

**SİNİRSEL AĞLARI KULLANARAK PROJELERDE
EFOR TAHMİNİ**

Burcu ŞENGÜNEŞ



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SİNİRSEL AĞLARI KULLANARAK PROJELERDE EFOR TAHMİNİ

Burcu ŞENGÜNEŞ
0000-0002-4737-8671

Prof. Dr. Nursel ÖZTÜRK
0000-0002-9835-0783
(Danışman)

DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2023
Her Hakkı Saklıdır

TEZ ONAYI

Burcu ŐENGÜNEŐ tarafından hazırlanan ‘‘SİNİRSEL AĖLARI KULLANARAK PROJELERDE EFOR TAHMİNİ’’ adlı tez alıŐması aŐaĖıdaki jüri tarafından oy birliĖi ile Bursa UludaĖ Üniversitesi Fen Bilimleri Enstitüsü Endüstri MühendisliĖi Anabilim Dalı’nda **DOKTORA TEZİ** olarak kabul edilmiŐtir.

DanıŐman: Prof. Dr. Nursel ÖZTÜRK

- Başkan** : Prof. Dr. Nursel ÖZTÜRK İmza
0000-0002-9835-0783
Bursa UludaĖ Üniversitesi, Mühendislik
Fakültesi,
Endüstri MühendisliĖi Anabilim Dalı
- Üye** : Do. Dr. Aslı AKSOY İmza
0000-0002-2971-2701
Bursa UludaĖ Üniversitesi, Mühendislik
Fakültesi,
Endüstri MühendisliĖi Anabilim Dalı
- Üye** : Do. Dr. Göknur Arzu AKYÜZ İmza
0000-0003-2024-5884
Türk Hava Kurumu Üniversitesi, İŐletme Fakültesi,
Lojistik Yönetimi Anabilim Dalı
- Üye** : Do. Dr. Ayta YILDIZ İmza
0000-0002-0729-633X
Bursa Teknik Üniversitesi, Mühendislik ve DoĖa Bilimleri
Fakültesi,
Endüstri MühendisliĖi Anabilim Dalı
- Üye** : Dr. Öğr. Üyesi Emre BULUT İmza
0000-0001-9159-5000
Bursa UludaĖ Üniversitesi, Mühendislik Fakültesi,
Otomotiv MühendisliĖi Anabilim Dalı

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Aksel EREN
Enstitü Müdürü

.././.....

B.U.Ü. Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

23/02/2023

Burcu ŞENGÜNEŞ

TEZ YAYINLANMA FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayımlanan “**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

Prof. Dr. Nursel Öztürk
20/01/2023

Burcu Şengüneş
20/01/2023

İmza

Bu bölüme kişinin kendi el yazısı ile okudum
anladım yazmalı ve imzalanmalıdır.

İmza

Bu bölüme kişinin kendi el yazısı ile okudum
anladım yazmalı ve imzalanmalıdır.

ÖZET

Doktora Tezi

SİNİRSEL AĞLARI KULLANARAK PROJELERDE EFOR TAHMİNİ

Burcu ŞENGÜNEŞ

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Nursel ÖZTÜRK

Proje efor tahmini, proje yönetiminde kritik öneme sahiptir. Proje için gereken eforun tahmin edilmesi, özellikle projenin erken aşamalarında belirsizlik seviyesinin yüksek olmasından dolayı oldukça zordur. Bu tez çalışmasının amacı, fotonik sektörde müşteriye özel makinelerin geliştirildiği projeler için efor tahmininde kullanılacak bir sistemin geliştirilmesidir. Bu çalışma kapsamında, iki ayrı yapay sinir ağı (YSA) modeli önerilmiştir. Birinci model, makine geliştirme projesinin otomasyon aşaması için gereken eforu tahminlemek üzere geliştirilmiştir. Tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşan makine geliştirme projelerinde, otomasyon aşaması yüksek belirsizlik içermektedir. Müşterinin üretimini manuel olarak gerçekleştirdiği fotonik ürünün üretim sürecinin, bir makine tarafından otomatik bir şekilde gerçekleştirilmesi için gereken süreç gereksinimlerini belirlemek oldukça zordur. Bu belirsizlik ortamında projenin otomasyon aşaması için gereken eforun tahmin edilebilmesi önem arz etmektedir. İkinci YSA modeli ise makine geliştirme projesi için gereken toplam proje eforunu tahminlemek üzere önerilmiştir. Geliştirilen iki model, gereken eforu proje ve makine karakteristiklerinden yola çıkarak tahmin etmektedir. YSA modelleri, 11 adet gerçek hayat makine geliştirme projesi ile test edilmiştir. Tahmin doğruluğunu ölçümlemek için PRED(%25) değeri kullanılmış olup, bu değer otomasyon eforu ve toplam proje eforu tahmini modelleri için sırası ile %73 ve %91 olarak hesaplanmıştır. Elde edilen veriler göz önünde bulundurulduğunda geliştirilen modeller ile umut verici sonuçlar elde edildiği sonucuna varılmıştır. Ayrıca proje yöneticileri için bu sistemin kullanımını kolaylaştırmak amacıyla YSA tabanlı bir karar destek sistemi geliştirilmiştir.

Anahtar Kelimeler: Yapay sinir ağları, Proje efor tahmini, Bayes optimizasyonu, Özelleştirilmiş makine geliştirme

2023, ix + 111 sayfa.

ABSTRACT

PhD Thesis

PROJECT EFFORT ESTIMATION USING NEURAL NETWORKS

Burcu ŞENGÜNEŞ

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Prof. Dr. Nursel ÖZTÜRK

Estimating project effort is a crucial concern for project managers. Estimation of required work is challenging, especially in the early phases of a project, due to the high level of uncertainty and lack of experience. This study aims to develop a system for estimating project effort for customized machine development in the photonics industry. This study proposes two artificial neural networks (ANN) models, one for estimating the effort of the automation phase and the other for estimating the total project effort. Design, procurement, production, automation, and testing are the phases of machine development projects, and the automation phase is highly unpredictable. Automating a machine includes uncertainty since customers make photonic products manually, and the process is not always clear at the beginning. In such a high-uncertainty environment, estimating how much effort is required to automate a machine is crucial. The second model was developed for estimating the total project effort. The proposed models were tested on 11 real-life machine development projects. The accuracy measure, PRED(%25), showed promising results with 73% and 91% for automation effort and total project effort, respectively. Additionally, an ANN-based decision support system has been developed to make this system easier for project managers.

Key words: Artificial neural network, Project effort estimation, Bayesian optimization, Customized machine development

2023, ix + 111 pages.

ÖNSÖZ ve TEŞEKKÜR

Doktora tez sürecimde bana yardımlarını esirgememiş ve teşvik etmiş kişilere olan teşekkürlerimi sunmak istiyorum. Hayatımda iş ve ülke değişikliği gibi önemli değişikliklerin de yer aldığı bu uzun doktora süreci sonucunda ciddi bir emek ve özveri ile hazırlamış olduğum doktora tezimi tamamlamanın heyecanını ve gururunu yaşıyorum.

Öncelikle doktora tez çalışması süreci boyunca desteğini esirgemeyen ve her daim beni cesaretlendiren değerli danışmanım Prof. Dr. Nursel ÖZTÜRK'e sonsuz teşekkürlerimi sunarım.

Bu çalışmanın veri toplama ve analizi süreçlerinin her bir aşamasında büyük emeği geçen eşim Mehmet İnanç ŞENGÜNEŞ'e ve model konseptinin oluşturulmasında yardımlarını esirgemeyen değerli eski iş arkadaşım Matthieu STOECKEL'a teşekkürlerimi sunarım.

Ayrıca tez jürimde bulunan ve çalışmalarımı değerlendirerek önerilerini paylaşan ve desteklerini esirgemeyen Doç. Dr. Göknur Arzu AKYÜZ ve Doç. Dr. Aslı AKSOY hocalarıma teşekkürlerimi sunarım.

Son olarak, hayatımın her aşamasında olduğu gibi doktora tez çalışmamda da beni destekleyen ve bana olan güvenlerini hiç kaybetmeyen annem Dr. Ecz. Nezahat KAYA, babam Dr. Abdullah KAYA ve kardeşim Ecz. Duygu KAYA'ya teşekkür ederim.

Burcu ŞENGÜNEŞ
20/01/2023

İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ ve TEŞEKKÜR.....	iii
SİMGELER VE KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ.....	1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI.....	5
2.1. Proje Yönetiminin Tarihçesi.....	5
2.2. Proje Kavramı.....	6
2.3. Proje Yaşam Döngüsü ve Aşamaları.....	7
2.4. Proje Yönetimi.....	10
2.5. Proje Yönetim Süreç Grupları.....	12
2.6. Proje Yönetim Bilgi Alanları.....	14
2.7. Proje Efor Tahmini ve Önemi.....	16
2.8. Proje Efor Tahmini için Kullanılan Faktörler.....	17
2.9. Proje Efor Tahmini için Kullanılan Yöntemler.....	19
2.10. Yapay Sinir Ağları.....	21
2.11. Yapay Sinir Ağları için Öğrenme Algoritmaları.....	23
2.12. Yapay Sinir Ağı Hiperparametreleri.....	29
2.13. Yapay Sinir Ağı için Hiperparametre Belirleme Yöntemleri.....	30
2.14. Bayes Optimizasyonu.....	33
2.15. Proje Efor Tahmininde Kullanılan Performans Değerlendirme Kriterleri.....	38
3. MATERYAL ve YÖNTEM.....	41
3.1. Kullanılan Veri Seti.....	41
3.2. Önerilen Tahmin Modelleri için Girdi Seçimi.....	46
3.3. Veri Ön İşleme.....	53
3.4. Önerilen YSA Modelinin Mimarisi.....	55
3.5. YSA Modeli Geliştirme Prosedürü.....	55
4. BULGULAR ve TARTIŞMA.....	65
4.1. Otomasyon Efor Tahmini Modeli için Yapılan Deneyler.....	65
4.2. Otomasyon Efor Tahmini Modeli için En iyi Ağ Yapısı ve Performansı.....	70
4.3. Toplam Proje Eforu Tahmini Modeli için Yapılan Deneyler.....	73
4.4. Toplam Proje Eforu Tahmini Modeli için En iyi Ağ Yapısı ve Performansı.....	78
4.5. Geliştirilen Yapay Sinir Ağlarını Temel Alan Efor Tahminleme Sistemi.....	82
5. SONUÇ.....	87
KAYNAKLAR.....	91
EKLER.....	97
EK 1 Girdi Seçimi için MATLAB Kodu.....	98
EK 2 Veri Ön İşleme için MATLAB Kodu.....	99
EK 3 BO-1 MATLAB Kodu.....	100
EK 4 BO-2 MATLAB Kodu.....	101
EK 5 BO-3 MATLAB Kodu.....	102
EK 6 Model 1 için Seçilen Hiperparametreler ile İleri Denemeler ve Test MATLAB Kodu.....	103

EK 7 BO-1 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler	106
EK 8 BO-2 Sonuçlarının Adımlar Halinde Listelenmesi.....	107
EK 9 BO-3 Sonuçlarının Adımlar Halinde Listelenmesi.....	108
EK 10 BO-2 ve BO-3 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler.....	109
ÖZGEÇMİŞ	111

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklama
α	Edinim fonksiyonu
b	<i>Bias</i> değeri
d	Hedef çıktı
ϵ	YSA toplam hata değeri
$e_j(n)$	j nöronuna ait n . iterasyondaki hata sinyali
δ	Lokal gradyan
n	İterasyon sayısı
η	Öğrenme oranı
$\varphi(\cdot)$	Aktivasyon fonksiyonu
u_k	Birleştirme fonksiyonu ile k . nöronda elde edilen net girdi
v_k	k . nöronda üretilen net çıktı
w	Nöronlar arası bağlantı ağırlıkları
$y_j(n)$	n . iterasyondaki bir j çıktı nöronu tarafından üretilen çıktı sinyali
Δw_{ji}	Ağırlık düzeltilmesi

Kısaltmalar	Açıklama
ACO	Karınca Koloni Optimizasyonu (Ant Colony Optimization)
BFG	Broyden-Fletcher-Goldfarb-Shanno Algoritması
BO	Bayes Optimizasyonu (Bayesian Optimization)
BP	Geri Yayılım (Backpropagation)
CG	Eşlenik Gradyan Algoritması (Conjugate Gradient)
CGB	Powell-Beale Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Powell-Beale Restarts)
CGF	Fletcher-Reeves Güncellemeleri ile Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Fletcher-Reeves Updates)
CGP	Polak-Ribière Güncellemeleri ile Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Polak-Ribière Updates)
COCOMO	Yapıcı Maliyet Modeli (Constructive Cost Model)
ELM	Aşırı Öğrenme Makineleri (Extreme Learning Machines)
FFNN	İleri Beslemeli Sinirsel Ağlar (Feedforward Neural Network)
FLANN	Fonksiyonel Link Sinirsel Ağları (Functional Link Artificial Neural Network)
GD	Gradyan Azalma (Gradient Descent)
GDX	Değişken Öğrenme Oranlı ve Momentumlu Gradyan Azalma (Gradient Descent with Momentum & Adaptive Backpropagation)
GRNN	Genelleştirilmiş regresyon sinir ağı (Generalized Regression Neural Network)
HPO	Hiperparametre Optimizasyonu (Hyperparameter Optimization)
LM	Levenberg-Marquardt Algoritması
MAPE	Ortalama Mutlak Yüzde Hatası (Mean Absolute Percentage Error)

MLP	Çok Katmanlı Algılayıcı (Multilayer Perceptron)
MMER	Ortalama Tahmine göre Bağlı Hata Büyüklüğü (Mean of Magnitude Error Relative to the Estimate)
MMRE	Ortalama Bağlı Hata Büyüklüğü (Mean Magnitude Relative Error)
NCA	Komşuluk Bileşenleri Analizi (Neighborhood Component Analysis)
OSS	Bir Adım Sekant Geri Yayılım Algoritması (One-step Secant Backpropagation)
PMBOK	Proje Yönetimi Bilgi Birikimi Kılavuzu (Project Management Body of Knowledge)
PMI	Proje Yönetim Enstitüsü (Project Management Institute)
PRED	Kestirim Düzeyi (Prediction Level)
PSO	Parçacık Sürü Optimizasyonu
RBNN	Radyal Temel Fonksiyonu Sinirsel Ağı (Radial Basis Function Neural Network)
RPROP	Esnek Geri Yayılım Algoritması (Resilient Backpropagation)
SCG	Ölçekli Eşlenik Gradyan Algoritması (Scaled Conjugate Gradient)
YSA	Yapay Sinir Ağı (Artificial Neural Network, ANN)

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Proje süreç gruplarına göre maliyet ve personel düzeyinin değişimi.....	8
Şekil 2.2. Proje süresine bağlı olarak risk ve maliyet etkilerinin değişimi.....	9
Şekil 2.3. Proje süresince paydaşların ilgisi, belirsizlik ve yaratıcılık gibi faktörlerin değişimi.....	10
Şekil 2.4. Proje süreç grupları.....	13
Şekil 2.5. Yapay sinir ağında bir nöron gösterimi.....	21
Şekil 2.6. j. çıktı nöronuna ait sinyal akışı.....	24
Şekil 2.7. Izgara arama ve rastgele arama.....	31
Şekil 2.8. BO prosedürü için bir örnek – artarda 3 iterasyon.....	38
Şekil 3.1. Makine geliştirme projesi adımları.....	43
Şekil 3.2. Otomasyon eforu tahmini için belirlenen faktörlere ait NCA ağırlıkları.....	52
Şekil 3.3. Logaritmik transformasyon öncesi ve sonrası otomasyon eforu çıktı değişkeninin veri dağılımı.....	54
Şekil 3.4. Logaritmik transformasyon öncesi ve sonrası toplam proje eforu çıktı değişkeninin veri dağılımı.....	55
Şekil 3.5. YSA geliştirme prosedürü.....	56
Şekil 3.6. Tez kapsamında kullanılan aktivasyon fonksiyonları.....	60
Şekil 4.1. BO-1 sonuç çıktısı (MATLAB)	66
Şekil 4.2. BO-1 denemesinde her bir adımda gözlemlenen amaç fonksiyonu değeri.....	67
Şekil 4.3. Model 1 için seçilen aday modellerin PRED (%25) değerleri.....	69
Şekil 4.4. Proje otomasyon eforu (Model 1) için önerilen YSA yapısı.....	70
Şekil 4.5. Model 1 – YSA eğitim ve doğrulama hatalarının her bir iterasyonundaki değişimi.....	71
Şekil 4.6. Model 1 - YSA eğitim ve doğrulama korelasyonu (R) değeri sonuçları.....	71
Şekil 4.7. Otomasyon eforu tahminleri ve gerçek değerleri.....	72
Şekil 4.8. BO-3 denemesinde her bir adımda gözlemlenen amaç fonksiyonu değeri	76
Şekil 4.9. Model 2 için seçilen aday modellerin MMRE değerleri.....	78
Şekil 4.10. Proje toplam eforu (Model 2) için önerilen YSA yapısı.....	79
Şekil 4.11. Model 2 – YSA eğitim ve doğrulama hatalarının her bir iterasyonundaki değişimi.....	80
Şekil 4.12. Model 2 - YSA eğitim ve doğrulama korelasyonu (R) değeri sonuçları.....	80
Şekil 4.13. Toplam proje eforu tahminleri ve gerçek değerleri.....	81
Şekil 4.14. Otomasyon ve toplam proje eforu tahminleme sisteminin iş-akış diyagramı.....	82
Şekil 4.15. Otomasyon ve toplam proje eforu tahminleme sistemi – eksik giriş uyarıları.....	83
Şekil 4.16. Otomasyon ve toplam proje eforu tahminleme sisteminde yapılan tahminlerin gösterimi.....	86

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1. Proje yönetim süreçleri.....	15
Çizelge 2.2 BO sözde kodu.....	37
Çizelge 3.1. Tahmin modelleri ve çıktı değişkenleri.....	45
Çizelge 3.2. Proje efor tahmini için belirlenen faktörler ve seviyeleri.....	46
Çizelge 3.3. Müşteri ürünü, süreci ve makine konfigürasyonu yeniliği girdileri ile ilgili tanımlamalar.....	48
Çizelge 3.4. Makine ana fonksiyonu girdi değişkeninin aldığı değerlerin tanımları.....	49
Çizelge 3.5. Makine dış kasa büyüklüğü girdisinin düzeyleri ile ilgili tanımlamalar.....	50
Çizelge 3.6. YSA modelleri için belirlenen girdiler ve çıktılar.....	53
Çizelge 3.7. Gizli katman sayısı belirlemek için literatürde önerilen denklemler.....	58
Çizelge 3.8. Model 1 ve Model 2 için literatürdeki denklemler kullanıldığında önerilen değerler.....	58
Çizelge 3.9. BO denemelerinde optimize edilmek üzere kullanılan hiperparametreler ve olası değerleri.....	59
Çizelge 3.10. Koşullu değişkenler içeren BO ile HPO prosedürüne dair sözde kod.....	63
Çizelge 4.1. BO-1 denemesi için seçilen hiperparametreler ve olası değerleri....	65
Çizelge 4.2. BO-1 sonucuna göre ileri denemeler için seçilen YSA modelleri...	67
Çizelge 4.3. Model 1 Testi için Kullanılan Test Veri Seti.....	68
Çizelge 4.4. Model 1 için bazı aday denemeler ve en iyi deneme.....	69
Çizelge 4.5. Model 1 için elde edilen en iyi ağın hata performans değerleri.....	72
Çizelge 4.6. Proje efor tahmini alanında geliştirilen YSA modellerinin hata değerleri.....	73
Çizelge 4.7. BO-2 denemesi için seçilen hiperparametreler ve olası değerleri....	74
Çizelge 4.8. BO-2 sonucuna göre otomasyon eforu tahmini modeli için ileri denemelerde kullanılmak üzere seçilen YSA modelleri.....	74
Çizelge 4.9. BO-3 denemesi için seçilen hiperparametreler ve olası değerleri....	75
Çizelge 4.10. BO-3 sonucuna göre ileri denemeler için seçilen YSA modelleri....	76
Çizelge 4.11. BO-2 ve BO-3 sonucuna göre toplam proje efor tahmini modeli için ileri denemelerde kullanılmak üzere seçilen YSA yapıları.....	77
Çizelge 4.12. Model 2 için bazı aday denemeler ve en iyi deneme.....	78
Çizelge 4.13. Model 2 için elde edilen en iyi ağın hata performans değerleri.....	81
Çizelge 4.14 Otomasyon ve proje eforu tahminleme sistemi örnek bir kullanıcı girişi.....	85

1. GİRİŞ

Proje eforu, bir aktiviteyi tamamlamak için gereken çalışan süresi (*staff time*) olarak tanımlanır (Trendowicz ve Jeffery, 2014). Ortalama bir çalışanın bir saatte gerçekleştirdiği iş miktarı genellikle adam-saat olarak ölçümlenmektedir. Proje efor tahmini, proje çizelgesi ve bütçesinin geliştirilmesi açısından kritik öneme sahiptir. Proje efor tahminindeki başarısızlık, projenin çizelge ve bütçe açısından başarısızlığı ile sonuçlanmaktadır (Usman ve diğerleri, 2018). Örneğin, proje eforunun gerekenden daha az tahmin edilmesi, proje süresince planlanmayan maliyetlerin ortaya çıkmasına ve bütçenin aşılmasına neden olmaktadır. Eforun yetersiz tahmin edilmesi sonucu, projeye gerekenden daha az kaynak atanmaktadır. Bu durum, kaynakların planlanandan daha fazla kullanılmasına ve çalışanların stres düzeyinde artışa neden olmaktadır. Öte yandan, projeye gereğinden fazla efor atanması sonucu süre ve kaynak fazlalığı oluşmakta olup, bu durum ürün kapsamı dışında gerekenden fazla özellikler eklenerek kapsam dışına çıkılmasına neden olmaktadır. Proje kapsamının genişlemesi, planlanan bütçenin aşılmasına neden olabilmektedir. Bunun yanında, bir aktivite üzerinde gerekenden fazla zaman harcayan kaynakların, daha önemli fırsat elde edilebilecek olan diğer projelerde bu süre zarfında kullanılamaması fırsat kaybına neden olabilmektedir (Hameed ve diğerleri, 2023) ve bu durum ilerleyen projelerin gecikmesine yol açabilmektedir. Kısaca, proje efor tahminindeki başarısızlık, çalışan memnuniyetsizliği, ürün kalitesinde düşüklük ve nihayetinde müşteri memnuniyetinde azalma gibi birtakım ciddi sorunlara sebep olmaktadır.

Belirsizliğin oldukça yüksek olduğu projenin ilk aşamalarında, gerçeğe yakın tahminler yapmak oldukça zordur (Carvalho ve diğerleri, 2021). Karar vericilerin belirsizlik ortamında genellikle iyimser tahminler yapmaya eğilimlerinin olduğu (Prater ve diğerleri, 2016) ve iyimser tahminlerin en önemli proje başarısızlık sebeplerinden biri olduğu düşünülmektedir (Nassif ve diğerleri, 2013). Diğer firmalarla rekabet edebilme açısından da efor tahmininde gerçeği ortaya koymak oldukça zordur (Jørgensen ve Sjøberg, 2014).

Literatürde proje efor tahmini ile ilgili çeşitli yöntemler öneren birçok çalışmaya rastlanmış olup, bu çalışmaların çoğunlukla yazılım projeleri üzerine odaklandığı

gözlemlenmiştir. Proje efor tahmini ile ilgili önerilen yöntemler üç kategori altında toplanmaktadır: uzman görüşü tabanlı yöntemler, algoritmik modeller ve makine öğrenmesi yöntemleri (Tronto ve diğerleri, 2008). Uzman görüşü tabanlı yöntemlerde değerlendirme uzman kişinin benzer projelerdeki tecrübesine bağlı olup, öznellik içermektedir (López-Martín ve diğerleri, 2011). Algoritmik modeller, örneğin doğrusal veya doğrusal olmayan regresyon modelleri, proje efor tahmininde sıklıkla kullanılan yöntemlerdir. Özellikle yazılım proje efor ve maliyet tahmini alanında, yapıcı maliyet modeli (Constructive Cost Model: COCOMO) sıklıkla karşılaşılmış bir algoritmik yöntem olup Boehm ve diğerleri tarafından 1981 yılında önerilmiştir (Arora ve Mishra, 2018). 63 adet yazılım projesinden toplanan verilerle geliştirilmiş olan bu yöntem, kod satırı (Lines of Code: LOC) verisini kullanarak proje eforunu adam-saat cinsinden tahminleyen bir regresyon modelidir (Heiat, 2002).

Son yıllarda, proje efor tahmini ile ilgili yapılan çalışmalarda, makine öğrenmesi gibi daha objektif yöntemlere yönelinmiştir (Monika ve Sangwan, 2017). Bulanık mantık, yapay sinir ağı (YSA), genetik algoritma gibi evrimsel teknikler ve karar ağaçları gibi makine öğrenme teknikleri özellikle yazılım proje efor tahmini alanında sıklıkla kullanılmıştır (López-Martín ve diğerleri, 2011; Nassif ve diğerleri, 2013). Yurt ve diğerleri (2019) ar-ge faaliyetleri için gereken mühendislik eforu ile ilgili geliştirdikleri efor tahmini modeli için karar ağaçları ve k-en yakın komşu algoritmaları kullanmışlardır. Karmaşık problemlerin çözümünde oldukça sıklıkla kullanılan bir makine öğrenmesi tekniği olan YSA modelleri özellikle yazılım efor tahmini ile ilgili birçok tahmin modeli uygulamasında kullanılmıştır (Ali ve Gravino, 2019). Yazılım projeleri efor tahmini ile ilgili yapılan birçok çalışmada YSA modelleri, geleneksel tahmin yöntemlerinden biri olan regresyon ile karşılaştırılmıştır ve regresyona göre daha iyi sonuçlar elde edilmiştir (Heiat, 2002; Tronto ve diğerleri, 2008; Park ve Baek, 2008; Dave ve Dutta, 2011; López-Martín ve diğerleri, 2011).

Son 20 yılda, proje efor tahmini ile ilgili özellikle yazılım sektöründe azımsanmayacak sayıda çalışma yapılmıştır (Pospieszny ve diğerleri, 2018). Yazılım efor tahmini alanında yapılan bu çalışmalarda YSA tabanlı yöntemlere sıklıkla rastlanırken, mühendislik tabanlı ürün geliştirme projeleri için efor tahmini konusunda YSA gibi makine öğrenme

yöntemlerinin kullanımına aynı sıklıkla rastlanamamıştır. Literatürdeki bu boşluğu doldurmayı amaçlayarak bu tez çalışmasında, özelleştirilmiş makine geliştirme projeleri için efor tahmini gerçekleştirilmiştir. Bu kapsamda, Almanya’da fotonik cihazlara yönelik mikro-montaj ve test makineleri geliştiren bir firmada yürütülen makine geliştirme projeleri ele alınmıştır. Geliştirilen makineler, müşterinin fotonik ürünün üretim süreçlerini yarı ya da tam otomatik olarak gerçekleştirilmeye yöneliktir. Makine geliştirme projeleri genellikle tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşmaktadır. Makinenin müşteri ihtiyacına yönelik geliştirildiği bir ortamda belirsizlik oldukça yüksek olup, proje için efor tahmini yapmak oldukça zordur. Özellikle otomasyon aşamasında bu belirsizlik oldukça yüksektir. Otomasyon aşamasında, müşterinin ürünün üretim süreci makine ile otomatize edilmektedir. Eğer müşterinin üretim sürecine yönelik daha önce benzer bir makine geliştirilmiş ve bu konuda deneyim söz konusu ise otomasyon için gereken efor nispeten az olabilmektedir. Müşteri ürünü ve süreci ile ilgili teknolojik olarak belirsizliklerin söz konusu olduğu durumlarda ise gereken efor daha fazla olabilmektedir.

Söz konusu makine geliştirme projelerinde efor tahmini önemli bir konu olup, literatürdeki bu boşluğu doldurma amacıyla bu çalışmada toplam proje eforu ve otomasyon eforu tahmini için iki ayrı YSA modeli geliştirilmiştir. Birinci model, projenin yüksek belirsizlik içeren ve gereken efor açısından öne çıkan aşamalarından biri olan otomasyon aşaması eforunu tahminlemek üzerinedir. İkinci model ise tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşan makine geliştirme projelerinin toplam proje eforunu tahminlemeye yöneliktir. Proje otomasyon eforu ve toplam proje eforu değerlerini tahminlemek üzere, makinenin karmaşıklığı, firmaya olan yeniliği gibi birtakım özellikler ile ilgili toplamda 11 adet girdi konunun uzmanları ile belirlenmiştir. Proje otomasyon eforu modeli için bu girdilerin otomasyon eforu olan bağımlı değişkenine etkileri Komşuluk Bileşenleri Analizi (Neighborhood Component Analysis: NCA) ile incelenmiştir. Bazı girdilerin otomasyon eforlarına etkilerinin olmadığı sonucuna varılmıştır. Bunun sonucunda, toplam 11 adet girdiden 8 adeti otomasyon eforu tahmini için kullanılmıştır.

Bu tez çalışması kapsamında geliştirilen YSA modelinin literatürde daha az sıklıkla karşılaşılan bir sektöre yönelik olması ve yeni girdiler barındırması açısından literatüre katkılar sağladığı düşünülmektedir.

Literatürde proje efor tahmini ile ilgili önerilen diğer YSA modellerinin genellikle deneme yanılma ile elde edildiği gözlemlenmiş olup, bu tez çalışmasında Bayes Optimizasyonu (Bayesian Optimization: BO) kullanılmıştır. BO'nun hiperparametre optimizasyonu (HPO) için kullanıldığı durumda, optimize edilmeye çalışılan değişkenler YSA hiperparametreleri olup amaç fonksiyonu YSA hata performansıdır. YSA hata performansı bir çapraz doğrulama yöntemi kullanılarak doğrulama veri seti üzerinden hesaplanmıştır. BO her bir aşamada, o ana kadarki gerçekleştirilen deneylerin sonuçlarını kullanarak, YSA performansını iyileştirecek yönde yeni hiperparametre değerleri seçmektedir (Snoek ve diğerleri, 2012). Daha karmaşık olan 11 adet girdili proje efor tahmini yapan ikinci model için gerçekleştirilen hiperparametre optimizasyonunda, öğrenme algoritması hiperparametre olarak ele alınmıştır. BO algoritmasına, seçilen öğrenme algoritmasına göre bazı hiperparametreleri aktifleştiren koşullar da katılmıştır. Son olarak, BO algoritması ile elde edilen veriler ışığında umut verici parametreler ile ek denemeler yapılarak en iyi ağlar elde edilmiştir.

Tezin ilerleyen kısımları dört ana başlık altında toplanmıştır. Bölüm 2'de proje yönetim, proje efor tahmini ve YSA ile ilgili kuramsal temeller ve kaynak araştırmasına, Bölüm 3'te toplam proje eforu ve otomasyon eforu tahmini için önerilen YSA modellerini geliştirmek için kullanılan yöntemlere yer verilmiştir. "Bulgular ve Tartışma" bölümünde ise geliştirilen YSA modellerinin yapılarına, performanslarına ve literatürde proje efor tahmini konusunda geliştirilmiş modeller ile karşılaştırılmasına, son olarak Bölüm 5'te ise sonuç kısmına yer verilmiştir.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Bu bölümde, öncelikle proje yönetimi ile ilgili kuramsal temellere yer verilmiştir. İlerleyen alt bölümlerde ise literatürde proje efor tahmini ile ilgili kullanılan faktörler, yöntemler, performans değerlendirme kriterleri ile ilgili kaynak araştırmasına ve son olarak kısaca yapay sınır ağları ve optimizasyonu ile ilgili kuramsal temellere yer verilmiştir.

2.1. Proje Yönetiminin Tarihçesi

1940'larda işletmelerde proje yönetim görevini, bölüm yöneticileri üstlenmişlerdir (Kerzner, 2003). Projede yer alan bir bölüm yöneticisinin, sorumluluğundaki işleri tamamladıktan sonra projeyi, sıradaki görevi gerçekleştirecek başka bir bölümün yöneticisine aktardığı belirtilmektedir. Bu durumda, projenin tümünü sahiplenen bir kişiden ziyade, projeyi kısmi olarak sahiplenen birçok kişi söz konusu olmuştur. II. Dünya Savaşı sırasında, özellikle savunma sanayi ile ilgili projelerde, her projeden belirli bir proje yöneticisinin sorumlu olması tercih edilmiştir ve bahsi geçen proje yönetim tarzı, bu talepleri karşılayamamıştır (Kerzner, 2003). 1960'larda özellikle havacılık ve uzay, savunma ve inşaat sanayide proje yönetimi, proje yöneticisinin otorite düzeyi daha yüksek olacak şekilde uygulanır hale gelmiştir. Proje yönetimi, geleneksel organizasyon şemalarının yeniden yapılandırılmasını gerektirmiştir ve yöneticilerin çoğu, proje yöneticisi olgusunun kendi yetki düzeylerine ve maaşlarına etki gösterip göstermeyeceği konularında endişelere düşmüşlerdir. 1990'larda büyük sermaye harcanan projelerin varlığı, müşteri beklentileri, rekabet koşulları, araştırma ve geliştirme (Ar-Ge) projelerinin çoğaltılması ve ticarileştirilmesi gereklilikleri gibi nedenlerden dolayı, firmalar proje yönetiminin bir seçenek değil, gereklilik olduğunu anlamaya ve sağladığı faydaların farkına varmaya başlamışlardır. 1996'larda proje kapsamında risklerin de göz önünde bulundurulma gereklilikleri ortaya çıkmıştır. O yıllarda proje yönetimi profesyonel kariyer yolu olarak tanınmıştır. Dünya çapında 700,000'den fazla üye sayısına sahip olan Proje Yönetim Enstitüsü (Project Management Institute: PMI) tarafından, 1996 yılında dünyaca kabul gören proje yönetim standartlarını barındıran Proje Yönetim Bilgi Birikimi Kılavuzu (Project Management Body of Knowledge:

PMBOK) yayınlanmıştır. Proje yönetimi ile ilgili birçok yaklaşım ve metodolojinin sunulduğu günümüzde, en kapsamlı çerçeveyi sunan yapı PMI olarak bilinmektedir. PMI'a göre proje yönetimi, proje ihtiyaçlarını karşılamak üzere proje yönetimi alanındaki bilgi birikiminin, yeteneklerin, araç ve tekniklerin proje aktivitelerine uygulanmasıdır (PMI, 2013).

2.2. Proje Kavramı

Proje kavramının temel özelliklerini anlamak amacıyla, bu bölümde öncelikle proje kavramı açıklanmıştır. Belirli bir başlangıç ve bitiş tarihleri arasında; zaman, maliyet ve kalite gibi parametreler için belirlenmiş hedefleri karşılamak amacıyla gerçekleştirilen özgün girişimler proje olarak tanımlanır (Gilbreath, 1988). Kerzner (2003) ise proje kavramını tanımlarken belirli bir süre içerisinde belirli hedefleri karşılama özelliğine dikkat çekmenin yanı sıra sınırlı sayıda insan, para, ekipman gibi kaynaklar kullanımı gerektiren aktiviteler bütünü olma özelliğine değinmiştir. PMI ise proje kavramını, özgün bir ürün, hizmet ya da sonuç yaratmak amacıyla yürütülen geçici bir girişim olarak tanımlar.

Yukarıdaki tanımlardan yola çıkarak proje kavramının genel özelliklerine aşağıda yer verilmiştir:

- Zaman, maliyet ve kalite gibi parametreler için belirlenmiş hedefler
- Belirli başlangıç ve bitiş tarihleri
- Özgün bir ürün, hizmet ya da sonuç yaratma çabası
- Daha önce yapılmamış olma
- Geçmiş tecrübe azlığı
- Zamanla gelişime açık
- Kaynak ve efor kullanımını gerektirme

Proje kapsamında gerçekleştirilecek işlere atanacak kaynakların kısıtlı sayıda olması göz önüne alındığında proje planlama ve kontrolünün gerekliliği ortadadır. Kısıtlı kaynakların planlanması ve kontrolü özelliği akla operasyonel süreçleri getirirse de, proje ve operasyon kavramları birbirleri ile karıştırılmamalı ve yönetimleri ayrı ele alınmalıdır (PMI, 2013).

Proje ve operasyon kavramlarının farklı birçok özelliği söz konusu olup burada en önemli farklılıklarına dikkat çekilmiştir. Operasyon; var olan sistemin, özelliklerin, süreçlerin ve kabiliyetlerin sürekli olarak en iyi şekilde kullanımını hedefler (Gilbreath, 1988). Operasyon kapsamında sürekli olarak gerçekleştirilen süreçlerin daha önce gerçekleştirilmiş olması sonucu oluşan geçmiş bilgi ve tecrübenin varlığı projeye nazaran çoğunlukla daha fazladır. Operasyonda süreklilik söz konusu iken, bir defaya mahsus olarak daha önceden yapılmamış yeni bir girişim olan projeler geçicidir (Gilbreath, 1988). Bu geçici olma özelliği projeyi operasyona göre geçmiş tecrübe varlığı açısından daha yetersiz yapmaktadır. Projeler, yaşam döngüsü boyunca belirsizlikler ortadan kalktıkça ve tecrübeler edinildikçe olgunlaşacak olup gelişmeye açıktır (PMI, 2013). Operasyon sürekli olarak tekrarlandığı için projeye nazaran optimizasyona daha açıktır (Gilbreath, 1988). Proje ve operasyon kavramlarının bazı özelliklerinin ortak olmasının yanı sıra birçoğunun farklılık göstermesi sonucu yönetimleri için gereken faaliyet ve yetenekler de farklılık göstermektedir.

