partial L}{\partial \xi_i^+} = 0 \rightarrow C - (\alpha_i^+ + \beta_i^+) = 0 \quad \forall_i \quad (3.21)$$

$$\frac{\partial L}{\partial \xi_i^-} = 0 \rightarrow C - (\alpha_i^- + \beta_i^-) = 0 \quad \forall_i \quad (3.22)$$

Eş. (3.19) - (3.22)'in kullanılması ile dual optimizasyon problemi Eş. (3.23)'e dönüşür:

$$maks - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^- - \alpha_i^+) (\alpha_j^- - \alpha_j^+) \langle x_i, x_j \rangle - \varepsilon \sum_{i=1}^n (\alpha_i^- + \alpha_i^+) - \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) \quad (3.23)$$

Kısıtlar;

$$\sum_{i=1}^n (\alpha_i^- - \alpha_i^+) = 0 \quad (3.24)$$

$$0 \leq \alpha_i^-, \alpha_i^+ \leq C$$

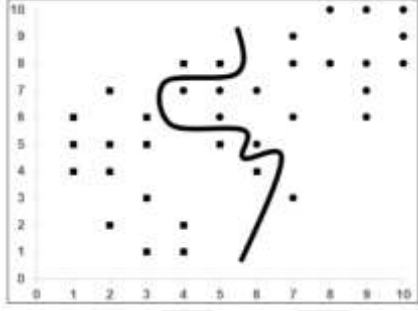
$w = \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) x_i$ ifadesi kullanılarak $f(x)$ fonksiyonu Eş.(3.25)'teki ifade ile gösterilir:

$$f(x) = \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) \langle x_i, x_j \rangle + b \quad (3.25)$$

Destek vektör açılımı olarak bilinen bu ifade, x_i örneklerinin doğrusal kombinasyonu olarak w ağırlıkları şeklinde yazılabilmektedir. Bu sayede fonksiyonun ne kadar kompleks olduğu sadece destek vektörlerinin sayısına bağlı olacak olup, girdi örneklerinin boyutundan bağımsız olacaktır. Böylelikle $f(x)$ fonksiyonu bulunurken w ağırlıklarını hesaplamaya gerek kalmamaktadır. Bu özellik algoritmanın doğrusal olmayan destek vektör regresyona genelleştirilmesinde kullanılmaktadır.

3.2.3. Doğrusal olmayan destek vektör regresyonu

Verilerin doğrusal olarak ayrılabilirdiği durumlarda veriler iki sınıfa doğrusal bir düzlem ile ayrılabilir. Fakat gerçek hayatta yapılan uygulamalarda bu durum her zaman geçerli olmayabilir (Şekil 3.11). Bu durumlarda doğrusal olmayan destek vektör regresyonuna ihtiyaç duyulmaktadır.



Şekil 3.11. Doğrusal ayrılamayan veriler [49]

Doğrusal olmayan destek vektör regresyonunun geliştirilmesi için x_i örneklerini özellik uzayına $\Phi: R^D \rightarrow H$ fonksiyonu yardımı ile haritalandırmaktadır. Sadece x_i örneklerinin iç çarpımına bağlı olan destek vektör regresyonu için $k(x_i, x_j) = \Phi(x_i), \Phi(x_j)$ fonksiyonunun hesaplanması yeterli olacaktır. Eş. (3.26)'da optimizasyon problemi verilmiştir [48].

$$\text{maks} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^- - \alpha_i^+) (\alpha_j^- - \alpha_j^+) k(x_i, x_j) - \varepsilon \sum_{i=1}^n (\alpha_i^- + \alpha_i^+) - \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) \quad (3.26)$$

Kısıtlar;

$$\sum_{i=1}^n (\alpha_i^- - \alpha_i^+) = 0 \quad (3.27)$$

$$0 \leq \alpha_i^-, \alpha_i^+ \leq C$$

Tahmin fonksiyonu ve ağırlıklar Eş. (3.28) - (3.29)'daki gibi ifade edilir:

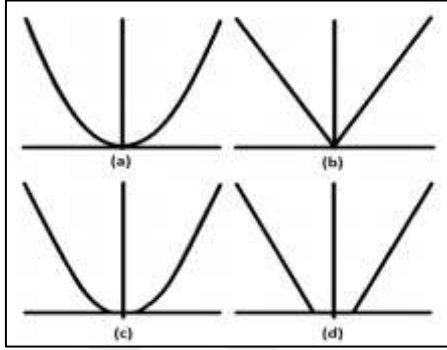
$$w = \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) \Phi(x_i) \quad (3.28)$$

$$f(x) = \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) k(x_i, x_j) + b \quad (3.29)$$

Doğrusal olmayan destek vektör regresyonunun doğrusal durumlardan farkı, w ağırlıklarının net bir şekilde elde edilmemesidir. Ayrıca doğrusal olmayan durumda optimizasyon problemi girdi uzayında değil özellik uzayında en düz fonksiyonu bulmaktadır.

3.2.4. Kayıp fonksiyonları

Destek vektör makinesi modeline kayıp fonksiyonu dâhil edilerek regresyon problemlerine uygulanabilir. Yaygın olarak kullanılan kayıp fonksiyonları Şekil 3.12’de verilmiştir.



Şekil 3.12. Kayıp fonksiyonları [51]

Klasik en küçük kareler hata kriterine karşılık gelen kayıp fonksiyonu Kuadratik (a)’da verilmiştir. (b)’de Laplace kayıp fonksiyonu verilmiştir. Bu fonksiyon aykırı değerlere karşı kuadratik fonksiyona göre daha az duyarlıdır. Verinin bilinmediği durumlarda optimal özelliklere sahip olan Huber kayıp fonksiyonu (c)’de verilmiştir [52]. Verilen üç fonksiyonun destek vektör sayısının az olmasına herhangi bir katkısı bulunmamaktadır. Buna karşılık destek vektör sayısının az olmasını sağlayan ve Huber kayıp fonksiyonu ile benzerlik gösteren ε -duyarsız kayıp fonksiyonu (d)’de verilmiştir [46].

3.2.5. Çekirdek fonksiyonları

Benzerlik ölçüsü olarak kullanılan çekirdek fonksiyonlarının esnekliği arttıkça, tekniğin daha geniş bir çözüm aralığında araması sağlanmış olur. Çekirdek fonksiyonları ile $Q(x)$ olarak belirtilen açık harita hesaplanmasına gerek kalmadan tüm hesaplamalar doğrudan girdi alanına yapılabilir [53].

Destek vektör regresyonları için yaygın olarak kullanılan kernel fonksiyonları Eş (3.30) (3.32)’de paylaşılmıştır [48]:

$$\text{Radyal Tabanlı kernel fonksiyonu: } \Phi_G(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad \gamma = 1/(2\sigma^2) \quad (3.30)$$

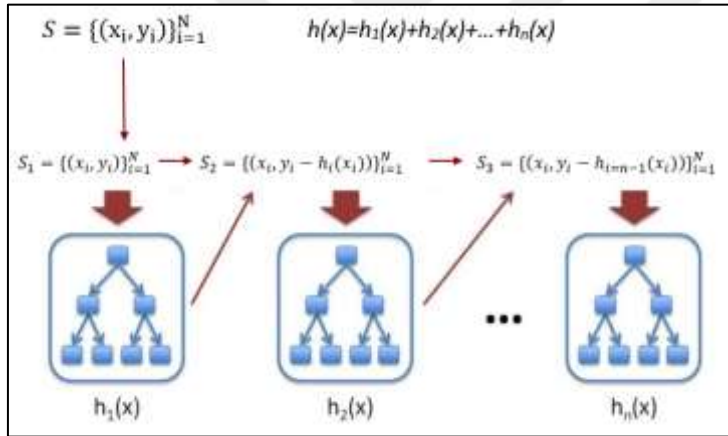
$$\text{Doğrusal kernel fonksiyonu: } \Phi_D(x_i, x_j) = x_i \cdot x_j \quad (3.31)$$

$$\text{Polynomial kernel fonksiyonu: } \Phi_p(x_i, x_j) = ((x_i, x_j) + 1)^p, \quad p \in \mathbb{N} \quad (3.32)$$

3.3. Gradyan Artırma Makinesi

Gradyan artırma algoritmaları ilk olarak sınıflandırma problemleri için makine öğrenimi topluluğu tarafından tanıtılmıştır [54-56].

Gradyan artırma yöntemi, makine öğrenimi literatüründe en yaygın kullanılan bir diğer model türüdür ve en çok kullanılan tekniklerden biridir. Aynı zamanda farklı zayıf öğrencilerden (ağaçlardan) oluşan bir topluluk modelidir. Rassal ormanda, ağaçlar paralel bağlanır ve nihai kararlar ağaçların ortalama sonuçlarına göre verilir. Bu öğrenciler bağımsız öğrencilerdir. Bununla birlikte, güçlendirmede, ağaçlar sırayla birbirine bağlanır (Şekil 3.13).



Şekil 3.13. Basit gradyan artırma modeli [57]

Tüm ağaçlar önceki ağaçların hatasına göre öğrenmektedir (Şekil 3.14, Şekil 3.15, Şekil 3.16). Bu nedenle modeller bağımsız değildir ve tüm ağaçların sonuçları toplanarak nihai tahmin yapılır. Ezberleme olabileceğinden, kullanıcı durdurma koşullarını dikkatlice tanımlamalıdır. Karar ağaçları öğrenciler olarak kullanılırsa, kategorik verileri de işleyebilir. Ağaç sayısı, öğrenme hızı, etkileşim derinliği gibi ayarlanması gereken önemli parametreler vardır.