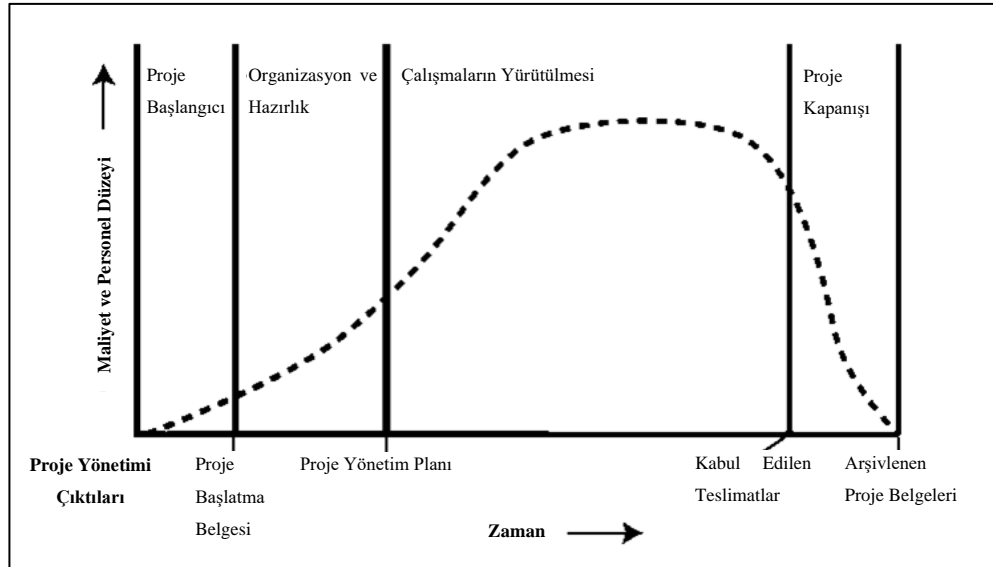
2.3. Proje Yaşam Döngüsü ve Aşamaları

Proje yaşam döngüsü boyunca birtakım aşamalardan geçer. Bu aşamalar, projenin ana teslimatlarının tamamlanması amacıyla mantıksal olarak birbirine bağlanmış aktivite topluluklarıdır (PMI, 2013). Küçük boyutlu ve karmaşık olmayan projeler tek aşamalı olarak yönetilebilir. Daha büyük boyutlu ve karmaşık projeleri aşamalara bölmek projenin daha verimli bir şekilde yönetilmesi, planlanması ve kontrolünü sağlar. Projelerin aşamalara ayrılmasında her projeye uygulanabilecek ideal bir yapı olmamakla birlikte çeşitli ölçütler göz önünde bulundurulur. Projenin geçeceği aşamaların isimleri ve sayıları, projenin gerçekleştirildiği organizasyon yapısı ve ihtiyaçları, uygulama alanı, hedefleri, ara teslimatları, kilometre taşları ve finansal kısıtları gibi çeşitli ölçütlerine göre değişiklik gösterebilir. Diğer aktivitelerden farklı odağı olan, farklı bir organizasyon bünyesinde ya da lokasyonda gerçekleştirilen, farklı yetenekleri gerektiren aktiviteler bir aşama altında toplanabilir. Gereksinim belirleme, fizibilite, planlama, tasarım, üretim, test proje aşamalarına örnek olarak verilebilir (PMI, 2013). Başka bir kaynağa göre bu aşamalar, kavramsallaştırma, planlama, uygulama, sonlandırma olarak belirlenmiştir (Pinto, 2016). Proje kapsamında üretilecek olan ürüne ait kavramsallaştırma, tasarım,

planlama, atama, gerçekleştirme, teslimat, gözden geçirme, destekleme gibi aşamalar da söz konusu olabilir (Chapman ve Ward, 2003). Projeler; büyüklük ve karmaşıklık bakımından farklılık gösterse de genel olarak başlangıç, organizasyon ve hazırlık, çalışmaların yürütülmesi, kapanış gibi aşamalardan oluşan bir yaşam döngüsü çerçevesi kapsamında yürütülmektedir. PMI (2013) tarafından, proje aşamaları “başlangıç”, “planlama”, “yürütme”, “izleme, kontrol” ve “kapanış” şeklinde ele alınmıştır.

Proje yaşam döngüsü boyunca, projeye etki eden bazı karakteristiklerde değişimler görülebilir. Aşağıda bu değişimlerden en önemlilerine yer verilmiştir:

- **Proje Maliyetleri:** Şekil 2.1’de de görüldüğü üzere proje başlarında maliyetler daha düşüktür, çalışmalar gerçekleştirildikçe proje giderlerinde artış görülürken, projenin kapanışına doğru maliyetlerde hızlı bir şekilde azalış görülür. Fakat, her proje tipinde bu şekilde olmayabilir. Bazı projelerde başlangıçta güvenlik ihtiyaçları gibi projeye özgü ihtiyaçlara yönelik kullanılan kaynaklar göz önünde bulundurulduğunda maliyetler başlangıçta yüksek olabilir (PMI, 2013).

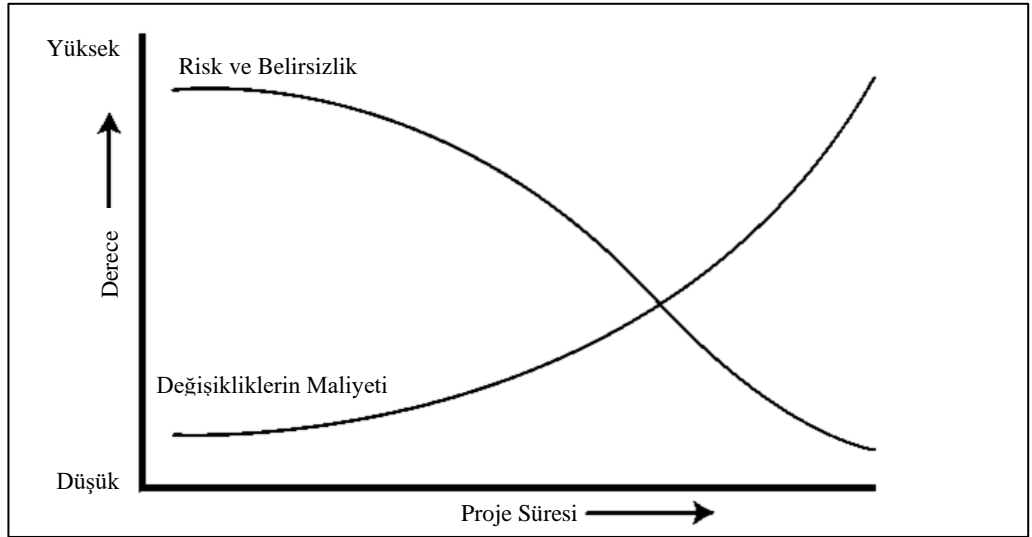


Şekil 2.1. Proje süreç gruplarına göre maliyet ve personel düzeyinin değişimi (PMI, 2013)

- **Risk ve Belirsizlik:** Şekil 2.2’de de görüldüğü üzere her projenin doğası gereği başlangıcında risk ve belirsizlik daha fazladır (PMI, 2013). Projede teslimatlar onaylanıp

gerçekleştirildikçe ve birtakım kararlar alındıkça belirsizlikler azalacaktır. Projenin karşı karşıya gelebileceği belirsizlikler, teknik zorluklar olabilir. Teknik açıdan belirsizlikler proje başlarında daha fazla olup, proje gereksinimleri netleştikçe ve araştırmalar yapıldıkça azalacaktır (Pinto, 2016).

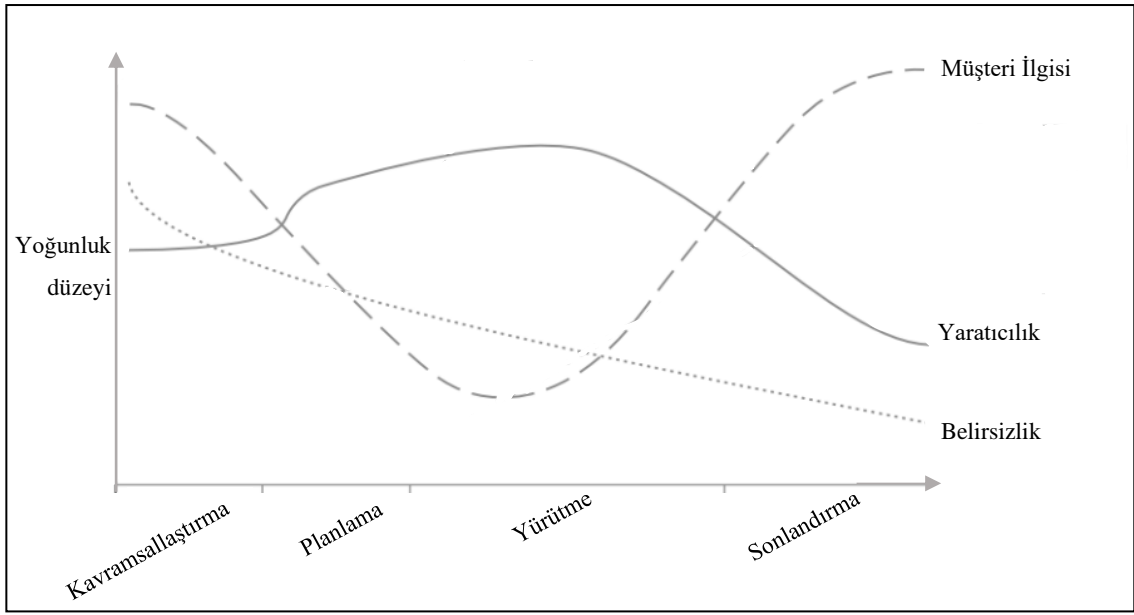
- Değişiklik Maliyetinde Değişim: Şekil 2.2’de de görüldüğü üzere proje başlarında proje sonucunda üretilecek olan ürünün karakteristiklerini etkileyecek değişimleri yapmanın maliyeti daha az iken, sonlara yaklaştıkça değişiklik maliyetlerinde artış görülmektedir. Diğer bir deyişle, proje ya da ürün kapsamında yapılacak olan değişiklikler ya da hataların giderilmesi sonucu ortaya çıkan maliyetler proje sonlarına doğru büyük ölçüde artacaktır (PMI, 2013).



Şekil 2.2. Proje süresine bağlı olarak risk ve maliyet etkilerinin değişimi (PMI, 2013)

- Müşteri İlgisi: Şekil 2.3’te de görüldüğü üzere, proje yaşam döngüsü boyunca müşterinin projeye olan ilgi miktarı, diğer bir deyişle heyecan ve endişe miktarlarındaki değişim U-tipinde bir eğri ile temsil edilir. Proje başlangıcında yüksek olan projeye olan ilgi, proje geliştirme aşamalarında azalırken, proje bitişine yakın tekrardan artış gösterir (Pinto, 2016).

- Yaratıcılık: Şekil 2.3'te de görüldüğü üzere, proje başlarında çalışanlarda daha çok yaratıcılık beklenmektedir. Sıklıkla yenilikçi düşünce ve özgün bakış açısının uygulanması olarak görülen yaratıcılık, proje amaçları ve kapsamı netleştirilip, planları oluşturulduktan sonra, uygulama aşamasında yaratıcılığa verilen önem azalmakta olup, çalışanlardan yaratıcılıktan ziyade görevlerine tamamlamak üzere konsantre olmaları beklenmektedir (Pinto, 2016).



Şekil 2.3. Proje süresince paydaşların ilgisi, belirsizlik ve yaratıcılık gibi faktörlerin değişimi (Pinto, 2016)

2.4. Proje Yönetimi

PMI proje yönetimini, proje ihtiyaçlarını karşılamak üzere proje yönetimi ile ilgili bilgi birikiminin, yeteneklerin, araç ve tekniklerin proje aktivitelerine uygulanması olarak tanımlamıştır. Projenin kapsam, maliyet ve zaman kısıtları arasında dengenin kurulması, proje yönetimi sürecinin en önemli amaçlarından biridir (Leach, 2005). Projenin kapsam, zaman, maliyet ve kalite hedeflerinden en az sapma göstererek başarı ile tamamlanmasından proje yöneticisi sorumludur (McCharty, 2016). Proje kapsamındaki aktivite sürelerinin kısaltılması için ek kaynak kullanımı yapılabilir ve bu durum da bütçede artmaya sebep olabilir. Eğer, bu durumda bütçenin arttırılması imkânsız ise kalite

hedefinde azaltmaya gidilebilecek olup bu da müşteri ihtiyaçları doğrultusunda belirlenen proje kapsamının karşılanmamasına sebebiyet verebilir. Projenin bu üç önemli kısıtı arasında denge kurmak ve hedeflerden sapmayı minimuma indirmek proje yönetimi kapsamında gerçekleştirilen en temel görevlerden bir tanesidir. Proje yönetimi kapsamında, kaynaklar en iyi şekilde kullanılarak dengelenmelidir. Proje yönetimi, firma kaynaklarının nispeten kısa dönemli hedefler ile planlaması, organize edilmesi, yönlendirilmesi ve kontrol edilmesi olarak tanımlanabilir (Kerzner, 2003).

Proje yönetimi, temel proje kısıtlarının, paydaş beklentilerinin ve kaynakların dengelenmesinin yanında genel olarak projenin planlanması ve kontrolü ile ilgilenir. Planlama kapsamında; işin gereksinimleri, miktarı, kalitesi ve kaynak gereklilikleri tanımlanır (Kerzner, 2003), proje yönetim planları oluşturulur, proje riskleri değerlendirilir, iletişimin gerekleri belirlenir (PMI, 2013). Projenin kontrolünde ise, projenin işleyişi izlenir, tahminlenen ve gerçekleşen çıktılar karşılaştırılır, etkiler analiz edilir ve düzenlemeler yapılır (Kerzner, 2003).

Proje başarısı, projenin belirlenen kapsam, zaman, maliyet dahilinde, etkin ve etkili kaynak kullanımı yapılarak üretilen çıktılarının müşteri tarafından onaylanması ile tamamlanması olarak tanımlanır (Kerzner, 2003). PMI (2013) kapsam, zaman ve maliyet olmak üzere projeye ait üç kısıtın dengelenmesine dikkat çekmiştir. Proje zamanının kısaltılması için kullanılan ek kaynaklar bütçede artma yaratır. Bütçede artma olmaması için kalite ve kapsam hedeflerinde azaltmaya gidilebilir. Bu durum müşteri ihtiyaçlarının karşılanmamasına sebep olabilir. Projenin süre açısından uzunluğu, maliyeti ve kapsam değişikliği riskini artırır. Kapsam değişikliğinin artması, maliyet ve süreyi artırır (PMI, 2013). Kapsam değişikliği projeler için kaçınılmaz olup, çok az proje belirlendiği kapsam dahilinde gerçekleştirilir. (Kerzner, 2003). Fakat bu kapsam değişikliğinin kontrolünün sağlanarak en az düzeyde tutulması proje yönetim başarısı için önem arz etmektedir. Sonuç olarak, projenin kapsam, zaman ve maliyet hedeflerinden kabul edilebilir bir sapma ile tamamlanması, proje yönetim başarısı olup, bu başarıda bahsi geçen üç hedefin dengelenmesi önem kazanmaktadır.

2.5. Proje Yönetim Süreç Grupları

Uygulama alanları ve odaklandığı sektörlerden bağımsız olarak projelerin yönetim süreçleri beş farklı süreç grubu altında toplanmıştır: “Başlangıç”, “planlama”, “yürütme”, “izleme, kontrol” ve “kapanış” (PMI, 2013).

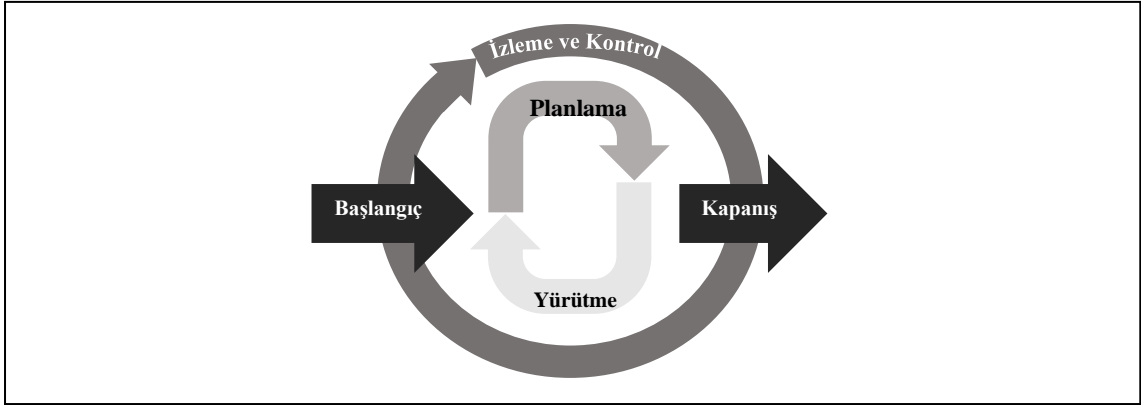
Başlangıç süreç grubu kısaca, proje ya da aşamanın tanımlanması, proje yöneticisinin seçilmesi, projenin başlatılma yetkisinin verilmesi, iç ve dış paydaşların belirlenmesi gibi süreçlerden oluşmaktadır. Başlangıç süreç grubundaki süreçler, bir proje aşamasının başlatılması için de gerçekleştirilebilir. Bir aşamanın başlatılmasında, projenin odaklandığı iş ihtiyaçları gözden geçirilir, başarı kriterleri doğrulanarak diğer aşamaya geçme, erteleme ya da geçmeme kararı alınır.

Planlama süreç grubu kısaca, kapsamın oluşturulması, hedeflerin geliştirilmesi ve bu hedeflere ulaşmak için yönetim planlarının geliştirilmesi gibi süreçlerden oluşmaktadır. Her projenin doğası gereği başlangıcında belirsizlikler fazladır. Proje yaşam döngüsü boyunca proje ilerledikçe sahip olunan bilgi artacak, bilgi arttıkça plan detaylanacak ve daha gerçeğe yakın tahminler yapılabilecektir. Bu yaklaşımı PMI (2013) aşamalı olgunlaşma olarak tanımlamaktadır. Proje yaşam döngüsü boyunca gerçekleşen önemli değişiklikler ve sahip olunan bilgiler, planları gözden geçirmek ve detaylandırmak üzere planlama süreçlerine, hatta başlatma süreçlerine geri dönmeye itebilir. Planlama ve dokümantasyon proje yaşam döngüsü boyunca yinelenen ve devam eden bir süreçtir.

Yürütme kısaca, proje gereksinimlerinin karşılanması için geliştirilen yönetim planlarında tanımlanmış olan işleri tamamlamak için gerçekleştirilen süreçleri kapsayan ve proje bütçesinin büyük bir miktarının harcandığı süreç grubudur. Proje yönetim planlarına bağlı kalarak, aktivitelerin gerçekleştirilmesini sağlamak, çalışanların ve kaynakların koordinasyonunu yönetmek, paydaşların beklentilerini yönetmek gibi süreçler yürütme süreç grubu kapsamında ele alınır. Yürütme sonucu ortaya çıkan çalışma performans bildirimleri, aktivite sürelerinin yeniden tahmini, kaynak kullanılabilirliklerinin yeniden planlanması ve beklenmeyen risklerin kayıt altına alınması gibi planlama süreçlerinin tekrarlanmasını gerektirebilir. Yürütme sürecinde

değişiklik talepleri oluşabilir, değişiklik kontrolü süreci sonucunda onaylanan değişiklikler de planlama süreçlerinin tekrarlanmasını gerektirebilir.

İzleme ve kontrol, proje sürecinin işleyişinin ve performansının belirli periyotlarda izlenmesi, gözden geçirilmesi, sapmaların analiz edilmesi, raporlanması, değişiklik gerektiren alanların tespiti, değişiklik kontrol sürecinin yönetilmesi, düzeltici/önleyici faaliyetlerin önerilmesi, çalışmalar gerçekleştikçe kapsam onayının alınması gibi süreçleri içeren bir süreç grubudur. İzleme ve kontrol süreçleri diğer süreç gruplarının paralelinde ilerlemektedir. Diğer bir deyişle projenin başlangıç, planlama, yürütme ve kapanış süreçleri; Şekil 2.4'te de görüldüğü üzere izleme ve kontrol süreçleri ile desteklenmelidir.



Şekil 2.4. Proje süreç grupları (PMI, 2013)

Kapanış süreç grubu, proje kapsamında gerçekleştirilecek olan aktivitelerin tamamlanması sonucu projenin resmi olarak kapanışı ve öğrenilmiş derslerin raporlanması gibi projenin sonlandırılması ile ilgili süreçlerden oluşmaktadır. Proje kapsamında gerçekleştirilecek faaliyetler tamamlanmadan, çeşitli nedenlerde durdurulma kararı alınmış projelerde de kapanış süreçleri uygulanmalıdır (PMI, 2013). Projenin bir aşaması için de kapanış süreçleri uygulanabilir. Aşama gözden geçirme toplantısı gerçekleştirilerek, projenin bu aşamasında proje çıktısı ile ilgili tamamlanan özellik ve fonksiyonlar başarı kriterleri göz önünde bulundurularak ana paydaşlar tarafından doğrulanır ve aşamaya ait öğrenilmiş dersler raporlanır. Proje ya da aşama ile ilgili dokümanlar ve kayıtlar diğer projelerde kullanılmak üzere arşivlenir. Proje takım üyeleri

değerlendirilir ve serbest bırakılır. Süreç grupları proje süresince birbirleri ile etkileşim içerisindedir. Bu etkileşimlere, bir sürecin çıktısının başka bir sürecin girdisi olması ya da bir sürecin bir projenin/alt projenin/aşamının teslimatı olması örnek olarak gösterilebilir. Planlama süreç grubu çıktısı olan proje planları ve dokümanları yürütme süreç grubunun girdisi iken proje yürütme sürecinde toplanan çalışma performans verilerine göre proje planı ve dokümanlarında güncelleme olabilir.

Proje yaşam döngüsü boyunca bir aşama içerisinde tüm proje süreç grupları gerçekleştirilebilir. Örneğin, kavram geliştirme, fizibilite çalışması, prototip üretim, test gibi proje aşamalarından oluşan bir projede bütün süreç grupları her bir aşamada tekrarlanabilir. Proje süreç grupları birbirleri ile ilişkili süreçlerden oluşturulmuştur. Proje yönetiminin yinelemeli yapısı gereği, proje yaşam döngüsü boyunca proje süreç grupları tekrardan gerçekleştirilebilir. Örneğin, yürütme aşamasında risk yanıtlarının gerçekleştirilmesi, planlama süreç grubunda yer alan risklerin tekrardan tanımlanması ve analizi ile ilgili süreçleri tetikleyebilir.

2.6. Proje Yönetim Bilgi Alanları

PMI (2013) tarafından proje yönetim süreçleri; aynı zamanda entegrasyon, kapsam, zaman, maliyet, kalite, insan kaynakları, iletişim, risk, tedarik ve paydaş olmak üzere on bilgi alanı dahilinde gruplandırılmıştır ve 10 bilgi alanının her biri bir uzmanlık alanı olarak sunulup, daha önce belirtilen başlangıç, planlama, yürütme, izleme, kontrol ve kapanış süreç gruplarına nasıl entegre edileceği ile ilgili yaklaşımlar ele alınmıştır. Çizelge 2.1’de PMI (2013)’ün sunduğu proje yönetim süreçleri yer almaktadır.

Proje yönetim bilgi alanları da birbiri ile etkileşimli olup, her bir bilgi alanına ait süreçler arasındaki koordinasyonu sağlamak, proje yöneticisinin en temel görevlerinden bir tanesidir. Proje yöneticisi, paydaş ihtiyaçlarını göz önünde bulundurup, proje hedefleri ve yönetim süreçleri arasında dengeyi sağlamakla görevlidir. Örneğin, kapsam değişimi maliyet veya kalite planlarını etkilerken, iletişim veya risk planlarını etkilemeyebilir. Ya da yeni eklenen risk üzerine risk planındaki değişim, zaman ve maliyet planlarını etkileyebilir. Değişiklik yönetimi sürecinden geçip onaylanan bir değişiklik; kapsam,

zaman ve maliyet planlarında değişiklikler oluşturabilir. Yönetim planları güncel tutulmalı ve aralarındaki etkileşimler dengelenmelidir.

Çizelge 2.1. Proje yönetim süreçleri (PMBOK, 2013)

		Süreç Grupları				
		Başlangıç	Planlama	Yürütme	İzleme ve Kontrol	Kapanış
Bilgi Alanları	Entegrasyon	Proje başlatma belgesinin geliştirilmesi	Proje yönetim planının geliştirilmesi	Çalışmaların yönlendirilmesi ve yönetilmesi	Proje çalışmalarının izlenmesi ve kontrolü Entegre değişiklik kontrolünün gerçekleştirilmesi	Projenin kapatılması
	Kapsam		Kapsam yönetiminin planlanması Gereksinimlerin toplanması Kapsamın tanımlanması İş kırılım yapısı oluşturulması		Kapsamın onaylanması Kapsamın kontrolü	
	Zaman		Zaman yönetiminin planlanması Aktivitelerin tanımlanması Aktivitelerin sıralanması Aktivite kaynaklarının tahmin edilmesi Aktivite sürelerinin tahmin edilmesi Çizelgenin geliştirilmesi		Zaman çizelgesinin kontrolü	
	Maliyet		Maliyet yönetiminin planlanması Maliyetlerin tahmin edilmesi Bütçenin belirlenmesi		Maliyetlerin kontrolü	
	Kalite		Kalite yönetiminin planlanması	Kalite güvencesinin sağlanması	Kalitenin kontrolü	
	İnsan Kaynakları		İK yönetiminin planlanması	Proje ekibinin oluşturulması Proje ekibinin geliştirilmesi Proje ekibinin yönetilmesi		
	İletişim		İletişim yönetiminin planlanması	İletişim yönetilmesi	İletişimin kontrolü	
	Risk		Risk yönetiminin planlanması Risklerin belirlenmesi Niteliksel risk analizinin gerçekleştirilmesi Niceliksel risk analizinin gerçekleştirilmesi Risk yanıtlarını planlanması		Risklerin kontrolü	
	Tedarik		Tedarik yönetiminin planlanması	Tedariklerin yürütülmesi	Tedariklerin kontrolü	Tedariklerin kapanışı
	Paydaş	Paydaşların belirlenmesi	Paydaş yönetiminin planlanması	Paydaş katılımının yönetilmesi	Paydaş katılımının kontrolü	

2.7. Proje Efor Tahmini ve Önemi

Bir proje ile ilgili merak edilen en önemli noktalardan biri proje maliyetinin ne kadar olacağıdır. Çoğu durumda en büyük proje maliyet kalemi işgücü olduğu için proje eforunun tahmin edilmesi proje yönetimi için en önemli odak noktalarından biridir. Proje efor tahmini, proje çizelgesi ve bütçesinin geliştirilmesi gibi proje planlama aktiviteleri için en önemli girdilerden bir tanesidir. Proje aktivite sürelerinin belirlenebilmesi için öncelikle ilgili aktiviteyi tamamlamak üzere gereken efor miktarının tahmin edilmesi gerekir. Proje eforu kısaca, bir projeyi tamamlamak üzere gereken toplam işgücü miktarıdır ve genellikle kişi-saat cinsinden ölçülür.

Proje efor tahmini, proje yönetimi alanında karşılaşılan en önemli zorluklardan biridir. Proje efor tahminindeki başarısızlık proje başarısını olumsuz yönde etkiler (Hameed ve diğerleri, 2023). Bu durum, çoğunlukla projenin çizelge ve bütçe açısından başarısızlığı ile sonuçlanmaktadır (Usman ve diğerleri, 2018).

Özellikle projenin ilk aşamalarında, belirsizliğin yüksek olmasından dolayı gerçeğe yakın tahminler yapılması oldukça zordur (Carvalho ve diğerleri, 2021). Projenin hedefleri ve öncelikleri (Chapman ve Ward, 2003), müşteri beklentileri (Pinto, 2016) ve kaynakların kullanılabilirlik durumları (Pinto, 2016) gibi birtakım belirsizlikler söz konusudur. Proje ortamındaki belirsizlikler ve geleceğin büyük ölçüde belirsiz olması göz önüne alındığında, projelerin geleceğe yönelik beklentilere, varsayımlara ve tahminlere dayanarak gerçekleştirilmesi kaçınılmaz olup, bu da risk almayı gerektirir (Raftery, 1994). Bu belirsiz ortamda, geçmiş tecrübe yetersizliği ve yapılan analizlerin yetersizliği de söz konusu olduğunda efor, süre, kaynak, maliyet gibi tahminlerde başarısızlık kaçınılmaz olmaktadır.

Belirsizliğin yüksek olduğu ortamlarda, karar vericilerin insan doğası gereği iyimser tahminler yapma eğiliminde olduğu (Prater ve diğerleri, 2017) ve iyimser tahminlerin en önemli proje başarısızlık sebeplerinden biri olarak düşünüldüğü belirtilmektedir (Nassif ve diğerleri, 2013). Bunun yanında, diğer firmalarla rekabet edebilme açısından da efor

tahmininde gerçeği ortaya koymak oldukça zor olup (Jørgensen ve Sjøberg, 2014), proje için genellikle iyimser tahminler yapılır.

Proje eforunun gerekenden daha az tahmin edilmesi, projeye yetersiz kaynak atanması ve bunun sonucunda atanan kaynakların planlanandan daha fazla kullanılmasına neden olmaktadır. Bu da çalışanların stres düzeyinde artışa ve çalışan memnuniyetsizliğine neden olabilir. Öte yandan, proje eforunun gerekenden fazla tahmin edilmesi de birtakım problemler oluşturabilir. Gereğinden fazla efor ataması sonucu süre ve kaynak fazlalığı oluşur ve bu durum ürün kapsamı dışında gerekenden fazla özellikler eklenerek kapsam dışına çıkılmasına, diğer bir deyişle kapsam genişlemesine ve planlanan bütçenin aşılmasına neden olabilir. Bunun yanında, bir aktivite üzerinde gerekenden fazla zaman harcayan kaynakların, daha önemli fırsat elde edilebilecek olan diğer projelerde bu süre zarfında kullanılamaması fırsat kaybına neden olabilir (Hameed ve diğerleri, 2023) ve bu durum ilerleyen projelerin gecikmesine yol açabilir.

Kısaca, proje efor tahminindeki başarısızlık, çalışan memnuniyetsizliği, ürün kalitesinde düşüklük ve nihayetinde müşteri memnuniyetinde azalma gibi birtakım ciddi sorunlar doğurabileceğinden dolayı, proje yönetimi için odaklanması gereken en önemli noktalardan biridir.

2.8. Proje Efor Tahmini için Kullanılan Faktörler

Literatürde proje efor tahmini için çeşitli faktörler kullanılmıştır. Projede üretilecek üründe kullanılan teknoloji birçok çalışmada ele alınmıştır. Jaifer ve diğerleri (2021) teknolojinin olgunluk düzeyini, ürün karmaşıklığını belirlemek üzere ele alırken, Bashir ve Thomson (2004) yeni teknoloji kullanımını projenin teknik zorluğu ile ilişkilendirmiştir. Bashir ve Thomson (2004) projenin teknik zorluğunu ölçümlerken, ayrıca gereksinimlerin karmaşıklığını da ele almıştır. Yurt ve diğerleri (2019) de proje efor tahmini için ürün karmaşıklığını ölçümlerken gereksinimlerin karmaşıklığına yer vermiştir. Literatürde proje efor tahmini için üretilecek ürün ile ilişkili ürün karmaşıklığının yanında (Bashir ve Thomson, 2004; Arundacahawat ve diğerleri, 2013; Yurt ve diğerleri, 2019; Jaifer ve diğerleri, 2021), ürün büyüklüğü (Pollmanns ve

diğerleri, 2013), alt sistemler arasındaki hareketler (Arundacahawat ve diğerleri, 2013), inovasyon düzeyi (Pollmanns ve diğerleri, 2013) gibi faktörler de kullanılmıştır.

Proje takımı kabiliyetleri ve deneyimlerine de bazı çalışmalarda proje eforunu etkileyen faktörler arasında yer verilmiştir (Bashir ve Thomson, 2004; Salam ve diğerleri, 2007; Pollmanns ve diğerleri, 2013; Jaifer ve diğerleri, 2021). Proje takım büyüklüğü, disiplin çeşitliliği, lokasyonları (Jaifer ve diğerleri, 2021), tedarikçi deneyimi (Arundacahawat ve diğerleri, 2013; Yurt ve diğerleri, 2019) ürün geliştirme eforunu tahminlerken kullanılmıştır.

Literatürde proje efor tahmini ile ilgili birçok çalışma yazılım proje eforu üzerine yoğunlaşmıştır. Yazılım proje efor tahmini alanında makine öğrenme teknikleri sıklıkla kullanılmış olup, özellikle YSA daha çok tercih edilmiştir. YSA modelinde girdi sayısının artması, modeli daha karmaşık hale getirir ve model performansında azalma meydana gelir (Park ve Baek, 2008). Dolayısıyla, YSA modellerinde kullanmak üzere, proje eforunu tahmin etmek için en kritik olan girdileri seçmek önemlidir.

Yazılım proje efor tahmini alanında birçok çalışma, YSA girdisi olarak yazılım büyüklüğünü belirlemek için yazılımdaki kod satırı sayısını kullanmıştır (Heiat, 2002; Park ve Baek, 2008; Dave ve Dutta, 2011; López-Martin ve diğerleri, 2011; Rankovic ve diğerleri, 2021). Yazılım büyüklüğünü belirlemek için ayrıca aktör sayısı ve kullanım şekli (use case) diyagramları arasındaki hareketler (Nassif ve diğerleri, 2013; Singh ve Kumar, 2020), yazılımdaki girdi sayısı ve işlev puanı (function points) (Koch ve Mitlöhrer, 2009; Nassif ve diğerleri, 2013) gibi faktörler de sıklıkla kullanılmıştır. Yazılım büyüklüğü yanı sıra proje takımının uygulama üzerindeki deneyimi ve kabiliyeti de efor tahmininde kullanılan faktörler arasındadır (Tronto ve diğerleri, 2008; López-Martin ve diğerleri, 2011; Arora ve Mishra, 2018; Goyal ve Bhatia, 2020; Singh ve Kumar, 2020). Ürün yeniliği faktörü de yazılım proje efor tahmin çalışmalarında kullanılmıştır. Park ve Baek (2008) ürün yeniliği faktörünü kullanmış olup, projeleri, yeni, geliştirme ve bakım projeleri olarak üç kategori altında toplarken, Pospieszny ve diğerleri (2018) projeleri yeni, iyileştirme ve yeniden geliştirme projeleri olarak ayırmıştır.

Yukarıdaki girdileri kullanarak geliştirilen YSA modellerinde, çıktı olarak genelde toplam proje eforu ya da ürün geliştirme eforu belirlenmiş olup, çeşitli birimler kullanılmıştır: Kişi- veya adam-saat (Heiat, 2002; Tronto ve diğerleri, 2008; Nassif ve diğerleri, 2013; Azzeh ve Nassif, 2016; Singh ve Kumar, 2020), kişi- ya da adam-ay (Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Rijwani ve Jain, 2016; Pospieszny ve diğerleri, 2018; Goyal ve Bhatia, 2020; Pandey ve diğerleri, 2020; Rankovic ve diğerleri, 2021).

2.9. Proje Efor Tahmini için Kullanılan Yöntemler

Son 20 yılda proje efor tahmini özellikle yazılım sektöründe birçok araştırmacı tarafından çalışılmış olup, bu konu ile ilgili birçok yöntem önerilmiştir. Proje efor tahmini ile ilgili önerilen yöntemler üç kategori altında toplanmış olup, bunlar uzman görüşü tabanlı yöntemler, algoritmik modeller ve makine öğrenmesi yöntemleridir (Tronto ve diğerleri, 2008). Uzman görüşü tabanlı yöntemlerde değerlendirme, uzman kişinin benzer projelerdeki tecrübesine bağlı olup, öznellik içerir (López-Martín ve diğerleri, 2011). Algoritmik modeller, örneğin doğrusal veya doğrusal olmayan regresyon modelleri, proje efor tahmininde oldukça sıklıkla kullanılan tekniklerdir. Bashir ve Thomson (2004) hidroelektrik jeneratörlerin tasarım eforunu tahmin etmeye yönelik gerçekleştirilmiş olan 15 adet projenin verileri ile parametrik bir model geliştirmişlerdir. Salam ve diğerleri (2007) uçak motoru geliştirme projelerinin tasarım efor tahminleri için parametrik bir yöntem tercih etmişlerdir.

Ürün geliştirme, mühendislik projelerin ile ilgili yapılan çalışmalarda daha çok algoritmik modeller tercih edilirken, yazılım proje efor tahmini çalışmalarında araştırmacılar makine öğrenmesi gibi objektif yöntemlere yönelmişlerdir (Monika ve Sangwan, 2017). Literatürde bulanık mantık, YSA, genetik algoritma gibi evrimsel teknikler, karar ağaçları gibi makine öğrenme teknikleri özellikle yazılım proje efor tahmini alanında sıklıkla kullanılmıştır (López-Martín ve diğerleri, 2011; Nassif ve diğerleri, 2013). Yurt ve diğerleri (2019) tekerlekli zırhlı araçlar ve silah sistemleri ürün geliştirme projelerinde ar-ge faaliyetleri için gereken mühendislik eforu için geliştirdikleri efor tahmini modeli için

karar ağaçları ve k-en yakın komşu algoritmalarını kullanmışlardır. Karmaşık problemlerin çözümünde oldukça sıklıkla kullanılan bir makine öğrenmesi tekniği olan YSA modelleri özellikle yazılım efor tahmini ile ilgili birçok tahmin modeli uygulamasında kullanılmıştır (Ali ve Gravino, 2019). Birçok araştırmacı tarafından YSA modelleri, özellikle regresyon gibi diğer geleneksel tahmin yöntemleri ile kıyaslanmış olup, YSA modellerinin geleneksel tahmin yöntemlerine göre daha iyi sonuçlar verdiği sonucuna varılmıştır (Heiat, 2002; Tronto ve diğerleri, 2008; Park ve Baek, 2008; Dave ve Dutta, 2011; López-Martín ve diğerleri, 2011).

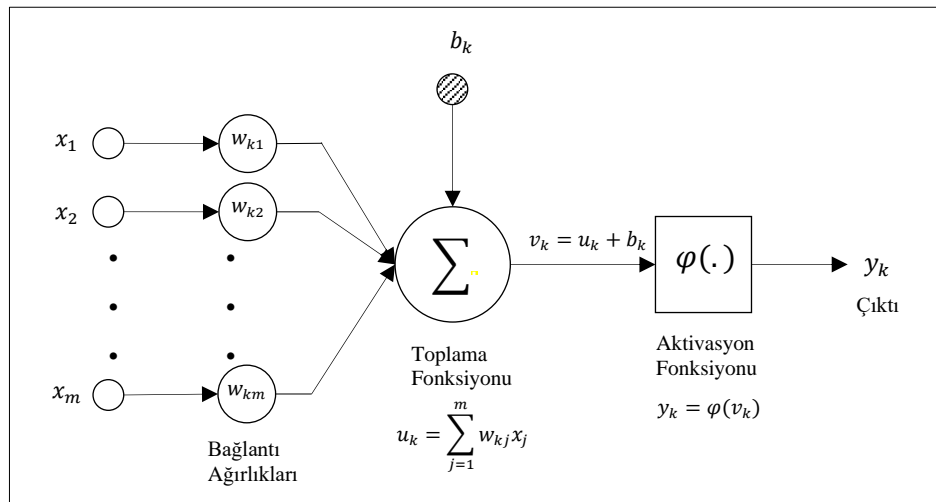
Literatürde, proje efor tahmini alanında YSA kullanımına özellikle yazılım projeleri üzerine yapılan çalışmalarda rastlanmıştır. Proje efor tahmini alanında en çok kullanılan YSA yapısı, ileri beslemeli sinir ağıdır (Feedforward Neural Network: FFNN) (Tronto ve diğerleri, 2008). Dave ve Dutta (2011) yazılım proje eforu üzerine gerçekleştirdikleri çalışmalarında, FFNN yapısının diğer tahmin yöntemlerine göre daha iyi sonuç verdiğini rapor etmişlerdir. FFNN girdi, gizli ve çıktı katmanları olmak üzere üç farklı katman tipinden oluşmaktadır ve her bir katmandan sadece ilerleyen katmanlara bağlantı bulundurmaktadır. FFNN yapısının farklı tipleri mevcuttur: çok katmanlı algılayıcı (Multilayer Perceptron: MLP), radyal temel fonksiyonu sinirsel ağı (Radial Basis Function Neural Network: RBFNN) ve genelleştirilmiş regresyon sinirsel ağı (Generalized Regression Neural Network: GRNN) (Nassif ve diğerleri, 2013).

En az bir gizli katman içeren MLP tam bağlantılı (fully connected) yapıdadır. RBFNN ise, MLP'den farklı olarak gizli katmanlarında radyal temel fonksiyonuna yer vermektedir. Yazılım efor tahmini için RBFNN kullanan ve regresyona göre daha iyi sonuçlar elde eden Heiat (2002), RBFNN'nin eğitiminin MLP'den daha kolay olduğuna dikkat çekmiştir. FFNN'nin diğer bir türü olan GRNN, MLP ve RBFNN'den farklı olarak dört katmandan oluşmaktadır (Kumar ve diğerleri, 2020) ve proje efor tahmini ile ilgili çalışmalarda kullanılmıştır (Quah ve Thwin, 2003; Nassif ve diğerleri, 2013). López-Martín ve diğerleri (2011) efor tahmini için GRNN ağ yapısını kullanmışlar ve regresyondan daha iyi sonuçlar elde etmişlerdir.

Proje efor tahmini ile ilgili bazı çalışmalarda YSA yapısına bulanık mantık entegre edilmiştir. Idri ve diğerleri (2002) tarafından gerçekleştirilen bir araştırmada, bulanık kurallı YSA önerilmiştir. Kumar ve diğerleri (2020) tarafından yapılan proje efor tahmininde YSA ile ilgili bir araştırmada hibrit yöntemlerin kullanımına da dikkat çekilmiştir. Sürü zekâsı optimizasyon algoritmaları (Parçacık Sürü Optimizasyonu: PSO; Karınca Koloni Optimizasyonu, Ant Colony Optimization: ACO; Yapay Arı Kolonisi, Artificial Bee Colony: ABC) ile YSA'nın kombine edildiği hibrit algoritmalara rastlanmıştır. Örneğin, PSO ve fonksiyonel link YSA (Functional Link Artificial Neural Network: FLANN) sinir ağı yapısının kombine edildiği hibrit yöntem literatürde sıklıkla rastlanmıştır (Kumar ve diğerleri, 2020).

2.10. Yapay Sinir Ağları

Yapay sinir ağları (YSA) biyolojik nöron hücrelerinden esinlenerek geliştirilen düğümlerin birbiri ile bağlanması ile oluşan ağ yapılarıdır. İnsan beyninin öğrenme yolu ile yeni bilgiler elde edebilme, hatırlama ve genelleme özelliklerinden ilham alınarak, bu öğrenme sürecinin matematiksel olarak modellenmesi sonucu ortaya çıkmıştır. Genellikle bir girdi katmanı, gizli katmanlar ve bir çıktı katmanından oluşur. Veri YSA'ya girdi katmanı aracılığıyla girer ve ilerleyen katmanlara iletilir. Nöron, bir YSA'nın işleyişi için temel olan bir bilgi işleme birimidir (Haykin, 2008). Yapay bir nöron Şekil 2.5'te örnek olarak gösterilmiştir.



Şekil 2.5. Yapay sinir ağında bir nöron gösterimi (Haykin, 2008)

YSA'da bulunan her bir nöron için üç temel eleman söz konusudur:

- Her biri kendi ağırlığı veya gücü ile nitelendirilmiş bir dizi bağlantılar (k nöronuna bağlı j 'nin girdi sinyali, x_j , w_{kj} bağlantı ağırlığı ile çarpılır)
- Girdi sinyallerinin ilgili bağlantı ağırlıkları ile ağırlıklandırılarak birleşimi, birleştirme fonksiyonu (linear combiner)
- Nöron çıktısının büyüklüğünü limitleyen aktivasyon fonksiyonu.

YSA'da bulunan her bir k nöronunun bir veya birden fazla girdisi (x_1, x_2, \dots, x_m) ve bunlara ilişkin bağlantı ağırlıkları ($w_{k1}, w_{k2}, \dots, w_{km}$) mevcuttur. Bağlantı ağırlıkları ile birleşen girdiler birleştirme fonksiyonu ile toplanarak, her bir nörona ait net girdi, u_k , denklem (2.1) ile hesaplanır:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.1)$$

Birleştirme fonksiyonundan elde edilen net girdi, u_k , aktivasyon fonksiyonunun net girdisini artırma ya da azaltma etkisi olan *bias* değeri b_k de eklenerek k nöronunda üretilen toplam net çıktı denklem (2.2) kullanılarak hesaplanır:

$$v_k = u_k + b_k \quad (2.2)$$

Bu ağırlıklı toplam değer, v_k , denklem (2.3)'de gösterildiği üzere, aktivasyon fonksiyonundan, $\varphi(\cdot)$, geçerek k . Nöronuna ait y_k çıktı sinyali üretilir.

$$y_k = \varphi(v_k) \quad (2.3)$$

Bu işlem katmandan katmana tekrarlanır ve çıktı katmanından sonuç çıktı olarak elde edilir (Haykin, 2008).

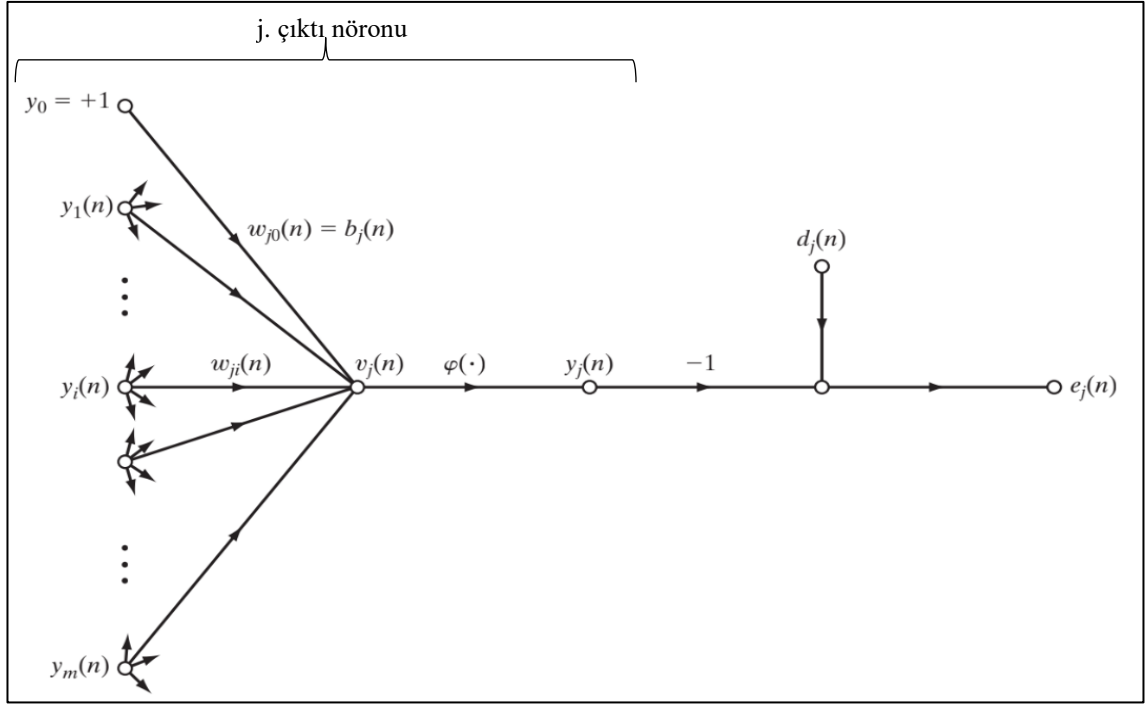
2.11. Yapay Sinir Ağları için Öğrenme Algoritmaları

YSA eğitimi, ağ yapısındaki nöronlar arasındaki bağlantıların ağırlık değerlerinin, eğitim kümesindeki girişleri çıkış değerlerine dönüştürecek şekilde ardışık olarak güncellenmesidir. Ağın öğrenme süreci klasik doğrusal olmayan bir problem olarak görülebilir. Çok katmanlı ileri beslemeli bir ağda ağırlıklar bağımsız değişken olarak düşünülebilir. Çıktı hata değeri toplamı ise amaç fonksiyonu olarak düşünülebilir. YSA eğitimindeki amaç, ağın çıktısının beklenen çıktı ile uyumlu hale getirecek şekilde ağırlıkların güncellenmesi olarak düşünülebilir (Johansson ve diğerleri, 1992).

YSA ağırlıklarının güncellenme yöntemleri ile ilgili literatürde birçok algoritma geliştirilmiş olup, özellikle FFNN yapısındaki ağlarda sıklıkla geri yayımlı (backpropagation: BP) öğrenme algoritması kullanılmaktadır. BP algoritması, makine öğrenmesinde oldukça popüler bir yöntem olan Gradyan Azalma (Gradient Descent: GD) tekniği ile hata fonksiyonunun en küçük değerini arayarak, ağırlık güncellemesi yapmaktadır (Haykin, 2008). BP algoritmasında her iterasyon, ileri ve geri yayılım olmak üzere iki aşamadan oluşur. İleri yayılım aşamasında, o anki n durumunda YSA'ya uygulanan giriş sinyallerine karşı YSA'nın ürettiği çıkış sinyalleri bulunur. Geri yayılımda ise, çıkışlarda oluşan hata değerlerini kullanarak, ağırlıklar güncellenir.

YSA'nın ağırlıkları başlangıçta rasgele seçilir, her iterasyonda her nöron için o nörona uygulanan girdilerden gelen fonksiyon sinyalleri toplanarak ve aktivasyon fonksiyonu ile işlenerek toplam net çıktı hesaplanır. Şekil 2.6'da görüldüğü üzere çıktı nöronu j 'nin solunda bulunan bir katmandan m adet girdi sinyali ile beslendiği durum ele alındığında, j nöronu ile ilişkili aktivasyon fonksiyonunun girdisinde üretilen net çıktı denklem (2.4)'teki gibi hesaplanır:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (2.4)$$



Şekil 2.6. j . çıktı nöronuna ait sinyal akışı (Haykin, 2008)

j çıktı nöronunda görünen fonksiyon sinyali, y_j denklem (2.5)'teki gibi hesaplanır:

$$y_j(n) = \varphi(v_j(n)) \quad (2.5)$$

n . iterasyondaki bir j çıktı nöronu tarafından üretilen çıktı sinyali $y_j(n)$ ve gerçek (hedef) çıktı, $d_j(n)$ değerleri karşılaştırılarak, j nöronunun n . iterasyondaki hata sinyali, $e_j(n)$, denklem (2.6)'daki gibi hesaplanır:

$$e_j(n) = d_j(n) - y_j(n) \quad (2.6)$$

j . nöronun anlık hata değeri, $\frac{1}{2}e_j^2(n)$ olarak tanımlanır. Ağın çıktı katmanındaki her bir çıktı nöronu $j \in C$, için bu değerler toplanarak, toplam hatanın anlık değeri, $\varepsilon(n)$ denklem (2.7)'deki gibi hesaplanır:

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.7)$$

BP algoritmasının amacı olan toplam hatanın anlık değerini $\varepsilon(n)$ en küçüklemek için gereken ağırlık düzeltmesini hesaplamak için, $\Delta w_{ji}(n)$, $\partial \varepsilon(n)/w_{ji}(n)$ kısmi türevini kullanır. Bu gradyan değeri, denklem (2.8)'deki zincir kuralı ile hesaplanarak elde edilir:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (2.8)$$

$\partial \varepsilon(n)/w_{ji}(n)$ kısmi türevi ağırlık uzayındaki arama yönünü belirleyen bir duyarlılık faktörünü temsil eder. Kısmi türev hesaplamaları yapıldığında kısaca denklem (2.9) kullanılarak elde edilir:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'(v_j(n)) y_j(n) \quad (2.9)$$

YSA'daki her w_{ji} ağırlık değerinin düzeltme miktarı, $\Delta w_{ji}(n)$, denklem (2.10)'daki gibi delta kuralı ile belirlenir.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \quad (2.10)$$

Burada η değeri, BP'nin öğrenme oranı parametresidir. YSA eğitiminde önemli noktalardan biri bu değer seçilmesidir. Eşitlikten de görüldüğü üzere bu katsayı ağırlıklardaki düzeltmeyi kontrol eder. Öğrenme oranı $[0,1]$ aralığında değerler alır ve yüksek η değerleri seçildiğinde eğitim süresi kısalmır. Bununla birlikte, bu değer küçük tutulursa, daha çok iterasyon gerekmesinin yanında optimal sonucun atlanma riski azaltılmış olur. Eşitlikte eksi işaretinin kullanımı ağırlık uzayında gradyan azalanı ifade eder ve hata fonksiyonunu $\varepsilon(n)$ 'i azaltacak yönde bir ağırlık değişimi için yön aranır. Yukarıdaki denklem (2.9) ve denklem (2.10) kullanılarak denklem (2.11) elde edilir:

$$\Delta w_{ji}(n) = \eta \times \delta_j(n) \times y_i(n) \quad (2.11)$$

Burada lokal gradyan (δ) değeri denklem (2.12)'deki gibi tanımlanır:

$$\delta_j(n) = \frac{\partial \varepsilon(n)}{\partial v_j(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \varphi'(v_j(n)) \quad (2.12)$$

Bu lokal gradyan değeri bağlantı ağırlıklarındaki gerekli olan değişikliği ifade eder. Çıktı katmanını besleyen tüm bağlantıların ağırlık düzeltilmesi $\Delta w_{ji}(n)$, her bir çıktı nöronunun hata sinyali $e_j(n)$ ile aktivasyon fonksiyonunun birinci derece türevinin $\varphi'(v_j(n))$ çarpımı ile gerçekleştirilir. Bu durum j nöronunun çıktı nöronu olduğu durumdur. j nöronunun gizli nöron olduğu durumda ise, doğrudan erişilebilir olmamasına rağmen, ağırlık düzeltilmesinde yapılmış herhangi bir hatanın sorumluluğunu paylaşırlar. Bu durum ile ilgili hesaplamaların ayrıntısı (Haykin, 2008)'de verilmiştir. Bu şekilde katmandan katmana ağırlıklardaki güncellemeler yayılarak devam eder. Ağırlıklar için optimal çözüme yakınsamada öğrenme oranı (η) parametresi önemli bir etkiye sahiptir. Öğrenme oranı düşük tutulup, eşitliğe momentum parametresi de eklenerek, ağırlık uzayı boyunca gerçekleşen hareketi ile ilgili dalgalanmalar hafifletilerek daha kararlı bir güncelleme gerçekleştirilebilir ve minimuma doğru hızla yakınsama sağlanabilir (Haykin, 2008).

GD tekniğini kullanan BP öğrenme algoritmalarının birtakım dezavantajları mevcuttur. GD tekniğinde, YSA ağırlıkları performans fonksiyonunun hızlı bir şekilde düştüğü yönde, diğer bir deyişle gradyanın negatif yönünde güncellenir, fakat bu hızlı yakınsamayı sağlamayabilir. Öğrenme oranı, momentum ve gizli katman sayısı gibi bazı parametrelerin seçimi, sonucu önemli derecede etkileyebilir, hata fonksiyonunun yerel minimuma takılma olasılığı söz konusu olabilir ve ağırlık güncellemelerinde çok sayıda iterasyon gerektirebilir (Chang ve Mak, 1999). Bu dezavantajların üstesinden gelmek için, hata fonksiyonunun sadece yerel gradyan değerine odaklanmayıp, ikinci dereceden türevini de dikkate alarak, en iyi güncelleme yönüne karar veren Eşlenik Gradyan (Conjugate Gradient, CG) algoritması geliştirilmiştir. Aramayı gradyanın eşlenik yönlerinde gerçekleştiren bu teknik, YSA ağırlık değerlerinin ayarlanmasında daha hızlı yakınsamaya olanak sağlar. CG yönteminin en önemli avantajları hızlı ve basit olmasıdır (Johansson ve diğerleri, 1992).

Wolpert ve Macready'e (1996) göre hangi problem tipinde hangi öğrenme algoritmasının daha iyi performans göstereceği hakkında karar vermek için deneyler gerçekleştirmek gerekir. Literatürde yer alan öğrenme algoritmalarından bazıları aşağıda maddeler halinde verilmiştir:

- Fletcher-Reeves Güncellemeleri ile Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Fletcher–Reeves Updates: CGF)
- Polak–Ribière Güncellemeleri ile Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Polak–Ribière Updates: CGP)
- Powell-Beale Eşlenik Gradyan Algoritması (Conjugate Gradient Backpropagation with Powell–Beale restarts: CGB)
- Ölçekli Eşlenik Gradyan Algoritması (Scaled Conjugate Gradient: SCG)
- Broyden-Fletcher-Goldfarb-Shannoo Algoritması (BFG)
- Levenberg–Marquardt Algoritması (LM)
- Değişken Öğrenme Oranlı ve Momentumlu Gradyan Azalma (Gradient Descent with Momentum and Adaptive Backpropagation: GDX)
- Bir Adım Sekant Geri Yayılım Algoritması (One-step Secant Backpropagation: OSS)
- Esnek Geri Yayılım Algoritması (Resilient Backpropagation: RPROP)

Fletcher-Reeves, Polak–Ribière, Powell-Beeale ve ölçekli eşlenik gradyan yöntemleri eşlenik gradyan temelli yöntemler olup; esnek geri yayılım ve bir adım sekant yöntemleri geri yayımlı yöntemlerdir (Zhu ve diğerleri, 2021).

Eşlenik eğimi temel alan algoritmalarından olan CGF, Fletcher ve Reeves tarafından 1964 yılında önerilmiştir. Algoritma, ağırlık güncellemede mevcut gradyanın norm karesi ile önceki gradyanın norm karesi arasındaki oranı temel alır (Zhu ve diğerleri, 2021). CGF'ye yakın performans gösterdiği, fakat bellek gereksiniminin CGF'ye nispeten daha fazla olduğu düşünülen CGP algoritması ise, bir önceki gradyanın norm karesine bölünen o anki gradyan ile bir öncekinin içler çarpımını temel alarak ağırlık güncellemesi yapar (Zhu ve diğerleri, 2021). CGB'nin ise CGP'den bazı durumlarda daha üstün olduğu fakat bellek gereksinimi daha fazla olduğu düşünülmektedir. Bu algoritma, bir önceki gradyan ile o anki gradyan arasındaki ortogonalite çok küçülmüşse, ağ arama yönünü başlangıçtaki gradyanın negatifi yönüne geri döndürür (Zhu ve diğerleri, 2021). SCG ise

Moller (1993) tarafından geliştirilmiş olup, eşlenik gradyan yönteminin ölçeklendirilmiş adım büyüklüğüne sahip bir çeşididir. CGP, CGF ve CGB algoritmalarındaki gibi, ağırlıklar eşlenik yönler boyunca aranır, farklı olarak SCG algoritması her bir iterasyonda adım boyutunu ölçeklendirerek zaman alıcı hat aramalarını gerçekleştirmez. Ağırlıklar, girdiler ve aktivasyon fonksiyonları türevlenebilir olduğu sürece herhangi bir ağ eğitimi bu algoritma ile gerçekleştirilebilir (Zhu ve diğerleri, 2021). Riedmiller ve Braun tarafından geliştirilmiş ve geri yayımlı algoritmalarından biri olan RP ise genel olarak CG temelli algoritmalarından biri olan CGF'den daha hızlıdır. GD'den daha çok bellek gerekir (Zhu ve diğerleri, 2021). Battiti tarafından 1992'de önerilen OSS algoritmasının ise en önemli avantajı, yeni arama yönü belirlenirken, Hessian matris hesaplamak yerine bir önceki iterasyondaki Hessian matrisini kullanarak, Hessian matrisin tersinin alınması gibi bir işleme gerek kalmamasıdır. Fakat, eşlenik azaltma algoritmalarına göre, hesaplama süresi ve bellek miktarı biraz daha fazladır (Zhu ve diğerleri, 2021). Geri yayılım algoritmasının lokal minimuma yakalanma riski gibi birtakım dezavantajlarına yönelik geliştirilen LM algoritması ise hesaplama belleği açısından GD gibi yöntemleri geride bırakır. Temel olarak maksimum komşuluk fikri üzerine kurulmuş olan LM algoritması, bir en küçük kareler hesaplama yöntemidir. Hessian matrisine küçük bir sabit eklenerek, Gauss Newton metodunda karşılaşılan problemlerden biri olan Hessian matrisinin tersinin alınması durumunun da üstesinden gelir.

Proje efor tahmini alanında, birçok araştırmacı geri yayımlı algoritmaları seçerken (Jun ve Lee, 2001; Dave ve Dutta, 2011; Attarzadeh ve diğerleri, 2012; Rijwani ve Jain, 2016), bazı araştırmacılar Levenberg-Marquardt öğrenme algoritmasını kullanmışlardır (Nassif ve diğerleri, 2013; Pai ve diğerleri, 2013; Goyal ve Bhatia, 2020).

2.12. Yapay Sinir Ağı Hiperparametreleri

Yazılım proje efor tahmini ile ilgili birçok çalışmada YSA, diğer tahmin yöntemlerine göre daha iyi sonuçlar vermiştir. Fakat, YSA'nın performansı dışarıdan girilmesi gereken parametreler olan hiperparametre değerlerine bağlı olup (Makarova ve diğerleri, 2021), hiperparametrelerde gerçekleştirilen en ufak bir ayar YSA performansını anlamlı derecede etkileyebilir. En iyi YSA yapısını elde etmek, modelin genelleme yaparak proje karakteristikleri ve eforu arasındaki karmaşık problemleri çözmesi adına önem arz etmektedir.

YSA modelinde kullanılan parametreler model parametreleri ve hiperparametreler olmak üzere iki ana grupta toplanır. Model parametreleri, YSA eğitim sürecinde ağın veriden öğrenerek elde ettiği ve tasarımcının herhangi bir ayarlama yapmasına gerek olmadığı parametrelerdir. Örneğin YSA'daki bağlantı ağırlıkları, YSA'nın öğrenme algoritmasına göre eğitim esnasında elde edilir. Fakat gizli katman sayısı, gizli katmandaki nöron sayısı ve aktivasyon fonksiyonları gibi hiperparametreler ise tasarımcı tarafından eğitim öncesinde belirlenir. Öğrenme oranı, momentum gibi hiperparametrelerin kullanımı, kullanılan öğrenme algoritmasına göre değişiklik gösterir. Örneğin, SCG algoritmasında Marquart Adjustment parametresi kullanılırken, diğer öğrenme algoritmalarında bu parametre kullanılmaz.

Proje efor tahmininde YSA kullanımına daha çok yazılım projelerinde rastlanmış olup, genellikle gizli katman sayısı, her bir gizli katmandaki nöron sayısı ve aktivasyon fonksiyonu tipi, öğrenme oranı ve momentum gibi hiperparametreler ayarlanmıştır. Proje efor tahmini ile ilgili YSA modeli kullanan birçok çalışma hiperparametre ayarlamasında deneme yanılma (trial and error) yöntemini tercih etmiştir (Rijwani ve Jain, 2016; Rankovic ve diğerleri, 2021). En çok ayarlanan hiperparametre gizli katman sayısı ve her bir gizli katmandaki gizli nöron sayılarıdır. Jun ve Lee (2001) proje efor tahmini için geliştirdiği YSA modelinde, bir gizli katman kullanmış olup, gizli katmandaki nöron sayısını 12'den 47'ye kadar değiştirerek deneyler gerçekleştirmişlerdir. 23 adet girdi ile proje eforu tahmini ile ilgili önerdikleri en iyi ağ yapısına bir gizli katman ve 17 gizli nöron ile ulaşmışlardır. Proje efor tahmini alanında daha çok bir gizli katman

kullanılmıştır (Tronto ve diğerleri, 2008; Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Rijwani ve Jain, 2016; Pospieszny ve diğerleri, 2018; Goyal ve Bhatia, 2020). Goyal ve Bhatia (2020) 9 adet girdi ile yazılım proje efor tahmini yaptıkları çalışmalarında, 8 gizli nörondan oluşan bir gizli katman ile en iyi yapıya ulaşırken, Tronto ve diğerleri (2008) yazılım geliştirme eforunu 17 girdi ile tahmin ettikleri modele ait en iyi yapıyı 23 gizli nöron ile elde etmiştir. Rijwani ve Jain (2016) çalışmalarında, yazılım eforunu 23 girdi ile tahminleyen bir FFNN ağ yapısı için, bir adet gizli katman kullanmış olup 2'den 20'ye kadar değişen gizli nöron sayıları ile denemeler yapmışlardır. En iyi ağ yapısına 10 gizli nöron ile ulaşmışlardır. Pai ve diğerleri (2013) 8 adet girdi ile yazılım proje eforu tahmin ettikleri model için, 2, 5, 10, 15, 20, 25 gizli nöron ile denemeler yapmış olup, gizli nöron sayısının 10'dan daha büyük olduğu durumlarda anlamlı bir iyileşme elde edemediklerini raporlamıştır. Rao ve Kumar (2015) 17 girdi değişkeni ile yazılım proje eforu tahmin ettikleri model ile ilgili, iki gizli katmanlı yapıyı tercih etmişlerdir.

Proje efor tahmini alanında, aktivasyon fonksiyonu sıklıkla ayarlanan bir diğer hiperparametredir. Gizli katman aktivasyon fonksiyonu olarak literatürde sıklıkla, sigmoid (Tronto ve diğerleri, 2008; Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Rijwani ve Jain, 2016), hiperbolik tanjant (Rao ve Kumar, 2015; Pospieszny ve diğerleri, 2018; Rankovic ve diğerleri, 2021), tansig (Goyal ve Bhatia, 2020) kullanılmıştır. Çıktı katmanı aktivasyon fonksiyonu olarak ise purelin (Goyal ve Bhatia, 2020) tercih edilmiştir. Öğrenme oranı ve momentum da sıklıkla kullanılan hiperparametreler arasındadır. Proje efor tahminine yönelik YSA modeli öneren bazı çalışmalarda öğrenme oranı 0.1 olarak seçilirken (Rao ve Kumar, 2015; Arora ve Mishra, 2018), Dave ve Dutta (2011) öğrenme oranını 0,85 olarak seçmiştir. Rao ve Kumar (2015) momentum değerini 0.5 olarak almıştır.

2.13. Yapay Sinir Ağı için Hiperparametre Belirleme Yöntemleri

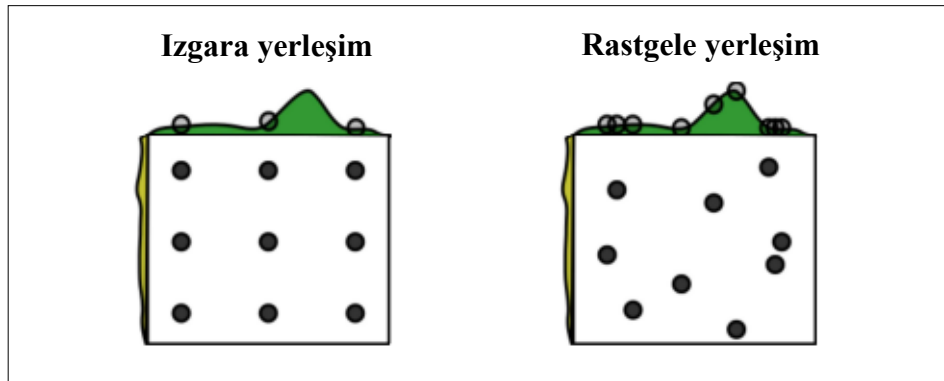
Proje efor tahmini alanındaki çalışmalarda, en iyi YSA yapısına ulaşmak için hiperparametre değerlerini bulma yöntemi olarak genelde, deneme yanılma yöntemi tercih edilmiştir.

YSA hata performansını eniyileyecek hiperparametre deęerlerini bulma problemi hiperparametre optimizasyonu olarak isimlendirilir ve denklem (2.13)'teki gibi ifade edilir:

$$x^* = \operatorname{argmin}_{x \in X} f(x) \quad (2.13)$$

Burada $f(x)$ en küçüklenmek istenen aęın hata deęeri olup, x^* ise $f(x)$ deęerini en küçükleyen hiperparametre deęerleridir.

HPO'nun kapalı denklem formunun bilinmedięi bir kara kutu (black-box) fonksiyonunun optimizasyon problemi olmasından dolayı, kara kutu problemlerinin çözümünde kullanılan birçok yöntem HPO'ya uygulanabilir (Hutter ve dięerleri, 2019). Tam faktöriyel tasarım (full factorial design) olarak da bilinen ızgara arama (grid search) en temel HPO yaklaşımıdır (Hutter ve dięerleri, 2019). Probleme özgü literatürde benzer konularda yapılan çalıřmalardan faydalanılarak ya da problem ile ilgili eęer önceden sahip olunan bilgiler varsa bunların ışığında hiperparametrelerin alabilecekleri deęerler için belirli ana deęerler belirlenebilir. Bu belirlenen ana deęerlerden oluşan ızgarada bulunan tüm noktaları, dięer bir deyiřle hiperparametrelerin tüm kombinasyonları için YSA eęitimi gerçekleştirilir. Izgara aramada her bir hiperparametre için belirlenen sonlu deęer kümelerinin kartezyen çarpımlarından oluşan kümedeki her bir eleman için deneylerin yapılması ve amaç fonksiyonlarının hesaplanması gerekir (Hutter ve dięerleri, 2019). Őekil 2.7'de solda görünen Őekilde gösterilen, tüm olası hiperparametre deęerlerine ait her bir kombinasyon için hesaplama yapılır.



Şekil 2.7. Izgara arama ve rastgele arama (Bergstra ve Bengio, 2012)

Izgara arama hesaplama süresi açısından en pahalı yöntemdir (Zheng, 2015). Bu yöntemde her hiperparametre kombinasyonu için hesaplama yapılır ve performans kriterine göre en uygun hiperparametre kombinasyonu seçilir (Zheng, 2015). Bunun yanında, küçük bir ızgara ile başlanıp, daha iyi YSA performansı elde edilen noktalarda ızgara genişletilebilir. Bu yöntem de el ile ızgara arama (manual grid search) olarak adlandırılır (Zheng, 2015). Deneyin her bir kombinasyon için yapılıyor olmasından dolayı, diğer bir deyişle bütün olası hiperparametre değerleri için YSA'nın eğitiminin tekrarlanması ve hata değerlerinin hesaplanması maliyet ve süre kaybına neden olmaktadır (Zheng, 2015). Izgara aramaya göre nispeten daha az maliyet gerektiren bir yöntem olan rastgele arama (random search) yönteminde, ızgarada belirlenen tüm köşe noktalar için hesaplama yapmak yerine, Şekil 2.7'de sağda görünen şekildeki gibi rasgele seçilen noktalar için hesaplama yapılır. Tüm kombinasyonlar için hesaplama yapılmadığından dolayı en iyi çözümü bulması açısından ızgara aramaya göre dezavantaja sahip olsa da, ızgara arama kadar başarılı olduğunu gösteren birçok çalışmanın olduğu belirtilmektedir (Zheng, 2015). Bergstra ve Bengio (2012) rastgele seçilen denemelerin ızgara aramaya göre daha etkin olduğunu göstermişlerdir.

Model karmaşıklığı arttıkça yukarıda belirtilen yöntemler süre ve maliyet açısından dezavantajlı olmaktadır ve son zamanlarda bu yöntemler yerine daha akıllı yöntemler önerilmiştir. Akıllı yöntemler, genelde rastgele bir başlangıç değerinden başlar ve ilerleyen denemedeki hiperparametre değerlerini seçerken önceki denemelerde elde edilen hata değerlerini kullanır (Zheng, 2015). Bu akıllı yöntemlerden biri olan Bayes Optimizasyonu (BO) ilk defa, Snoek ve diğerleri tarafından 2012 yılında yapılan bir çalışmada HPO problemlerini çözmek için önerilmiştir. Birçok araştırmada, bu yöntemin HPO alanında kullanılabilirliği vurgulanmıştır (Nguyen ve diğerleri, 2017; Makarova ve diğerleri, 2021). HPO probleminde, ayarlanacak hiperparametre sayısının artması, arama uzayını genişleteceğinden, ızgara ve rastgele arama gibi yöntemler süre ve maliyet açısından dezavantajlı olmaktadır. Bu yöntemi diğer optimizasyon yöntemlerinden ayıran en önemli özellik, hiperparametrelerin belirli değerleri ile gerçekleştirilen deneylerin sonuçlarının sonraki adımlardaki hiperparametre seçimlerinde kullanılmasıdır (Snoek ve diğerleri, 2012). Optimize edilmek istenen fonksiyonun en iyi değerini bulmak için birçok hesaplama yapmak yerine, bu hesaplamayı daha çok sıradaki deneyin hangi noktada

olacağını belirlemek için, sıradaki deneyde hangi parametrelerin kullanılacağını belirlemek için gerçekleştirir (Snoek ve diğerleri, 2012). Bu da fonksiyon hesaplama sayısı açısından BO'yu diğer tekniklere göre daha etkin kılar (Brochu ve diğerleri, 2010). HPO problemlerinin BO kullanarak çözümünde, amaç fonksiyonu YSA'nın hata performansı olup, buradaki amaç bu fonksiyonu en küçükleyen hiperparametre değerlerini bulmaktır. HPO'da, girdisi hiperparametreler olan ve çıktısı YSA'nın eğitim performansı olan, kapalı denklem formunun bilinmediği bir kara kutu fonksiyonunun optimizasyon problemi söz konusudur. Bu tip bir problemde BO'nun kullanılmasındaki amaç özellikle, her bir deneyin gerçekleştirilmesi ile hesaplanması oldukça maliyetli olan amaç fonksiyonunun en iyi değerinin, daha kısa süre ve maliyet harcanarak birkaç adımda bulunmasıdır (Nguyen ve diğerleri, 2017). Hiperparametrelerin seçilen değerleri ile modelin verdiği hata değerinin hesaplanması için gereken süreler deneyden deneye çeşitlilik gösterebilir (Snoek ve diğerleri, 2012). Örneğin gizli nöron birimi sayısı 10 olan bir ağı eğitimi, 1000 adet nöron sayısına sahip bir ağı eğitime göre daha az süre ve maliyet (öğrenme için gereken bellek büyüklüğü) alacaktır (Snoek ve diğerleri, 2012). Diğer global optimizasyon algoritmalarına göre oldukça iyi performans gösterdiği düşünülen BO'nun, hiperparametre optimizasyonu için iyi bir tercih olduğu düşünülmektedir (Snoek ve diğerleri 2012).

2.14. Bayes Optimizasyonu

BO'da, kapalı denklem formunun bilinmediği ve optimize edilmek istenen bir kara kutu fonksiyonu, f , optimize edilmeye çalışılır. Herhangi bir bilginin söz konusu olmadığı bu fonksiyonun alabileceği değerler, rassal değişken olarak modellenen ve bu değişkenlerin olasılık dağılımları bulunabilmektedir. BO'da bilinmeyen f fonksiyonunu tahmin etmek için kullanılan olasılıksal modele vekil model (surrogate model) denir. Vekil model, genellikle Gaussian süreci (Gaussian Process: GP) kullanılarak elde edilir (Sicard ve diğerleri, 2022). BO, bilinmeyen f fonksiyonunun GP'den örneklendiğini ve ilerleyen adımlarda da bu dağılımın aynı kalacağını varsayarak çalışır (Snoek ve diğerleri, 2012). GP, fonksiyonlar üzerinde bir olasılık dağılımını temsil eden bir olasılık modelidir.

Denklem (2.14)'te gösterildiği üzere, GP süreci f fonksiyonun herhangi bir t noktasındaki $f(x_t)$ değerlerin; ortalaması, μ , ve varyansı, σ^2 olan bir normal dağılım içinde olduğu varsayımına dayanmaktadır.

$$f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix} \sim N \quad (2.14)$$

BO'da vekil model oluşturma aşamasında, veri noktalarının kullanılması ile fonksiyonun ortalama ve kovaryansını hesaplayan bir GP modeli oluşturulur. BO genellikle çok boyutlu optimizasyon problemlerinde kullanıldığı için söz konusu GP'nin ortalama ve varyans değerleri çok boyutlu olup, çok değişkenli normal dağılım olarak tanımlanabilir (Ramussen ve Williams, 2006).

GP modelinin bir öncül (prior) dağılımı vardır. Öncül dağılım gözlem verisi toplamadan önce, optimize edilmek istenen amaç fonksiyonu hakkında sahip olunan ön bilgiyi temsil etmekte olup, bu dağılım genellikle bir normal dağılımdır. BO'nun her bir t adımında, elde edilen gözlem verileri ile bu dağılım güncellenir ve amaç fonksiyonu hakkında güncellenmiş olan sonsal (posterior) dağılım elde edilir (Brochu ve diğerleri, 2010). Kısaca, öncül dağılım amaç fonksiyonu hakkında önceden sahip olunan bilgiyi temsil ederken, sonsal dağılım gözlem verilerine dayanarak güncellenmiş bilgiyi temsil eder. Algoritma her bir adımda f fonksiyonu ile ilgili bilgiler elde edildikçe, sonraki adımlarda fonksiyonun alabileceği değerler ve olasılıklarını barındıran olasılıksal modeli iteratif olarak günceller. Çoğunlukla GP aracılığıyla oluşturulan vekil model sıralı olarak güncellenir (Nguyen ve diğerleri, 2017).

BO bu güncellemeyi yaparken, koşullu olasılık kavramından türetilmiş olan Bayes teoremini kullanmaktadır (Brochu ve diğerleri, 2010). B olayının gerçekleştiği koşulda A'nın olma olasılığı, $P(A|B)$ şeklinde gösterilir.