Karar ağaçlarının sayısına da karşılık gelen yineleme sayısı, modele kaç ağacın dâhil edileceğini belirler. Her ağaç, önceki kümülatif öğrencilerden kalan kalıntıları modellemek için art arda inşa edilir. Öğrenme oranı parametresi, genellikle 0 ile 1 arasında küçük bir değer alır. Modele yeni eklenen bir öğrencinin ağırlığını belirler. Çok hızlı öğrenmek aşırı

uyuma neden olabilir, o zaman öğrenme sürecini yavaşlatmak daha iyidir. Her yeni ağaç modeli kalıntılar, ancak bu ağırlık ile çarpıldıktan sonra modele eklenir. Öğrenme oranının daha büyük değerleri, ezberlemeye neden olur, ancak daha az zaman alır. Daha küçük öğrenme oranı değerleri, fazla uydurmayı engelleyebilir ve modeli sorunsuz bir şekilde inşa edebilir, ancak daha fazla zaman alır. Etkileşim derinliği, gradyan artırma yönteminde her ağaçta yapılması gereken bölme sayısıdır. Ayrıca, bu bölünmeler sırasında etkileşime girecek maksimum değişken sayısını da belirler. Derinlik bir olduğunda, her ağaçta yalnızca bir bölünme gerçekleştirilir ve yalnızca bir değişken dâhil edilir. Derinlik birden fazla olduğunda, her ağaçta birden fazla bölünme gerçekleştirilir. Bu bölünmelerin aynı değişken üzerinde olması gerekmez. Değişkenler arasındaki etkileşim de düşünülebilir. Bu, bölünmelerde farklı değişkenlerin birleştirilebileceği anlamına gelir [58].

Sınıflandırma problemleri için gradyan artırma algoritması gözlem ağırlığı $w_i = 1/N$, $i=1,2,...,N$ eşitliği ile başlar. Örnek ağırlıklar her bir yinelemeli iterasyon $m=2,3,...,M$ için değiştirilir ve ağırlıklı sonuçlara göre sınıflandırma algoritması yeniden doldurulur. W_i ağırlıkları $G_m(x)$ sınıflandırıcısını eğitmek için kullanılır:

$$G(x) = \text{sign}(\sum_{m=1}^M a_m G_m(x)) \quad (3.33)$$

Burada $G(x)$ son sınıflandırıcıdır. Bu hesaplamayı elde etmek için aşağıdaki eşitlikler kullanılır [60]:

$$err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i} \quad (3.34)$$

$$a_m = \log \left(\frac{1-err_m}{err_m} \right) \quad (3.35)$$

$$\exp[a_m \cdot I(y_i \neq G_m(x))] \rightarrow w_i \quad i = 1, 2, \dots, N$$

Regresyon problemleri için gradyan artırma algoritması ayarı, $L(y, f(x)) = (y - f(x))^2$ hata karesi kaybı ile $L(y, f(x)) = |y - f(x)|$ mutlak kayıp arasındaki bağlantı, üstel kayıp ile y 'nin gerçek veri ve x 'in tahmin edilen değerleri olduğu binom log-olabilirlik arasındaki bağlantıya benzer. Kayıp fonksiyonlarından biri, regresyon ve hesaplama için Huber kaybıdır [59].

$$L(y, f(x)) = \begin{cases} (y - f(x))^2, & |y - f(x)| \leq \delta, \\ 2\delta|y - f(x)| - \delta^2, & \text{dd} \end{cases} \quad (3.36)$$

3.4. Model Performans Değerlendirme Ölçütleri

Regresyon problemlerinde elde edilen modelin başarısını ölçmek için çeşitli ölçüm yöntemleri kullanılır. Bunlardan bazıları;

3.4.1. Determinasyon katsayısı (R^2)

Determinasyon katsayısı yani R^2 , bağımsız bir değişkenle veya bir regresyon modelindeki değişkenlerle açıklanan bağımlı değişken için varyans oranını temsil eden istatistiksel bir ölçüdür. Korelasyon, bağımsız ve bağımlı değişken arasındaki ilişkinin gücünü açıklarken, R^2 , bir değişkenin varyansının ikinci değişkenin varyansını ne ölçüde açıkladığını açıklar. Dolayısıyla, örneğin bir modelin R^2 değeri 0,50 ise, gözlemlenen varyasyonun yaklaşık yarısı modelin girdileri ile açıklanabilir anlamına gelmektedir. R^2 değerinin matematiksel gösterimi Eş. (3.37)'de gösterilmiştir.

$$R^2 = 1 - \frac{\sum(Y_i - Y_i^*)^2}{\sum(Y_i - Y_{ort})^2} \quad (3.37)$$

Y_i gözlem değeri, Y_i^* tahmin edilen değer, Y_{ort} gözlem değerlerinin ortalaması

3.4.2. Hata kareleri ortalaması (MSE)

Hata kareleri ortalaması, gerçek ve tahmini değerler arasındaki farklar karesinin ortalamasıdır (Eş. 3.38).

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - Y_i^*)^2 \quad (3.38)$$

Y_i gözlem değeri, Y_i^* tahmin edilen değer, n örneklem sayısı

3.4.3. Ortalama mutlak hata (MAE)

Ortalama mutlak hata, gerçek ve tahmini değerler arasındaki farkın mutlak değerlerinin ortalamasıdır (Eş.3.39).

$$MAE = \frac{\sum_{i=1}^n |Y_i - Y_i^*|}{n} \quad (3.39)$$

Y_i gözlem değeri, Y_i^* tahmin edilen değer, n örneklem sayısı

3.4.4. Ortalama mutlak yüzde hata (MAPE)

Ortalama mutlak yüzde hata, tahminlerin mutlak yüzde hatalarının ortalamasıdır. Bu metrik, hatayı yüzde olarak sağladığından anlaşılması kolaydır. Ayrıca, mutlak yüzde hataları kullanıldığından, birbirini iptal eden pozitif ve negatif hatalar problemi önlenir. Sonuç olarak, MAPE'nin yönetsel çekiciliği vardır ve tahminlerde yaygın olarak kullanılan bir ölçüdür. MAPE ne kadar küçükse tahmin o kadar iyidir (Eş. 3.40).

$$MAPE = \frac{\sum_{i=1}^n \left| \frac{Y_i - Y_i^*}{Y_i} \right|}{n} \quad (3.40)$$

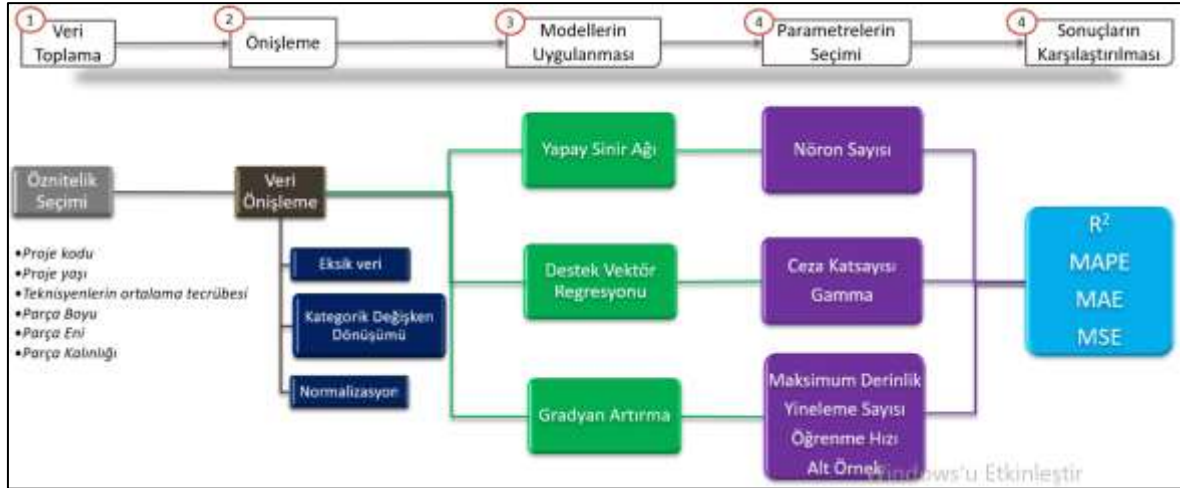
Y_i gözlem değeri, Y_i^* tahmin edilen değer, n örneklem sayısı

Ortalama mutlak yüzde hata değeri %10'un altında olan modeller “çok iyi”, %10-%20 arasında olan modeller “iyi”, %20-%50 arasında olan modeller “kabul edilebilir” ve ortalama mutlak yüzde hata değeri %50'den fazla olan modeller ise “yanlış ve hatalı” olarak değerlendirilmektedir [60].



4. UYGULAMA

Çalışmada havacılık ve savunma sanayide faaliyet gösteren bir firmaya ait detay üretim alanlarından talaşlı imalat, sac metal imalat, kompozit imalat ve kalite alanlarında üretim süresi tahminleme çalışması yapılmıştır. Uygulama adımlarını gösteren akış şeması Şekil 4.1’de paylaşılmıştır.



Şekil 4.1. Akış Şeması

4.1. Öznitelik Seçimi

Çalışmada, belirlenen talaşlı imalat, sac metal, kompozit ve kalite detay alanlarına ait 2’şer iş merkezi olmak üzere toplam 8 iş merkezinde çalışma yapılmıştır. Belirlenen 8 iş merkezinin son 5 yıla ait gerçekleştirme verileri veri tabanından çekilmiştir. Gerçekleşme işçilikleri; teknisyenlerin her parça için üretime başlamadan önce sisteme okutmaları gereken iş emirleri üzerinden veri tabanına aktarılıp hesaplanmaktadır. Çalışmanın yapıldığı üretim tesisinde montaj üretim alanları, işin akışı ve seriliğinden dolayı detay üretim alanlarına kıyasla üretim süresi tahmin etmeye daha uygun bir ortama sahiptir. Bundan dolayı montaj üretim alanlarına ait belirlenen iş merkezleri üzerinden uzman görüşü ve literatür taraması ile aday nitelikler belirlenmiştir [13]. Belirlenen aday nitelikler;

- Teslimat (projeye ait üretilen kaçınıcı parça olduğu bilgisi)
- Proje yılı
- Proje % bitiş
- Parça üzerinde çalışan teknisyen sayısı

- Teknisyenlerin ortalama tecrübesi

Seçilen aday niteliklerin üretim süresi ile olan ilişkisini görebilmek amacıyla korelasyon analizleri yapılmıştır (Çizelge 4.1). Korelasyon rakamlarına göre üretim süresi ile yüksek ilişkili öznitelikler belirlenmiştir.