Bayes teoremi ile bu koşullu olasılık kavramından yola çıkılarak, iki koşullu olasılık arasındaki ilişki denklem (2.15)'teki gibi ifade edilir.

$$P(A | B) = (P(B | A) * P(A)) / P(B) \quad (2.15)$$

Denklemde $P(B | A)$, B olayının A olayı gerçekleştiği koşulda olma olasılığı, $P(A)$ ve $P(B)$ sırası ile A ve B olaylarının olma olasılıklarıdır. Bayes teoremi sonsal bilginin hesaplanmasında öncül bilgiyi kullanır. $P(A | B)$ ise sonsal dağılımını elde etmek için $P(B | A)$ koşullu olasılığı ve $P(A)$ öncül dağılımı kullanır. Bayes teoreminden yola çıkarak, BO'da amaç fonksiyonu f 'in t . adıma kadar gözlemlenen verilerin birikimi ile oluşan $D_{1:t} = \{x_{1:t}; f(x_{1:t})\}$ veri kümesi altında sonsal dağılımı, $P(f | D_{1:t})$, denklem (2.16)'daki gibi güncellenir.

$$P(f | D_{1:t}) = P(D_{1:t} | f) * P(f) / P(D_{1:t}) \quad (2.16)$$

Burada $P(f)$, f amaç fonksiyonunun öncül olasılık dağılımıdır. $P(D_{1:t})$ ise $D_{1:t}$ veri kümesinin önceden bilinen olasılık dağılımını ifade eder ve genellikle BO'da sabit bir değer olarak alınır. Gözlemler $D_{1:t}$ kümesinde biriktikçe, amaç fonksiyonunun öncül dağılımı $P(f)$, olasılık fonksiyonu $P(D_{1:t} | f)$ ile birleştirilir (Shin ve diğerleri, 2020). 1'den t . adıma kadar biriken gözlemlerin oluşturduğu $D_{1:t}$ kümesi bilindiğinde f fonksiyonunun koşullu olasılık değeri $P(f | D_{1:t})$ hesaplanarak, sonsal dağılım denklem (2.17)'deki gibi elde edilir.

$$P(f | D_{1:t}) \propto P(D_{1:t} | f) P(f) \quad (2.17)$$

Öncül olasılık dağılımı başlangıçta, örneklem uzayında rastgele noktalar seçilerek veya önceden belirlenmiş bir dağılım kullanılarak elde edilebilir. Sonsal olasılık dağılımı ise, öncül olasılık dağılımına ve edinim fonksiyonu (acquisition function) tarafından önerilen bir sonraki noktanın fonksiyon değerini kullanarak, örneklem uzayındaki her noktanın fonksiyon değerinin güncellenmiş olasılıklarını ifade eder. Edinim fonksiyonu, α , optimize edilmek istenen f fonksiyonunun dağılımını gösteren vekil model kullanılarak,

hangi noktanın amaç fonksiyonunun en iyilemesine ne kadar katkı sağlayacağını ölçümlemek için kullanılır. Literatürde sıklıkla kullanılan edinim fonksiyonu Beklenen Gelişme (Expected Improvement: EI) fonksiyonudur (Nguyen ve diğerleri, 2017). Buradaki amaç, sıradaki deneyi gerçekleştirmek üzere en çok kazanım elde edilecek noktanın, x_t , denklem (2.18)'deki gibi seçilerek sıradaki deneyin o noktada gerçekleştirilmesidir.

$$x_t = \operatorname{argmax}_x \alpha(x|D_{1:t-1}) \quad (2.18)$$

BO algoritmasının her bir iterasyonunda, seçilen yeni bir nokta ile, amaç fonksiyonunun değeri gözlemlenir. Bu gözlem değeri, öncül olasılık dağılımı ve edinim fonksiyonu ile birlikte Bayes teoremi kullanılarak sonsal olasılık dağılımı güncellenir. Bayes teoremi, bu şekilde, BO'da, amaç fonksiyonunun en iyi değerinin bulunmasına yardımcı olur.

BO prosedürü için sözde kod Çizelge 2.2'de verilmiştir. Algoritma başlangıçta örneklem uzayından rastgele bir x_0 değeri seçer. Bu değer ve ilgili gözlemlenen amaç fonksiyonu değerinden, $f(x_0)$, oluşan başlangıç veri kümesi denklem (2.19)'daki gibi ifade edilmektedir.

$$D_0 = \{x_0, y_0\} \quad (2.19)$$

f fonksiyonun öncül dağılımının, $P(f)$, GP sürecinden geldiği varsayılarak algoritma başlatılır. HPO'da öncelikle hiperparametrelerin olası değerlerinin aralıkları belirlenir ve rasgele bir hiperparametre kombinasyonu, x_0 , seçilir. Bu değerler kullanılarak YSA eğitilir ve hata performansı değeri, y_0 , hesaplanır. İlk gözlem sonucu, $\{x_0, y_0\}$, kullanılarak $f(x)$ amaç fonksiyonunun olasılık dağılımı hesaplanır.

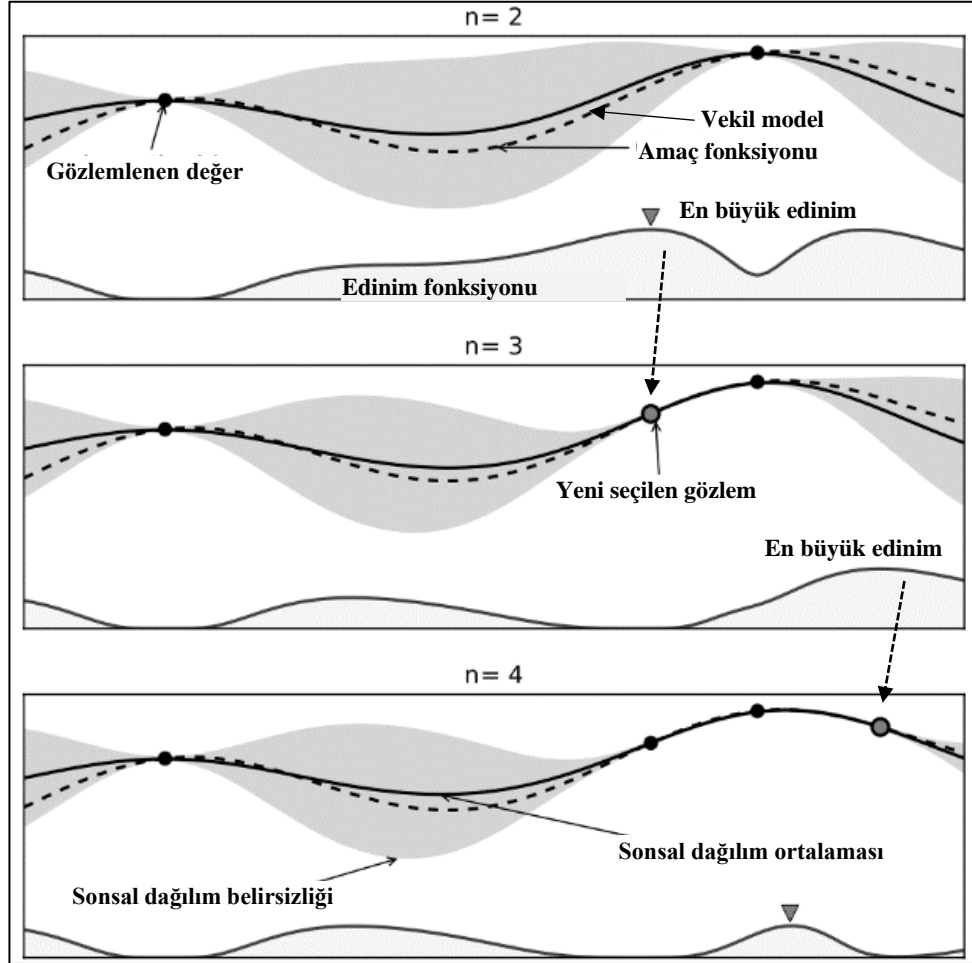
Çizelge 2.2. BO sözde kodu (Sicard ve diğerleri, 2022)

Bayes Optimizasyonu
for $t = 1, 2, \dots, k$ do
Adım 1: α edinim fonksiyonunu iyileştirecek yönde yeni x_t seç: $x_t = \operatorname{argmax}_x \alpha(x D_{1:t-1})$
Adım 2: x_t noktasında gözlem yap ve $f(x_t)$ hesapla
Adım 3: Veri setini çoğalt: $D_{1:t} = \{D_{1:t-1}, (x_t, f(x_t))\}$
Adım 4: Vekil modeli güncelle: $P(f D_{1:t}) \propto P(D_{1:t} f)P(f)$
end for
Amaç fonksiyonunu minimize eden en iyi x değerleri elde edilir: $x^* = \operatorname{argmin}_{x \in X} f(x)$

HPO probleminde bu girdiler, x_t , hiperparametrelerin t . adımda seçilen değerleri ve amaç fonksiyonu ise $f(x_t)$, seçilen hiperparametre değerleri ile ağ eğitildiğinde elde edilen hata performans değeridir.

Şekil 2.8’de örnek olarak BO’nun üç iterasyonu gösterilmiştir. İlk şekil ikinci iterasyonda elde edilen amaç fonksiyonunun olasılıksal modelin (vekil modelin) ortalama ve güven aralığı tahminleri görülmektedir. Bu adıma kadar, iki adet nokta için gözlemlenen değer hesaplanmıştır. Siyah kesikli çizgi amaç fonksiyonudur. Buna göre bu adımdaki, fonksiyonlar için yapılan tahminlerin ortalaması siyah çizgi ile gösterilmiştir. Siyah çizgi etrafında koyu gri ile gösterilen alan ise vekil modelin tahmin belirsizliğini ifade etmektedir. Gözlemlenen iki nokta ile ilgili herhangi bir belirsizlik yoktur, çünkü daha önce bu noktalardan örnekleme yapılmıştır. Şeklin alt tarafında açık gri ile renklendirilmiş edinim fonksiyonu görülmektedir. Edinim fonksiyonu, modelin amaç fonksiyonunu yüksek olarak öngördüğü (exploitation, sömürme) ve belirsizliğin yüksek olduğu yerlerde (exploration, keşif) yüksek değer almaktadır. Diğer bir deyişle, GP’nin ortalama ve standart sapmasının yüksek olduğu yerde, edinim fonksiyonu yüksek değer almaktadır. Edinim fonksiyonu temel alınarak, seçilen sıradaki noktanın gözlemlenen amaç fonksiyonu hesaplanır (Çizelge 2.2’de Adım 1). HPO’da bu gözlemlenen değer t . iterasyonda seçilen hiperparametre ayarı ile YSA’nın eğitimi sonucu elde edilen hata performansı olarak düşünülebilir (Adım 2). t . adımda gözlemlenen $f(x_t)$ değeri $D_{1:t-1}$ kümesine eklenerek, küme $D_{1:t}$ güncellenir (Adım 3). Bayes teoremi kullanılarak f fonksiyonuna ilişkin öncül dağılım kullanılarak sonsal dağılım güncellenir (Adım 4) ve

iterasyon 3'e geçilir. Şekil 2.8'de ikinci şekilde yeni seçilen gözlem değeri ve amaç fonksiyonu için güncellenen vekil dağılım görülmektedir. Çizelge 2.2'deki algoritmanın 1-4 arası adımları BO'nun durdurma kriterine kadar tekrarlanır.



Şekil 2.8. BO prosedürü için bir örnek – artarda 3 iterasyon (Brochu ve diğerleri, 2010)

2.15. Proje Efor Tahmininde Kullanılan Performans Değerlendirme Kriterleri

Proje efor tahmini için önerilen YSA modellerinin performansını, diğer bir deyişle YSA çıktısının beklenen çıktıya ne derece uyduğunu, değerlendirmek için literatürde birçok kriter kullanılmıştır. Ortalama bağıl hata büyüklüğü (Mean Magnitude Relative Error, MMRE), ortalama tahmine göre bağıl hata büyüklüğü (Mean of Magnitude Error Relative to the Estimate: MMER) ve kestirim düzeyi (Prediction Level: $PRED(\%x)$) proje efor tahmini alanında YSA model doğruluğunu değerlendirmek için sıklıkla kullanılan

tekniklerdir (Pospieszny ve diğeri, 2018; Kumar ve diğeri, 2020). MMRE değeri diğeri disiplinlerde ortalama mutlak yüzde hatası (Mean Absolute Percentage Error, MAPE) olarak geçmekte olup (Tofallis, 2015), proje efor tahmini alanında daha çok MMRE terimi kullanılmaktadır. MMRE değerini hesaplamak için öncelikle ortalama bağıl hata (Mean Relative Error, MRE) değeri denklem (2.20) kullanılarak hesaplanır. Sonrasında, her bir i projesine ait MRE değerleri üzerinden, MMRE değeri denklem (2.21) kullanılarak hesaplanır.

$$MRE_i = \left(\frac{|Y_i - \hat{Y}_i|}{Y_i} \right) \quad (2.20)$$

Y_i : Gerçekleşen efor değeri

\hat{Y}_i : Tahmini efor değeri

$$MMRE = \left(\frac{1}{N} \right) \sum_{i=1}^N MRE_i \quad (2.21)$$

N : örnek sayısı

Proje efor tahmini literatüründe bir diğeri sıklıkla karşılaşılan değerlendirme kriteri MMER'dir (Nassif ve diğeri, 2013). Bazı yazarlar MRE yerine MER'nin daha iyi sonuçlar verdiği dikkat çekmişlerdir (López-Martín ve diğeri, 2011). MMER değerinin hesaplanması için öncelikle her bir i projesi için ortalama tahmine göre bağıl hata (Mean Of Magnitude Of Relative Error, MER) denklem (2.22) kullanılarak hesaplanır. Her bir i projesine ait MER değerleri üzerinden, MMER değeri denklem (2.23) kullanılarak hesaplanır.

$$MER_i = \left(\frac{|Y_i - \hat{Y}_i|}{\hat{Y}_i} \right) \quad (2.22)$$

Y_i : Gerçekleşen efor değeri

\hat{Y}_i : Tahmini efor değeri

$$MMER = \left(\frac{1}{N}\right) \sum_{i=1}^N MER_i \quad (2.23)$$

N : örnek sayısı

Kestirim düzeyi, $PRED(x)$ ise belirlenen x değerine eşit ya da küçük MRE ya da MER değerlerinin ortalaması alınarak hesaplanır (Nassif ve diğerleri, 2013). Hesaplama MRE kullanıldığında, proje eforunun gerçekleşen efor değerinin en fazla yüzde x 'i kadar sapma ile tahminlendiği yüzde miktarıdır (Tronto ve diğerleri, 2008). $PRED(x)$ değerinin hesaplanması denklem (2.24)'te verilmiştir.

$$PRED(x) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{eğer } MER_i \leq x \\ 0 & \text{diğer durumlarda} \end{cases} \quad (2.24)$$

YSA modelinin tahmin doğruluğu $PRED(\%x)$ değeri ile doğru orantılı olup, MMER veya MMRE değerleri ile ters orantılıdır (Nassif ve diğerleri, 2013). Birçok çalışmada $x = \%25$ olarak kullanılır (Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Araujo ve diğerleri, 2017; Pandey ve diğerleri, 2020). Örneğin, yazılım proje efor tahmini alanında $MMRE \leq 0.25$ ve $PRED(\%25) \geq \%75$ değerleri YSA modelinin tahmin doğruluğu için yeterli olarak kabul edilir (Pospieszny ve diğerleri, 2018).

3. MATERYAL ve YÖNTEM

Tez çalışmasının bu aşamasında, önerilen YSA modelleri için kullanılan veri seti, seçilen girdiler, çıktılar ve ağın hiperparametre değerlerinin optimizasyon yöntemi ile ilgili açıklamalara yer verilmiştir.

3.1. Kullanılan Veri Seti

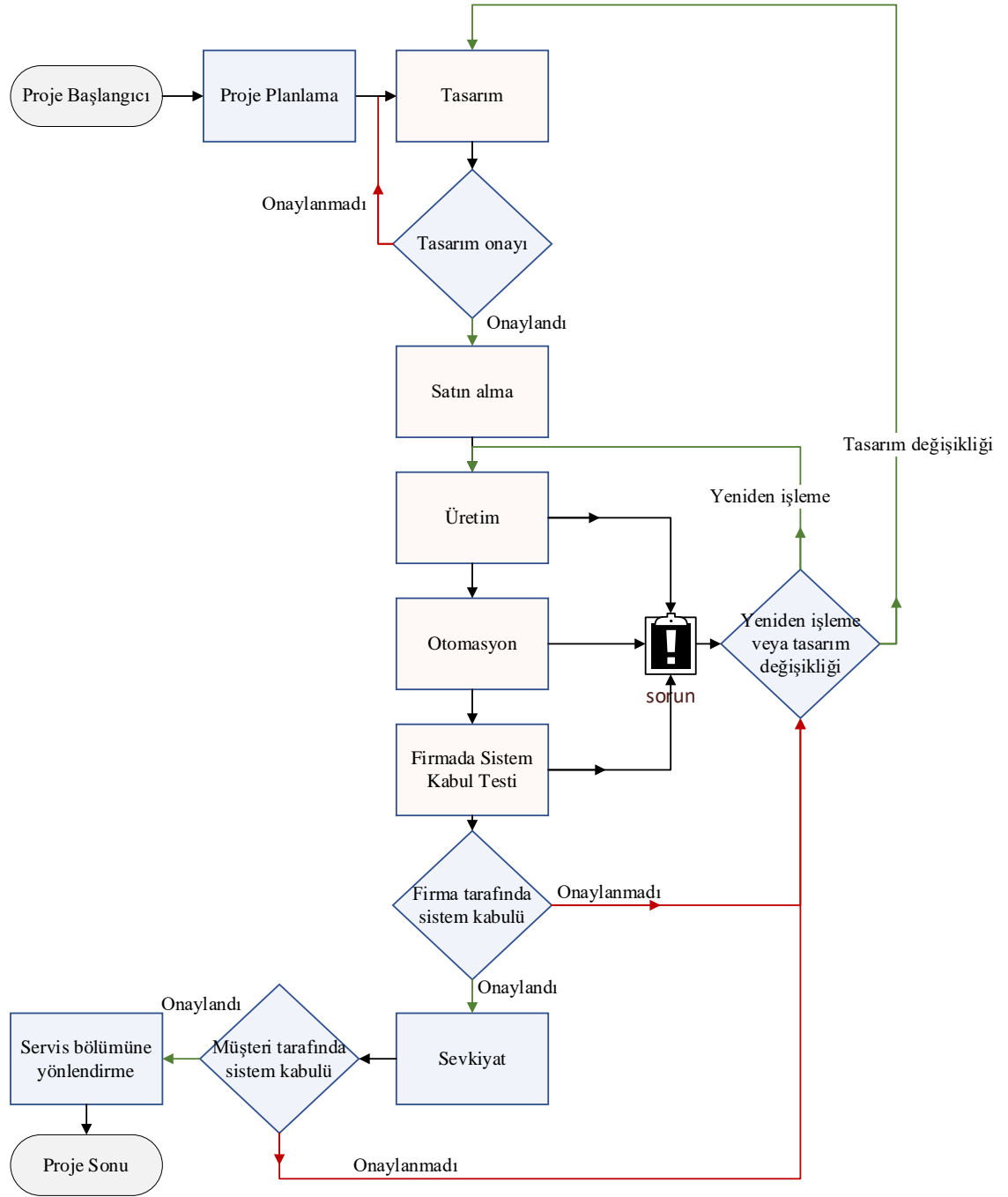
Tez kapsamında, proje efor tahmini için geliştirilen YSA modelleri, fotonik sektörüne özelleştirilmiş makine üreten bir firmada tamamlanmış olan projelerden toplanan veriler ile geliştirilmiştir. Literatürde proje efor tahmini ile ilgili geliştirilen YSA modellerinin eğitiminde genellikle mevcut veri tabanları kullanılmıştır. Yazılım projeleri için özellikle COCOMO veri tabanına birçok çalışmada rastlanmıştır (Attarzadeh ve Ow, 2011; Tronto ve diğerleri, 2008; Rankovic ve diğerleri 2021; Rijwani ve Jain, 2016). Bunun yanında, yazılım projelerinin efor tahmini ile ilgili çalışmalarda, NASA (Attarzadeh ve diğerleri, 2012; Rankovic ve diğerleri, 2021), Kemerer (Heiat, 2002; Rankovic ve diğerleri, 2021), ve Desharnais (Predescu ve diğerleri, 2019, Rankovic ve diğerleri, 2021; Carvalho ve diğerleri, 2021) veri tabanlarına sıklıkla rastlanmıştır. Bu tez çalışması kapsamında önerilen YSA modelleri, fotonik sektöründe makine üretimi yapan bir firmadaki uzmanlar ile gerçekleştirilen görüşmeler ile toplanan veriler ile geliştirilmiştir. Dolayısıyla bu tez çalışmasının, literatürde yer alan çalışmalardan farklı bir sektörde yapılması ve gerçek hayat verisinin kullanılması açısından literatüre katkı sağladığı düşünülmektedir.

Tez kapsamında, fotonik cihaz üretimi alanında kullanılmak üzere müşteriye özel mikro-montaj ve test makinelerinin üretimi projelerinin efor tahmini üzerinde çalışılmıştır. Veri setinin elde edildiği firma, fotonik sektöründe kullanılmak üzere müşteriye özgü mikro-montaj ve test makine üretimi yapmakta olup, merkezi Almanya'dadır ve 500'den fazla çalışana sahiptir. Birçok ülkede tesisleri bulunan firma, bugüne kadar 900'den fazla sistem teslim etmiştir. Firma, Ar-Ge'den yüksek hacimli üretime kadar birçok farklı üretim tipine çözümler sunmaktadır. Firmada üretilen makineler, datakom, telekomünikasyon, otomotiv, uzay ve havacılık sektörleri gibi birçok alanda

kullanılmaktadır. Firma bünyesinde, mikro lens, fiber, lazer, silikon çip ve sensör gibi fotonik ve optik bileşenlerin; yüksek güçlü lazer diyotlar, alıcı-vericiler (*transceiver*) gibi fotonik cihazları oluşturmak üzere hizalanması, montajlanması ve testlerinde kullanılacak olan makinelerin üretimi gerçekleştirilmektedir.

Firmada gerçekleştirilen projeler kapsamında, müşterinin fotonik ve optik bileşenler içeren ürününün üretim sürecinin otomatik hale getirilmesini sağlayacak makineler üretilmektedir. Makine özetle, üretimi gerçekleştirmek üzere gereken yer değiştirmeleri sağlayacak hareket sistemleri, ürün bileşenlerini tutmaya yönelik tutucular (pick-up tool), bileşenlerin hareketlerini gözlemlemek amacıyla görüntü sistemlerinden oluşmaktadır. Makinenin ana sistemi, makine kasası çıkarıldığında kalan kısım, müşteri ürününe göre farklılık göstermekte olup, üretim sürecine göre değişen ek özelliklere sahip olabilmektedir (optik bileşenlerin yapıştırılması için gereken epoksi yapıştırma sistemi, ürün montajlanması sırasında ürünün tutulacağı tabla, ısıtma ve soğutma sistemleri, araç değiştirme sistemleri gibi). Ayrıca makineye müşteri gereksinimine göre; güç ölçer, spektrum ölçer, ek soğutucu, lazer sürücü gibi cihazlar entegre edilebilmektedir. Makinede bulunan hareket, görüntü sistemlerinin, ek cihazların koordineli bir şekilde çalışmasını sağlayan yazılım, firma tarafından geliştirilmiş olup entegrasyonu otomasyon aşamasında sağlanmaktadır.

Tez kapsamında ele alınan projelerde üretilen makineler, müşteri ürün ve sürecini otomatikleştirmeye yönelik geliştirilmektedir. Firmada gerçekleştirilen makine geliştirme projeleri tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşmaktadır. Proje aşamaları ile ilgili akış şeması Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Makine geliştirme projesi adımları

Proje planlama aşamasından sonra proje, tasarım aşaması ile başlar. Bu aşamada makinenin mekanik, elektronik, mikro-optik bileşenler ve alt montaj parçaları müşteri ihtiyaçlarına göre tasarlanır. Müşteri ürün ve süreci ile ilgili gereksinimler göz önünde bulundurularak ve müşteri ürününün çizimleri gibi gerekli dokümanlar kullanılarak

makineye yönelik sistem/alt sistem çizimleri gerçekleştirilir. Tasarımın müşteri ile gözden geçirilmesi ve müşteri onayından sonra, ürün reçetesi (sistem ve alt sistemler için gereken malzeme listesi) satın alma takımına iletilir. Gereken bileşenler ve alt montaj parçaları için tedarikçi araştırması ve seçimi ile devam edilir.

Üretim planına göre gereken bileşenler gerektiği zamanlarda temin edilir ve makinenin mekanik, elektronik ve mikro-optik bileşenlerinin tasarım dokümanlarına ve entegrasyon planına göre montajlanması gerçekleştirilir. Son olarak makine ile entegre olarak çalışacak ekipmanlar ve cihazlar da makine ile entegre hale getirilerek testleri yapılır ve makine otomasyon için hazır hale getirilir. Bazı durumlarda, müşteriler daha önceden sipariş verdikleri makinenin aynısını tekrardan sipariş verebilir ya da aynı tip makineden bir kalemde birden fazla sipariş verebilir ve talep edilen proje hızı yüksek olabilir. Bu durumda, makine dış kaynak kullanılarak üretilebilir. Makinenin ana hatları dış kaynak tarafından üretilmiş olsa da tekrardan üretim personeli tarafından gözden geçirilir ve bazı düzeltmeler gerekebilir. Bu tip projelerde üretim eforunun nispeten daha az olması beklenir. Ya da başka bir proje için, başka bir amaçla daha önceden üretilmiş fakat sipariş iptali, ürün veya süreçte değişiklik gibi birtakım nedenlerden ötürü kullanılmamış olan makinenin, tekrardan ele alınıp farklı fonksiyonlar eklenip başka tip bir makineye dönüştürülmesi söz konusu olabilir. Bu tip durumlarda harcanan üretim eforu, sıfırdan makine üretimine göre daha az olabilir.

Otomasyon aşamasında, makinenin müşteri ürününün otomatik olarak üretilmesi için gereken süreç geliştirilir. Daha önceden geliştirilmiş olan genel makine süreci kontrol yazılımı, ilgili makinenin süreçlerini kontrol etmeye yönelik düzenlenir ve üretim süreci ile ilgili adımlar tanımlanır. Müşterinin manuel olarak gerçekleştirdiği üretim sürecini otomatikleştirecek makine geliştirilir. Müşterinin manuel olarak gerçekleştirdiği kendi ürününe özgü olan üretim sürecinin makine tarafından nasıl otomatikleştirileceği konusu çoğunlukla yüksek düzeyde belirsizlik içerir. Makine tasarımı veya otomasyonu sırasında makinenin kabiliyetleri göz önünde bulundurularak, müşterinin üretim sürecinde değişikliklere gidilebilir. Bazı durumlarda ise, üretim süreci firmanın daha önce geliştirdiği makinelerin üretim süreci ile benzer olabilir. Bu tip durumlarda gereken otomasyon eforu daha az olabilir. Sonuç olarak gereken efor müşteri ürünü ve üretim

sürecine göre farklılıklar gösterebilir. Otomasyon aşaması ayrıca firma bünyesinde gerçekleştirilecek olan kabul testi için makinenin hazır hale getirilmesi, birtakım testlerinin gerçekleştirilmesi süreçlerini de içermektedir. Başarılı bir kabul testi sonrasında makine, son kontrolleri yapılarak müşterinin tesisine sevk edilir. Müşteri tesisinde genelde firma bünyesinde yapılan test tekrarlanarak, saha kabul testi gerçekleştirilir. Bu test ortalama bir hafta sürmekte olup, sınırlı bir süre olması ve bu sürecin efor kaydının sağlıklı olarak gerçekleştirilmemesinden dolayı bu kısım tez kapsamına alınmamıştır.

Tez kapsamında yukarıda bahsi geçen tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşan makine geliştirme projesi için gereken toplam efor değerini tahmin eden bir model geliştirilmiştir. Bu modelde çıktı değişkeni, projenin toplam eforu olup, kişi-saat cinsinden ölçümlenmiştir. Bunun yanında, projenin yüksek belirsizlik içeren ve gereken efor açısından öne çıkan aşamalarından biri olan otomasyon aşaması için gereken efor değerini de tahmin etmek üzere bir tahmin modeli geliştirilmesi amaçlanmıştır. Tez kapsamında geliştirilen tahmin modelleri ve çıktı değişkenleri Çizelge 3.1’de gösterilmiştir.

Çizelge 3.1. Tahmin modelleri ve çıktı değişkenleri

Model No	Model Adı	Çıktı Değişkeni
Model 1	Otomasyon Eforu Tahmin Modeli	Otomasyon Eforu (kişi-saat)
Model 2	Toplam Proje Eforu Tahmin Modeli	Toplam Proje Eforu (kişi-saat)

Efor tahmini modellerinin geliştirilmesinde kullanılan veri seti, firmada gerçekleştirilen 101 adet tamamlanmış proje kullanılarak elde edilmiştir. Firma bünyesinde çalışanlar tarafından, atandıkları her bir proje için harcadıkları efor değerleri günlük olarak sisteme girilmektedir. Toplam projeye harcanan ve otomasyon aşamasına harcanan efor değerleri, girişler ile ilgili olası hatalar göz önünde bulundurularak, uzman kişilerle birlikte gözden geçirilmiş olup çıktı veri seti oluşturulmuştur.

101 adet proje Bölüm 3.2’de detaylandırılmış olan makine özellikleri ile ilgili 11 girdi değişkeni açısından uzmanlar ile değerlendirilerek, girdi verileri elde edilmiştir.

3.2. Önerilen Tahmin Modelleri için Girdi Seçimi

Firmadaki uzmanlar ile yapılan beyin fırtınası seansları sonucu ve literatürdeki çalışmalarda kullanılan faktörler de göz önüne alınarak proje toplam eforu ve otomasyon aşaması için gereken eforu tahmin etmek üzere belirlenen 11 adet girdi Çizelge 3.2’de seviyeleri ile verilmiştir.

Çizelge 3.2. Proje efor tahmini için belirlenen faktörler ve seviyeleri

No	Faktör Adı	Faktör Seviyesi
1	Donanım karmaşıklığı	1-Düşük 2-Yüksek
2	Süreç karmaşıklığı	1-Düşük 2-Yüksek
3	Müşteri tipi	1-Eski 2-Yeni
4	Müşteri ürünü yeniliği	1-Değişiklik yok 2-Güncellenmiş 3-Yeni
5	Müşteri ürün süreci yeniliği	1-Değişiklik yok 2-Güncellenmiş 3-Yeni
6	Makine konfigürasyonu yeniliği	1-Değişiklik yok 2-Güncellenmiş 3-Yeni
7	Makine ana fonksiyonu	Genel montaj, fiber montaj, ısıl bağlama, müşteriye özel veya test
8	Makine ana sistem büyüklüğü	6 kategori
9	Makine kasa büyüklüğü	6 kategori
10	Makinenin hat tipi	1-Giriş düzeyi 2-Tek başına 3-Hat içerisinde
11	Makine üretim sürecinde dış kaynak kullanımı	1-Dış kaynak kullanımı var 2-Dış kaynak kullanımı yok

İlk girdi donanım karmaşıklığı olup, makinenin mekanik tasarım açısından ne kadar karmaşık olduğu ile ilgilidir. Bu faktöre ait seviyeler düşük ve yüksek olarak belirlenmiştir. Makineler girdi 7’deki ana fonksiyonlarına göre kategorilere ayrılmaktadır. Makine donanımı ile ilgili, her bir kategori için ayrı ayrı donanım özelliklerinin (hareket, görüntü, veri edinimi gibi) hangi durumlarda yüksek hangi durumlarda düşük olduğuna dair ayrıntılı açıklamalar belirlenmiştir. Fakat, bilgi gizliliğinden dolayı burada yer verilmemiştir. Örnek olarak genel montaj makine tipi için açıklama yapılmıştır. Girdi 7’nin ilk seçeneğinde yer alan genel montaj makineleri, mikro lenslerin aktif olarak hizalanması ve montajlanmasını içeren fotonik cihaz üretimi

gerçekleştirilen sistemlerdir. Bu tip bir makinede genelde 6 eksenli hareket sistemi, görüntü, epoksi yapıştırma ve kütleme sistemleri yer aldığı düşük karmaşıklık seviyesi seçilir. Eğer bu tip bir makine birden fazla 6 eksenli ya da daha farklı hareket sistemleri bulunuyorsa veya araç değiştirme, *wafer* tablası gibi müşterinin isteği üzerine ek özellikler içeriyor ise yüksek karmaşıklık düzeyi seçilir.

İkinci girdi süreç karmaşıklığı olup, makinenin gerçekleştireceği sürecin, diğer bir deyişle müşteri ürününün makine tarafından üretilme sürecinin ne kadar karmaşık olduğu ile alakalı olup, düşük ve yüksek olmak üzere iki seviye ile tanımlanmaktadır. Müşteri ürünü birden fazla farklı süreç gerektirebilir. Bazı süreçler teknolojinin olgunluk düzeyinden dolayı ve firmanın sürece aşına olmamasından dolayı daha karmaşık ve zor olabilir. Makine süreci ile ilgili de düşük ve yüksek düzeyler için herhangi bir öznel değerlendirmeyi engellemek amacıyla belirli ifadeler belirlenmiştir. Bilgi gizliliğinden dolayı burada sadece bir makine tipi için açıklama yapılmıştır. Örneğin girdi 7’de 2. seçenek olan fiber montaj makinelerinde, eğer tek bir fiberin silikon çipe ya da müşteri ürününe hizalanması söz konusu ise, süreç karmaşıklığı düşük seçilir. Fakat fiberin çipe hizalanıp, sonrasında da bu alt montaj ürününün müşteri ürününe hizalanması gibi birden fazla hizalama ve montaj sürecini içeriyorsa yüksek karmaşıklık seçilir.

Sırası ile müşteri tipi, müşteri ürün yeniliği, müşteri süreç yeniliği ve makine konfigürasyonu yeniliği olan üçüncü, dördüncü, beşinci ve altıncı girdiler ise, makinenin üreteceği ürünün ve sürecin, daha önce aynı müşteriye ait ürüne ve üretim sürecine benzerliği ile alakalıdır. Diğer bir deyişle, makinenin üreteceği ürünün, sürecin ve makine konfigürasyonunun firmaya ne derece yeni olduğu ile ilgilidir. Eğer müşteri, firmadan daha önce en az bir defa makine satın almış ise, müşteri tipi eski olarak değerlendirilir. Müşteri ürün ve üretim sürecinin yeniliği için üç düzey belirlenmiştir: “değişiklik yok”, “güncellenmiş” ve “yeni”. Değerlendiriciler için bu durumlara ait tanımlamalar ve örnekler belirlenmiş olup, bilgi gizliliğinden dolayı Çizelge 3.3’te bu bilgilere genel ifadeler ile yer verilmiştir.

Çizelge 3.3. Müşteri ürünü, süreci ve makine konfigürasyonu yeniliği girdileri ile ilgili tanımlamalar

Girdi Adı	Düzy ve Açıklama		
	1-Değişiklik yok	2-Güncellenmiş	3-Yeni
Girdi 4 Müşteri ürünü yeniliği	Müşteri ürününde önemli düzeyde bir değişiklik yok	Müşteri ürününde aşağıdaki gibi güncellemeler söz konusu ise: - Müşteri ürününe yerleştirilecek olan optik bileşenin büyüklüğünde güncelleme, - Optik bileşen sayısında değişiklik, - Farklı bir optik bileşenin yine aynı müşteri ürününe hizalanması.	Eski ya da yeni müşterinin tamamen yeni bir ürünü söz konusu. (Örn. eski bir müşteri daha önce lazer diyot ürünü için makine ürettirmiş, şu anki siparişte otomotiv Lidar ürünü ürettirmek üzere makine talep ediyor ise)
Girdi 5 Müşteri ürün süreci yeniliği	Müşteri ürününe ait üretim sürecinde önemli düzeyde bir değişiklik yok	Müşteri ürününe üretim süreci ile ilgili aşağıdaki durumlar söz konusu ise: - Üretim adımlarının sırasında değişiklik, - Üretim hızında artış talebi, - Daha önce bu üretim sürecine benzer bir üretim sürecinin başka bir makine için ya da ar-ge amaçlı geliştirilmiş olma durumu.	Eski ya da yeni müşterinin ürününe üretim süreci için aşağıdaki durumlar söz konusu ise: - Tamamen yeni bir süreç geliştirilme durumu, - Müşteri ürünüdeki güncellemelerden kaynaklı üretim sürecinde değişimler, - Aynı tip ürüne farklı tipte bir bileşenin hizalanması, montajlanması, - Daha önce bu tipte bir ürüne ar-ge amaçlı bir makine geliştirilmiş fakat süreç geliştirilmemiş olma durumu.
Girdi 6 Makine konfig. yeniliği	Makine konfigürasyonu daha önce aynı tipte bir müşteri ürünü ve süreci için geliştirilmiş ve iyileştirmeye gerek olmadan tekrardan aynı şekilde üretilecek ise	Makine konfigürasyonunda aşağıdaki sebeplerden dolayı güncelleme gerekiyor: - Müşteri tarafından daha önce aynı tipte bir makine alınmış, fakat makine konfigürasyonu ile ilgili iyileştirme talepleri söz konusu, - Müşteri ürünüdeki ve/veya üretim sürecindeki güncellemelerden dolayı makine konfigürasyonunda birtakım güncellemeler söz konusu, - Yeni bir müşteri ya da yeni bir ürün söz konusu, fakat daha önce benzer bir ürüne dair bir makine aynı ya da farklı bir müşteriye geliştirilmiş.	Aşağıdaki sebeplerden dolayı yeni bir makine konfigürasyonu gerektirme durumu: - Daha önce bu tipte bir müşteri ürünü ve süreci ile ilgili benzer bir makine konfigürasyonu geliştirilmemiş, - Daha önce aynı müşteri ürününe farklı bir optik bileşen (lens) hizalama makinesi üretilmiş, şu anki siparişte farklı bir optik bileşen (fiber) hizalanması talep ediliyor, - Daha önce iki uygulama (örn. lens ve fiber hizalama) için bir makine üretilmiş, şu anki siparişte bu uygulamalar için ayrı ayrı makineler isteniyor ise.

Eğer müşteri, firmadan daha önce aynı tipte bir makine satın almış ise ve müşteri ürün/üretim sürecinde anlamlı bir değişiklik yok ise “değişiklik yok” düzeyi seçilir. Eğer

müşteri için daha önceden aynı tipte bir makine üretilmiş fakat müşteri ürününde ve üretim sürecinde bir iyileştirme veya bir değişiklik var ise "güncellenmiş" seçeneği seçilir. Bu durumda özellikle otomasyon aşamasında, müşteri ürününün üretim süreci ile ilgili değişiklikten dolayı ek efor gerekebilir. Fakat daha önceden benzer tipte makine geliştirildiği için, bu eforun, tamamen yeni bir ürün geliştirmek için harcanacak olan efordan daha az olması beklenir. Eğer müşteri tamamen yeni ve firma daha önce bu tipte bir ürün için bir makine geliştirmemiş ise "yeni" seçeneği seçilir. Makine konfigürasyonu yeniliği faktörü ise müşteri ürün ve üretim süreci yeniliği ışığında geliştirilecek olan sistem konfigürasyonunun, alt sistemlerin yenilik düzeyi ile ilgili olup, "değişiklik yok", "güncellenmiş" ve "yeni" düzeyleri ile tanımlanır. Müşteri ürünü, üretim süreci ve makine konfigürasyonu (girdi 4, girdi 5 ve girdi 6) girdilerine dair açıklamalara Çizelge 3.3'te yer verilmiştir.

Makine ana fonksiyonu olan yedinci girdi, makinenin gerçekleştireceği ana fonksiyonuna göre makinenin tiplere ayrılması sonucu oluşturulmuştur: genel montaj, fiber montaj, ısıl bağlama, müşteriye özel veya test makineleri gibi. Makine ana fonksiyonu girdisinin düzeyleri ile ilgili açıklamalar Çizelge 3.4'te yer almaktadır.

Çizelge 3.4. Makine ana fonksiyonu girdi değişkeninin aldığı değerlerin tanımları

Düzyey	Açıklama
1 Genel montaj	Fotonik cihazların çoklu eksen hareket sistemleri ile otomatik olarak hizalanmasının ve montajının firma tarafından geliştirilen bir yazılım ile kontrol edilerek yapıldığı makinelerdir. Mikro lensleri aktif olarak hizalama, esnek yapıştırma ve montajlama özelliklerine sahip makinelerdir. Otomatik araç değiştirme, test etme ve <i>wafer</i> işleme gibi ek özellikleri de müşterinin isteğine göre eklenebilir.
2 Fiber montaj	Yüksek-çözünürlükte, özellikle fiber-optik ve lazer bileşenlerden oluşan fotonik cihazların üretiminde kullanılan, fiberin fotonik ve optik bileşen içeren müşteri ürününe yapıştırılmasına yönelik üretilen makinelerdir. Genel montaj sistemlerdeki gibi, otomatik hizalama yapan çoklu-eksen hareket sistemlerini kullanır.
3 Isıl Bağlama	Lazer diyot, silikon çip gibi fotonik bileşenlerin mekanik olarak, ısıl işleme yüksek hassasiyetle hizalanması ve montajlanmasını gerçekleştiren sistemlerdir.
4 Müşteriye özel	Isıl montaj, <i>wafer</i> işleme ve test gibi fonksiyonları içeren müşteriye özel üretilen makinelerdir. Özellikle yeni ürün geliştirme için uyumludur.
5 Test	Lazer, silikon çip gibi fotonik bileşenlerin güç-akım-voltaj, spektral testleri ve muayeneleri için üretilen tam otomatik test sistemleridir.

Sekizinci girdi makinenin esas fonksiyonu gerçekleştirecek ana sisteminin büyüklüğü ile ilgilidir. Sistemin üretilecek ürün ve süreç karmaşıklığına göre içereceği alt sistemlerin karmaşıklık düzeyine göre firmanın belirlediği 6 adet kategori mevcut olup, bilgi gizliliğinden dolayı burada yer verilmemiştir. Dokuzuncu girdi ise makinenin iç kısmında bulunan alt sistemlerini koruyan dış kasasının büyüklüğü ile alakalıdır. Bu iki girdi ile ilgili büyüklük ölçmek adına 6 kategori belirlenmiştir. Makine dış kasa büyüklüğü girdisinin düzeyleri ile ilgili ifadeler Çizelge 3.5’te yer verilmiştir.

Çizelge 3.5. Makine dış kasa büyüklüğü girdisinin düzeyleri ile ilgili tanımlamalar

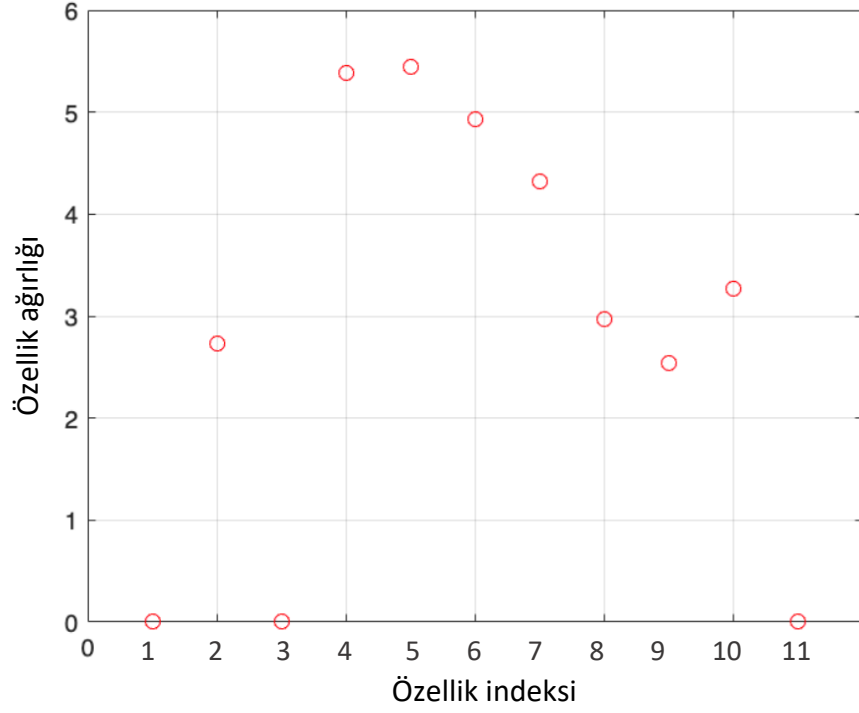
Düzyey	Açıklama
1 Çok Küçük-açık	Nispeten daha az karmaşık fotonik veya optik bileşenlerden oluşan alt montaj ürünlerin, Ar-Ge amacıyla üretimi için kullanılmak üzere tasarlanan makinelerde kullanılan kasa tipidir. Makine kasası üstü açık yapıdadır.
2 Çok Küçük-kapalı	Birden fazla fonksiyona sahip, daha düşük hacimli üretim gereksinimlerini, tek başına karşılayan makinelerde kullanılan kasa tipidir. Makine kasası tamamen kapalı yapıdadır.
3 Küçük	Birden fazla fonksiyona sahip karmaşık üretim gereksinimlerini, tek başına karşılayan ve gerektiğinde farklı makine tiplerinden oluşan bir fotonik cihaz üretim hattının bir hücresi olarak da kullanılabilen makinelerde kullanılan kasa tipidir. Makine kasası tamamen kapalı yapıdadır.
4 Orta	Yüksek hacimli üretimlerde, farklı makine tiplerinden oluşan bir fotonik cihaz üretim hattının bir hücresi olarak tasarlanmış makinelerde kullanılan kasa tipidir. Nispeten daha modern ve yeni formattır. Makine kasası tamamen kapalı yapıdadır.
5 Büyük	Yüksek boyutlu üretimlerde kullanılan, besleme (<i>feed-in/out</i>) sistemleri ile de çalışabilen, esnek fonksiyonlara sahip, modern ve büyük boyutlu sistemlerdir. Makine kasası tamamen kapalı yapıdadır.
6 Çok büyük	Çok karmaşık ürünlerin, müşteriye özel birden çok fonksiyonu gerçekleştiren, esnek fonksiyonlara sahip, çok büyük boyutlu sistemlerdir.

Onuncu girdi makinenin hat tipi ile ilgilidir. Giriş düzey makineler, nispeten daha az karmaşık ürünlerin, daha düşük hacimli üretim veya araştırma amacıyla üretimin yapıldığı durumlar olup; tek başına (standalone) makineler, kendi başına bir fonksiyonu (fiber hizalama, ısıl bağlama, test gibi) gerçekleştirebilen makinelerdir. Son olarak hat içerisinde makineler ise birden fazla farklı tipte makinenin birbiri ile entegre edilerek sunulması ile elde edilen hat içerisindeki makinelerdir.

Son girdi olan 11. girdi deęiřkeni ise makinenin satın alınan alt sistemleri ve parçalarının montajlanması aşamasında dış kaynak kullanımı ile ilgilidir. Eęer makine montajında dış kaynak kullanımı var ise ilk düzey seçilir. Bazı durumlarda daha önceden başka bir amaçla üretilmiş bir makinenin dönüřtürülmesi söz konusudur, bu tip durumlarda da ilk düzey seçilir. Dięer durum ise, “dış kaynak kullanımı yok”, makine montajının sıfırdan firma içerisinde gerçekleştirilme durumudur. Kullanılan dış kaynağın kabiliyetleri firmanın üretim kabiliyetinden farklılık göstereceęi için, üretim için gereken efor deęeri farklılık gösterebilir.

Toplam proje eforu için geliştirilen Model 2’de 11 adet deęiřkenin hepsi kullanılmıştır. Otomasyon eforu için geliştirilen Model 1’de ise, belirlenen 11 adet girdi ile (bağımsız deęiřkenler), otomasyon eforu çıktısının (bağımlı deęiřkeni) ne derece iyi tahmin edilebileceęi ile ilgili bir ön bilgi edinmek amacıyla, regresyon problemlerinde parametrik olmayan bir boyut azaltma (öznitelik çıkartma) teknięi olarak kullanılan Komşuluk Bileşenleri Analizi (Neighborhood Component Analysis: NCA) yöntemi kullanılmıştır. Goyal ve Bhatia (2019) bu yöntemi yazılım efor tahmini için geliřtirdikleri YSA modelinin girdileri üzerinde uygulamışlardır.

NCA faktör azaltma yöntemi, her bir efor tahmin modeli için 11 girdi deęiřkeni ve ilgili efor deęerleri alınarak MATLAB uygulaması ile uygulanmış olup, analiz sonuçları Şekil 3.2’de gösterilmiştir. Ek 1’de NCA analizi için MATLAB kodu verilmiştir. Görüldüğü üzere, 1, 3 ve 11. faktörlerin otomasyon eforu üzerinde etki ağırlıkları sıfıra yakındır. Donanım karmaşıklığı, müşteri tipi ve dış kaynak kullanımı olan bu faktörler veri setinden çıkarılmıştır. Otomasyon aşaması, makinenin sürecinin geliřtirildięi aşama olduğundan dolayı, donanımın karmaşıklığının etkisinin olmaması, bunun yerine makine süreç karmaşıklığının etkisinin daha yüksek olması beklenen bir durumdur. Bunun yanında makinenin tedarik yönteminin de otomasyon eforunu etkilememesi normaldir. Makinenin tümünün sıfırdan üretilmesi, ya da stoktan hazır olarak temini ile ilgili olan deęiřkenin üretim eforuna daha çok etkisi olması beklenen bir durumdur. Otomasyon modeli için, 11 adet özellikten geri kalan dięer 8 adet özellik kullanılmıştır.



Şekil 3.2. Otomasyon eforu tahmini için belirlenen faktörlere ait NCA ağırlıkları

Tez kapsamında geliştirilen proje otomasyon eforu tahmini modeli için seçilen girdiler Çizelge 3.6'da gösterilmiştir. Bundan sonraki aşamalarda gerçekleştirilen YSA eğitimi ve hiperparametre optimizasyonu çalışmalarına, otomasyon modeli için bu kısımda belirlenen girdiler ile devam edilmiştir. Proje toplam eforu için 11 adet girdinin hepsi kullanılmıştır. Otomasyon eforu (Model 1) ve toplam proje eforu (Model 2) tahmin modelleri için ele alınan girdiler ve çıktılar Çizelge 3.6'da özetlenmiştir.

Çizelge 3.6. YSA modelleri için belirlenen girdiler ve çıktıları

Model Adı	Faktör Adı	Rolü
Model 1 Otomasyon Eforu Tahmin Modeli	Girdi 2-Süreç karmaşıklığı	Girdi
	Girdi 4-Müşteri ürün yeniliği	
	Girdi 5-Müşteri ürün süreci yeniliği	
	Girdi 6-Makine konfigürasyonu yeniliği	
	Girdi 7-Makine ana fonksiyonu	
	Girdi 8-Makine ana sistem büyüklüğü	
	Girdi 9-Makine kasa büyüklüğü	
	Girdi 10-Makine hat tipi	
Projenin otomasyon aşaması için gereken efor		Çıktı
Model 2 Toplam Proje Eforu Tahmin Modeli	Girdi 1-Donanım karmaşıklığı	Girdi
	Girdi 2-Süreç karmaşıklığı	
	Girdi 3- Müşteri tipi	
	Girdi 4-Müşteri ürün yeniliği	
	Girdi 5-Müşteri ürün süreci yeniliği	
	Girdi 6-Makine konfigürasyonu yeniliği	
	Girdi 7-Makine ana fonksiyonu	
	Girdi 8-Makine ana sistem büyüklüğü	
	Girdi 9-Makine kasa büyüklüğü	
	Girdi 10-Makine hat tipi	
	Girdi 11- Makine üretim sürecinde dış kaynak kullanımı	
Projenin tasarım, satın alma, üretim, otomasyon ve test aşaması için gereken toplam eforu		Çıktı

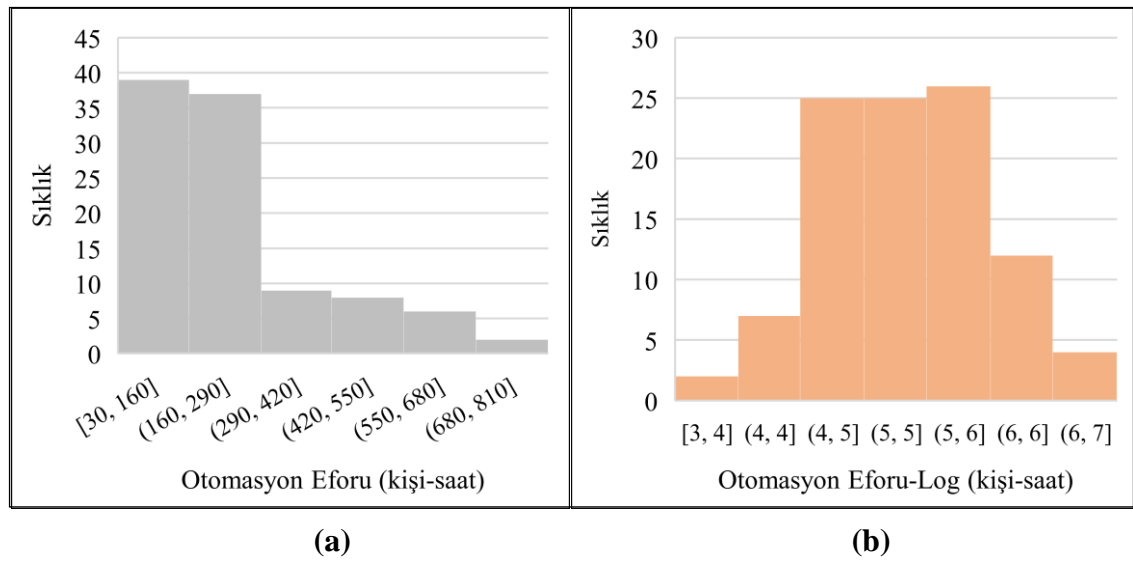
3.3. Veri Ön İşleme

Modellerin geliştirmesine başlamadan önce, veriler bir ön işlemden geçirilmişlerdir. Bu adımda girdi ve çıktı verilerine sırası ile normalizasyon ve logaritmik veri dönüştürme uygulanmıştır. Girdi veri seti aşağıdaki denklem (3.1) ile min-max normalizasyon yöntemi uygulanarak normalize edilmiştir. Böylece tüm girdi değişkenleri orijinal dağılımları korunarak [0, 1] aralığına getirilmiştir.

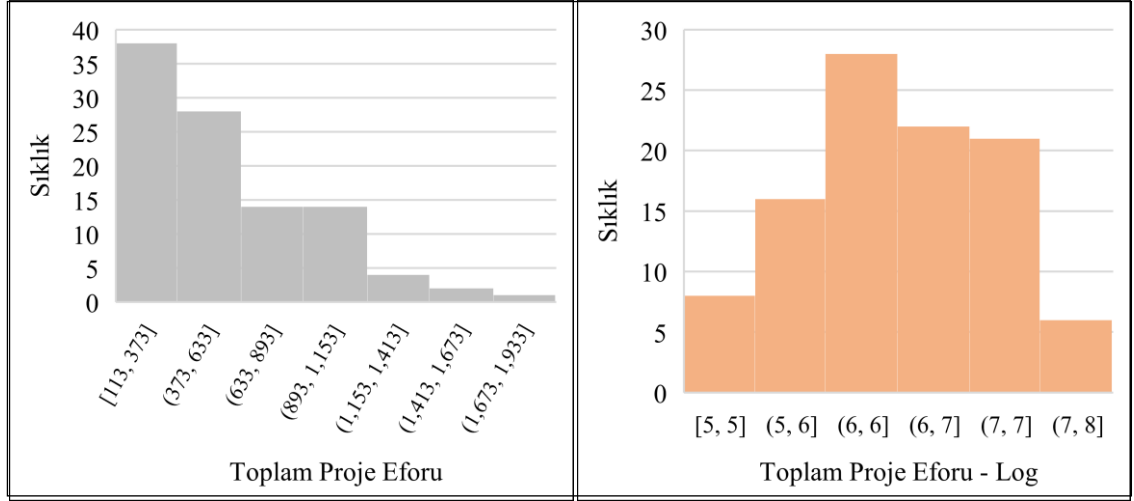
$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

Çıktı veri setine ise dağılım çarpıklıklarının düzeltilmesi ve daha normale yakın bir dağılım elde edilmesi adına logaritmik transformasyon uygulanmıştır. Çıktı değeri y , $\log(y)$ formülü ile dönüştürülmüştür. Verilerin önışleme süreci ile ilgili MATLAB kodu Ek 2’de yer almaktadır. YSA modelinde sonuç hata değeri hesaplanırken, veriler tekrar eski haline dönüştürölüp değerdendirmeler yapılmaktadır.

Proje otomasyon eforu olan çıktı değışkeninin veri dağılımlarının orijinal ve dönüştürölmüş halleri sırası ile Şekil 3.3 (a) ve Şekil 3.3 (b)’de gösterilmiştir. Şekillerde göröldüğü üzere veri dağılımında efor değeri arttıkça sıklığın oldukça azaldığı gözlemlenmiş olup, logaritmik dönüştürme uygulanarak verinin normal dağılıma yaklaşması sağlanmıştır. Aynı işlem toplam proje eforu için de gerçekleştirilmiş olup, ilgili grafikler Şekil 3.4 (a) ve Şekil 3.4 (b)’de gösterilmiştir.



Şekil 3.3. Logaritmik transformasyon öncesi ve sonrası otomasyon eforu çıktı değışkeninin veri dağılımı



a)

b)

Şekil 3.4. Logaritmik transformasyon öncesi ve sonrası toplam proje eforu çıktı değişkeninin veri dağılımı

3.4. Önerilen YSA Modellerinin Mimarisi

Tez kapsamında ele alınan problem tahmin problemi olup, kısaca proje karakteristiklerini girdi olarak alan ve proje eforunu da çıktı olarak alan YSA yapıları geliştirilmiştir. Sinirsel ağ yapısı olarak, literatürde de sıklıkla kullanılan ileri beslemeli sinir ağı (FFNN) kullanılmıştır. FFNN YSA mimarisi, giriş katmanı, gizli katman /gizli katmanlar ve çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı veri setinin özellik sayısı kadardır. Çıkış katmanı ise, problem sınıflandırma problemi ise sınıf sayısı kadar, regresyon problemi ise çıkış sayısı kadar nöron içerir. Bunun yanında giriş ve gizli katman bias nöronlarını içerir (Haykin, 2008). Otomasyon eforunu tahmin eden modelde (Model 1), 8 adet girdi ve 1 adet çıktı; toplam proje eforunu tahmin eden modelde (Model 2), 11 adet girdi ve 1 adet çıktı yer almaktadır.

3.5. YSA Modeli Geliştirme Prosedürü

Yazılım proje efor tahmini ile ilgili yapılan birçok çalışmada YSA, diğer tahmin yöntemlerine göre daha iyi sonuçlar vermiştir. Fakat, YSA'nın performansı dışarıdan girilmesi gereken parametreler olan hiperparametre değerlerine göre farklılık

YSA geliştirme prosedürü, veri önışleme, hiperparametre optimizasyonu, MSE hata performansı açısından umut vadeden YSA yapıları ile çok sayıda denemeler ve test veri seti ile modellerin değeriendirilmesi aşamalarından oluşmaktadır. Prosedür ilk olarak, Bölüm 3.3'te bahsi geçen veri önışleme ile başlamaktadır. Veri önışleme sonrası 101 adet proje verisi rastgele olarak eğitim/doğrulama ve test veri setlerine bölünmüştür. Farklı veri setlerinde üzerinde YSA'nın performansının gözlemlenebilmesi adına, 11 adet test verisi YSA'nın eğitimi süresince ağa gösterilmemiş olup, ileriki adımlarda test amacıyla kullanılmıştır. Geriye kalan 90 projeden oluşan eğitim ve doğrulama veri seti kullanılarak YSA için hiperparametre optimizasyonu gerçekleştirilmiştir. Bu aşamada BO algoritması ile MSE doğrulama hata performansı açısından umut verici ağ yapılarının bulunması hedeflenmiştir. Tez kapsamında otomasyon eforu (Model 1) ve toplam proje eforu tahmini (Model 2) olmak üzere iki ayrı model geliştirilmiştir. Bu iki ayrı model için seçilen hiperparametreler ve olası değerieleri ile çeşitli BO denemeleri gerçekleştirilmiştir.

Literatürde en çok ayarlanan hiperparametreler, gizli katman sayısı ve her bir gizli katmandaki gizli nöron sayılarıdır. Proje efor tahmini alanında, özellikle yazılım projeleri için YSA modelleri geliştirilmiştir. Proje efor tahmini alanında daha çok bir gizli katman kullanılmıştır (Tronto ve diğerieleri, 2008; Attarzadeh ve Ow, 2011; Attarzadeh ve diğerieleri, 2012; Pospieszny ve diğerieleri, 2018). Goyal ve Bhatia (2020) 9 girdi ile yazılım proje efor tahmini yaptıkları çalışmalarında, 8 gizli nörondan oluşan bir gizli katman ile en iyi yapıya ulaşırken, Pai ve diğerieleri (2013) 8 adet girdi kullandıkları model için, 2, 5, 10, 15, 20, 25 gizli nöron ile denemeler yapmış olup, gizli nöron sayısının 10'dan daha büyük olduğu durumlarda anlamlı bir iyileşme elde edemediğini raporlamıştır. Tronto ve diğerieleri (2008) 17 girdi kullandıkları modele ait en iyi yapıyı 23 gizli nöron ile elde etmiştir. Jun ve Lee (2001), 23 girdiden oluşan YSA için, bir gizli katman kullanmış olup gizli nöron sayısını 12'den 47'ye kadar değeriştirirken; Rijwani ve Jain (2016) 23 girdi için bir adet gizli katman ve 2'den 20'ye kadar değerişen gizli nöron sayıları ile denemeler yapmışlardır. Yazılım efor tahmini için 23 girdi kullanılarak geliştirdikleri YSA modellerinde, Jun ve Lee (2001) en iyi YSA yapısına 17 gizli nöron sayısı ile ulaşırlarken, Rijwani ve Jain (2016) en iyi ağ yapısına 10 gizli nöron ile ulaşmışlardır.

YSA yapısında gizli nöron sayısı arttıkça model performansının iyileşmesinin yanı sıra aşırı öğrenme (overfitting) durumu ile de karşılaşmaktadır. Gizli nöron sayısının seçimi için literatürde önerilen birçok yöntem bulunmaktadır. s örnek sayısı, n girdi sayısı ve l çıktı sayısı olmak üzere m gizli nöron sayısı ile ilgili literatürde önerilen veya kullanılan denklemlere Çizelge 3.7’de yer verilmiştir.

Çizelge 3.7. Gizli katman sayısı belirlemek için literatürde önerilen denklemler

No	Tarih	Yazar	m 'in Alabileceği Değer
1	1992	Blum	n ve l arasında
2	1997	Berry ve Linoff	$\leq 2n$
3	2000	Chan ve diğerleri	$(n + l)/2$
4	2022	Zhao ve diğerleri	$\sqrt{n + l} + \alpha$ $\alpha: [1,10]$
5	2022	Zhao ve diğerleri	$\log_2 n$
6	2022	Zhao ve diğerleri	\sqrt{nl}
7	2012	Chow ve diğerleri	En büyük $m = \frac{n+l}{2} + \sqrt{s}$

Literatürde yer alan Çizelge 3.7’deki çalışmaların denklemleri kullanılarak Model 1 ve 2 için gizli nöron sayılarının önerilen değerleri hesaplanmıştır, Çizelge 3.8’de yer almaktadır.

Çizelge 3.8. Model 1 ve Model 2 için literatürdeki denklemler kullanıldığında önerilen değerler

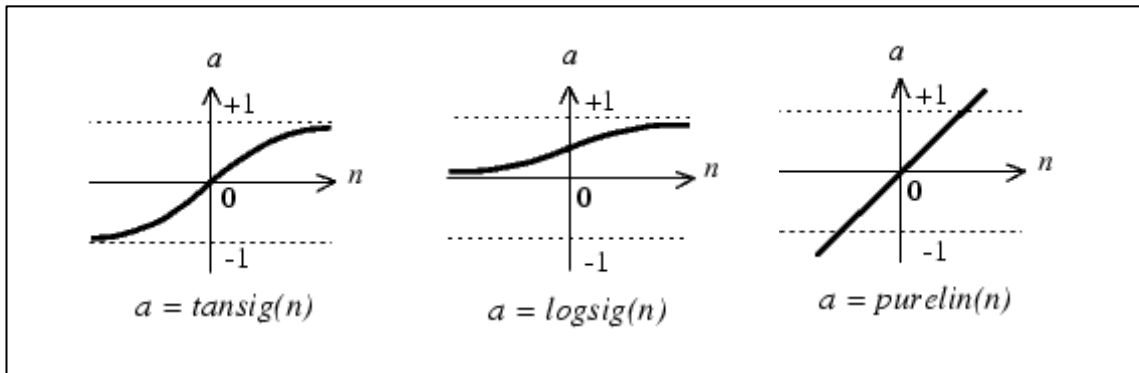
Değişken Adı	Model 1	Model 2
s : Örnek sayısı	90	90
n : Girdi sayısı	8	11
l : Çıktı sayısı	1	1
1.çalışmaya göre önerilen m aralığı	1-8	1-11
2.çalışmaya göre önerilen en büyük m	16	22
3.çalışmaya göre önerilen m	5	6
4.çalışmaya göre önerilen m	4-13	4-13
5.çalışmaya göre önerilen m	3	3
6.çalışmaya göre önerilen m	3	3
7.çalışmaya göre önerilen en büyük m	14	15

Proje efor tahmininde kullanılan YSA yapıları ve Çizelge 3.8'deki bilgiler ışığında bu tez kapsamında seçilen gizli nöron değer aralıkları Çizelge 3.9'da, Model 1 ve Model 2 için gerçekleştirilen BO denemelerinde kullanılan tüm hiperparametreler ve değer aralıkları ile birlikte verilmiştir.

Çizelge 3.9. BO denemelerinde optimize edilmek üzere kullanılan hiperparametreler ve olası değerleri

Model Adı	BO Deneme Numarası	Optimize Edilen Hiperparametre	Olası Değerleri
Model 1 - Otomasyon Eforu Tahmin Modeli	BO – 1 Tek Gizli Katmanlı YSA	Gizli nöron sayısı	2-8
		Gizli katmandaki aktivasyon fonksiyonu	tansig, logsig, purelin
		Çıktı katmanındaki aktivasyon fonksiyonu	tansig, logsig, purelin
Model 2 - Toplam Proje Eforu Tahmin Modeli	BO – 2 Tek Gizli Katmanlı YSA	Öğrenme algoritması	SCG, LM, GDX
		Tek gizli katmandaki nöron sayısı	3-11
		Tek gizli katmandaki aktivasyon fonksiyonu	Tansig, logsig, purelin
		Öğrenme oranı (GDX)	0.1-0.5
		Momentum (GDX)	0.6-0.9
	BO – 3 İki Gizli Katmanlı YSA	Öğrenme algoritması	SCG, LM, GDX
		1. gizli katmandaki nöron sayısı	3-7
		2. gizli katmandaki nöron sayısı	3-7
		1. gizli katmandaki aktivasyon fonksiyonu	tansig, logsig, purelin
		2. gizli katmandaki aktivasyon fonksiyonu	tansig, logsig, purelin
		Öğrenme oranı (GDX)	0.1-0.5
		Momentum (GDX)	0.6-0.9

Proje efor tahmini alanında, aktivasyon fonksiyonu sıklıkla ayarlanan bir diğer hiperparametredir. Gizli katman aktivasyon fonksiyonu olarak literatürde sıklıkla, “sigmoid” (Tronto ve diğerleri, 2008; Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Rijwani ve Jain, 2016), “hiperbolik tanjant” (Rao ve Kumar, 2015; Pospieszny ve diğerleri, 2018; Rankovic ve diğerleri, 2021), “tansig” (Goyal ve Bhatia, 2020) kullanılmıştır. Çıktı katmanı aktivasyon fonksiyonu olarak ise purelin (Goyal ve Bhatia, 2020) tercih edilmiştir. BO-2 ve BO-3 denemelerinde gerek proje efor tahmini literatürü gerekse de ele alınan problemin bir regresyon problemi olduğu göz önüne alındığında çıktı katman aktivasyon fonksiyonu doğrusal bir fonksiyon olan “purelin” olarak tercih edilmiştir. BO-1 denemesinde ise çıktı katmanının aktivasyon fonksiyonu da hiperparametre olarak ele alınmış olup, olası değerlerinde “purelin” fonksiyonunun yanı sıra “tansig” ve “logsig” fonksiyonlarına da yer verilmiştir. Gizli katmanların aktivasyon fonksiyonu tüm BO denemelerinde hiperparametre olarak ele alınmış olup, denemelerde Şekil 3.6’da gösterilen “tansig”, “logsig” ve “purelin” aktivasyon fonksiyonları değerlendirilmiştir.



Şekil 3.6. Tez kapsamında kullanılan aktivasyon fonksiyonları

YSA eğitiminde bir diğer önemli konu öğrenme algoritmasıdır. Proje efor tahmini alanında, FFNN sinirsel ağ yapısını kullanan algoritmalarda, genellikle gradyan azalma tabanlı yöntemler kullanılmıştır (Dave ve Dutta, 2011; Rijwani ve Jain, 2016; Attarzadeh ve diğerleri, 2012). Fakat gradyan azalma tabanlı yöntemlerin gradyan hesaplamalarından dolayı yoğun matris hesaplamaları gerektirmesi, YSA performansını oldukça etkileyen öğrenme oranı ve momentum gibi hiperparametrelerin ayarlanması gerekliliği gibi bir takım dezavantajlarından dolayı öncelikle otomasyon eforu tahmin modeli (Model 1) için

Levenberg-Marquardt (LM) algoritması kullanılmıştır. Gradyan azalma tabanlı teknikler kadar karmaşık gradyan hesaplamaları gerektirmeden daha hızlı yakınsama sağlayan ve öğrenme oranı ve momentum gibi hiperparametre ayarlamaları gerektirmeyen bir yöntem olan LM öğrenme algoritmasının, FFFN sinirsel ağ yapıları için de oldukça tercih edilen etkili bir yöntem olduğu düşünülmektedir.

Birinci model olan otomasyon efor tahmini modeline göre daha fazla girdiye sahip Model 2, toplam proje eforu tahmini modeli, için öğrenme algoritması olarak literatürde sıklıkla kullanılan aşağıdaki algoritmalar denenmiştir.

- Ölçekli Eşlenik Gradyan Algoritması (SCG)
- Levenberg–Marquardt Algoritması (LM)
- Değişken Öğrenme Oranlı ve Momentumlu Gradyan Azalma (GDX)

Bu algoritmalar BO’da hiperparametre olarak ele alınmıştır. GDX algoritması, öğrenme oranı ve momentum hiperparametrelerini içermektedir. BO algoritması öğrenme algoritmasını GDX algoritması seçtiğinde, öğrenme oranı ve momentum hiperparametrelerinin aktifleşmesi ve değer ataması yapılması gerekmektedir. Bu amaçla BO’ya koşullar eklenmiştir. Proje efor tahminine yönelik YSA modeli öneren bazı çalışmalarda öğrenme oranı 0,1 olarak seçilirken (Rao ve Kumar, 2015; Arora ve Mishra, 2018), Dave ve Dutta (2011) öğrenme oranını 0,85 olarak seçmiştir. Rao ve Kumar (2015) momentum değerini 0,5 olarak almıştır. Bu bilgiler ışığında öğrenme oranı 0,1-0,5 aralığında, momentum ise 0,6-0,9 aralığında seçilmiştir.

Seçilen hiperparametre değerleri ile her bir deneme için 90 veri rastgele olarak eğitim ve doğrulama veri setlerine %90-%10 oranında bölünmüştür. 90 projeden oluşan veri setinin %90’ını rastgele seçilerek eğitim verisi olarak kullanılmıştır. Geri kalan %10 oranında veri ise doğrulama verisi olarak kullanılmıştır. YSA eğitiminin durdurma kriteri olarak ise, doğrulama performansı seçilmiştir. Aşırı öğrenmeyi önlemek amacıyla, eğitimin genellenin durduğu yerde, diğer bir deyişle eğitim hatasının oldukça iyi fakat doğrulama hatasının fazla olduğu anda, durdurulması amaçlanmıştır. Bu amaçla durdurma kriteri olarak “Artarda 6 iterasyonda doğrulama hatası, en küçük hatadan daha küçük değil ise DUR.” seçilmiştir. Burada hata değeri ortalama karesel hata (mean

squared errors, MSE) olarak seçilmiştir. MSE aşağıdaki denklem (3.2) kullanılarak hesaplanır:

$$MSE = \left(\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n} \right) \quad (3.2)$$

Y_i : Gerçekleşen efor değeri

\hat{Y}_i : Tahmini efor değeri

n : veri setindeki toplam veri sayısı

BO prosedürü hiperparametreler için başlangıç değer atamaları ile başlar. Belirlenen hiperparametre değerleri ile YSA eğitimi gerçekleştirilir. Belirlenen hiperparametreler ile eğitim 100 defa tekrarlanır ve ortalama doğrusal hata değeri hesaplanır. Her bir eğitim denemesinde YSA farklı başlangıç ağırlıkları ve eğitim/doğrulama veri setleri için rastgele olarak seçilen veri kombinasyonları ile eğitime başlatılır. Eğitim ve doğrulama olmak üzere %90-%10 oranında rastgele seçilmiş farklı veriler kullanılarak, YSA modelinin belirlenen hiperparametre değerleri ile farklı veri setlerinde verdiği performans gözlemlenir. Bu teknik literatürde tekrarlanan rassal örnekleme doğrulaması (repeated random subsampling validation) ya da Monte Carlo çapraz doğrulaması olarak geçmektedir. Veri seti n defa rastgele olarak eğitim ve doğrulama veri setlerine bölünür ve bu n denemenin ortalama doğrulama hata değeri alınarak YSA modelinin performansı belirlenir. Denklem (3.2) doğrulama veri seti için hesaplanır ve doğrulama hatası, $MSE_{doğrulama}$ elde edilir. Sonrasında denklem (3.3) kullanılarak, 100 deneme üzerinden bu değerlerin ortalaması alınarak $ortalama(MSE_{doğrulama})$ değeri elde edilir.

$$ortalama(MSE_{doğrulama}) = \frac{\sum_{j=1}^{100} MSE_{doğrulama}(j)}{100} \quad (3.3)$$

Bu değer BO'nun amaç fonksiyonu olarak alınır ve BO algoritması ile bu hata değeri en küçüklenmeye çalışılarak $ortalama(MSE_{doğrulama})$ değerini en iyileyen hiperparametre değerleri elde edilir. BO algoritması her bir adımda bu hata değerini hesapladıktan sonra, o adıma kadar hesaplanmış olan YSA model performans değerlerini

de kullanarak amaç fonksiyonun dağılımını tahminler. Global en iyiye ulaşmak için sıradaki iterasyonda seçilecek olan hiperparametre değerlerini bulmaya çalışır.

BO algoritması tabanlı YSA modeli geliştirme prosedürüne ait sözde kod Çizelge 3.10'da verilmiştir. Edinim fonksiyonu (acquisition function), α , ile bir sonraki adım için seçilecek olan aday noktalardan elde edilecek fayda değerleri tahminlenir. Maksimum faydayı sağlayacak olan h_t noktası, sıradaki iterasyonda kullanılacak olan hiperparametre değerleri olarak belirlenir. Bu işlem BO'nun sonlandırma kriteri olarak seçilen maksimum adım sayısı kadar tekrarlanır ve maksimum adıma ulaştığında durur. Maksimum adım sayısı Model 1 için yapılan BO-1 deneyinde 30 adım olarak belirlenmiştir. Model 2 için tek gizli katmanlı YSA yapıları için yapılan BO-2 deneyinde 50 adım, iki katmanlı YSA yapıları için yapılan BO-3 deneyi için 70 adım olarak belirlenmiştir. Prosedür MATLAB ortamında kodlanmış olup, Model 1 ve Model 2 için yapılan BO denemeleri (BO-1, BO-2, BO-3) ile ilgili MATLAB kodu sırası ile Ek 3, Ek 4 ve Ek 5'te verilmiştir.

Çizelge 3.10. Koşullu değişkenler içeren BO ile HPO prosedürüne ait sözde kod

BO temelli YSA geliştirme algoritması	
<hr/>	
1:	X : Normalize edilmiş girdi matrisi
2:	Y : Logaritmik dönüştürülmüş çıktı vektörü
3:	h_t : BO algoritmasının t . adımında seçilen hiperparametre değerlerine ait vektör $t: 1, \dots, k$.
4:	$D_{1:t}$: iterasyon 1'den t 'ye kadarki gözlemlenen değerler
5:	for $t = 1, 2, \dots, k$ do
6:	$h_t = \text{argmax}_h \alpha(h D_{1:t-1})$ /* Edinim fonksiyonunu α iyileştirecek yönde h_t aşağıdaki koşullar altında seçilir */
7:	for $i = 1, 2, \dots, 100$ do /* Seçilen h_t değerleri kullanılarak YSA performansı aşağıdaki gibi hesaplanır, y_t */
8:	X, Y, h_t ile YSA eğitimi gerçekleştirilir ve YSA'nın hata değeri hesaplanır, y_i
9:	$\text{toplama} \leftarrow y_i + \text{toplama}$
10:	end for
11:	$y_t \leftarrow \text{toplama}/100$ /* 100 deneme üzerinde ortalama YSA hata değeri hesaplanır */
12:	$D_{1:t} = \{D_{1:t-1}, (h_t, y_t)\}$ /* Amaç fonksiyonunun gözlemlenen değerini, y_t , $D_{1:t}$ veri setine eklenir */
13:	Vekil model güncellenir.
14:	end for

BO ile elde edilen sonuçlar ışığında belirlenen hiperparametre kombinasyonları ile çok sayıda deneme yapılmıştır. Eğitimlerin tekrar edilmesindeki amaç, her bir denemede farklı başlangıç değer atamaları ve rastgele olarak bölünmüş eğitim/doğrulama veri setleri kullanılacağı için YSA performanslarına ait genel bir fikir edinmektir. Her bir tekrarda YSA performansının test hata performansı, eğitimde kullanılmamış olan 11 adet projeye ait test veri seti ile hesaplanmıştır.

YSA performanslarını değerlendirmek için proje efor tahmini alanında daha çok kestirim düzeyi, $PRED(\%x)$ kullanılmış olup birçok çalışmada bu değer $x = \% 25$ olarak alınmıştır (Attarzadeh ve Ow, 2011; Attarzadeh ve diğerleri, 2012; Araujo ve diğerleri, 2017; Pandey ve diğerleri, 2020). $PRED(\%x)$, modelin efor değerini gerçekleştiren efor değerinden en fazla $\%x$ 'i kadar sapma ile tahmin edildiği yüzde miktarı anlamına gelir. Her bir i verisi için, öncelikle MRE değerleri hesaplanır. Bu değer belirli bir $\%x$ değerinin altında ise 1, değilse 0 değerini alarak, bu değerlerin ortalaması kullanılarak $PRED(\%x)$ kestirim düzeyi değerine erişilir. Yazılım proje efor tahmini alanında $MMRE \leq 0.25$ ve $PRED(\%25) \geq 75\%$ değerleri YSA modelinin tahmin doğruluğu için yeterli olarak kabul edilir (Pospieszny ve diğerleri, 2018). Çıktı değerleri eğitim öncesi logaritmik dönüşüme uğradıkları için, karşılaştırma yapabilmek adına orijinal değerlere tekrar dönüştürülmüştür. MMRE ve PRED değerleri göz önüne alınarak, proje otomasyon eforu tahmini için en iyi YSA yapısı seçilmiştir.