Çizelge 4.1. Üretim süresinin aday parametreler ile korelasyonu

İş Merkezi	Teslimat	Proje Yılı	Proje % Bitiş	Teknisyen Sayısı	Teknisyenlerin Ortalama Tecrübesi
RD1	-0,51	-0,52	-0,52	0,28	-0,66
RD2	-0,66	-0,67	-0,67	0,12	-0,66

Korelasyon rakamlarına göre seçilen öznitelikler:

- Teslimat
- Proje yılı
- Proje % Bitiş
- Teknisyenlerin ortalama tecrübesi

Teslimat, Proje Yılı ve Proje % Bitiş öznitelikleri direkt olarak birbirleri ile ilişkili olduğundan dolayı yalnızca proje yılı özniteliği ile ilerlenmesine karar verilmiştir.

Korelasyon analizi ile seçilen özniteliklere ek olarak uzman görüşleri de alınmıştır.

Detay üretim alanlarına ait yapılacak olan üretim süresi tahminlemesi çalışmasında kullanılacak özniteliklerin nihai hali şu şekildedir:

- Proje Yılı
- Teknisyenlerin Ortalama Tecrübesi
- Proje Kodu
- Parça boyu
- Parça eni
- Parça kalınlığı

4.2. Veri Ön İşleme

Çalışmada özniteliklerin belirlenmesinden sonraki adım olarak SQL server ile sistemden son 5 yıla ait gerçekleştirmeler seçilen 8 detay iş merkezi için çekilmiştir. Veri ön işleme olarak öncelikle eksik gelen veriler ve aykırı değerler ayıklanmıştır. Veri setine ait kategorik değişkenler binary yani ikili (0-1) olacak şekilde dönüştürülmüştür. Nümerik değişkenler arasındaki ölçek farklılıklarının modelin tahmin etme performansını olumsuz etkilememesi ve yanlış yorumlanmaması için normalizasyon işlemi yapılmıştır. Min-Max tekniği ile bütün veriler [0.1-0.9] arasında ölçeklenerek normalize edilmiştir. Normalizasyon işlemi sadece nümerik olan girdi değişkenlerine uygulanmış olup ikili dönüşüme uğramış olan kategorik değişkenlere ve çıktı değişkenlerine uygulanmamıştır. Veri ön işlemeden önceki veri seti ile sonraki veri seti örnekleri Çizelge 4.2 ve Çizelge 4.3'te paylaşılmıştır.

Çizelge 4.2. Veri ön işleme öncesi veri seti örneği

Proje	Parça No	Proje Yılı	Tecrübe	En	Kalınlık	Boy	Toplam Saat
A1	X1	3,93	2,58	241,30	1,14	444,50	20,50
A1	X1	4,04	2,87	241,30	1,14	444,50	15,50
A1	X1	4,07	1,68	241,30	1,14	444,50	17,75
A1	X1	4,08	0,08	241,30	1,14	444,50	25,00
A1	X1	4,29	2,68	241,30	1,14	444,50	13,00

Çizelge 4.3. Veri ön işleme sonrası veri seti örneği

Parça No	Proje Yılı	Tecrübe	En	Kalınlık	Boy	Toplam Saat	Proje_ A1	Proje_A2	Proje_A3
X1	0,10	0,34	0,90	0,10	0,66	20,50	1	0	0
X1	0,11	0,36	0,90	0,10	0,66	15,50	1	0	0
X1	0,11	0,25	0,90	0,10	0,66	17,75	1	0	0
X1	0,11	0,10	0,90	0,10	0,66	25,00	1	0	0
X1	0,12	0,34	0,90	0,10	0,66	13,00	1	0	0

4.3. Modellerin Uygulanması ve Hiperparametrelerin Belirlenmesi

Modellerin uygulanmasında Python programından yararlanılmıştır. Her iş merkezine ait veri setinin %75'i eğitim, %25'i test verisi olarak ayrılmıştır. Yapay Sinir ağları, destek vektör regresyonu ve gradyan artırma modelleri her iş merkezi için uygulanmıştır.

4.3.1. Yapay sinir ağıları uygulaması

Yapay sinir ağlarında çok katmanlı algılayıcılar modeli kullanılmış olup, Python uygulamasının “Scikit-learn” kütüphanesinden “MLPRegressor” fonksiyonu kullanılmıştır. Maksimum iterasyon sayısı 200 olarak alınmıştır. Python uygulamasında yapay sinir ağıları modeli için yazılan kodlar detaylı olarak EK-1’de paylaşılmıştır. YSA modellerinde performans temel olarak aktivasyon fonksiyonu, gizli katman sayısı ve gizli katmanlarda bulunan nöron sayısı ile ilişkilidir.

Yapay sinir ağlarında nöron sayısı ne kadar fazla olur ise genelleme yeteneği o kadar az olmaktadır, bu nedenle optimum hiperparametre sayısı belirlenirken nöron sayısının olabildiğince az olması gerekmektedir [14]. Nöron sayısının sınırlandırılabilmesi için nöron sayısı belirlenmesi üzerine en çok kullanılan geometrik piramit kuralından faydalanılmıştır. Geometrik piramit kuralına göre girdi katmanından çıktı katmanında doğru nöron sayısı azalmalıdır ve nöron sayısı girdi sayısının 2 katından fazla olmamalıdır. Ek olarak nöron sayısı, girdi ve çıktı sayısının çarpımının karekökünden az olmamalıdır [14]. Geometrik kural ile belirlenen kısıtlara göre, modelde en iyi sonucu verecek nöron sayısının belirlenmesi için Scikit-learn kütüphanesinden gridsearchcv fonksiyonu kullanılmıştır. Kullanılan fonksiyon sayesinde 10 kat ağ arama ile YSA modellerinde en iyi sonucun verilmesini sağlayan nöron sayısının belirlenmesi hedeflenmiştir. Grid arama fonksiyonu ile, gizli katman sayısı için aday hiperparametreler 1 ve 2 alınmıştır. Aktivasyon fonksiyonu olarak ise relu ve sigmoid fonksiyonları aday hiperparametreler olarak değerlendirilmiştir. Her bir iş merkezi için geometrik kural kısıtlarına göre belirlenen 3 ile 12 arası tüm nöron sayıları denenmiştir. Her bir iş merkezi için modellerin en başarılı sonuçları vermiş olduğu nöron sayıları aşağıda Çizelge 4.4’te belirtilmiştir.

Çizelge 4.4. YSA modelleri için optimum nöron sayıları

Üretim Grubu	İş Merkezi	Gizli Katman Sayısı	Nöron Sayısı	Aktivasyon Fonksiyonu
Sac Metal	Gerdirmeli Şekil Verme Presi 1	2	12,12	Relu
	Gerdirmeli Şekil Verme Presi 2	2	10,9	Relu
Talaşlı İmalat	5 Eksenli İşleme Tezgâhı 1	2	11,12	Relu
	5 Eksenli İşleme Tezgâhı 2	2	12,12	Relu
Kompozit	Hassas Frezeleme Tezgâhı	2	12,9	Relu
	5 Eksenli Frezeleme ve Su Jeti	2	12,10	Relu
Kalite	Ultrasonik Kontrol Tezgâhı 1	2	12,12	Relu
	Ultrasonik Kontrol Tezgâhı	2	12,11	Relu

4.3.2. Destek vektör regresyonu uygulaması

Destek vektör regresyonu uygulamasında Python uygulamasının “Scikit-learn” kütüphanesinden “SVR” fonksiyonu kullanılmıştır. Python uygulamasında destek vektör regresyonu modeli için yazılan kodlar detaylı olarak EK-2’de paylaşılmıştır. Kernel fonksiyonu olarak doğrusal olmayan regresyon modellerinde kullanılan radyan tabanlı fonksiyon (RBF) verilerin doğrusal olmamasından dolayı seçilmiştir.

Destek vektör regresyonu modellerinde modelinin performansı C ve γ hiperparametrelerinden etkilenmektedir [61, 62]. C hiperparametresi modeldeki ceza katsayısını ifade etmektedir. Ceza katsayısı, modeldeki Lagrange çarpanının alabileceği maksimum değeri gösteren ifadedir [63]. γ hiperparametresi ise modelin eğriliğini ifade eden değerdir. Modelde γ değeri arttıkça, modelin eğriliğinin arttığı anlamına gelmektedir [64]. YSA modellerinde olduğu gibi, bu modelde de C ve γ değerlerinin belirlenmesi için literatürde genel geçer kurallar yoktur. Yine YSA modellerinde uygulandığı gibi destek vektör regresyonu modellerinde en iyi sonucu veren hiperparametrelerin belirlenmesi için 10 kat ağ arama fonksiyonu kullanılmıştır. Grid arama algoritması ile her bir iş merkezi için C ve γ ($C = 2, 2^2, 2^4, 2^5$; $\gamma = 2, 2^2, 2^4, 2^5$)’nin üssel sonuçları değerlendirilmiştir. Değerlendirme sonucu modellerde en iyi sonucu veren C ve γ çiftleri Çizelge 4.5’te paylaşılmıştır.