4. BULGULAR ve TARTIŞMA

Tez kapsamında fotonik cihazlar için yarı veya tam otomatik makineler geliştiren Almanya merkezli bir firma bünyesinde tamamlanmış 101 adet makine geliştirme projesinden elde edilen veriler toplanmıştır. Bu veriler ile 3. Bölüm’de bahsi geçen yöntem uygulanarak otomasyon efor tahmini (Model 1) ve toplam proje eforu tahmini (Model 2) için YSA modelleri geliştirilmiş olup, elde edilen sonuçların literatürdeki benzer modeller ile karşılaştırmalarına yer verilmiştir.

4.1. Otomasyon Efor Tahmini için Yapılan Deneyler

Otomasyon eforu tahmin modeli olan Model 1 için yapılan BO denemesi için seçilen hiperparametre değerleri Çizelge 4.1’de verilmiştir. Çizelgede görüldüğü ve Bölüm 3’te de anlatıldığı üzere gizli nöron sayıları 2-8 arasında değişmekte olup, gizli ve çıktı katmanları için aktivasyon fonksiyonları “tansig”, “logsig” ve “purelin” olarak belirlenmiştir.

Çizelge 4.1. BO-1 denemesi için seçilen hiperparametreler ve olası değerleri

Model Adı	BO Deneme Numarası	Optimize Edilen Hiperparametre	Olası Değerler
Model 1	BO – 1	Gizli nöron sayısı	2-8
Otomasyon Eforu Tahmin Modeli	Tek Gizli Katmanlı YSA	Gizli katmandaki aktivasyon fonksiyonu Çıktı katmanındaki aktivasyon fonksiyonu	tansig, logsig, purelin tansig, logsig, purelin

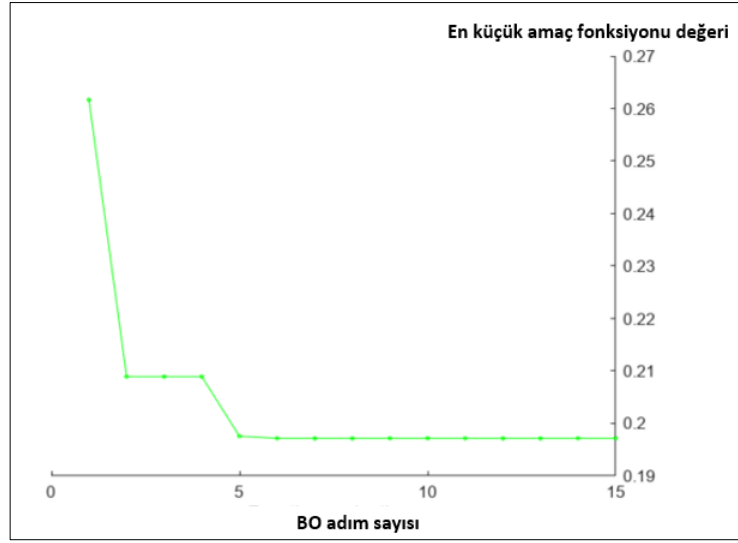
101 adet tamamlanmış proje ile ilgili toplanan verinin 90’ı eğitim ve doğrulama için kullanılmış olup, geri kalan 11 proje verisi eğitim sırasında kullanılmamıştır. HPO için BO algoritması uygulanmış olup, BO’nun her bir adımında seçilen hiperparametre değerleri ile ağız eğitimi 100 kere tekrarlanmıştır. Her tekrarda 90 adet projeden oluşan eğitim ve doğrulama verisi %90-%10 oranında eğitim ve doğrulama veri setlerine rasgele bölünmüştür. Her bir adımda algoritma tarafından belirlenen hiperparametre değerlerini

kullanarak rastgele olarak seçilmiş eğitim ve doğrulama veri setleri ile YSA'nın ortalama hata performansı hesaplanmıştır. BO sonuçları Şekil 4.1'de yer almaktadır.

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	HLN1	tranFcn	tr_ou
1	Best	0.26169	816.46	0.26169	0.26169	3	purelin	purelin
2	Best	0.20893	223.38	0.20893	0.20893	7	logsig	purelin
3	Accept	0.27712	259.82	0.20893	0.20893	4	purelin	logsig
4	Accept	0.23455	483.57	0.20893	0.20893	5	logsig	logsig
5	Best	0.19756	295.04	0.19756	0.19756	6	logsig	purelin
6	Best	0.19718	203.55	0.19718	0.19718	4	logsig	purelin
7	Accept	0.20445	204.24	0.19718	0.19718	5	logsig	purelin
8	Accept	0.19865	313.91	0.19718	0.19718	3	logsig	purelin
9	Accept	0.19934	428.14	0.19718	0.19718	4	tansig	purelin
10	Accept	0.20332	994.78	0.19718	0.19718	6	tansig	purelin
11	Accept	0.2303	872.02	0.19718	0.19718	2	tansig	purelin
12	Accept	0.21819	620.62	0.19718	0.19718	5	tansig	purelin
13	Accept	0.23603	298.17	0.19718	0.19718	7	tansig	purelin
14	Accept	0.24078	271.59	0.19718	0.19718	2	logsig	purelin
15	Accept	0.22467	218.02	0.19718	0.19718	3	tansig	purelin
16	Accept	0.20882	323.46	0.19718	0.19718	8	logsig	purelin
17	Accept	0.25216	286.08	0.19718	0.19718	5	tansig	logsig
18	Accept	0.20936	759.77	0.19718	0.19718	8	logsig	tansig
19	Accept	0.24213	586.31	0.19718	0.19718	7	logsig	tansig
20	Accept	0.28669	383.86	0.19718	0.19718	8	purelin	tansig
21	Accept	0.22323	276.1	0.19718	0.19718	8	logsig	logsig
22	Accept	0.27223	301.95	0.19718	0.19718	8	tansig	tansig
23	Accept	0.276	1125.5	0.19718	0.19718	2	logsig	logsig
24	Accept	0.2783	2387.6	0.19718	0.19718	2	tansig	tansig
25	Accept	0.24461	1865.6	0.19718	0.19718	7	logsig	logsig
26	Accept	0.26785	623.18	0.19718	0.19718	8	purelin	purelin
27	Accept	0.27247	247.68	0.19718	0.19718	2	purelin	tansig
28	Accept	0.24629	314.32	0.19718	0.19718	2	logsig	tansig
29	Accept	0.26789	352.7	0.19718	0.19718	8	tansig	logsig
30	Accept	0.26848	449.16	0.19718	0.19718	2	tansig	logsig

Şekil 4.1. BO-1 sonuç çıktısı (MATLAB)

BO sonucuna göre en iyi ağı gözlemlenen hata değeri 0.19718 olarak elde edilmiştir. Bu değer, belirlenen hiperparametreler ile ağı eğitiminin farklı %90-%10 eğitim ve doğrulama veri set kombinasyonları ile tekrarlanması sonucu elde edilen ortalama MSE log değeridir. BO algoritmasının durdurma kriteri 30 adım olarak seçilmiş olup, algoritma amaç fonksiyonunu en küçükleyen değere Şekil 4.1 ve Şekil 4.2'de görüldüğü üzere 6. adımda ulaşmıştır.



Şekil 4.2. BO-1 denemesinde her bir adımda gözlemlenen amaç fonksiyonu değeri

En iyi hata performansını veren ağ yapısı, dört adet gizli nöron barındıran bir gizli katmana sahip olup, gizli katman aktivasyon fonksiyonu logsig, çıktı katmanı aktivasyon fonksiyonu ise purelin olarak seçilmiştir. İkinci en iyi çözüme ise yine aynı aktivasyon fonksiyonlarını kullanarak 6 gizli nöron ile erişilmiştir. Bunun yanında, yine aynı aktivasyon fonksiyonlarını kullanarak 3 ve 5 arası gizli katmanlı ağ yapıları ile de en iyi çözüme yakın çözümler elde edilmiştir. BO algoritması en iyi değere 0.19718’de ulaştığı için, 0.20500 ve altında olan YSA yapıları (Şekil 4.1’de mavi çerçevede gösterilen) bir sonraki aşamalarda manuel denemeler için ele alınmıştır. Çizelge 4.2’de bu YSA yapılarına ve BO tarafından hesaplanan amaç fonksiyonu değerlerine yer verilmiştir. Bütün yapılarda çıktı katmanındaki aktivasyon fonksiyonu için en iyi değer ‘purelin’ olarak elde edilmiştir.

Çizelge 4.2. BO-1 sonucuna göre ileri denemeler için seçilen YSA modelleri

BO-1’de bulunduğu iterasyon numarası	Gözlemlenen Amaç Fonksiyonu Değeri	Gizli Nöron Sayısı	Gizli Katmandaki Aktivasyon Fonksiyonu
5	0.19756	6	logsig
6	0.19718	4	logsig
7	0.20445	5	logsig
8	0.19865	3	logsig
9	0.19934	4	tansig
10	0.20332	6	tansig

BO sonucunda seçilen, Çizelge 4.2’de gösterilmiş olan YSA yapıları için denemeler tekrarlanmış olup, yapılan her bir deneme sonucu elde edilen YSA, Çizelge 4.3’te gösterilen 11 test veri seti ile test edilmiştir. Bu veri setinde bulunan girdiler Ek 2’de gösterilen kod ile normalize edilmiştir. Aynı şekilde çıktılar da Ek 2’de gösterilen kod ile logaritmik olarak dönüştürülmüştür. Test aşamasına ait MATLAB kodu Ek 6’da gösterilmiştir. Her bir ağ yapısında eğitim ve doğrulamalar yine Ek 6’da verilen MATLAB kodu ile yapılmıştır. Çizelge 4.2’de seçilen hiperparametre değerleri girilerek ağ eğitilmiş ve doğrulaması yapılmıştır. Eğitimi tamamlanmış ağa ait özellikler *net* objesi olarak MATLAB’da kaydedilmiş olup, Ek 6’nın devamında gösterilen kod aracılığıyla, ağın test veri seti girdilerine ürettiği çıktılar gözlemlenmiştir.

Çizelge 4.3. Model 1 Testi için Kullanılan Test Veri Seti

Test Veri Numarası	Süreç Karmaşıklığı	Müşteri Ürün Yeniliği	Müşteri Ürün Süreci Yeniliği	Makine Konfigurasyon Yeniliği	Makine Ana Fonksiyonu	Makine Ana Sistem Büyüklüğü	Makine Kasa Büyüklüğü	Makine Hat Tipi	Gerçekleşen Otomasyon Efor Değeri
	Girdi 1	Girdi 2	Girdi 3	Girdi 4	Girdi 5	Girdi 6	Girdi 7	Girdi 8	Çıktı
1	1	3	3	3	2	1	1	1	30
2	2	1	1	1	1	2	4	3	103
3	2	1	1	1	1	4	4	3	103
4	2	1	2	2	1	2	4	3	123
5	1	3	3	3	3	6	6	2	147
6	2	3	3	2	4	5	6	2	184
7	1	2	2	2	1	4	4	3	220
8	2	2	2	2	1	2	4	2	232
9	2	3	3	3	5	4	4	2	246
10	2	2	2	3	5	6	6	2	470
11	2	3	3	3	1	4	4	3	586

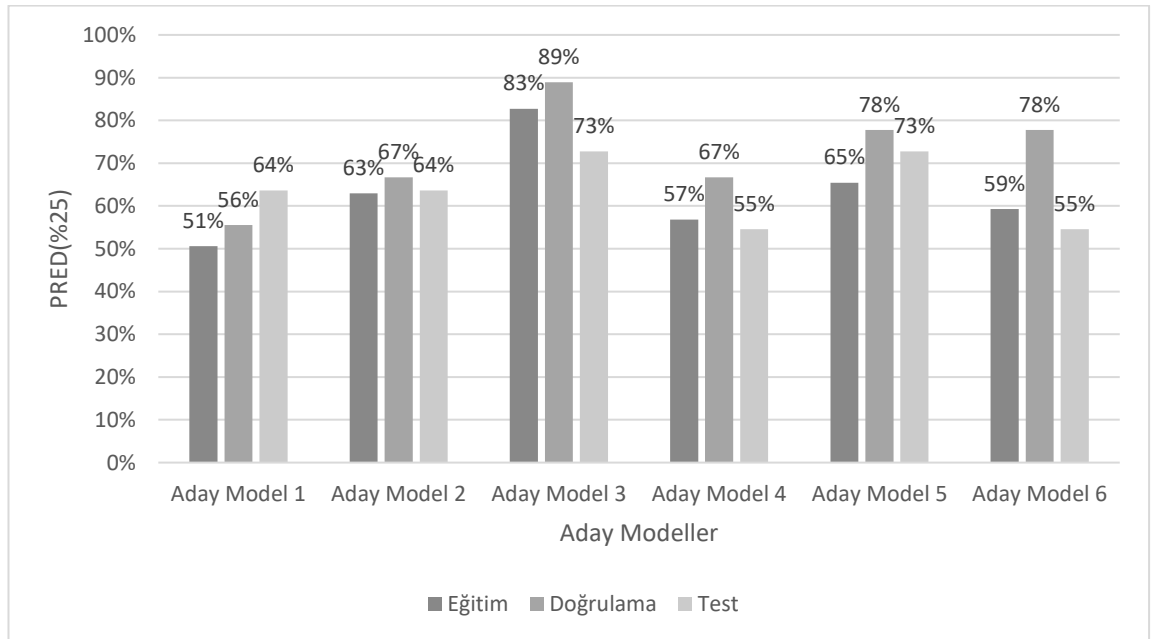
Yapılan denemelerden her bir ağ yapısı için en iyi çıkan ağ performansları Ek 7’de verilmiştir. Çizelge 4.2’de bulunan her bir hiperparametre kombinasyonuna ait Ek 7’deki denemeler arasından MMRE değerleri 0.45 altında olan ve PRED(%25) değerleri %50 üzerinde olan umut vadeden adaylar Çizelge 4.4’te özetlenmiştir.

Çizelge 4.4. Model 1 için bazı aday denemeler ve en iyi deneme

Gizli Nöron Sayısı	Aktivasyon fonksiyonu	Deney No (Ek 7)	Aday Model Numarası	MMRE			PRED(%25)		
				E	D	T	E	D	T
3	logsig	5	1	0.31	0.28	0.40	%51	%56	%64
4	logsig	5	2	0.27	0.22	0.40	%63	%67	%64
5	logsig	4	3	0.15	0.13	0.30	%83	%89	%73
6	logsig	3	4	0.28	0.33	0.43	%57	%67	%55
4	tansig	3	5	0.24	0.20	0.41	%65	%78	%73
6	tansig	1	6	0.27	0.26	0.42	%59	%78	%55

E: Eğitim, D: Doğrulama, T: Test

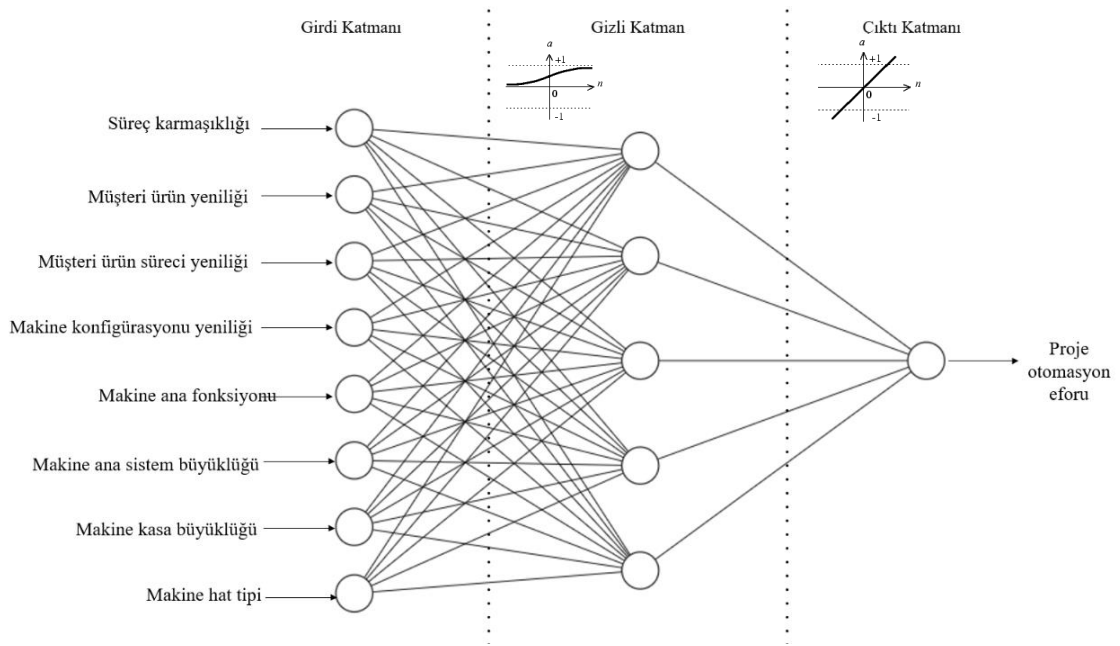
Şekil 4.3'te görüldüğü üzere aday model 3, diğer bir deyişle Çizelge 4.4'teki 5 gizli nörona sahip ve "logsig" aktivasyon fonksiyonunu kullanan YSA modeli; eğitim, doğrulama ve test veri setlerine ait MMRE değerleri 0.30'un altında olması ve PRED(%25) değerlerinin %70'in üstünde olmasından dolayı en iyi ağ olarak seçilmiştir.



Şekil 4.3. Model 1 için seçilen aday modellerin PRED (%25) değerleri

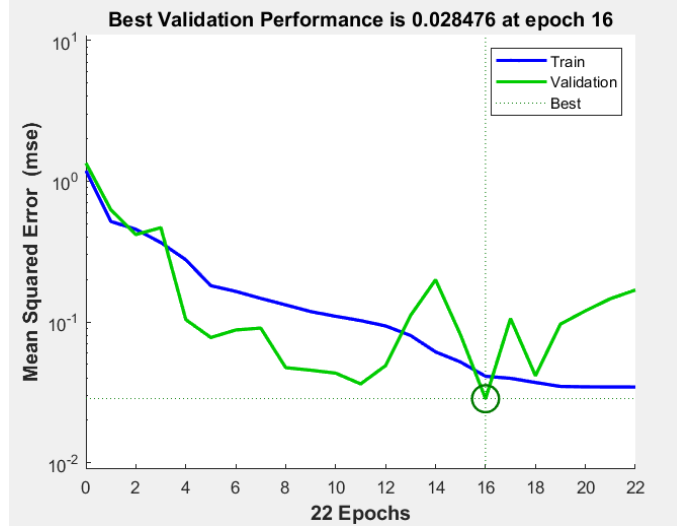
4.2. Otomasyon Efor Tahmini Modeli için En iyi Ağ Yapısı ve Performansı

Otomasyon eforu tahmin modeli olan Model 1 için seçilen YSA yapısı Şekil 4.4'te gösterilmiştir. 8 adet girdi ve 1 adet çıktıdan oluşan model 5 adet gizli nöron yapısına sahip olup, aktivasyon fonksiyonları gizli katman için “logsig” ve çıktı katmanı için “purelin” olarak belirlenmiştir. Ağın eğitimi Bölüm 3’te belirtildiği üzere LM algoritması ile gerçekleştirilmiştir.



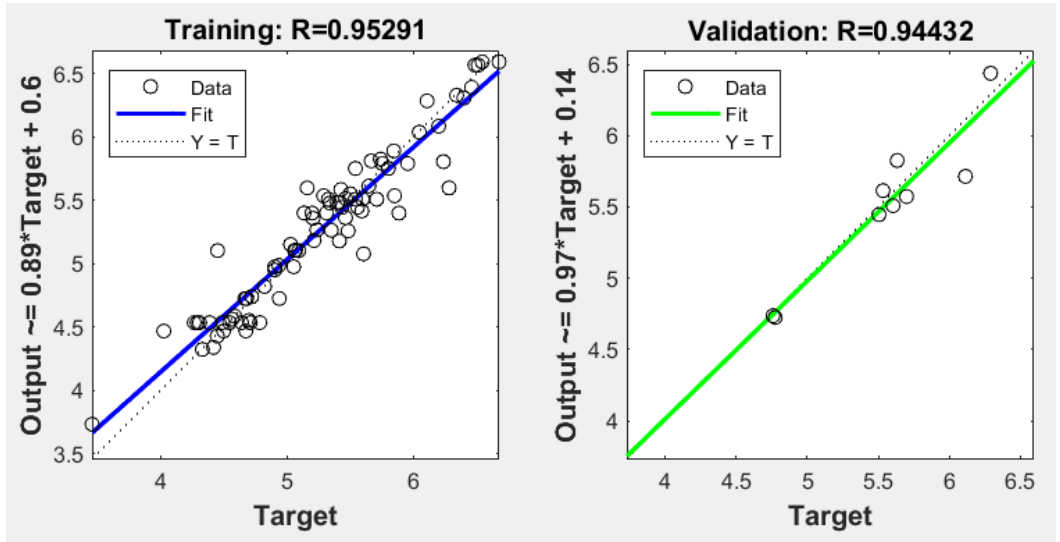
Şekil 4.4. Proje otomasyon eforu (Model 1) için önerilen YSA yapısı

Belirlenen en iyi ağ yapısının eğitim grafikleri Şekil 4.5’te verilmiştir. Eğitim, 16. iterasyondan sonraki artarda 6 iterasyonda elde edilen doğrulama hatasının, 16. iterasyona kadar elde edilmiş en küçük hatadan daha küçük olmadığı için, diğer bir deyişle doğrulama hatası daha çok iyileşmediği için eğitim 16. iterasyonda durdurulmuştur. Şekil 4.5’te eğitim ve doğrulama hatalarının eğitimin her bir iterasyondaki değişimi ve eğitimin durdurulduğu nokta gözlemlenmektedir.



Şekil 4.5. Model 1 – YSA eğitim ve doğrulama hatalarının eğitimin her bir iterasyonundaki değişimi

Ağın korelasyon (R) değeri sonuçları Şekil 4.6’da gösterilmiş olup; eğitim, doğrulama ve test veri setlerine ait hata değerleri Çizelge 4.5’de verilmiştir.

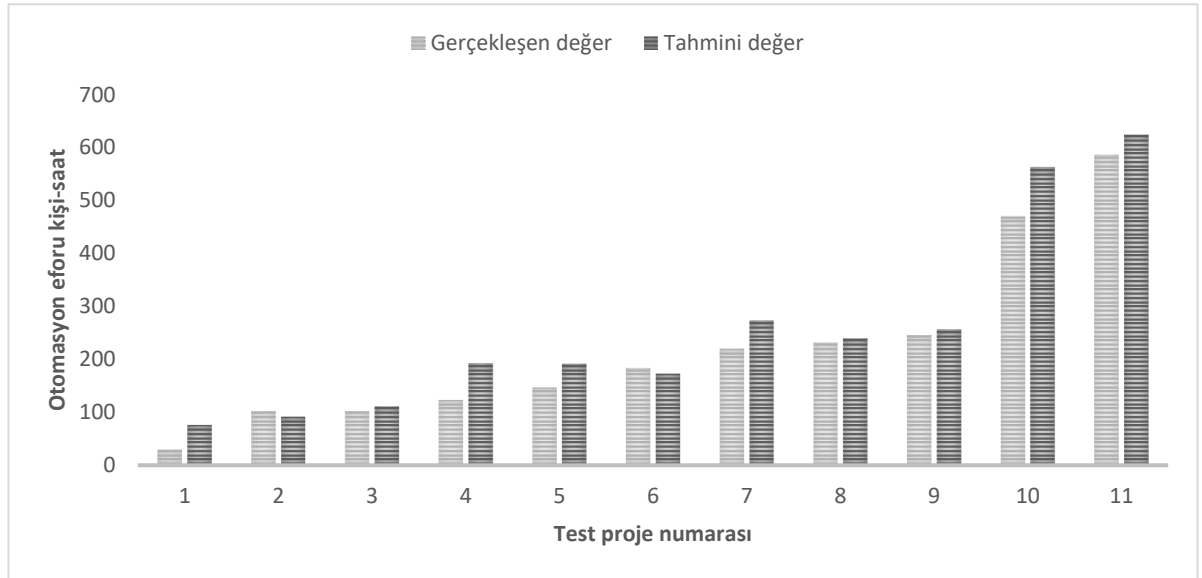


Şekil 4.6. Model 1 - YSA eğitim ve doğrulama korelasyon (R) değeri sonuçları

Çizelge 4.5. Model 1 için elde edilen en iyi ağın hata performans değerleri

	MSE_{\log}	R-değeri	MMRE	PRED (%25)
Eğitim	0.04	0.95	0.15	%83
Doğrulama	0.03	0.94	0.13	%89
Test	0.11	0.94	0.30	%73

Önerilen YSA modeli ile 11 adet test projesinin otomasyon eforu ile ilgili yapılan tahminler ve gerçekleşen değerler Şekil 4.7’de verilmiştir. Otomasyon eforu tahmini değerleri logaritmik olup, gerçekleşen değerler ile karşılaştırılmak üzere geri dönüştürülmüştür.



Şekil 4.7. Otomasyon eforu tahminleri ve gerçek değerleri

Proje efor tahmini alanında geliştirilen modellerde genelde MMRE değeri 0.25’in altında olan ve PRED(%25) değeri %75’in üstünde olan modeller, tahmin için yeterli kabul edilmektedir (Pospieszny ve diğerleri, 2018). Proje efor tahmini ile ilgili YSA modeli öneren çalışmalardan bazıları Çizelge 4.6’da özetlenmiş olup, bu çalışmalarda bulunan en iyi MMRE değerleri 0.14-0.59, PRED(%25) değerleri ise %42-%65 arasında değişmektedir.

Çizelge 4.6. Proje efor tahmini alanında geliştirilen YSA modellerinin hata değerleri

Yıl	Yazarlar	Yöntem	MMRE ve PRED Değerleri
2008	Tronto ve diğerleri	MLP	MMRE: 0.25-0.63
2008	Park ve Baek	ANN	MRE : 0.59-0.63
2010	Reddy ve diğerleri	RBFNN, GRNN	MMRE: 0.17-0.35
2010	Kalichanin-Balich ve Lopez-Martin	FFNN	MMER: 0.22
2011	Lopez-Martin ve diğerleri	GRNN	MMER: 0.26
2011	Attarzadeh ve Ow	MLP	MMRE: 0.45, PRED(%25): %43
2012	Attarzadeh ve diğerleri	FFNN	MMRE: 0.46, PRED(%25): %46
2013	Nassif ve diğerleri	MLP	MMER: 0.40, PRED(%25): %46
2013	Pai, McFall ve Subramanian	MLP	MMRE: 0.59
2016	Rijwani ve Jain	FFNN	MMRE: 0.14
2018	Pospieszny ve diğerleri	MLP	MMRE: 0.21, PRED(%25): %65
2020	Goyal ve Bhatia	FFNN	MMRE: 0.26
2020	Pandey ve diğerleri	MLP	MMRE: 0.24, PRED(%25): %42
2021	Rankovic ve diğerleri	MFFN (ANN-L36, test)	MMRE: 0.16-0.57
2021	Carvalho ve diğerleri	Aşırı Öğrenme Makineleri (<i>Extreme Learning Machine</i>)	MMRE: 0.18
2022	Sharma ve Vijayvargiya	Neuro-Fuzzy Network	MMRE: 0.17, PRED(%25)=%84

Bu tez çalışmasında otomasyon eforu tahmini geliştirilen modele ait MMRE değerleri 0.14-0.63 ve PRED(%25) değerlerinin %42-%84 arasında değiştiği düşünüldüğünde, geliştirilen modelin yeterli doğrulukta olduğu sonucuna varılmıştır.

4.3. Toplam Proje Eforu Tahmini Modeli için Yapılan Deneyler

Tez kapsamında önerilen ikinci model olan toplam proje eforu tahmini modeli için öncelikle tek katmanlı ağ yapısı denenmiştir. BO-2 kodlu BO denemesi için alınan hiperparametreler Çizelge 4.7’de özetlenmiştir.

Çizelge 4.7. BO-2 denemesi için seçilen hiperparametreler ve olası değerleri

Model Adı	BO Deneme Numarası	Optimize Edilen Hiperparametre	Olası Değerler
Model 2	BO – 2	Öğrenme algoritması	SCG, LM, GDX
Toplam Proje Eforu Tahmin Modeli	Tek Gizli Katmanlı YSA	Tek gizli katmandaki nöron sayısı	3-11
		Tek gizli katmandaki aktivasyon fonksiyonu	Tansig, logsig, purelin
		Öğrenme oranı (GDX)	0.1, 0.2, 0.3, 0.4, 0.5
		Momentum (GDX)	0.6, 0.7, 0.8, 0.9

BO'nun her bir adımında elde edilen amaç fonksiyonu değerleri Ek 8'de verilmiştir. En iyi hata performansını veren ağ yapısına 20. adımda ulaşılmış olup, YSA yedi adet gizli nöron barındıran bir gizli katmandan oluşmaktadır. Gizli katman aktivasyon fonksiyonu “tansig” olup öğrenme algoritması SCG seçildiği için momentum ve öğrenme oranları hiperparametrelerine değer ataması yapılmamıştır. İkinci en iyi çözüme ise yine SCG öğrenme algoritması kullanılarak, aynı aktivasyon fonksiyonuna sahip 3 gizli nöronlu ağ yapısı ile ulaşılmıştır. Bunun yanında, yine SCG öğrenme algoritması kullanılarak, aynı aktivasyon fonksiyonuna sahip 8 gizli nöronlu ağ yapısı ile de en iyi üçüncü çözüm elde edilmiştir. BO algoritması en iyi amaç fonksiyonu değerine 0.11106'da ulaştığı için, 0.12000 ve altında olan YSA yapıları bir sonraki aşamalarda manuel denemeler için ele alınmıştır. Çizelge 4.8'de bu YSA yapılarına ve BO tarafından hesaplanan amaç fonksiyonu değerlerine yer verilmiştir.

Çizelge 4.8. BO-2 sonucuna göre otomasyon eforu tahmini modeli için ileri denemelerde kullanılmak üzere seçilen YSA modelleri

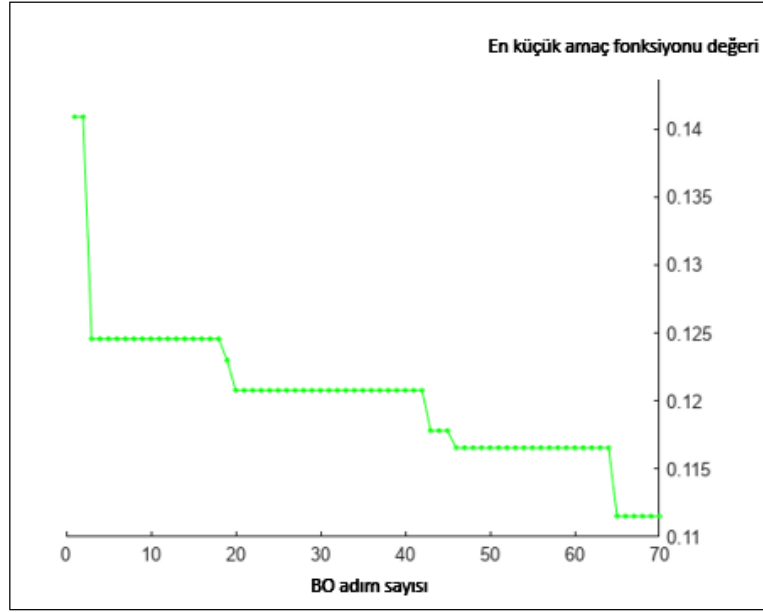
BO-2'de bulunduğu iterasyon numarası	Gözlemlenen Amaç Fonksiyonu Değeri	Öğrenme Algoritması	Gizli Nöron Sayısı	Gizli Katmandaki Aktivasyon Fonksiyonu
8	0.11592	SCG	3	tansig
20	0.11106	SCG	7	tansig
30	0.11651	SCG	8	tansig

Toplam proje eforu modeli için ayrıca iki katmanlı yapılar için de BO uygulanmıştır ve bu uygulama BO-3 olarak isimlendirilmiştir. Tek katmanlı yapıları içeren BO-2 uygulamasının sonuçları (Çizelge 4.8) göz önünde bulundurularak BO-3 deneyinde 1. ve 2. gizli katmandaki nöron sayısı değer aralığı 3-7 olarak belirlenmiştir. BO-3 denemesinde ele alınan hiperparametreler ve olası değerleri Çizelge 4.9’da verilmiştir.

Çizelge 4.9. BO-3 denemesi için seçilen hiperparametreler ve olası değerleri

Model Adı	BO Deneme Numarası	Optimize Edilen Hiperparametre	Olası Değerler
Model 2 Toplam Proje Eforu Tahmin Modeli	BO – 3 İki Gizli Katmanlı YSA	Öğrenme algoritması	SCG, LM, GDX
		1. gizli katmandaki nöron sayısı	3-7
		2. gizli katmandaki nöron sayısı	3-7
		1. gizli katmandaki aktivasyon fonksiyonu	tansig, logsig, purelin
		2. gizli katmandaki aktivasyon fonksiyonu	tansig, logsig, purelin
		Öğrenme oranı (GDX)	0.1, 0.2, 0.3, 0.4, 0.5
	Momentum (GDX)	0.6, 0.7, 0.8, 0.9	

BO sonucuna göre en iyi ağın gözlemlenen hata değeri 0.11153 olarak elde edilmiştir. Bu değer, daha önce de belirtildiği üzere belirlenen hiperparametreler ile ağın eğitiminin farklı %90-%10 eğitim ve doğrulama veri set kombinasyonları ile tekrarlanması sonucu elde edilen ortalama MSE log değeridir. BO algoritmasının durdurma kriteri 70 adım olarak seçilmiş olup, algoritma amaç fonksiyonunu en küçükleyen değere Şekil 4.8’de ve Ek 9’da görüldüğü üzere 65. adımda ulaşmıştır. Her adımda ele alınan hiperparametre değerleri ve gözlemlenen BO amaç fonksiyonu değeri Ek 9’da gösterilmiştir.



Şekil 4.8. BO-3 denemesinde her bir adımda gözlemlenen amaç fonksiyonu değeri

En iyi hata performansını veren ağ yapısı, iki katmanlı 7-4 gizli katman nöronlu ve gizli katman aktivasyon fonksiyonları sırası ile tansig ve purelin olan yapı olmuştur. İkinci en iyi çözüme ise 2 gizli katmanlı 6-4 gizli katman nöronlu ve iki katmanında da purelin aktivasyon fonksiyonunu barındıran yapı ile erişilmiştir. Bunun yanında, yine aynı iki gizli katman aktivasyon fonksiyonları purelin olan 7-4 gizli katman nöronlu ağ yapıları ile de en iyi çözüme yakın çözüm elde edilmiştir. BO algoritması en iyi amaç fonksiyonu değerine 0.11153'te ulaştığı için, 0.12000 ve altında olan YSA yapıları bir sonraki aşama için denemeler yapmak üzere ele alınmıştır. Çizelge 4.10'da bu YSA yapılarına ve BO tarafından hesaplanan amaç fonksiyonu değerlerine yer verilmiştir. Bütün yapılarda çıktı katmanındaki aktivasyon fonksiyonu 'purelin olup, öğrenme algoritmaları BO tarafından SCG seçilmiştir.

Çizelge 4.10. BO-3 sonucuna göre ileri denemeler için seçilen YSA modelleri

BO-3'te bulunduğu iterasyon numarası	Gözlemlenen Amaç Fonksiyonu Değeri	Öğrenme Algoritması	1-2. Gizli Katman Nöron Sayıları	1-2. Gizli Katman Aktivasyon Fonksiyonları
43	0.11782	SCG	7-4	purelin-purelin
46	0.11655	SCG	6-4	purelin-purelin
65	0.11153	SCG	7-4	tansig-purelin

Toplam proje eforu modeli (Model 2) için BO-2 ve BO-3 denemeleri sonucu elde edilen ortalama MSE-log doğrulama performansı açısından umut verici ağ yapıları (Çizelge 4.8 ve Çizelge 4.10'da bulunan) Çizelge 4.11'de özetlenmiştir.

Çizelge 4.11. BO-2 ve BO-3 sonucuna göre toplam proje efor tahmini modeli için ileri denemelerde kullanılmak üzere seçilen YSA yapıları

BO ile elde edildiği iterasyon numarası	Gözlemlenen		Gizli Nöron Sayısı	Gizli Katmandaki Aktivasyon Fonksiyonu
	Amaç Fonksiyonu Değeri	Öğrenme Algoritması		
BO-2'nin 8. adımında	0.11592	SCG	3	tansig
BO-2'nin 20. adımında	0.11106	SCG	7	tansig
BO-2'nin 30. adımında	0.11651	SCG	8	tansig
BO-3'ün 43. adımında	0.11782	SCG	7-4	purelin-purelin
BO-3'ün 46. adımında	0.11655	SCG	6-4	purelin-purelin
BO-3'ün 65. adımında	0.11153	SCG	7-4	tansig-purelin

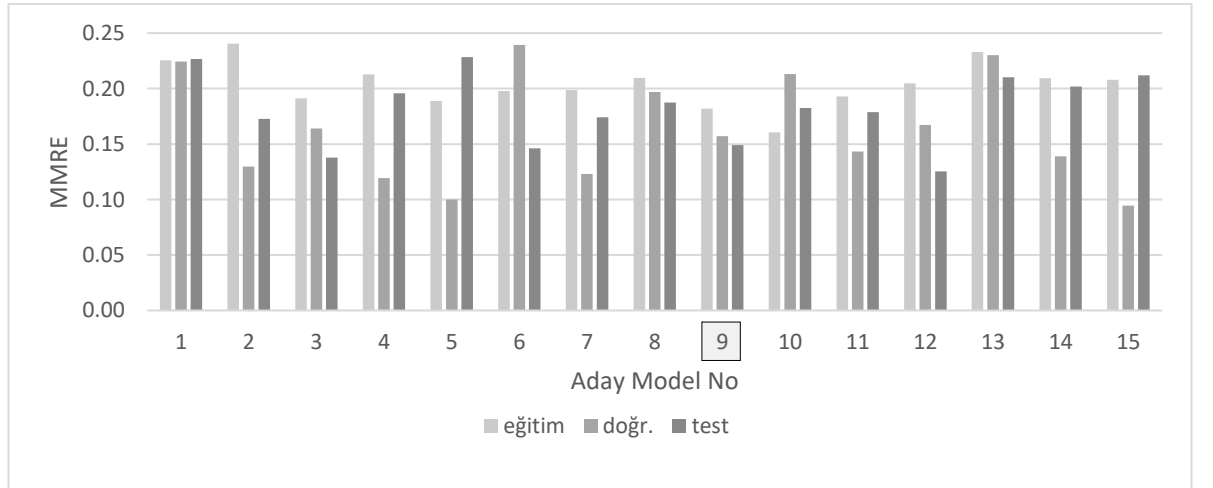
Çizelge 4.11'de yer alan yapılar ve ek olarak tek gizli katmanlı, gizli katman nöron sayısı 4, 5 ve 6 olan, gizli katman aktivasyon fonksiyonu tansig olan yapılar ele alınarak, her bir ağ yapısı için 500'den fazla deneme yapılmış olup, her bir ağ yapısı için elde edilen en iyi 10 denemeye ait performans değerleri Ek 10 ve Ek 11'de listelenmiştir. Bütün denemelerde "SCG" öğrenme algoritması kullanılmıştır. Çizelge 4.10'da bulunan her bir hiperparametre kombinasyonu için Ek 10 ve Ek 11'deki denemeler arasından MMRE değerleri 0.25 altında olup PRED(%25) değerleri %65 üzerinde olan umut vadeden adaylar Çizelge 4.12'de özetlenmiştir. 3, 4 ve 6 gizli katman nöronlu, tansig aktivasyonuna sahip tek gizli katmanlı yapılar ve 6-4 ve 7-4 nöronlu iki gizli katmanında da purelin fonksiyonu barındıran yapılarda MMRE ve PRED değerlerine ait MMRE değerleri 0.25 altında olup PRED(%25) değerleri %65 üzerinde olan herhangi bir ağ bulunamamıştır. MMRE ve PRED değerlerine göre seçilen denemelere aday model numarası verilerek çizelgede listelenmiştir. Çizelge 4.12'de de görüldüğü üzere daha çok tek gizli katmanlı, 7 nöronlu oluşan ve aktivasyon fonksiyonu tansig yapısında olan yapı ile iyi değerlere ulaşılmıştır.

Çizelge 4.12. Model 2 için bazı aday denemeler ve en iyi deneme

Gizli Nöron Sayısı	Aktivasyon fonksiyonu	Deney Numarası (Ek 10)	Aday Model Numarası	MMRE			PRED(%25)		
				E	D	T	E	D	T
5	tansig	3	1	0.23	0.22	0.23	%68	%67	%73
7	tansig	1	2	0.24	0.13	0.17	%65	%100	%73
7	tansig	2	3	0.19	0.16	0.14	%73	%78	%91
7	tansig	3	4	0.21	0.12	0.20	%70	%100	%73
7	tansig	4	5	0.19	0.10	0.23	%74	%100	%82
7	tansig	5	6	0.20	0.24	0.15	%72	%78	%82
7	tansig	6	7	0.20	0.12	0.17	%72	%89	%82
7	tansig	7	8	0.21	0.20	0.19	%72	%89	%82
7	tansig	8	9	0.18	0.16	0.15	%74	%78	%91
7	tansig	9	10	0.16	0.21	0.18	%81	%78	%73
7	tansig	10	11	0.19	0.14	0.18	%74	%78	%82
8	tansig	1	12	0.20	0.17	0.13	%77	%78	%82
8	tansig	4	13	0.23	0.23	0.21	%65	%67	%73
8	tansig	7	14	0.21	0.14	0.20	%67	%78	%73
7-4	tansig-purelin	1	15	0.21	0.09	0.21	%72	%100	%82

E: Eğitim, D: Doğrulama, T: Test

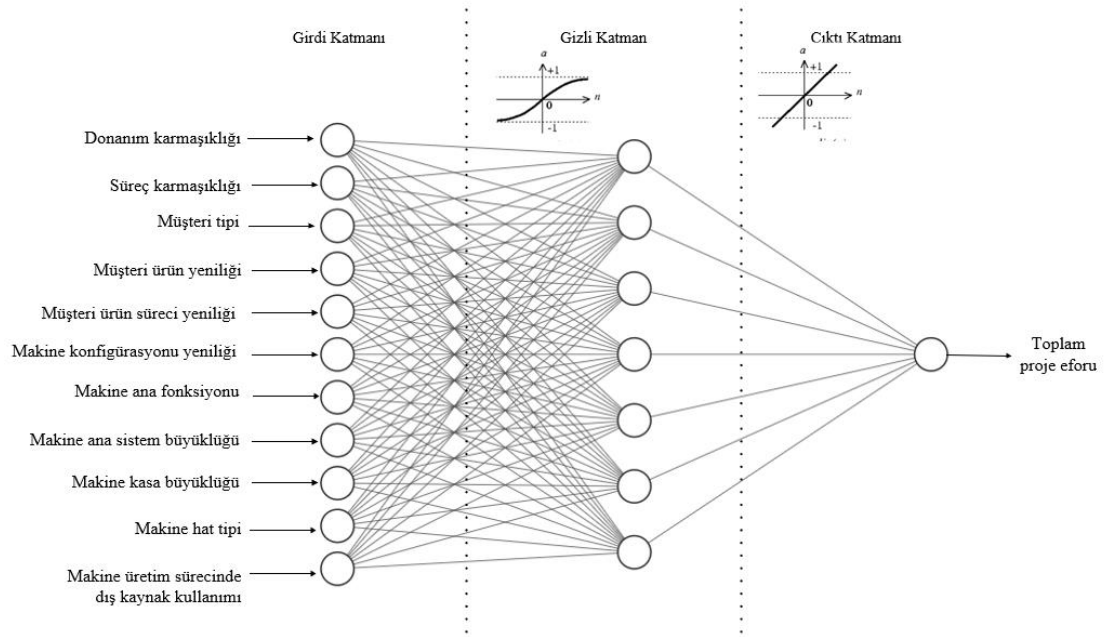
Şekil 4.9’da 15 adet aday modellere ait eğitim, doğrulama ve test MMRE değerleri karşılaştırılmıştır. Eğitim, doğrulama ve test MMRE değerlerinin diğer aday modellere göre daha dengeli ve düşük olmasından dolayı ve diğer hata performans değerleri de göz önünde bulundurularak Aday Model numarası 9 olan model seçilmiştir.



Şekil 4.9. Model 2 için seçilen aday modellerin MMRE değerleri

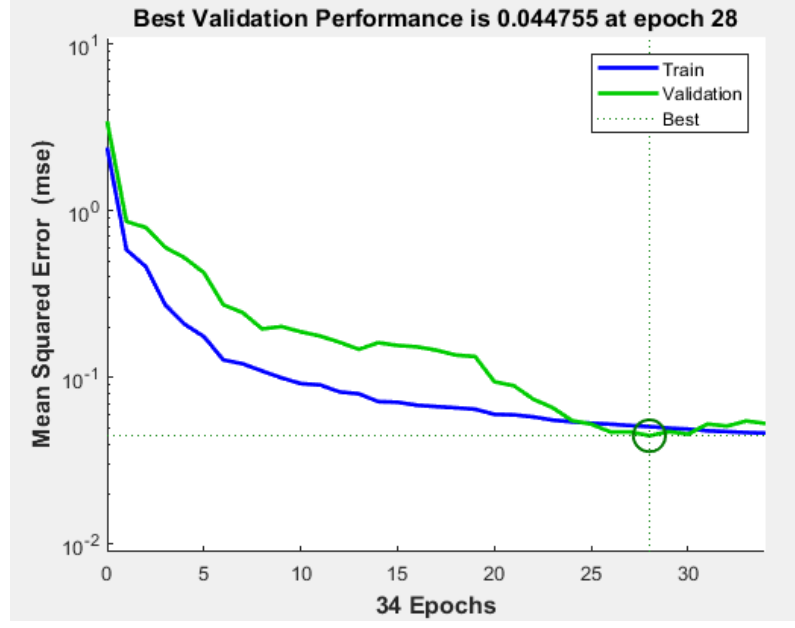
4.4. Toplam Proje Eforu Tahmini Modeli için En iyi Ağ Yapısı ve Performansı

Toplam proje eforunu tahmin etmek üzere seçilen YSA yapısı Şekil 4.10'da gösterilmiştir. 11 adet girdi ve 1 adet çıktıdan oluşan model tek gizli katmanlı ve 7 adet gizli nöron yapısına sahip olup, aktivasyon fonksiyonları gizli katman için “tansig” ve çıktı katmanı için “purelin” olarak belirlenmiştir. Ağın eğitimi SCG algoritması ile gerçekleştirilmiştir.



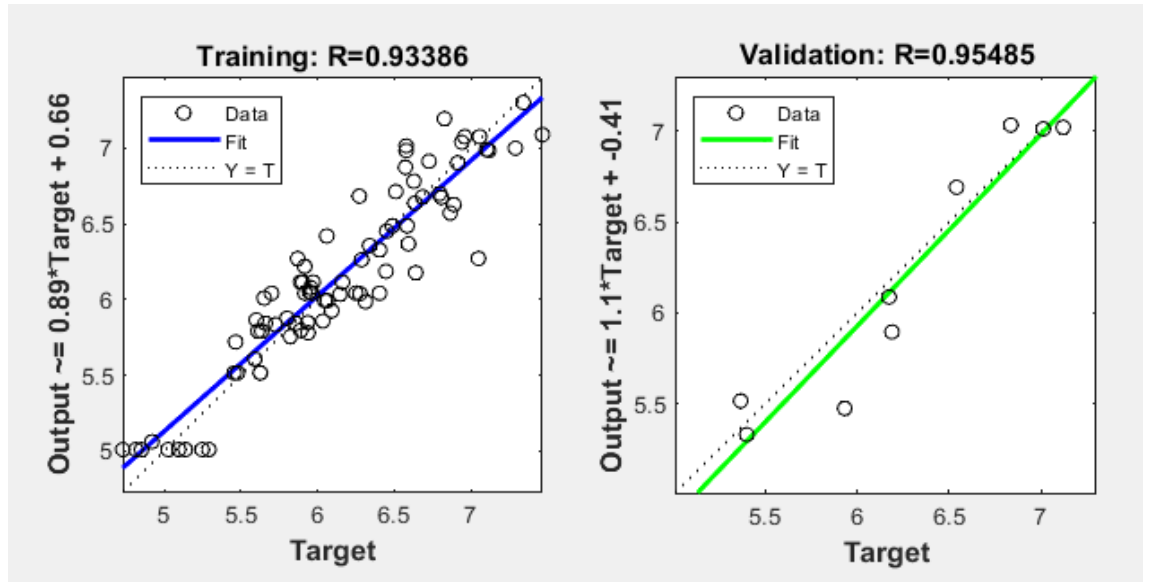
Şekil 4.10. Proje toplam eforu (Model 2) için önerilen YSA yapısı

Belirlenen en iyi ağ yapısının eğitim grafikleri Şekil 4.11'de verilmiştir. Eğitim, 28. iterasyondan sonraki artarda 6 iterasyonda elde edilen doğrulama hatası, 28. iterasyona kadar elde edilmiş en küçük hatadan daha küçük olmadığı için, diğer bir deyişle doğrulama hatası daha çok iyileşmediği için eğitim 28. iterasyonda durdurulmuştur. Şekil 4.11'de eğitim ve doğrulama hatalarının eğitimin her bir iterasyondaki değişimi ve eğitimin durdurulduğu nokta gözlemlenmektedir.



Şekil 4.11. Model 2 - YSA eğitim ve doğrulama hatalarının eğitimin her bir iterasyonundaki değişimi

Ağın korelasyon (R) değeri sonuçları Şekil 4.12’de gösterilmiş olup, eğitim doğrulama ve test veri setlerine ait hata değerleri Çizelge 4.13’de verilmiştir.

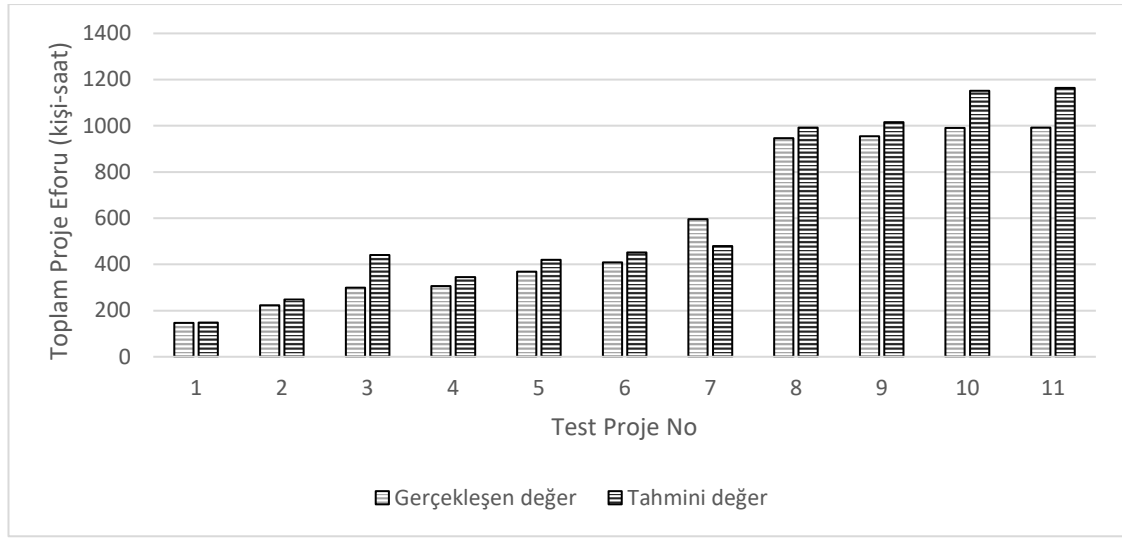


Şekil 4.12. Model 2 - YSA eğitim ve doğrulama korelasyon (R) değeri sonuçları

Çizelge 4.13. Model 2 için elde edilen en iyi ağın hata performans değerleri

	MSE_{log}	R-value	MMRE	PRED(%25)
Eğitim	0.05	0.93	0.18	%74
Doğrulama	0.04	0.95	0.16	%78
Test	0.03	0.98	0.15	%91

Toplam proje eforunu tahmin etmek üzere geliştirilen Model 2 için önerilen YSA modeli ile 11 adet test projesinin toplam proje eforuna ait yapılan tahminler ve gerçekleşen değerler Şekil 4.13'te verilmiştir. Toplam proje eforu tahmini değerleri logaritmik değerler olup, gerçekleşen değerler ile karşılaştırılmak üzere geri dönüştürülmüştür.

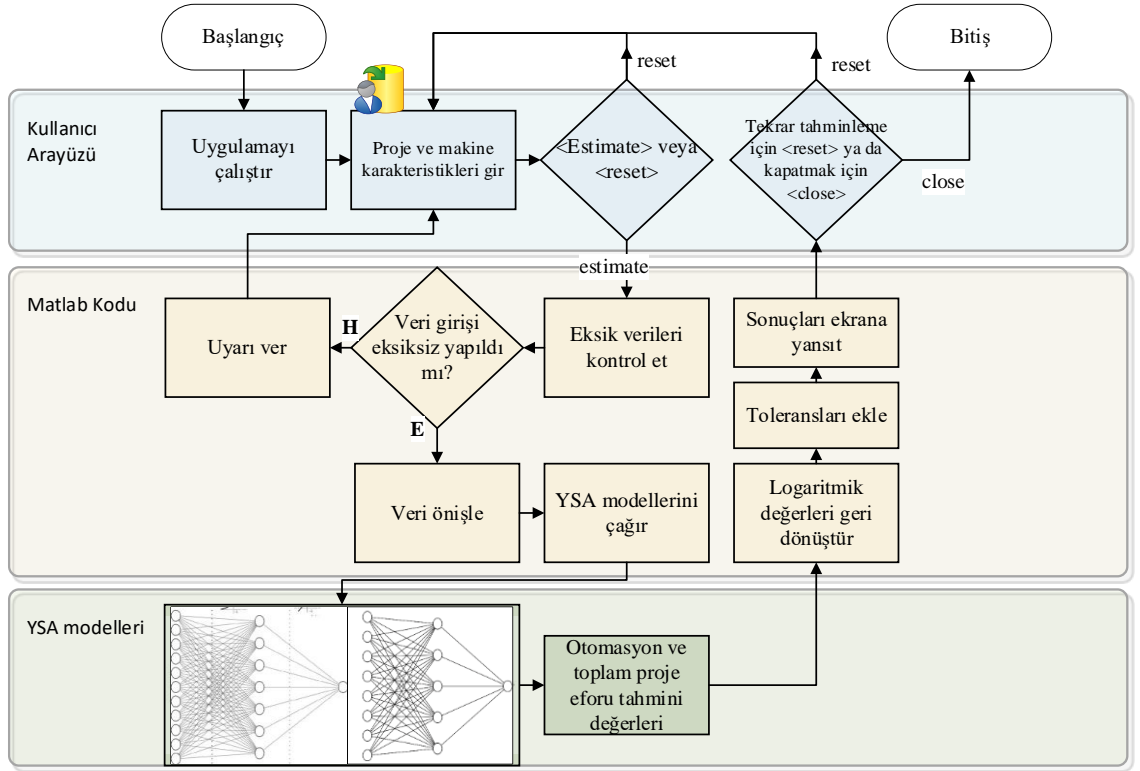


Şekil 4.13. Toplam proje eforu tahminleri ve gerçek değerleri

Proje efor tahmini alanında geliştirilen modellerde genelde MMRE değeri 0.25'in altında olan ve PRED(%25) değeri %75'in üstünde olan modeller, tahmin için yeterli kabul edilmektedir (Pospieszny ve diğerleri, 2018). Proje efor tahmini ile ilgili Çizelge 4.6'da özetlenen, YSA modeli öneren çalışmalara bakıldığında MMRE değerleri 0.14-0.59, PRED(%25) değerleri ise %42-%65 arasında değişmektedir. Bu tez çalışmasında toplam proje eforu tahmini için geliştirilen modele ait MMRE değerleri 0.15-0.18 ve PRED(%25) değerlerinin %74-%91 arasında değiştiği düşünüldüğünde, geliştirilen modelin yeterli doğrulukta olduğu sonucuna varılmıştır.

4.5. Geliştirilen Yapay Sinir Ağlarını Temel Alan Efor Tahminleme Sistemi

Proje otomasyon eforu ve toplam proje eforu için geliştirilen YSA modellerinin firmadaki proje yöneticileri ve efor tahminleyiciler tarafından kolaylıkla kullanımı açısından bir karar destek sistemi geliştirilmiştir. Bahsi geçen sistem, MATLAB 2022a App Designer kullanılarak kodlanmıştır. Geliştirilen efor tahminleme sistemi kısaca, makine geliştirme projelerinde makine süreç geliştirmesi, diğer bir deyişle otomasyonu için gereken eforu ve toplam proje eforunu tahminler. Bunu gerçekleştirirken geliştirilmiş olan iki YSA modelini kullanır. Uygulama süreç akış şeması Şekil 4.14'te gösterilmiştir.



Şekil 4.14. Otomasyon ve toplam proje eforu tahminleme sisteminin iş-akış diyagramı

Kullanıcı proje ve makine karakteristikleri ile ilgili Şekil 4.15'teki ekrandan girişler yapar. Kullanıcının makine özelliklerini olabildiğince subjektif değerlendirebilmesi açısından girdilerde yer alan her seçeneği ayrıntılı açıklayan, hangi tip makinede hangi tipte alt montaj parçaları olduğu durumlarda hangi seçeneğin seçileceği ile ilgili açıklamaların yer verildiği iki sayfalık bir açıklama metni hazırlanmıştır. Bu açıklama

metnine göre kullanıcı girişlerini sağlar. Eğer kullanıcı eksik giriş sağlar ya da bir girdi için birden fazla seçenek seçerse sistem Şekil 4.15'teki gibi uyarılar vermektedir.

Missing information!

MACHINE COMPLEXITY

- * Hardware Complexity Low High
- * Process Complexity Low High

MACHINE NOVELTY

- * Customer Old New
- Cust.Prod.Novelty Old Upgraded New *Only one option!!!*
- * Cust.Process Novelty Old Upgraded New
- * System Config. Novelty Old Upgraded New

MACHINE TYPE

- * Line Type AL FL BL CL TL
- * System Type
- * Housing Type
- * Machine Platform Entry Standalone Inline
- * Outsource Usage TG/InStock in-house prod

EFFORT ESTIMATION (man-hours)

Project Effort h - h

Automation Effort h - h

ESTIMATE

RESET

Şekil 4.15. Otomasyon ve toplam proje eforu tahminleme sistemi – eksik giriş uyarıları

Kullanıcı girişlerinin eksiksiz tamamlanmasından sonra, kullanıcı “ESTIMATE” tuşuna basarak, uygulamayı çalıştırır.

Kullanıcı girişleri ön işlemden geçirildikten sonra en iyi bulunan kaydedilmiş ağ yapıları kullanılarak efor tahmini değeri belirli bir tolerans değeri ile birlikte ekrana yansıtılır. Bu tolerans değerlerini belirlemek için, eğitim, doğrulama ve testte kullanılan veri setlerinin gerçekleşen ve tahmini değerleri kullanılmıştır. Veri setindeki toplam 101 proje için yüzde hata değerleri denklem (4.1) ile hesaplanmıştır:

$$\text{Yüzde hata} = \frac{\sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{Y_i}}{n} \quad (4.1)$$

Y_i : Gerçekleşen efor değeri

\hat{Y}_i : Tahmini efor değeri

n : veri setindeki toplam veri sayısı

Bu formül kullanarak otomasyon eforu için yüzde hata 0.160, toplam proje eforu için yüzde hata 0.175 olarak hesaplanmıştır. Kullanıcı girişleri sağlandıktan ve “ESTIMATE” tuşuna basıldıktan sonra, tahmini değerler toleransları ile ekranın sağ tarafında belirir. Proje efor tahminleme sistemi ile otomasyon eforu ve toplam proje eforu tahmini için bir gerçek hayat örneği ele alınmıştır. Firmanın veri gizliliğinden ötürü makine sistem büyüklüğü, kasa büyüklüğü gibi bilgiler açık bir şekilde burada verilmemiştir. Bu projede, yeni bir müşteri fotonik cihazı ile ilgili karmaşık test prosedürlerini çalıştıracak, firmaya yeni olan özellikte bir test makinesi talep etmektedir. Ayrıca, müşterinin üretim süreci için belirli bir çevrim süresi şartı da mevcuttur.

Müşteri ürününün testi için kullanılacak olan makinenin hareket ve görüş alt sistemleri diğer makinelerle göre karmaşık olduğundan donanım karmaşıklığı olan ilk girdi “yüksek” olarak belirlenmiştir. Üretim süreci karmaşıklığı olan ikinci girdi, müşteri ürünün üretim sürecini daha önce karşılaşılmayan ve çevrim süresi şartını içeren bir süreç olduğu için “yüksek” karmaşıklıkta olarak seçilir. Müşterinin test edilecek fotonik cihazı firmaya yeni olup, müşteri daha önce bu firmadan herhangi bir makine almamıştır. Dolayısıyla 4-6 arası girdiler “yeni” olarak işaretlenir. Müşteri test makinesi talep ettiği için 7. girdi “test” olarak seçilir. Makinenin test edilmek üzere müşteri ürününün yer alacağı ana sisteminin büyüklüğü (Girdi 8), müşteri ürünü ve sürecinin gereksinimlerine göre “orta”

olarak seçilmiştir. Müşteri ürünün test süreci gereksinimlerini karşılamak üzere gereken alt sistemleri içerecek olan makine kasanın büyüklüğü (Girdi 9) “orta” olarak seçilmiştir. Makine test etmek üzerine tasarlanacak olup, test sürecini başka herhangi bir makineye gerek olmadan gerçekleştirebildiği için Girdi 10 için de “tek başına” olarak seçilmiştir. Gerek makine konfigürasyonu firmaya göre yeni olduğu için gerekse de üretimde kullanılacak olan dış kaynağın kabiliyetleri ve kapasitesi düşünülerek makine üretiminin firma içinde gerçekleştirilmesine karar verilmiştir. Böylece, girdi 11 “firma içi üretim” olarak seçilir. Çizelge 4.14’te seçilen girdi değerleri listelenmiştir.

Çizelge 4.14. Otomasyon ve toplam proje eforu tahminleme sistemi örnek bir kullanıcı girişi

No	Girdi Adı	Seçilen Değer	Kullanıldığı YSA Modeli
1	Donanım karmaşıklığı	Yüksek	Toplam Proje Eforu
2	Süreç karmaşıklığı	Yüksek	Otomasyon/ Toplam Proje Eforu
3	Müşteri tipi	Yeni	Toplam Proje Eforu
4	Müşteri ürünü yeniliği	Yeni	Otomasyon/ Toplam Proje Eforu
5	Müşteri ürün süreci yeniliği	Yeni	Otomasyon/ Toplam Proje Eforu
6	Makine konfigürasyon yeniliği	Yeni	Otomasyon/ Toplam Proje Eforu
7	Makine ana fonksiyonu	Test	Otomasyon/ Toplam Proje Eforu
8	Makine ana sistem büyüklüğü	Orta	Otomasyon/ Toplam Proje Eforu
9	Makine kasa büyüklüğü	Orta	Otomasyon/ Toplam Proje Eforu
10	Makine hat tipi	Tek başına	Otomasyon/ Toplam Proje Eforu
11	Makine üretim sürecindeki dış kaynak kullanımı	Firma içi üretim	Toplam Proje Eforu

Örnekteki girişler ile sistem çalıştırıldığında, proje otomasyon efor değeri 257 kişi-saat olarak elde edilmiştir (Şekil 4.16). Bu tahmini değere, 0.16 olarak belirlenen yüzde hata değeri de eklendiğinde tolerans değeri ekrana $257 \times 0.16 \cong 41$ kişi-saat olarak yansıtılmıştır.

Örnekteki girişler ile toplam proje efor değeri 992 kişi-saat olarak elde edilmiştir. 0.175 olarak belirlenen yüzde hata değeri de eklendiğinde tolerans değeri ekrana $992 \times 0.175 \cong 174$ kişi-saat olarak yansıtılmıştır.

MATLAB App

MACHINE COMPLEXITY

Hardware Complexity Low High

Process Complexity Low High

MACHINE NOVELTY

Customer Old New

Cust. Prod. Novelty Old Upgraded New

Cust. Process Novelty Old Upgraded New

System Confg. Novelty Old Upgraded New

MACHINE TYPE

Line Type AL FL BL CL TL

System Type

Housing Type

Machine Platform Entry Standalone Inline

Outsource Usage TG/InStock in-house prod

EFFORT ESTIMATION (man-hours)

Project Effort h ± h

Automation Effort h - h

ESTIMATE

RESET

Şekil 4.16. Otomasyon ve toplam proje eforu tahminleme sisteminde yapılan tahminlerin gösterimi

5. SONUÇ

Proje efor tahmini, proje başarısı için kritik öneme sahiptir. Proje eforunun yetersiz tahminlenmesi; planlanmayan maliyetler, çalışan memnuniyetinde azalma, proje kapsamının dışına çıkılması, ürün kalitesinde düşüklük ve nihayetinde müşteri memnuniyetinde azalma gibi birtakım ciddi sorunlara sebep olmaktadır. Proje özgün ve bir kereye mahsus olması gibi özelliklerinden dolayı, doğası gereği yüksek belirsizlik içerir. Projenin ilk aşamalarında, gerçeğe yakın tahminler yapmak oldukça zordur. Karar vericilerin belirsizlik ortamında genellikle iyimser tahminler yapmaya eğilimlerinin olduğu ve iyimser tahminlerin de proje başarısızlıklarına neden olduğu düşünülmektedir.

Son 20 yılda, proje efor tahmini ile ilgili özellikle yazılım sektöründe azımsanmayacak sayıda çalışma yapılmış olup, araştırmacılar uzmanların subjektif kanaatlerine dayalı yaklaşımlardansa, daha objektif metotlara yönelmişlerdir. Yazılım efor tahmini alanında yapılan bu çalışmalarda YSA tabanlı yöntemlere oldukça sıklıkla rastlanırken, mühendislik tabanlı ürün geliştirme projelerinin efor tahmini konusunda YSA gibi makine öğrenme yöntemlerinin kullanımına aynı sıklıkla rastlanamamıştır. Literatürdeki bu boşluğu doldurmayı amaçlayarak bu tez çalışmasında, proje efor tahmini yapılmak üzere özelleştirilmiş makine geliştirme projeleri ele alınmıştır.

Bu tez çalışması kapsamında, fotonik sektöründe müşteriye göre özelleştirilmiş makinelerin tasarım, satın alma, üretim, otomasyon ve test aşamaları için gereken toplam proje eforu ve otomasyon aşaması için gereken eforu tahmin eden iki adet YSA modeli önerilmiştir. Müşteri üretim sürecini yarı veya tam otomatik gerçekleştirmeye yönelik geliştirilen makinelerin proje süreçlerinde yüksek belirsizlik söz konusudur. Bu durum özellikle proje başında müşterinin o ana kadar manuel olarak yaptığı sürecin makinede oluşturacağı gereksinimlerin belirlenmesinde zorluklar yaratabilir. Projenin özellikle yüksek efor gerektiren otomasyon aşaması için gereken eforun tahmin edilmesi de önem arz etmektedir. Bu noktada, proje efor tahmin modelinin yanı sıra bu aşamanın da ayrıca tahmin edilmesinin proje yöneticilerine ya da efor tahmincilerine yardımcı olacağı düşünülmektedir.

Birinci model, projenin yüksek belirsizlik içeren ve gereken efor açısından öne çıkan aşamalarından biri olan otomasyon aşaması için gereken eforu tahminlemek üzerinedir. İkinci model ise tasarım, satın alma, üretim, otomasyon ve test aşamalarından oluşan makine geliştirme projelerinin toplam proje eforunu tahminlemeye yöneliktir. Proje otomasyon eforu ve toplam proje eforu değerlerini tahminlemek üzere, makinenin karmaşıklığı, firmaya olan yeniliği gibi birtakım özellikler ile ilgili toplamda 11 adet girdi konunun uzmanları ile belirlenmiştir. Proje otomasyon eforu modeli için bu girdilerin otomasyon eforu olan bağımlı değişkenine etkileri Komşuluk Bileşenleri Analizi ile incelenmiştir. Bazı girdilerin otomasyon eforuna etkilerinin olmadığı sonucuna varılmış olup, bu girdiler veri setinden çıkartılmıştır. Bunun sonucunda, toplam 11 adet girdiden 8 adeti otomasyon eforu tahmini için kullanılmıştır. Literatürde proje efor tahmini ile ilgili önerilen diğer YSA modellerinin genellikle deneme yanılma ile elde edildiği gözlemlenmiş olup, bu tez çalışmasında Bayes Optimizasyonu (BO) kullanılmıştır. BO'nun hiperparametre optimizasyonu için kullanıldığı durumda, optimize edilmeye çalışılan değişkenler YSA hiperparametreleri olup amaç fonksiyonu YSA hata performansdır. YSA hata performansı bir çapraz doğrulama yöntemi kullanılarak doğrulama veri seti üzerinden hesaplanmıştır. BO her bir aşamada YSA performansını iyileştirecek yönde, o ana kadarki gerçekleştirilen deneyleri de dikkate alarak, yeni hiperparametre değerleri seçmektedir. Daha karmaşık olan 11 adet girdili proje efor tahmini yapan ikinci model için gerçekleştirilen hiperparametre optimizasyonunda, öğrenme algoritması da hiperparametre olarak ele alınmıştır. BO algoritmasına, seçilen öğrenme algoritmasına göre bazı hiperparametreleri aktifleştiren koşullar da katılmıştır. Son olarak, BO algoritması ile elde edilen veriler ışığında umut verici parametreler ile ek denemeler yapılarak en iyi ağlar elde edilmiştir.

Tez kapsamında, proje otomasyon eforunu ve toplam proje eforunu tahminlemek üzere geliştirilen iki adet YSA modeli, 90 adet proje verisi kullanılarak eğitilmiş olup, eğitimde kullanılmayan 11 adet gerçek hayat projesi ile test edilmiştir. Elde edilen en iyi YSA yapılarının sonuçları, proje efor tahmini alanında oldukça sıklıkla kullanılan MMRE ve PRED(%25) performans kriterleri ile değerlendirilmiştir. Literatürde, MMRE değerinin 0.25'ten az olduğu PRED(%25) değerinin ise %75'ten fazla olduğu modeller yeterli olarak görülmektedir. Bu çalışmada, eğitim ve doğrulama süreçlerinde modele

gösterilmemiş test veri seti ile model test edilmiştir. Test veri setine ait MMRE değeri otomasyon efor tahmin modeli (Model 1) için 0.30, toplam proje efor tahmini modeli (Model 2) için ise 0.15 olarak hesaplanmıştır. Modellerin PRED(%25) değerleri ise sırası ile %73 ve %91 olarak elde edilmiştir. Elde edilen veriler göz önüne alındığında geliştirilen modellerin efor tahmini için yeterli olduğu sonucuna varılmıştır.

Bu tez çalışması kapsamında geliştirilen YSA modelinin literatürde daha az sıklıkla karşılaşılan bir sektöre yönelik olması ve yeni girdiler barındırması açısından literatüre katkılar sağladığı düşünülmektedir. Bu tez çalışması literatüre sunduğu katkıların yanında, firmaya da önemli katkılar sunmaktadır. Firmadaki mevcut düzende, proje teklifi hazırlanırken deneyimli teknik satış ekibi proje için toplam süre tahmini yapmaktadır. Proje başlangıcında ise proje yöneticisi tarafından, tasarım, üretim ve otomasyon takımlarının liderlerinin desteği ile proje aşamaları için süre tahminleri yapılmaktadır. Bu tez kapsamında geliştirilen sistem ile firmada süre tahmininden önce efor tahmininin yapılması gerekliliği ile ilgili farkındalık yaratılmıştır. Geçmiş projelerde gerçekleşen proje efor değerleri kullanılarak geliştirilen bu sistem, proje karakteristikleri ile ilgili bilgilerin girilmesi ile nispeten deneyimsiz kişilerin bile projenin otomasyon efor tahminlerini uygun bir şekilde yapabilmelerine olanak sağlamıştır. Projelerin geçmiş kayıtlarından elde edilen bilgi birikiminin kullanılması ile geliştirilen sistemin, gelecek projelerin efor tahminlerinin daha sağlıklı ve doğru yapılmasına olanak sağlayacağı düşünülmektedir. Proje yöneticilerine proje için süre ve bütçe gibi planların geliştirilmesi aşamalarında oldukça yarar sağlayacağı düşünülmektedir.

Geliştirilen YSA modellerinin proje yöneticileri tarafından kolaylıkla kullanılabilmesi için, kullanıcı form ekranlarına sahip bir karar destek sistemi geliştirilmiştir. Proje yöneticileri ve proje efor tahmin edicileri benzer bir projede yeterli miktarda tecrübeye sahip olmasalar bile bu sistem aracılığıyla projenin en önemli aşamalarından biri olan otomasyon aşaması ve toplam proje eforu ile ilgili gereken efor değeri ile ilgili bir fikir elde edeceklerdir. Projelerin geçmiş kayıtlarından elde edilen bilgi birikiminin kullanılması ile geliştirilen YSA modellerini temel alan karar destek sisteminin, gelecek projeler için efor tahminlerinin daha sağlıklı ve doğru yapılmasına olanak sağlayacağı

düşünülmektedir. Sistem ile elde edilen efor tahminlerinin, proje ile ilgili süre ve bütçe gibi planların geliştirilmesinde de oldukça yarar sağlayacağı düşünülmektedir.

Firmada zamanla süreçlerde iyileşmeler söz konusu olabileceği için proje eforlarında iyi yönde gelişmeler beklenmektedir. Bu durumda gelecek projelerde modelin etkin bir şekilde kullanılması adına, belirli aralıklarla hata değerleri gözden geçirilip, belli bir eşik değeri geçtiğinde, tekrar eğitim gerekliliği düşünülecektir. Yeni projeler geldikçe, eski projelerin veri setinden çıkarılması durumu da söz konusu olabilir.

Bu çalışmada efor tahmininde daha çok makine karakteristikleri kullanılmıştır. Efor tahminlemek üzere YSA modelleri gerçekleşen efor değerleri ile eğitilmiştir. Bu gerçekleşen efor değeri çalışanın deneyimi, kabiliyeti ve disiplini gibi faktörlerden etkilenebilir. Gelecek çalışmalarda efor tahmini için makine karakteristiklerinin yanı sıra proje takım kabiliyetleri, deneyimleri, büyüklüğü, disiplin çeşitliliği ve lokasyonları gibi insan faktörleri de ele alınabilir.