Çizelge 4.5. DVR modelleri için optimum C ve Gamma değerleri

		Destek Vektör Reg.	
Üretim Grubu	İş Merkezi	C	γ
Sac Metal	Gerdirmeli Şekil Verme Presi 1	2^5	2^5
	Gerdirmeli Şekil Verme Presi 2	2^5	2^5
Talaşlı İmalat	5 Eksenli İşleme Tezgâhı 1	2^5	2
	5 Eksenli İşleme Tezgâhı 2	2^5	2
Kompozit	Hassas Frezeleme Tezgâhı	2^5	2^4
	5 Eksenli Frezeleme ve Su Jeti	2^5	2^5
Kalite	Ultrasonik Kontrol Tezgâhı 1	2^5	2^5
	Ultrasonik Kontrol Tezgâhı	2^5	2^5

4.3.3. Gradyan artırma modeli uygulaması

Gradyan artırma modeli uygulamasında Python uygulamasının “Scikit-learn” kütüphanesinden “GradientBoostingRegressor” modülü kullanılmıştır. Python

uygulamasında gradyan artırma modeli için yazılan kodlar detaylı olarak EK-3'te paylaşılmıştır. Gradyan artırma modelinin performansını etkileyen dört hiperparametre vardır:

D: Modelin maksimum etkileşim sırasını da kontrol eden karar ağaçlarının derinliği

K: Karar ağaçlarının sayısına da karşılık gelen yineleme sayısı

α : Genellikle 0 ile 1 arasında küçük bir pozitif değer olan öğrenme hızı,

H: Her yinelemeli adımda kullanılan veri oranı

Gradyan artırma algoritmasında da diğer iki algoritmada olduğu gibi en iyi sonucu veren hiperparametrelerin belirlenmesinde genel geçer bir kural yoktur. Bu sebeple yine diğer modellerde uygulandığı gibi burada da en iyi sonucu veren değerlerinin belirlenmesi için 10 kat ağ arama (grid search) algoritması D, K, α ve H hiperparametreleri üzerinde uygulanmıştır. Grid arama fonksiyonu her bir iş merkezi için D =2,4,8; K = 100,500,1000; α =0.01,0.1,0.2,0.5; H= 0.5,0.75,1 değerlerinin kombinasyonlarını kullanarak modelde en iyi sonucu veren hiperparametre değerlerini bulmayı hedeflemiştir. Grid arama fonksiyonu ile her bir iş merkezi için belirlenen optimum hiperparametreler Çizelge 4.6'da belirtilmiştir.

Çizelge 4.6. Gradyan artırma modelleri için optimum D, K, α ve H değerleri

Üretim Grubu	İş Merkezi	Gradyan Artırma			
		D	K	α	H
Sac Metal	Gerdirmeli Şekil Verme Presi 1	8	1000	0,01	0.5
	Gerdirmeli Şekil Verme Presi 2	2	1000	0,2	0.75
Talaşlı İmalat	5 Eksenli İşleme Tezgâhı 1	2	1000	0,1	1
	5 Eksenli İşleme Tezgâhı 2	4	1000	0,01	0.75
Kompozit	Hassas Frezeleme Tezgâhı	8	500	0,01	0.5
	5 Eksenli Frezeleme ve Su Jeti	8	1000	0,01	0.75
Kalite	Ultrasonik Kontrol Tezgâhı 1	4	500	0,01	0.5
	Ultrasonik Kontrol Tezgâhı	4	500	0,1	0.5

4.4. Modellerin Karşılaştırılması

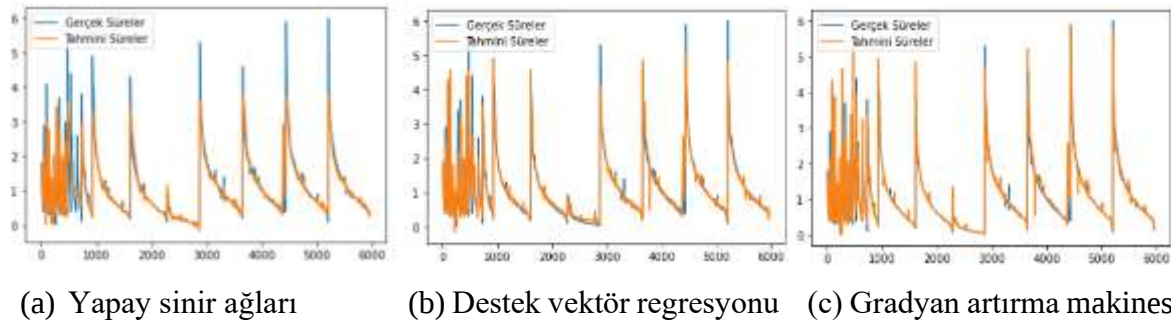
Bütün iş merkezlerine uygulanan modeller için en iyi sonucun verilmesini sağlayan hiperparametre değerlerinin belirlenmesinden sonra modellerin tekrar tahmin yapması sağlanmıştır. Elde edilen tahmin değerleri ile gerçek değerler karşılaştırılmış ve bu sayede her bir iş merkezi için en başarılı model belirlenmiştir. Modellerin tahmin başarısı R^2 , MSE

ve MAPE değerleri ile ölçülmüştür. Karşılaştırma sonuçları özeti Çizelge 4.7’de paylaşılmıştır.

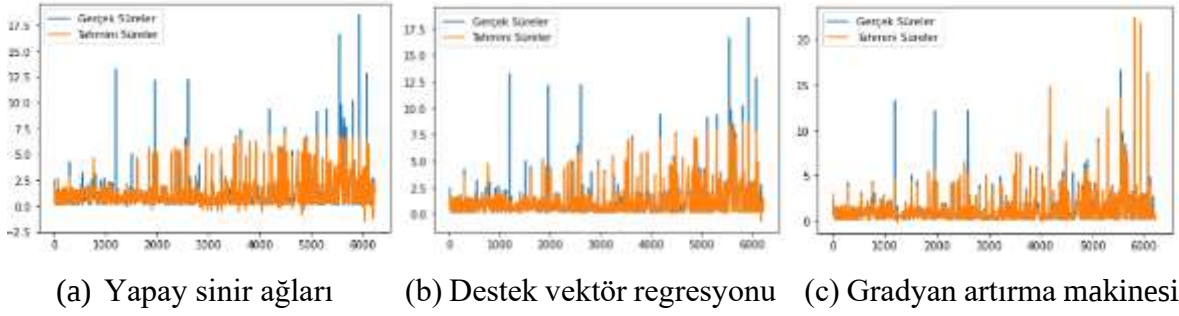
Çizelge 4.7. Model tahmin başarıları

Üretim Grubu	İş Merkezi	Yapay Sinir Ağı			Destek Vektör Reg.			Gradyan Artırma		
		R ²	MSE	MAPE %	R ²	MSE	MAPE %	R ²	MSE	MAPE %
Sac Metal	Gerdirmeli Şekil Verme Presi 1	0,89	0,29	25,62	0,93	0,23	20,21	0,96	0,16	6,94
	Gerdirmeli Şekil Verme Presi 2	0,75	0,79	41,85	0,79	0,74	21,84	0,86	0,60	32,42
Talaşlı İmalat	5 Eksenli İşleme Tezgâhı1	0,81	4,79	16,79	0,84	7,03	9,23	0,96	3,48	7,32
	5 Eksenli İşleme Tezgâhı2	0,81	9,90	18,11	0,98	3,52	6,74	0,98	3,15	6,01
Kompozit	Hassas Frezeleme Tezgâhı	0,94	1,44	26,72	0,95	1,29	14,75	0,96	1,21	15,11
	5 Eksenli Frezeleme ve Su Jeti	0,87	1,62	43,34	0,94	1,09	16,27	0,98	0,69	11,07
Kalite	Ultrasonik Kontrol Tezgâhı1	0,84	1,23	26,81	0,86	1,16	21,89	0,89	1,01	23,26
	Ultrasonik Kontrol Tezgâhı2	0,85	1,22	26,48	0,79	1,49	16,40	0,91	0,95	15,42

Modellerin tahmin başarıları, performans değerlendirme metriklerinden MAPE değerine göre belirlenmiştir. MAPE değerlerine göre: Sac metal üretim prosesi için seçilen “Gerdirmeli şekil verme presi 1” tezgâhına ait hata yüzdesi en az %6,94 ile gradyan artırma modelinde görülmüştür. Destek vektör regresyonu modeli %20,2 ile en düşük ikinci hata oranını vermiştir. En yüksek hata oranı ise %25,62 ile yapay sinir ağları modelinde görülmüştür. “Gerdirmeli şekil verme presi 2” incelendiğinde, %21,84 ile en küçük hata oranı destek vektör regresyonu modelinde görülmüştür. %32,42 hata oranı ile gradyan artırma modeli ikinci sırada olup, en yüksek hata oranı %41,85 ile yapay sinir ağları modelinde görülmüştür.

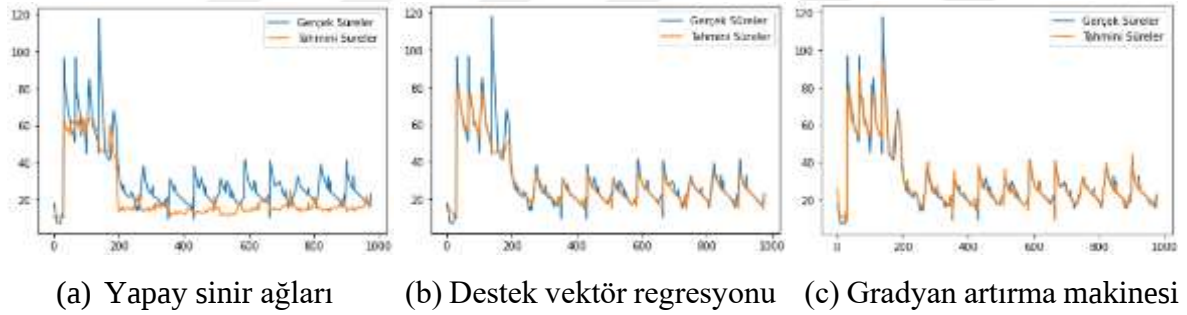


Şekil 4.2. Gerdirmeli şekil verme presi 1 gerçek süreler ve tahmini süreler

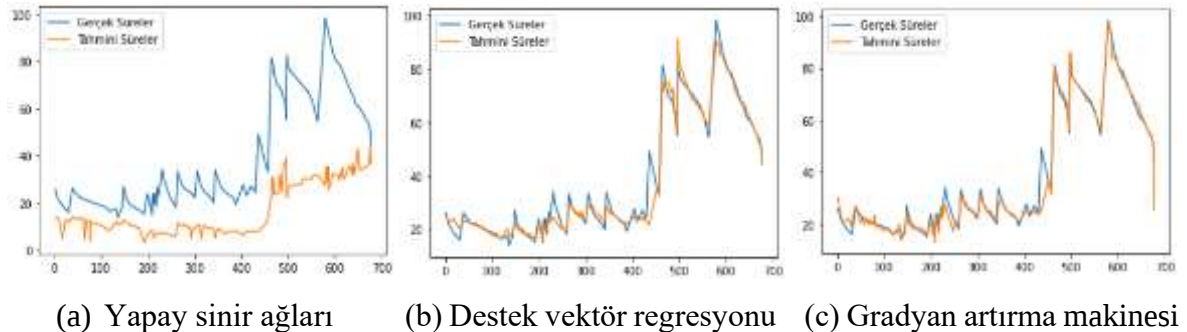


Şekil 4.3. Gerdirmeli şekil verme presi 2 gerçek süreler ve tahmini süreler

Talaşlı imalat prosesi için seçilen “5 eksenli işleme tezgâhı 1” tezgahına ait hata oranları incelendiğinde %7,32 ile gradyan artırma modeli en başarılı sonucu vermiştir. %9,23 ile destek vektör regresyonu en başarılı ikinci hata oranını veren model olarak karşımıza çıkarken, yapay sinir ağları %16,79 ile en yüksek hata oranını veren model olmaktadır. “5 eksenli işleme tezgâhı 2” için karşılaştırılan model performanslarına göre; gradyan artırma modeli %6,01 ile en düşük hata oranını vermiş olup, destek vektör regresyonu modeli %6,74 ile takip etmektedir. %18,11 hata oranı ile yapay sinir ağları modeli ise son sırada gelmektedir.



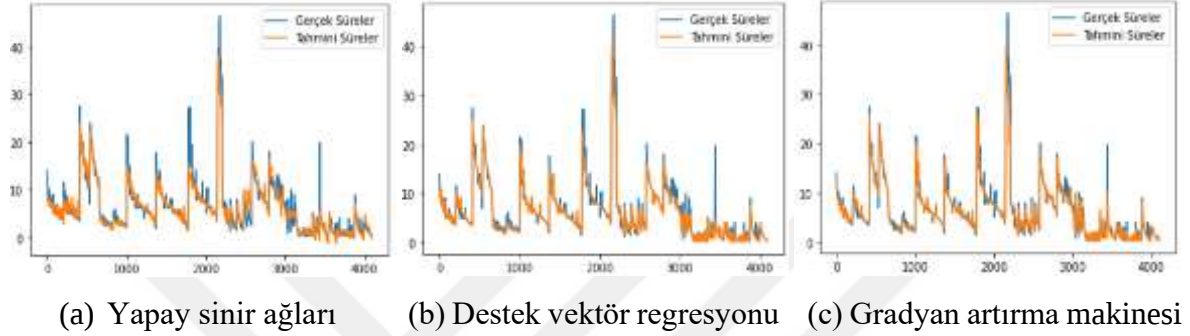
Şekil 4.4. 5 eksenli işleme tezgâhı 1 gerçek süreler ve tahmini süreler



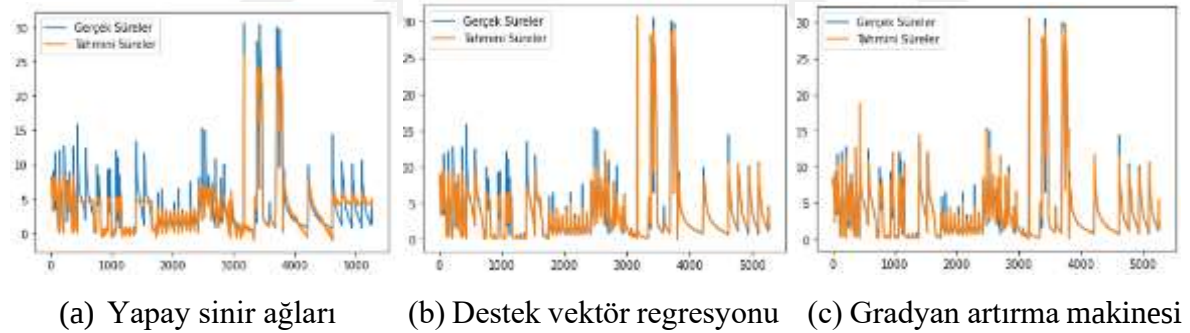
Şekil 4.5. 5 eksenli işleme tezgâhı 2 gerçek süreler ve tahmini süreler

Kompozit üretim prosesi için seçilen iş merkezlerinden “Hassas frezeleme tezgâhı”nda %14,75 ile en düşük hata oranı gradyan artırma modelinde görülmektedir. %15,11 hata oranı

ile destek vektör regresyonu takip etmekte olup, yapay sinir ağları %26,72 hata oranı ile en başarısız model olarak belirlenmektedir. “5 eksenli frezeleme ve su jeti tezgâhı” üzerinde uygulanan modellerin MAPE değerleri karşılaştırıldığında %11,07 oranı ile gradyan artırma modeli en başarılı model olmuştur. %16,40 oranı ile destek vektör regresyonu en başarılı ikinci model olurken yapay sinir ağları %43,34 hata oranı ile en başarısız model olarak karşımıza çıkmaktadır.

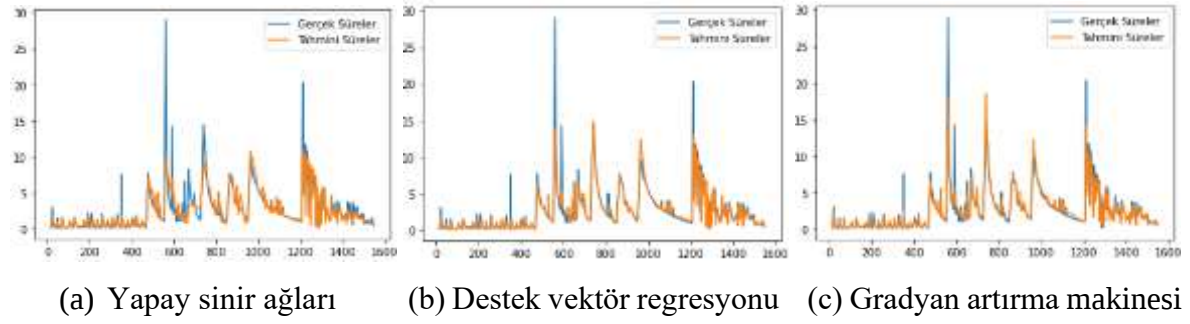


Şekil 4.6. Hassas frezeleme tezgâhı gerçek süreler ve tahmini süreler

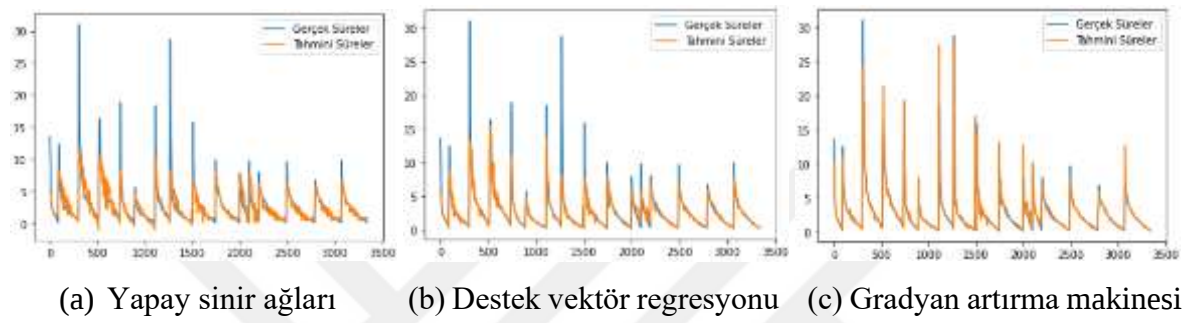


Şekil 4.7. 5 eksenli frezeleme ve Su Jeti gerçek süreler ve tahmini süreler

Kalite iş merkezlerinden “ultrasonik kontrol tezgâhı 1” için en düşük hata oranı %21,89 ile destek vektör regresyonunda olup en düşük ikinci hata oranı ise %23,26 ile gradyan artırma modelinde görülmektedir. Yapay sinir ağları modeli %26,81 hata oranı ile en yüksek hata sonucunu vermiştir. “Ultrasonik kontrol tezgâhı 2” tezgâhında ise %15,42 hata oranı ile en başarılı sonucu gradyan artırma modeli vermiştir. %16,40 hata oranı ile destek vektör regresyonu ikinci sırada yer alırken yapay sinir ağları modeli %26,48 hata oranı ile en yüksek hatayı vermektedir.



Şekil 4.8. Ultrasonik kontrol tezgâhı 1 gerçek süreler ve tahmini süreler



Şekil 4.9. Ultrasonik kontrol tezgâhı 2 gerçek süreler ve tahmini süreler

5. SONUÇ VE TARTIŞMA

Rekabetin ve maliyet kavramının çok önemli olduğu günümüz koşullarında, müşteri taleplerine hızlı ve doğru karşılık verip aksiyon alabilmek müşteri memnuniyeti açısından büyük öneme sahiptir. Kaynakların kısıtlı olduğu üretim ortamında verimliliği göz önünde bulundurarak üretim faaliyetlerine devam edebilmek, ileriye dönük yatırımların belirlenebilmesi için üretim sürelerinin doğru tahmin edilmesi rekabet avantajı sağlamayı hedefleyen firmalar için kritik bir öneme sahiptir. Zaman etüdü üretim süresi belirlenmesi açısından en yaygın kullanılan yöntem olmasına rağmen ürün çeşitliliğinin ve tezgah/proses çeşitliliğinin çok fazla olduğu tesislerde her bir parça için zaman etüdü yapmak büyük bir iş yükü gerektirmektedir. Bu durum zaman ve dolayısıyla maliyet kaybına yol açmaktadır. Zaman ve maliyet kaybının önüne geçmek için tek tek zaman etüdü yapmak yerine tahminleme modeli kurmak, ürün çeşitliliğinin çok fazla olduğu tesislerde büyük avantaj sağlamaktadır.

Bu çalışmada, detay üretim alanlarında üretim süresi tahminleme amacıyla makine öğrenmesi algoritmaları uygulanmıştır. Yapılan literatür çalışması sonucu tahminleme problemlerinde yüksek başarı sağlayan yapay sinir ağları, destek vektör regresyonu ve gradyan artırma modellerinin kullanılmasına karar verilmiştir. Bu amaçla öncelikle özniteliklerin belirlenmesi hedeflenmiştir. Öznitelikler belirlenirken tesiste daha düzenli bir akışa sahip olan montaj alanlarına ait veriler kullanılmıştır. Montaj alanlarına ait veriler ile üretim süresinin ilişkili olduğu öznitelikler belirlenmiştir. Literatür ve uzman görüşlerinin de katkısı ile öznitelikler nihai haline getirilmiştir. Tahminleme çalışmasının yapılacağı detay parça üretim alanlarında uzman görüşü ve iş yükü yoğunlukları dikkate alınarak 4 temel proses seçilmiştir. Seçilen prosesler; sac metal, kalite, talaşlı imalat ve kompozit imalat olarak belirlenmiştir. Seçilen proseslere ait 2'şer iş merkezinden geçmişe dönük 5 yıllık gerçekleşme verileri SQL ile veri tabanından çekilmiştir. Toplam 8 iş merkezine destek vektör regresyonu, yapay sinir ağları ve gradyan artırma algoritmaları uygulanmıştır. Algoritmaların 8 iş merkezinde olan farklı kombinasyonları ile toplam 24 adet üretim süresi tahmin modeli oluşturulmuştur. Modellerin başarıları MAPE değeri ile karşılaştırılmıştır.

Modellerin MAPE değerleri incelendiği zaman; gradyan artırma modelleri ortalamada %14,69 hata oranı ile en başarılı algoritma olmuştur ve 8 iş merkezinden 5'inde en iyi tahmin sonucunu vermiştir. Destek vektör regresyonu modelleri ise ortalamada %15,92 hata oranı

ile en başarılı ikinci algoritma olurken 8 iş merkezinden 3'ünde en iyi sonucu vermiştir. Yapay sinir ağları modellerinin ortalamada %28,22 hata oranı her iş merkezinde en yüksek hata oranını verdiği görülmüştür.

Çalışma sonucunda benzer veri setleri ile yapılan tahminleme çalışmaları incelenmiştir. Lim ve arkadaşları; 2019 yılında üretim iş emirlerini kullanarak destek vektör regresyonu ile üretim süresi tahminlemiş olup çalışma sonucunda %15,38 hata oranı elde etmişlerdir [9]. Kurnaz ise çalışmasında yapay sinir ağları kullanarak kablo takımlarının ihale için öngörülen üretim sürelerini tahmin etmiştir. İş emri miktarı ve takımlara ait fiziksel özelliklerin girdi olarak kullanıldığı çalışmada %14,8 hata oranı elde edilmiştir [14]. Bilekdemir 2011 yılında yaptığı çalışmada su sayacı üretim süresi tahmin etmek amacıyla C4.5 sınıflandırma algoritmasını kullanmıştır. Üretim hattından geçen bir önceki ürün, ürünün üretildiği vardiya gibi operasyonel girdiler ile yaptığı çalışmada ortalama %38 hata oranı elde etmiştir [13].

MAPE sonuçlarına göre hata oranı %10'un altında olan modeller "çok iyi", %10-20 aralığında olan modeller "iyi", %20-%50 arasında olan modeller "kabul edilebilir" ve hata oranı %50'nin üzerinde olan modeller ise "yanlış ve hatalı" olarak değerlendirilmektedir [60]. Çalışmanın yapıldığı firma MAPE değeri %20 altında olan modelleri başarılı kabul ederek o iş merkezine ait üretim süresi tahmin modelini kullanabilecektir. Firma, özellikle müşterilerden gelen potansiyel ürünler üzerine istenen teklif talepleri için değerlendirme yaparken önerilen üretim süresi tahmin modelini kullanarak iş gücü ve iş yükü değerlerine ulaşabilecektir. Ayrıca yine teklif paketlerinde değerlendirilmesi gereken tezgâh yatırımı ihtiyaçlarını da daha doğru analiz edebilecektir. Üretim süresine yönelik yapılan tahminlerin sapmalarından dolayı oluşabilecek maliyet farklılıklarının önüne geçebilecektir. Ek olarak, yine üretim süresi tahmin sapmalarından dolayı, müşteriye iletilen teslim tarihlerindeki gecikmelerin azaltılarak müşteri memnuniyetinde artış sağlayabilecektir. Ayrıca firma bu model ile fazla mesai ve vardiya planlarını daha doğru yaparak iş gücünün daha verimli bir şekilde kullanılmasını sağlayabilecektir.

Modeller karşılaştırıldığında gradyan artırma ve destek vektör regresyonu modellerinin üretim süresi tahminlemesinde başarılı sonuçlar verdiği ve uygulanabilir olduğu görülmüştür. Fakat uygulama bölümünde de özellikle ele alındığı gibi, modellerin en başarılı sonucu vermesini sağlayan optimal hiperparametrelerin seçilmesi adımı, aday hiperparametrelerin belirlenmesi üzerine literatürde genel geçer kurallar bulunmamaktadır. Bu sebeple aday hiperparametre sayısını artırmanın, yani optimal hiperparametrenin

aranacağı uzayın genişletilmesi, modellerin çalışma süresini uzatacak olmasına rağmen modelin tahmin performansını artırma ihtimalini de beraberinde getirdiği değerlendirilmektedir.

Gelecek çalışmalarda girdi çeşitliliğinin artırılması ve optimum hiperparametrelerin aranacağı uzayın genişletilmesi ile çalışmanın daha başarılı sonuçlar verebileceği öngörülmektedir. Ek olarak gelecek çalışmalarda yapay sinir ağları, destek vektör regresyonu ve gradyan artıma modelleri dışında da makine öğrenmesi algoritmaları çalıştırılabilir ve karar destek sistemi uygulanarak çalışma zenginleştirilebilir.





KAYNAKLAR

1. Altın, S.Ş. (2011). Benzer Süreçlerde Üretilen Ürünler için Yapay Zeka ile Zaman Tahmini. Başkent Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 44-48.
2. Enns, S.T. (1995). A Dynamic Forecasting Model for Job Shop Flowtime Prediction and Tardiness Control. *International Journal of Production Research*, 33(5), 1295-1312.
3. Raddon, A. and Grigsby, B. (1997, September). *Throughput Time Forecasting Model*. Paper presented at the IEEE/SEMI Conference and Workshop on Advanced Semiconductor Manufacturing, Cambridge, MA.
4. Şeker, A., Diri, B. ve Balık, H.H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
5. Janakiram, M., Backus, P., Movafagh, S. and Runger, G. (2003, October). *Data Mining for Cycle Time Prediction*. Paper presented at the IEEE Transactions on Circuits and Systems II, INFORMS Annual Conference, Atlanta, GA.
6. Janakiram, M. and Mowzoon, S. (2006). Factory Cycle-Time Prediction with Data-Mining Approach. *IEEE Transactions on Semiconductor Manufacturing*, 19(2), 252-258
7. Öztürk, A., Kayalığil, S. and Özdemirel, N.E. (2006). Manufacturing Lead Time Estimation Using Data Mining. *European Journal of Operational Research*, 173(2), 683-700.
8. Alenezi, A., Moses, S. A. and Trafalis, T. B. (2007). Real-time prediction of order flowtimes using support vector regression. *Computers & Operations Research*, 35(11), 3489-3503.
9. Lim, Z., Yusof, U. and Shamsudin, H. (2019, November). *Manufacturing Lead Time Classification Using Support Vector Machine*. Paper presented at the 6th International Visual Informatics Conference, Malaysia.
10. Meidan, Y., Lerner B., Hassoun M. and Rabinowitz G. (2009). Data Mining for Cycle Time Key Factor Identification and Prediction in Semiconductor Manufacturing. *IFAC Proceedings Volumes*, 42(4), 217-222.
11. Meidan, Y., Lerner B., Hassoun M. and Rabinowitz G. (2011). Cycle-Time Key Factor Identification and Prediction in Semiconductor Manufacturing Using Machine Learning and Data Mining. *IEEE Transactions on Semiconductor Manufacturing*, 24(2), 237-248.
12. Liao, D. and Wang, C. N. (2004). Neural-Network-Based Delivery Time Estimates for Prioritized 300-mm Automatic Material Handling Operations. *IEEE Transactions on Semiconductor Manufacturing*, 17(3), 324-332.

13. Bilekdemir, G. (2010). Veri Madenciliği Tekniklerini Kullanarak Üretim Süresi Tahmini ve Bir Uygulama. Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İzmir, 84-89.
14. Kurnaz, G. (2019). Kablo Takımı Üretim Süresinin ve Kusurlu Ürün Oluşumuna Yönelik Risk Faktörlerinin Makine Öğrenmesi Algoritmaları ile Belirlenmesi. Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Samsun, 200-220.
15. Zhang Y. and Haghani A. (2015). A Gradient Boosting Method to Improve Travel Time Prediction. *Transportation Research Part C: Emerging Technologies*, 58(2), 308-324.
16. Kuo, Y., Chan, N., Leung, J., Meng, H., So, A., Tsoi and K., Graham. C. (2020). An Integrated Approach of Machine Learning and Systems Thinking for Waiting Time Prediction in an Emergency Department. *International Journal of Medical Informatics*, 139, 104-143.
17. Gyulai, D., Pfeiffer, A., Bergmann, J. and Gallina, V. (2018). Online Lead Time Prediction Supporting Situation-Aware Production Control. *Procedia CIRP*, 78, 190-195.
18. Var, T. ve Türkay, B.E. (2014, Kasım). *Yapay Sinir Ağları Kullanılarak Kısa Dönem Elektrik Yükü Tahmini*. Eleco 2014 Elektrik – Elektronik – Bilgisayar ve Biyomedikal Mühendisliği Sempozyumunda sunuldu, Bursa.
19. Kartal, E. (2015). Siniflandırmaya Dayalı Makine Öğrenmesi Teknikleri Ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama. İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul, 130-133.
20. Sarı, M. (2016). Yapay Sinir Ağları Ve Bir Otomotiv Firmasında Satış Talep Tahmini Uygulaması. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sakarya, 99-101.
21. İmamoğlu, S. ve İnce H. (2016). Destek Vektör Regresyon ve İkiz Destek Vektör Regresyon Yöntemi ile Tedarikçi Seçimi. *Doğuş Üniversitesi Dergisi*, 2(2), 241-253.
22. Yang, S., Wu, J., Du, Y., He, Y. and Chen, X. (2017). Ensemble Learning for Short-Term Traffic Prediction Based on Gradient Boosting Machine. *Journal of Sensors*. 10, 1-15.
23. Bardak, E.S., Aydemir, D. ve Bardak, S. (2018). *Yapay Sinir Ağları ve Derin Öğrenme Algoritmaları Kullanarak Nanokompozitlerde Deformasyonun Tahmin Edilmesi*. *Bartın Orman Fakültesi Dergisi*, 20(2), 223-231.
24. Kurt, R., Karayılmazlar, S., Imren, E. ve Çabuk, Y. (2017). Yapay Sinir Ağları İle Öngörü Modellemesi: Türkiye Kâğıt-Karton Sanayi Örneği. *Bartın Orman Fakültesi Dergisi*, 19(2), 99-106.
25. Yiğit, İ.O., Çiftçi, S., Kalyoncu, F. ve Kaya, T. (2018, May). *Makine Öğrenmesi Yöntemleriyle Mobil Servis Deneyimi Tahmini*. Paper presented at the 26th Signal Processing and Communications Applications Conference, İzmir.

26. Touzani, S., Granderson, J. and Fernandes, S. (2018). Gradient Boosting Machine for Modeling the Energy Consumption of Commercial Buildings. *Energy and Buildings*, 158, 1533-1543.
27. Zhou, J., Li, E., Yang, S., Wang, M., Shi, X., Yao, S. and Mitri, H.S. (2019). Slope Stability Prediction for Circular Mode Failure Using Gradient Boosting Machine Approach Based on an Updated Database of Case Histories. *Safety Science*, 118, 505-518.
28. Öztemel, E. (2003). *Yapay Sinir Ağları*. Istanbul: PapatyaYayincilik, 40-44.
29. Hajek, M. (2005). *Neural Networks*. South Africa: University of KwaZulu-Natal, 114.
30. İnternet: Parikh, D. (2018). Neural Networks. URL: <https://medium.com/coinmonks/neural-networks-bb11fb9a8266>, (Son Erişim Tarihi: 11.05.2021).
31. Jha, G. K. (2007). *Artificial Neural Networks*. Indian Agricultural Research Inst., PUSA, New Delhi-110 012.
32. Yılmaz, Ş.K. (2014). Doğrusal Vektör Kuantizasyon Modeli Kullanılarak Yapay Sinir Ağlarıyla Kontrol Şemalarında Örüntü Tanıma: Hazır Beton Üreten Bir İşletmede Uygulama. *Akademik Bakış Dergisi*, 60, 431-451.
33. Kartalopoulos, S.V. and Kartakopoulos, S.V. (1997). *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*. New York: Wiley-IEEE Press, 232.
34. Hamid, S.A. and Iqbal, Z. (2004). Using Neural Networks for Forecasting Volatility of S&P 500 Index Futures Prices. *Journal of Business Research*, 57(10), 1116-1125.
35. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Ontario: Prentice-Hall, 830-837.
36. Kohonen, T. (1984). *Self-Organizing and Associate Memory*. Berlin: Springer, 301.
37. Nielsen, R.H. (1990). *Neurocomputing*. MA: Addison-Wesley, Reading, 415-433.
38. Buntine, W.L. and Weigend, A.S. (1994). Computing Second Derivatives in Feed-Forward Networks: A Review. *IEEE Transactions on Neural Networks*, 5(3), 480-488.
39. Hopfield, J.J. (1984). Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons. *Proceedings of the National Academy of Sciences*, 81(10), 3088-3089.
40. Hinton, G.E. and Sejnowski, T.J. (1983). Optimal Perceptual Inference. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, June, Washington DC, 448-453.
41. İnternet: Hinton, G.E. (2007). Boltzmann Machine. URL: http://var.scholarpedia.org/article/Boltzmann_machine, Son Erişim Tarihi: 11.05.2021.

42. İnternet: Boltzmann Makinesi. URL: <https://veribilimcisi.com/2017/09/26/boltzmann-makinesi-boltzmann-machine>, Son Erişim Tarihi: 11.05.2021.
43. Sevgi, P. (2012). Hazır Giyim Perakendeciliği Yapan Bir Firmada Yapay Sinir Ağları İle Satış Tahmini. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 130-133.
44. Khristodulo, O.I., Makhmutov, V.V. and Sazonova, T.V. (2017). Use Algorithm Based at Hamming Neural Network Method for Natural Objects Classification. *Procedia Computer Science*, 103, 388-395.
45. Bow, S.T. (2002). *Pattern Recognition and Image Preprocessing*. Illionis: CRC Press, 720.
46. Vapnik V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer, 300-314.
47. Smola A.J. and Schölkopf, B. (2002). *Learning with Kernels*. Cambridge: MIT Press, 626.
48. Smola, A. J. and Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(1), 199-222.
49. Acı, M., Avcı, M. ve Acı, Ç. (2017). Destek Vektör Regresyonu Yöntemiyle Karbon Nanotüp Benzetim Süresinin Kısaltılması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 32(3), 901-907.
50. Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine learning*, 20(3), 273-297.
51. Gunn, S.R. (1998). Support Vector Machines for Classification and Regression. University of Southampton, Technical Report, 60-66.
52. Huber, P.J. (1964). Robust Estimation of a Location Parameter. *The Annals 52 of Mathematical Statistics*, 35(1), 73-101.
53. Elattar, E.E., Goulernas, J. and Wu, Q.H., (2010). Electric Load Forecasting Based on Locally Weighted Support Vector Regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(4), 438-447.
54. Schapire, R.E. (1989, November). *The Strength of Weak Learnability*. Paper presented at the 30th Annual Symposium on Foundations of Computer Science, Triangle Park, NC.
55. Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, 121(2), 256-285.
56. Freund, Y. and Schapire, R.E. (1996, July). *Experiments with a New Boosting Algorithm*. Paper presented at the Thirteenth International Conference on International Conference on Machine Learning, Italy.

57. İnternet: Arora, U. (2016). Gradient Boosting Part1–Visual Conceptualization. URL: <https://dimensionless.in/gradient-boosting>, Son Erişim Tarihi: 11.05.2021.
58. Jiang, W. (2002). On Weak Base Hypotheses and Their Implications for Boosting Regression and Classification. *The Annals of Statistics*, 30(1), 51-73.
59. Friedman, J., Hastie, T. and Tibshirani, R. (2001). *The Elements of Statistical Learning*. New York: Springer Series in Statistics, 730-764.
60. Lewis, C. D. (1982). *Industrial and Business Forecasting Methods : A Practical Guide to Exponential Smoothing and Curve Fitting*. London: Butterworth Scientific, 143.
61. Cherkassky, V. and Ma, Y. (2004). Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. *Neural Networks : The Official Journal Of The International Neural Network Society*, 17(1), 113-26.
62. Lu, C.J., Lee,T.S. and Chiu, C.C. (2009). Financial Time Series Forecasting Using Independent Component Analysis And Support Vector Regression. *Decision support systems*, 47(2), 115-125.
63. Katagiri, S. and Abe, S. (2006). Incremental Training Of Support Vector Machines Using Hyperspheres. *Pattern Recognition Letters*, 27(13), 1495-1507.
64. İnternet: Kumar, A.M. (2018). *C and Gamma in SVM*. URL: <https://medium.com/@myselfaman12345>, Son Erişim Tarihi: 15.01.2021.





EK-1. (devam) Python ortamında kurulan yapay sinir ağıları modeli kodları

```

        random_state=42)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train_scaled=scaler.transform(X_train)
X_test_scaled=scaler.transform(X_test)

from sklearn.neural_network import MLPRegressor
mlp_model=MLPRegressor(hidden_layer_sizes=(100)).fit(X_train_scaled,y_train)
y_pred = mlp_model.predict(X_test_scaled)
np.sqrt(mean_squared_error(y_test, y_pred))

mlp_params={'hidden_layer_sizes':[(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(9,9),(10,9),(11,9),
(12,9),(10,10),(11,10),(12,10),(11,11),(11,12),(12,12)], 'activation':
['relu','sigmoid']}mlp_cv_model=GridSearchCV(mlp_model,mlp_params,cv=10,n_jobs=
-1,verbose = 2)

mlp_cv_model.best_params_

mlp_tuned=MLPRegressor(hidden_layer_sizes=(12))
mlp_tuned.fit(X_train_scaled,y_train)
y_pred=mlp_tuned.predict(X_test_scaled)
np.sqrt(mean_squared_error(y_test,y_pred))

def mape (y_test,y_pred):
    y_test,y_train=np.array(y_test),np.array(y_pred)
    return np.mean(np.abs((y_test-y_pred)/y_test))*100

mape (y_test,y_pred)

mean_absolute_error(y_test,y_pred)

[coef.shape for coef in mlp_tuned.coefs_]

r2_score(y_test,y_pred)

K_T=pd.DataFrame({"Gerçek Süreler": y_test,"Tahmini Süreler": y_pred})

K_T=K_T.sort_index()

K_T.plot();

```

EK-2. Python ortamında kurulan destek vektör regresyonu modeli kodları

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV,cross_val_score

from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error

import matplotlib.pyplot as plt

from sklearn.preprocessing import scale

from sklearn import model_selection

from warnings import filterwarnings

filterwarnings('ignore')

hit = pd.read_excel("971_sonn.XLSX")

df = hit.copy()

df = df.dropna()

dms = pd.get_dummies(df[['Proje']])

X_ = df.drop(['Proje','Parça No',"Toplam Saat"], axis=1).astype('float64')

y=df["Toplam Saat"]

from sklearn import preprocessing

x_n=preprocessing.MinMaxScaler(feature_range=(0.1,0.9))

x_n=x_n.fit_transform(X_)

x_n_f=pd.DataFrame(x_n,columns=['Proje Yılı','Tecrübe','En','Kalınlık','Boy'])

X=pd.concat([x_n_f,dms[['Proje_Y1','Proje_Y2','Proje_Y3']]],axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y,
```

test_size=0.25,

EK-2. (devam) Python ortamında kurulan destek vektör regresyonu modeli kodları

```

                                random_state=42)

from sklearn.svm import SVR

svr_rbf=SVR("rbf").fit(X_train,y_train)

y_pred=svr_rbf.predict(X_test)

np.sqrt(mean_squared_error(y_test,y_pred))

svr_params={"C":[2,4,16,32],"gamma":[2,4,16,32]}

svr_cv_model=GridSearchCV(svr_rbf,svr_params,cv=10,n_jobs=-1,verbose = 2)

svr_cv_model.fit(X_train,y_train)

pd.Series(svr_cv_model.best_params_)

y_pred=svr_tuned.predict(X_test)

np.sqrt(mean_squared_error(y_test,y_pred))

def mape (y_test, y_pred):

    y_test, y_train=np.array(y_test), np.array(y_pred)

    return np.mean(np.abs((y_test - y_pred)/ y_test))*100

mape (y_test,y_pred)

mean_absolute_error(y_test,y_pred)

r2_score(y_test,y_pred)

K_T=pd.DataFrame({"Gerçek Süreler": y_test,"Tahmini Süreler": y_pred})

K_T=K_T.sort_index()

K_T.plot();

```

EK-3. Python ortamında kurulan gradyan artırma makinesi modeli kodları

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV,cross_val_score

from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error

import matplotlib.pyplot as plt

from sklearn.preprocessing import scale

from warnings import filterwarnings

filterwarnings('ignore')

hit = pd.read_excel("971_sonn.XLSX")

df = hit.copy()

df = df.dropna()

dms = pd.get_dummies(df[['Proje']])

X_ = df.drop(['Proje','Parça No',"Toplam Saat"], axis=1).astype('float64')

y=df["Toplam Saat"]

from sklearn import preprocessing

x_n=preprocessing.MinMaxScaler(feature_range=(0.1,0.9))

x_n=x_n.fit_transform(X_)

x_n_f=pd.DataFrame(x_n,columns=['Proje Yılı','Tecrübe','En','Kalınlık','Boy'])

X=pd.concat([x_n_f,dms[['Proje_Z1','Proje_Z2','Proje_Z3']],axis=1)

X.head()

X_train, X_test, y_train, y_test = train_test_split(X, y,
```

EK-3. (devam) Python ortamında kurulan gradyan artırma makinesi modeli kodları

```
random_state=42)

from sklearn.ensemble import GradientBoostingRegressor

gbm_model = GradientBoostingRegressor()

gbm_model.fit(X_train, y_train)

y_pred = gbm_model.predict(X_test)

np.sqrt(mean_squared_error(y_test, y_pred))

def mape (y_test, y_pred):

    y_test, y_train=np.array(y_test), np.array(y_pred)

    return np.mean(np.abs((y_test - y_pred)/ y_test))*100

mape (y_test,y_pred)

gbm_params =
{ 'learning_rate': [0.01, 0.1,0.2,0.5],

  'max_depth': [2,4,8],

  'n_estimators': [100,500,1000],

  'subsample': [0.5,0.75,1],}

gbm = GradientBoostingRegressor()

gbm_cv_model = GridSearchCV(gbm, gbm_params, cv = 10, n_jobs = -1, verbose = 2)

gbm_cv_model.fit(X_train, y_train)

gbm_cv_model.best_params_

gbm_tuned = GradientBoostingRegressor(learning_rate = 0.1,

                                     max_depth = 4,
```

EK-3. (devam) Python ortamında kurulan gradyan artırma makinesi modeli kodları

```
n_estimators = 500,  
  
    subsample = 0.75)  
  
gbm_tuned = gbm_tuned.fit(X_train,y_train)  
  
y_pred = gbm_tuned.predict(X_test)  
  
np.sqrt(mean_squared_error(y_test, y_pred))  
  
def mape (y_test,y_pred):  
  
    y_test,y_train=np.array(y_test),np.array(y_pred)  
  
    return np.mean(np.abs((y_test-y_pred)/y_test))*100  
  
mape (y_test,y_pred)  
  
mean_absolute_error(y_test,y_pred)  
  
r2_score(y_test,y_pred)  
  
K_T=pd.DataFrame({"Gerçek Süreler": y_test,"Tahmini Süreler": y_pred})  
  
K_T=K_T.sort_index()  
  
K_T.plot();
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : YÜCE, Tuğçe
Uyruğu : T.C.

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Yönetim Bilişim Sistemleri	Devam ediyor
Lisans	Gazi Üniversitesi / Endüstri Mühendisliği	2016
Lise	Gölbaşı Anadolu Lisesi	2011

İş Deneyimi

Yıl	Yer	Görev
2018-Halen	Türk Havacılık ve Uzay Sanayi	Kurumsal Kaynak Yönetimi Mühendisi
2018-2018	Türk Traktör	Satınalma Mühendisi
2017-2018	Eczacıbaşı Vitra Karo	Satınalma Uzman Yardımcısı

Yabancı Dil

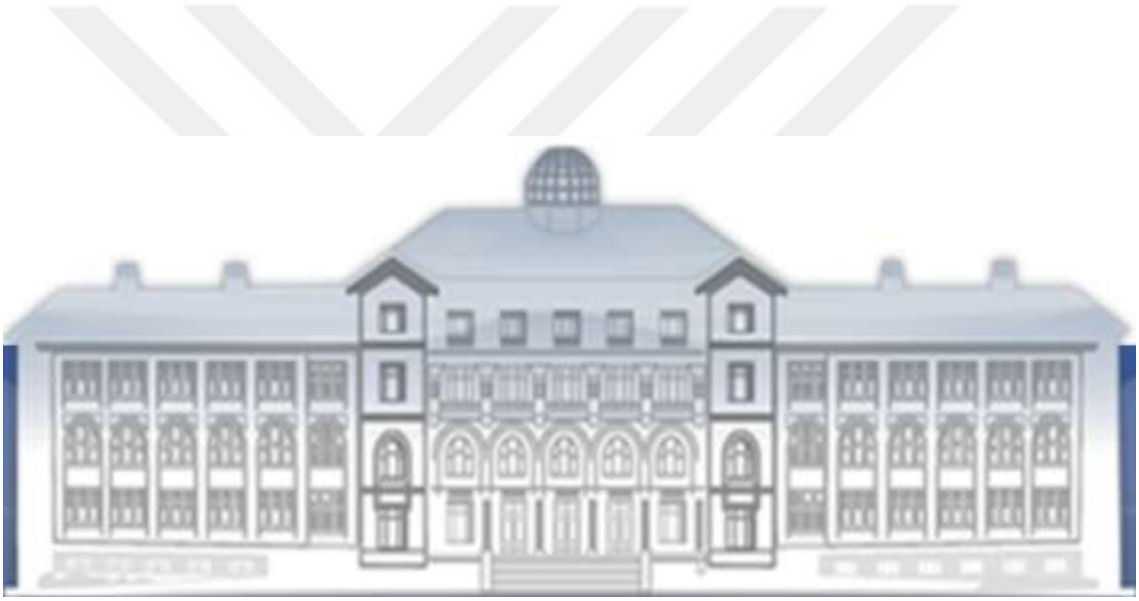
İngilizce

Yayınlar

Yüce, T., Kabak, M. (2021). Makine Öğrenmesi Algoritmaları ile Detay Üretim Alanları İçin İş Merkezi Kırılımında Üretim Süresi Tahminleme. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 37(1), 47-60.

Hobiler

Voleybol oynamak, seyahat etmek



GAZİLİ OLMAK AYRICALIKTIR...