KAYNAKLAR

- Ali, A., & Gravino, C. A. (2019). Systematic literature review of software effort prediction using machine learning methods. *Journal of Software: Evolution and Process*, 31 (10), 1-25. <https://doi.org/10.1002/smr.2211>
- Arora, S., & Mishra, N. (2018). Software cost estimation using artificial neural network. *Advances in Intelligent Systems and Computing*, 584, 51–58. https://doi.org/10.1007/978-981-10-5699-4_6
- Arundachawat, P., Roy, R., & Al-Ashaab, A. (2013). An analogy-based estimation framework for design rework efforts. *Journal of Intelligent Manufacturing*, 24(3), 625-639. <https://doi.org/10.1007/s10845-011-0605-6>
- Attarzadeh, I., & Ow, S.H. (2011). Software development cost and time forecasting using a high performance artificial neural network model. In *Intelligent Computing and Information Science, International Conference, Part I, CCIS* 134. 18–26. https://doi.org/10.1007/978-3-642-18129-0_4
- Attarzadeh, I., Mehrazadeh, A., & Barati, A. (2012). Proposing an enhanced artificial neural network prediction model to improve the accuracy in software effort estimation. *Proceedings of the International Conference on Computational Intelligence, Communication Systems and Networks*, 167–172. <https://doi.org/10.1109/CICSyN.2012.39>
- Bashir, H. A., & Thomson, V. (2004). Estimating design effort for GE hydro projects. *Computers & Industrial Engineering*, 45(2), 195-204. <https://doi.org/10.1016/j.cie.2003.12.005>
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2012), 281-305. <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>
- Berry, M. J., & Linoff, G. (1997). *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York, NY, USA: John Wiley & Sons, Inc.
- Blum A. (1992). *Neural networks in C++: an object-oriented framework for building connectionist systems*. John Wiley and Sons Inc, New York.
- Brochu, E., Cora, V. M., & Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. doi.org/10.48550/arXiv.1012.2599
- Carvalho, H. D. P., Fagundes, R., & Santos, W. (2021). Extreme learning machine applied to software development effort estimation. *IEEE Access*, 9, 92676-92687. <https://doi.org/10.1109/ACCESS.2021.3091313>

- Chan M. C., Wong, C. C., Lam, C. C. (2000). Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization. *Comput Econ Finance* 61:326–342.
- Chang, W. F., Mak, M. W. (1999). A conjugate gradient learning algorithm for recurrent neural networks. *Neurocomputing*, 24(1–3), 173-189. [https://doi.org/10.1016/S0925-2312\(98\)00104-0](https://doi.org/10.1016/S0925-2312(98)00104-0).
- Chapman, C., & Ward, S. (2003). Project risk management, processes, techniques and insights, 2nd ed. John Wiley&sons, Ltd, England, 389 pp.
- Chow, S. K. H., Lee, E. W. M., & Li, D. H. W. (2012). Short-term prediction of photovoltaic energy generation by intelligent approach. *Energy and Buildings*, 55 (2012), 660-607. <http://dx.doi.org/10.1016/j.enbuild.2012.08.011>
- Dave, V. S., & Dutta D. M. K. (2011). Application of feed-forward neural network in estimation of software effort. *IJCA Proceedings on International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC)*, 5, 5-9.
- Gelbart, M. A. (2015). Constrained bayesian optimization and applications. [Doctoral dissertation, Harvard University, Graduate School of Arts & Sciences]
- Gilbreath, R. D. (1988). Working with pulses not streams. Using projects to capture opportunity, in Cleland, I.C.&King, D.I. Project Management Handbook, 2nd ed. John Wiley&Sons, Inc., New York, 997 pp.
- Goyal, S., & Bhatia, P. K. (2019). Feature selection technique for effective software effort estimation using multi-layer perceptrons. Emerging trends in information technology, *Lecture Notes in Electrical Engineering*, 605, 183–194. https://doi.org/10.1007/978-3-030-30577-2_15
- Hameed, S., Elsheikh, Y., & Azzeh, M. (2023). An optimized case-based software project effort estimation using genetic algorithm. *Information and Software Technology*, 153. <https://doi.org/10.1016/j.infsof.2022.107088>
- Haykin, S. S. (2008). *Neural networks and learning machines*, 3rd ed.; Prentice Hall/Pearson.
- Heiat, A. (2002). Comparison of artificial neural network and regression models for estimating software development effort. *Information and Software Technology*, 44(15), 911-922. [https://doi.org/10.1016/S0950-5849\(02\)00128-3](https://doi.org/10.1016/S0950-5849(02)00128-3)
- Hutter, F., Kotthoff, J., & Vanschoren, J. (2019). Automated machine learning: Methods, systems, challenges. Springer (Chapter 1), <https://doi.org/10.1007/978-3-030-05318-5>

- Idri, A., Khoshgoftaar, T. M. ve Abran, A. (2002). Can neural networks be easily interpreted in software cost estimation? *IEEE World Congress on Computational Intelligence*. [doi:10.1109/FUZZ.2002.1006668](https://doi.org/10.1109/FUZZ.2002.1006668)
- Jaifer, R., Beauregard, Y., & Bhuiyan, N. (2021). New Framework for effort and time drivers in aerospace product development projects. *Engineering Management Journal*, 33(2), 76-95. <https://doi.org/10.1080/10429247.2020.1772950>
- Jørgensen, M., & Sjøberg, D. I. K. (2004). The impact of customer expectation on software development effort estimates. *International Journal of Project Management*, 22(4), 317-325. [https://doi.org/10.1016/S0263-7863\(03\)00085-1](https://doi.org/10.1016/S0263-7863(03)00085-1)
- Johansson, E. M., Dowla, F. U., & Goodman, D. M. (1992). Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, 2(4), 291-301. <https://doi.org/10.1142/S0129065791000261>
- Jun, E. S., & Lee, J. K. (2001). Quasi-optimal case-selective neural network model for software effort estimation. *Expert System with Applications*, 21, 1-14.
- Kerzner, H. (2003). *Project Management*, 8th ed. John Wiley&Sons, Inc., New York, 891 pp.
- Koch, S., & Mitlöhner, J. (2009). Software project effort estimation with voting rules. *Decision Support Systems*, 46(4), 895-901. <https://doi.org/10.1016/j.dss.2008.12.002>
- Kumar, P. S., Behera, H. S., Kumari, A. K., Nayak, J., & Naik, B. (2020). Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. *Computer Science Review*, 38. <https://doi.org/10.1016/j.cosrev.2020.100288>
- Leach, L.P. (2005). *Critical Chain Management*, 2nd ed. Artech House, Inc., London, 8 pp.
- López-Martín, C., Chavoya, A., & Meda-Campaña, M. E. (2011). Software development effort estimation in academic environments applying a general regression neural network involving size and people factors. *Pattern Recognition*, 3rd Mexican Conference, MCPRLNCS 6718 269–277. https://doi.org/10.1007/978-3-642-21587-2_29
- Makarova, A., Shen, H., Perrone, V., Klein, A., Faddoul, J. B., Krause, A., Seeger, M., & Archambeau, C. (2021). Overfitting in bayesian optimization: an empirical study and early-stopping solution. *ICLR 2021 Workshop on Neural Architecture Search*, <https://www.amazon.science/publications/overfitting-in-bayesian-optimization-an-empirical-study-and-early-stopping-solution>

MATLAB R2022b.

- McCharty C. C. (2016). Program Management in Defense and High-Tech Environments. CRC Press, USA, 267 pp.
- Monika, & Sangwan, O. P. (2017). Software effort estimation using machine learning techniques. *7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, India*, <https://doi.org/10.1109/CONFLUENCE.2017.7943130>
- Nassif, A. B., Ho, D., & Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, *86(1)*, 144–160. <https://doi.org/10.1016/j.jss.2012.07.050>
- Nguyen, V., Gupta, S., Rana, S., Li, C., & Venkatesh, S. (2017). Regret for expected improvement over the best-observed value and stopping condition. *Proceedings of the Ninth Asian Conference on Machine Learning PMLR*, *77*, 279-294.
- Pandey, M., Litoriya, R., & Pandey, P. (2020). Validation of existing software effort estimation techniques in context with mobile software applications. *Wireless Personal Communications*, *110(4)*, 1659–1677. <https://doi.org/10.1007/s11277-019-06805-0>
- Park, H., & Baek, S. (2008). An empirical validation of a neural network model for software effort estimation. *Expert Systems with Applications*, *35(3)*, 929–937. <https://doi.org/10.1016/j.eswa.2007.08.001>
- Pinto, J. K. (2016). Project Management, Achieving Competitive Advantage, 4th ed. Pearson Education, Inc., USA, 542 pp.
- PMI, (2013). A guide to the project management body of knowledge, 5th ed. Project Management Institute, Inc., USA, 589 pp.
- Pollmanns, J., Hohnen, T., & Feldhusen J. (2013). An information model of the design process for the estimation of product development effort. *Proceedings of the 23rd CIRP Design Conference, Smart Product Engineering, Bochum, Germany*, 885-894. https://doi.org/10.1007/978-3-642-30817-8_87
- Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, 184-196. <https://doi.org/10.1016/j.jss.2017.11.066>
- Prater, J., Kirytopoulos, K., & Ma, T. (2017). Optimism bias within the project management context. *International Journal of Managing Projects in Business*, *10(2)*, 370-385. <https://doi.org/10.1108/IJMPB-07-2016-0063>
- Predescu, E. F., Stefan, A., & Zaharia, A. V. (2019). Software effort estimation using multilayer perceptron and long short term memory. *Informatica Economică*, *23(2)*, 76-87. <https://doi.org/10.12948/issn14531305/23.2.2019.07>

- Quah, T. S. ve Thwin, M. M. T. (2003). Application of neural networks for software quality prediction using object-oriented metrics. *IEEE Computer Society International Conference on Software Maintenance (ICSM'03)*. [doi:10.1109/ICSM.2003.1235412](https://doi.org/10.1109/ICSM.2003.1235412)
- Raftery, J. (1994). Risk analysis in project management, 2nd Edition. Routledge, London and New York, 143 pp.
- Ramussen, C. E., Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press, Massachusetts Institute of Technology. www.GaussianProcess.org/gpml
- Rankovic, N., Rankovic, D., Ivanovic, M., & Lazic, L. (2021). Improved effort and cost estimation model using artificial neural networks and taguchi method with different activation functions. *Entropy*, 23(7), 854. <https://doi.org/10.3390/e23070854>
- Rao, P. S., & Kumar, R. K. (2015). Software effort estimation through a generalized regression neural network. In *Emerging ICT for Bridging the Future (1), Advances in Intelligent Systems and Computing*, 337, 19-30. https://doi.org/10.1007/978-3-319-13728-5_3
- Rijwani, P., & Jain, S. (2016). Enhanced software effort estimation using multi layered feed forward artificial neural network technique. *Procedia Computer Science*, 89, 307–312. <https://doi.org/10.1016/j.procs.2016.06.073>
- Salam, A., Bhuiyan, N. F., Gouw, G. J., & Raza, S. A. (2007). Estimating design effort in product development: A case study at Pratt & Whitney Canada. *International Conference on Industrial Engineering and Engineering Management*, <https://doi.org/10.1109/IEEM.2007.4419340>
- Shin, S., Lee, Y., Kim, M., Park, J., Lee, S., Min, K. (2020). Deep neural network model with Bayesian hyperparameter optimization for prediction of Nox at transient conditions in a diesel engine. *Engineering Applications of Artificial Intelligence*, 94 (2020). <https://doi.org/10.1016/j.engappai.2020.103761>
- Sicard, D., Brioris, P., Billard, A., Thevenot, J., Boichut, E., Chapellier, J., Bernard, F. (2022). Deep learning and Bayesian hyperparameter optimization: A data-driven approach for diamond grid segmentation toward grinding wheel characterization. *Applied Sciences*, 12, 12606. <https://doi.org/10.3390/app122412606>
- Singh, A. J., & Kumar, M. (2020). Comparative analysis on prediction of software effort estimation using machine learning techniques. *1st International Conference on Intelligent Communication and Computational Research (ICICCR-2020)*, *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3565822>
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Proceedings of Advances in Neural Information Processing Systems*. 25. <https://doi.org/10.48550/arXiv.1206.2944>

- Trendowicz, A., Jeffery, R. (2014). Software Project Effort Estimation: Foundations and Best Practice Guideliness for Success, *Springer International Publishing Switzerland*.
- Tronto, I. F. B., Silva, J. D. S., & Sant'Anna, N. (2008). An investigation of artificial neural networks based prediction systems in software project management. *Journal of Systems and Software*, 81(3), 356–367. <https://doi.org/10.1016/j.jss.2007.05.011>
- Tofallis, C. (2015). A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society* 66, 1352-1362. [doi:10.1057/jors.2014.103](https://doi.org/10.1057/jors.2014.103)
- Usman, M., Britto, R., Damm, L. O., & Börstler, J. (2018). Effort estimation in large-scale software development: An industrial case study. *Information and Software Technology*, 99, 21-40. <https://doi.org/10.1016/j.infsof.2018.02.009>
- Wolpert, D., Macready, W. (1995). No Free Lunch Theorems for Search. *Technical Report SFI-TR-95-02-010, Santa Fe Institute: Santa Fe, NM, USA, Volume 10*.
- Yurt, Z. O., Iyigun, C., & Bakal P. (2019). Engineering effort estimation for product development projects. *IEEE International Conference on Industrial Engineering and Engineering Management, China*, <https://doi.org/10.1109/IEEM44572.2019.8978764>
- Zhao, Y., Li, Y., Feng, C., Gong, C. & Tan, H. (2022). Early warning of systemic financial risk of local government implicit debt based on BP neural network model. *Systems*, 10 (207), <https://doi.org/10.3390/systems10060207>.
- Zheng, A. (2015). Evaluating machine learning models: A beginner's guide to key concepts and pitfalls, first ed., O'Reilly Media, Inc., Sebastopol, CA 95472, 27-37.
- Zhu, H, Leandro, J, & Lin, Q. (2021). Optimization of Artificial Neural Network (ANN) for Maximum Flood Inundation Forecasts. *Water* 13 (16), 2252. <https://doi.org/10.3390/w13162252>

EKLER

- EK 1** Girdi Seçimi için MATLAB Kodu
- EK 2** Veri Önışleme için MATLAB Kodu
- EK 3** BO-1 MATLAB Kodu
- EK 4** BO-2 MATLAB Kodu
- EK 5** BO-3 MATLAB Kodu
- EK 6** Model 1 için Seçilen Hiperparametreler ile ileri Denemeler ve Test MATLAB Kodu
- EK 7** BO-1 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler
- EK 8** BO-2 Sonuçlarının Adımlar Halinde Listelenmesi
- EK 9** BO-3 Sonuçlarının Adımlar Halinde Listelenmesi
- EK 10** BO-2 ve BO-3 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler

EK 1 Girdi Seçimi için MATLAB Kodu

Feature selection using neighborhood component analysis for regression

```
for l=1:o
    mdl=fsrnca(xp,yp(:,l));
    figure()
    plot(mdl.FeatureWeights,'ro')
    grid on
    xlabel('Feature index')
    ylabel('Feature weight')
end
```


EK 2 Veri Önışleme için MATLAB Kodu

```
data=xlsread('C:\...\xlsx');

model='aut';

if model=='aut'
    xp=data(:,[2 4 5 6 7 8 9 10]);yp=data(:,[14]);%net5 aut
elseif model=='prj'
    xp=data(:,[1 2 3 4 5 6 7 8 9 10 11]);yp=data(:,[15]); %net1 all project
end
[~,n]=size(xp); [m,o]=size(yp);
for i=1:o
    y2p(:,i)=log(yp(:,i)); %log transformation for the output
end
for i=1:n %input normalization for inputs
    x2p(:,i)=(xp(:,i)-min(xp(:,i)))/(max(xp(:,i))-min(xp(:,i)));
end

%TRAINING SET
for a=1:90
    for b=1:n
        x3p(a,b)=x2p(a,b);
    end
    for b=1:o
        y3p(a,b)=y2p(a,b);
    end
end

%TEST SET
for a=91:101
    for b=1:n
        x3p_test(a-90,b)=x2p(a,b);
    end
    for b=1:o
        y3p_test(a-90,b)=y2p(a,b);
    end
end
xtp=x3p'; ytp=y3p';
xtp_test=x3p_test'; ytp_test=y3p_test';
```

EK 3 BO-1 MATLAB Kodu

Ana Kod:

```
optimvars=[optimizableVariable('HLM1',[2,8],'Type','integer');
    optimizableVariable('tranFcn',{'tansig','logsig','purelin'},'Type','categorical');
    optimizableVariable('tr_ou',{'tansig','logsig','purelin'},'Type','categorical')];
Objfun = @(z)objfunc(xtp,ytp,z.HLM1,char(z.tranFcn),char(z.tr_ou));
results=bayesopt(Objfun,optimvars,'IsObjectiveDeterministic',true,...
    'AcquisitionFunctionName','expected-improvement-plus',...
    'MaxObjectiveEvaluations',30);
plot(results,'all');
```

BO Amaç Fonksiyonu Hesaplama Fonksiyonu:

objfcn_avg - MSE LOG avg of nn trials

```
function objective=objfunc(xtp,ytp,numhid,tranFcn,tr_ou)
nn=100;sum=0;
for count=1:nn
    net = fitnet(numhid,'trainlm');

    net.divideParam.trainRatio=0.9;
    net.divideParam.valRatio=0.1;net.divideParam.testRatio=0;
    net.layers{1}.transferFcn= tranFcn;
    net.layers{2}.transferFcn=tr_ou;
    net.performFcn= 'mse';net.trainParam.max_fail=6;
    [net,tr]=train(net,xtp,ytp);
    mse_test=perform(net,ytp(:,tr.valInd),net(xtp(:,tr.valInd)));
    sum=sum+mse_test; clear net; clear tr;
end
objective=sum/nn;
end
```

EK 4 BO-2 MATLAB Kodu

Ana Kod:

```
optimvars=[
    optimizableVariable('trfunc',{'trainscg','traingdx','trainlm'],'Type','categorical');% 'traincgf','traincgp'
    optimizableVariable('HLN1',[3,11],'Type','integer');%1. gizli katman nöron sayısı
    optimizableVariable('actf1',{'tansig','logsig','purelin'],'Type','categorical');%1.gizli katmandaki akt fonk
    optimizableVariable('mc',[6,9],'Type','integer');% gdx kullanılıyorsa momentum
    optimizableVariable('lr',[1,5],'Type','integer');% gdx kullanılıyorsa learning rate
Objfun = @(z)objfunc(xtp,ytp,char(z.trfunc),z.HLN1,char(z.actf1),z.mc,z.lr);
results=bayesopt(Objfun,optimvars,'IsObjectiveDeterministic', true,...
    'AcquisitionFunctionName', 'expected-improvement-plus', ...
    'MaxObjectiveEvaluations', 60,"ConditionalVariableFcn",@CVF);
zbest = bestPoint(results);
plot(results,'all');
```

Fonksiyonlar:

- BO Amaç Fonksiyonu Hesaplama Fonksiyonu
- Koşullara göre Hiperparametre Atama Fonksiyonu

```
function objective=objfunc(xtp,ytp,trfunc,numhid1,actf1,mc,lr)
nn=100;sum=0;
for count=1:nn

    net = fitnet(numhid1,trfunc);

    net.divideParam.trainRatio=0.9;
    net.divideParam.valRatio=0.1;
    net.divideParam.testRatio=0;

    net.layers{1}.transferFcn= actf1; %ilk gizli katman
    net.layers{2}.transferFcn='purelin';%çıkıktı katmanı akt fonk ataması

    if ~isnan(mc) % eğer mc hiperparametresine değer atanmışsa, o değeri 10'a böl
        net.trainParam.mc=mc/10;
    end
    if ~isnan(lr) % eğer lr hiperparametresine değer atanmışsa, o değeri 10'a böl
        net.trainParam.lr=lr/10;
    end

    net.performFcn= 'mse';
    net.trainParam.max_fail=6;

    [net,tr]=train(net,xtp,ytp);
    mse_test=perform(net,ytp(:,tr.valInd),net(xtp(:,tr.valInd)));
    sum=sum+mse_test;
    clear net; clear tr;
end
objective=sum/nn;
end

function X = CVF(X) % BO için koşullu atamalar
% eğer eğitim fonk gdx değil ise mc ve lr ataması yapma
X.lr(ismember(X.trfunc',{'trainscg','trainlm'})) = nan;
X.mc(ismember(X.trfunc',{'trainscg','trainlm'})) = nan;
end
```

EK 5 BO-3 MATLAB Kodu

Ana Kod:

```
optimvars=[
    optimizableVariable('trfunc',{'traingdx','trainscg','trainlm'],'Type','categorical');% 'traincgf','traincgp'
    optimizableVariable('HLN1',[3,7],'Type','integer');%1. gizli katman nöron sayısı
    optimizableVariable('HLN2',[3,7],'Type','integer');% var ise 2. gizli katman nöron sayısı
    optimizableVariable('actf1',{'tansig','logsig','purelin'],'Type','categorical');%1.gizli katmandaki akt fonk
    optimizableVariable('actf2',{'tansig','logsig','purelin'],'Type','categorical'); %var ise 2. gizli katmandaki akt fonk
    optimizableVariable('mc',[6,9],'Type','integer');% gdx kullanılıyorsa momentum
    optimizableVariable('lr',[1,5],'Type','integer');% gdx kullanılıyorsa learning rate
Objfun = @(z)objefunc(xtp,ytp,char(z.trfunc),z.HLN1,z.HLN2,char(z.actf1),char(z.actf2),z.mc,z.lr);
results=bayesopt(Objfun,optimvars,'IsObjectiveDeterministic', true,...
    'AcquisitionFunctionName', 'expected-improvement-plus', ...
    'MaxObjectiveEvaluations', 70,"ConditionalVariableFcn",@CVF);
zbest = bestPoint(results);
plot(results,'all');
```

Fonksiyonlar:

- BO Amaç Fonksiyonu Hesaplama Fonksiyonu
- Koşullara göre Hiperparametre Atama Fonksiyonu

```
function objective=objefunc(xtp,ytp,trfunc,numhid1,numhid2,actf1,actf2,mc,lr)
nn=100;sum=0;
for count=1:nn

    numhid=[numhid1 numhid2];
    net = fitnet(numhid,trfunc);

    net.divideParam.trainRatio=0.9;
    net.divideParam.valRatio=0.1;
    net.divideParam.testRatio=0;

    net.layers{1}.transferFcn= actf1; %ilk gizli katman
    net.layers{2}.transferFcn= actf2;%2. gizli katman
    net.layers{3}.transferFcn='purelin'; %çıkıktı katmanı akt fonk ataması

    if ~isnan(mc) % eğer mc hiperparametresine değer atanmışsa, o değeri 10'a böl
        net.trainParam.mc=mc/10;
    end
    if ~isnan(lr) % eğer lr hiperparametresine değer atanmışsa, o değeri 10'a böl
        net.trainParam.lr=lr/10;
    end

    net.performFcn= 'mse';
    net.trainParam.max_fail=6;

    [net,tr]=train(net,xtp,ytp);
    mse_test=perform(net,ytp(:,tr.valInd),net(xtp(:,tr.valInd)));
    sum=sum+mse_test;
    clear net; clear tr;
end
objective=sum/nn;
end

function X = CVF(X) % BO için koşullu atamalar
% eğer eğitim fonk gdx değil ise mc ve lr ataması yapma
X.lr(ismember(X.trfunc,{'trainscg','trainlm'})) = nan;
X.mc(ismember(X.trfunc,{'trainscg','trainlm'})) = nan;
end
```

EK 6 Model 1 için Seçilen Hiperparametreler ile İleri Denemeler ve Test MATLAB Kodu

Ana fonksiyon:

```
clearvars -except o xtp ytp xtp_test ytp_test;
```

EN İYİ PARAMETRE İLE BİRKAÇ KEZ DENEME

```
i=1;j=0;n_trial=0; zzkosul=0;
while (zzkosul<1)
    [net,tr,sumtab,j]=FINAL(o,j,xtp,ytp,'trainlm',5,'logsig','purelin',0.9,0.1,0,xtp_test,ytp_test);
    n_trial=n_trial+1;
    if sumtab(1,4)>0.10
        summ(i,:)=sumtab(1,:);
        netArray{i}=net;
        trArray{i}=tr;
        i=i+1;
    end
    ZMMREtr=sumtab(1,7); ZMMREval=sumtab(1,8); ZMMREtest=sumtab(1,9);
    zzPREdtr_25=sumtab(1,10);    zzPREdval_25=sumtab(1,11); zzPREdtest_25=sumtab(1,12);
    if ZMMREtr<=0.45 && ZMMREval<=0.45 && ZMMREtest<0.45 && zzPREdtr_25>=0.50 && zzPREdval_25>=0.50 && zzPREdtest_25>=0.50
        zzkosul=1;
    end
    clear net and tr and sumtab and gap;
end
```

Eğitim ve Test Fonksiyonu (Ağın Eğitimi Kısmı)

function

```
[net,tr,sumtab,j]=FINAL(o,j,xtp,ytp,trfunc,HN,TFcn1,tr_ou,tr_r,val_r,test_r,xtp_test,ytp_test)
net = fitnet(HN,trfunc);
net.divideParam.trainRatio=tr_r;
net.divideParam.valRatio=val_r;
net.divideParam.testRatio=test_r;
net.layers{1}.transferFcn=TFcn1; %gizli katman
net.layers{2}.transferFcn=tr_ou; %çıktı katmanı
net.performFcn= "mse";
net.trainParam.max_fail=6;
[net,tr]=train(net,xtp,ytp);
perf_val=perform(net,ytp(:,tr.valInd),net(xtp(:,tr.valInd)));
perf_tra=perform(net,ytp(:,tr.trainInd),net(xtp(:,tr.trainInd)));

%calculating R - grafik
ctrain=corrcoef(net(xtp(:,tr.trainInd)),ytp(:,tr.trainInd));
Rtr=ctrain(2,1);
cval=corrcoef(net(xtp(:,tr.valInd)),ytp(:,tr.valInd));
Rval=cval(2,1);
```

•
•
•

EK 6 – devam

Eğitim ve Test Fonksiyonu (Eğitim ve Doğrulama Hata Hesaplama Kısmı)

```
% function devamı
% calculating error of training
pt=exp(net(xtp(:,tr.trainInd))); %predicted train
rt=exp(ytp(:,tr.trainInd)); %real train
MAEtr=mean(mean(abs(pt-rt)));
RMSEtr=sqrt(mean(mean((pt-rt).^2)));
PREDtr_25=0; [~,z]=size(tr.trainInd);
for a=1:o
    for b=1:z
        MMREmat(a,b)=abs(pt(a,b)-rt(a,b))/rt(a,b);
        if MMREmat(a,b)<=0.25
            PREDtr(a,b)= 1;
        else
            PREDtr(a,b)= 0;
        end
    end
end
MMREtr=mean(mean(MMREmat));
PREDtr_25=mean(mean(PREDtr));

% calculating error of val
pv=exp(net(xtp(:,tr.valInd))); %predicted train
rv=exp(ytp(:,tr.valInd)); %real train
MAEval=mean(mean(abs(pv-rv)));
RMSEval=sqrt(mean(mean((pv-rv).^2)));
PREDval_25=0; [~,t]=size(tr.valInd);
for a=1:o
    for b=1:t
        MMREmat2(a,b)=abs(pv(a,b)-rv(a,b))/rv(a,b);
        if MMREmat2(a,b)<=0.25
            PREDval(a,b)= 1;
        else
            PREDval(a,b)= 0;
        end
    end
end
MMREval=mean(mean(MMREmat2));
PREDval_25=mean(mean(PREDval));
```

EK 6 – devam

Eğitim ve Test Fonksiyonu (Test Kısmı): Veri seti *predicted_test* adlı ayrı bir matriste tutularak hata hesaplaması ayrı bir şekilde gerçekleştirilir.

```
% function devamı

% calculating error of test
predicted_test=exp(net(xtp_test)); real_test=exp(ytp_test);
TEST_MSE_log=mean(mean((net(xtp_test)-ytp_test).^2));
TEST_MAE=mean(mean(abs(predicted_test-real_test)));
TEST_RMSE=sqrt(mean(mean((predicted_test-real_test).^2)));
TESTPRED_25=0;[~,mt]=size(predicted_test);
for i=1:o
    for g=1:mt
        MMREmat3(i,g)=abs(predicted_test(i,g)-real_test(i,g))/real_test(i,g);
        if MMREmat3(i,g)<=0.25
            PREDtest(i,g)= 1;
        else
            PREDtest(i,g)= 0;
        end
    end
end
end
TESTMMRE=mean(mean(MMREmat3));
TESTPRED_25=mean(mean(PREDtest));

ctest=corrcoef(net(xtp_test),ytp_test);
Rtest=ctest(2,1);
```

Eğitim ve Test Fonksiyonu (Tüm Sonuçları Ana Fonksiyona Döndürme Kısmı)

```
% function devamı

bepoch=tr.best_epoch;
sonuc(1,1)= perf_tra; %mse - log - tra
sonuc(1,2)= perf_val; %mse - log - val
sonuc(1,3)= TEST_MSE_log; %mse - log - test
sonuc(1,4)=Rtr; sonuc(1,5)=Rval; sonuc(1,6)=Rtest;
sonuc(1,7)=MMREtr; sonuc(1,8)=MMREval;
sonuc(1,9)=TESTMMRE;
sonuc(1,10)=PREDtr_25; sonuc(1,11)=PREDval_25; sonuc(1,12)=TESTPRED_25;
sonuc(1,13)=bepoch;
sumtab=sonuc;
end
```

EK 7 BO-1 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler

Gizli Nöron Sayısı	Gizli Katmandaki Aktivasyon Fonksiyonu	Deneme No	R-değeri Eğitim	R-değeri Doğrulama	R-değeri Test	MMRE Eğitim	MMRE Doğrulama	MMRE Test	PRED (%25) Eğitim	PRED (%25) Doğrulama	PRED (%25) Test	En iyi adım
3	logsig	1	0.88	0.89	0.89	0.24	0.24	0.37	%72	%56	%55	9
3	logsig	2	0.83	0.72	0.75	0.32	0.26	0.49	%53	%56	%55	4
3	logsig	3	0.84	0.93	0.82	0.28	0.23	0.36	%67	%56	%55	11
3	logsig	4	0.87	0.63	0.86	0.27	0.38	0.50	%67	%56	%64	9
3	logsig	5	0.82	0.71	0.89	0.31	0.28	0.40	%51	%56	%64	4
4	logsig	1	0.93	0.88	0.90	0.20	0.25	0.37	%65	%56	%64	13
4	logsig	2	0.88	0.90	0.91	0.26	0.27	0.38	%65	%44	%64	11
4	logsig	3	0.82	0.82	0.89	0.33	0.38	0.34	%53	%44	%64	4
4	logsig	4	0.88	0.90	0.91	0.26	0.27	0.38	%65	%44	%64	11
4	logsig	5	0.85	0.92	0.91	0.27	0.22	0.40	%63	%67	%64	5
5	logsig	1	0.93	0.96	0.87	0.19	0.17	0.32	%67	%89	%64	10
5	logsig	2	0.91	0.93	0.91	0.22	0.18	0.44	%74	%89	%64	5
5	logsig	3	0.92	0.94	0.91	0.21	0.15	0.33	%70	%89	%64	10
5	logsig	4	0.95	0.94	0.94	0.15	0.13	0.30	%83	%89	%73	16
5	logsig	5	0.84	0.95	0.79	0.31	0.14	0.43	%65	%89	%73	6
6	logsig	1	0.86	0.75	0.91	0.29	0.42	0.40	%56	%56	%55	4
6	logsig	2	0.89	0.60	0.86	0.25	0.44	0.41	%69	%56	%55	8
6	logsig	3	0.87	0.75	0.91	0.28	0.33	0.43	%57	%67	%55	5
6	logsig	4	0.90	0.65	0.91	0.24	0.50	0.39	%69	%56	%64	5
6	logsig	5	0.87	0.85	0.77	0.25	0.31	0.49	%67	%56	%64	6
4	tansig	1	0.82	0.95	0.79	0.31	0.19	0.46	%62	%78	%64	6
4	tansig	2	0.84	0.93	0.82	0.29	0.18	0.51	%63	%78	%64	5
4	tansig	3	0.88	0.95	0.87	0.24	0.20	0.41	%65	%78	%73	7
4	tansig	4	0.83	0.93	0.84	0.30	0.20	0.48	%65	%67	%73	5
4	tansig	5	0.92	0.86	0.87	0.21	0.26	0.44	%73	%67	%73	10
6	tansig	1	0.89	0.80	0.90	0.27	0.26	0.42	%59	%78	%55	6
6	tansig	2	0.93	0.91	0.76	0.18	0.34	0.45	%74	%56	%55	16
6	tansig	3	0.86	0.73	0.90	0.28	0.36	0.41	%57	%56	%55	5
6	tansig	4	0.90	0.94	0.90	0.23	0.20	0.33	%65	%56	%55	10
6	tansig	5	0.86	0.84	0.75	0.28	0.30	0.47	%62	%44	%64	5

EK 8 BO-2 Sonuçlarının Adımlar Halinde Listelenmesi

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	trfunc	HLM1	actfl	mc	lr
1	Best	0.15213	330.64	0.15213	0.15213	trainlm	9	tansig	-	-
2	Accept	0.1633	320.49	0.15213	0.15213	traingdx	3	tansig	6	3
3	Accept	0.40053	336.53	0.15213	0.15213	traingdx	4	purelin	9	2
4	Best	0.14443	321.57	0.14443	0.14443	trainlm	9	logsig	-	-
5	Best	0.12892	349.27	0.12892	0.12892	trainlm	5	logsig	-	-
6	Accept	0.19869	359.5	0.12892	0.12892	traingdx	11	tansig	8	3
7	Accept	0.17063	328.78	0.12892	0.12892	traingdx	3	tansig	6	1
8	Best	0.11592	331.83	0.11592	0.11592	traingdx	3	tansig	-	-
9	Accept	0.13837	394.97	0.11592	0.11592	traingdx	3	logsig	-	-
10	Accept	0.13831	342.53	0.11592	0.11592	trainlm	3	purelin	-	-
11	Accept	0.12595	336.35	0.11592	0.11592	traingdx	11	purelin	-	-
12	Accept	0.19771	329.29	0.11592	0.11592	traingdx	3	logsig	6	5
13	Accept	0.12321	361.74	0.11592	0.11592	traingdx	11	tansig	-	-
14	Accept	0.12437	370.68	0.11592	0.11592	traingdx	3	purelin	-	-
15	Accept	0.24667	345.54	0.11592	0.11592	traingdx	11	logsig	9	1
16	Accept	0.13976	347.63	0.11592	0.11592	trainlm	11	purelin	-	-
17	Accept	0.13097	371.98	0.11592	0.11592	traingdx	11	logsig	-	-
18	Accept	0.12988	351.99	0.11592	0.1229	traingdx	3	tansig	-	-
19	Accept	0.1304	347.22	0.11592	0.1229	trainlm	3	logsig	-	-
20	Best	0.11106	349.42	0.11106	0.11106	traingdx	7	tansig	-	-
21	Accept	0.12463	427.26	0.11106	0.11106	trainlm	3	tansig	-	-
22	Accept	0.13389	357.88	0.11106	0.11106	traingdx	7	purelin	-	-
23	Accept	0.1276	347.87	0.11106	0.11106	traingdx	6	tansig	-	-
24	Accept	0.18342	417.92	0.11106	0.11106	traingdx	3	tansig	9	5
25	Accept	0.12337	390.3	0.11106	0.11106	traingdx	9	tansig	-	-
26	Accept	0.12528	355.93	0.11106	0.11106	traingdx	7	logsig	-	-
27	Accept	0.13157	361.64	0.11106	0.11106	trainlm	7	purelin	-	-
28	Accept	0.12925	454.99	0.11106	0.11106	trainlm	5	tansig	-	-
29	Accept	0.12162	381.98	0.11106	0.11106	traingdx	9	logsig	-	-
30	Accept	0.11651	384.04	0.11106	0.11106	traingdx	8	tansig	-	-
31	Accept	0.12188	383.82	0.11106	0.11106	traingdx	5	purelin	-	-
32	Accept	0.15299	415.6	0.11106	0.11106	trainlm	11	logsig	-	-
33	Accept	0.1673	398.35	0.11106	0.11106	traingdx	7	tansig	6	5
34	Accept	0.43684	379	0.11106	0.11106	traingdx	11	purelin	6	5
35	Accept	0.16995	393.98	0.11106	0.11106	trainlm	11	tansig	-	-
36	Accept	0.13423	392.34	0.11106	0.11106	trainlm	5	purelin	-	-
37	Accept	0.21866	391.91	0.11106	0.11106	traingdx	3	tansig	9	1
38	Accept	0.26397	420.45	0.11106	0.11106	traingdx	3	logsig	9	1
39	Accept	0.16558	415.39	0.11106	0.11106	traingdx	8	logsig	6	5
40	Accept	0.12109	408.2	0.11106	0.11106	traingdx	5	logsig	-	-
41	Accept	0.18081	408.47	0.11106	0.11106	traingdx	11	logsig	6	5
42	Accept	0.13886	411.7	0.11106	0.11106	trainlm	9	purelin	-	-
43	Accept	0.22183	416.92	0.11106	0.11106	traingdx	7	tansig	9	5
44	Accept	0.16885	415.81	0.11106	0.11106	traingdx	11	tansig	6	5
45	Accept	0.1221	471.35	0.11106	0.11106	traingdx	9	purelin	-	-
46	Accept	0.17162	434.35	0.11106	0.11106	traingdx	9	tansig	6	1
47	Accept	0.1301	416.42	0.11106	0.11106	trainlm	7	logsig	-	-
48	Accept	0.16095	416.59	0.11106	0.11106	traingdx	6	logsig	6	1
49	Accept	0.14779	410.35	0.11106	0.11106	trainlm	7	tansig	-	-
50	Accept	0.1681	416.25	0.11106	0.11106	traingdx	9	logsig	6	1

EK 9 BO-3 Sonuçlarının Adımlar Halinde Listelenmesi

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	trfunc	HLN1	HLN2	actf1	actf2	mc	lr
1	Best	0.14089	164.44	0.14089	0.14089	trainscg	3	4	tansig	logsig	-	-
2	Accept	0.14108	168.06	0.14089	0.14089	trainlm	5	5	purelin	purelin	-	-
3	Best	0.12457	168.81	0.12457	0.12457	trainlm	5	4	tansig	logsig	-	-
4	Accept	0.14067	175.8	0.12457	0.12457	trainscg	3	6	tansig	purelin	-	-
5	Accept	0.14282	170.32	0.12457	0.12457	trainlm	5	7	tansig	logsig	-	-
6	Accept	0.135	166.86	0.12457	0.12457	trainlm	5	3	tansig	logsig	-	-
7	Accept	0.13377	167.61	0.12457	0.12457	trainlm	5	5	tansig	logsig	-	-
8	Accept	0.13546	169.87	0.12457	0.12457	trainlm	6	4	tansig	logsig	-	-
9	Accept	0.12707	178.26	0.12457	0.12457	trainlm	4	4	tansig	logsig	-	-
10	Accept	0.13976	176.17	0.12457	0.12457	trainlm	3	4	tansig	logsig	-	-
11	Accept	0.13347	174.33	0.12457	0.12457	trainlm	4	5	tansig	logsig	-	-
12	Accept	0.12565	167.4	0.12457	0.12457	trainlm	4	3	tansig	logsig	-	-
13	Accept	0.12463	175.18	0.12457	0.12457	trainlm	4	4	tansig	tansig	-	-
14	Accept	0.1431	173.32	0.12457	0.12457	trainlm	4	3	tansig	tansig	-	-
15	Accept	0.13464	173.44	0.12457	0.12457	trainlm	5	4	tansig	tansig	-	-
16	Accept	0.13849	174.18	0.12457	0.12457	trainlm	3	4	tansig	tansig	-	-
17	Accept	0.15915	171.15	0.12457	0.12457	trainlm	4	5	tansig	tansig	-	-
18	Accept	0.15468	170.79	0.12457	0.12457	trainlm	3	3	tansig	logsig	-	-
19	Best	0.12299	173.39	0.12299	0.12299	trainlm	4	4	logsig	tansig	-	-
20	Best	0.12079	175.41	0.12079	0.12079	trainlm	4	4	logsig	logsig	-	-
21	Accept	0.1591	169.68	0.12079	0.12079	trainscg	4	4	logsig	logsig	-	-
22	Accept	0.1404	175.99	0.12079	0.12079	trainlm	4	4	purelin	logsig	-	-
23	Accept	0.12756	179.2	0.12079	0.12079	trainlm	4	4	logsig	purelin	-	-
24	Accept	0.13634	175.31	0.12079	0.12079	trainlm	4	4	tansig	purelin	-	-
25	Accept	0.12226	180.45	0.12079	0.12079	trainlm	5	4	logsig	logsig	-	-
26	Accept	0.13995	171.54	0.12079	0.12079	trainlm	3	4	logsig	logsig	-	-
27	Accept	0.13702	173.44	0.12079	0.12079	trainlm	5	4	logsig	tansig	-	-
28	Accept	0.14029	196.41	0.12079	0.12079	trainlm	3	4	logsig	tansig	-	-
29	Accept	0.13327	185.01	0.12079	0.12079	trainlm	6	4	logsig	logsig	-	-
30	Accept	0.12266	186.02	0.12079	0.12079	trainlm	5	4	logsig	purelin	-	-
31	Accept	0.13196	194.11	0.12079	0.12079	trainlm	6	4	logsig	purelin	-	-
32	Accept	0.12239	182.78	0.12079	0.12079	trainlm	3	4	logsig	purelin	-	-
33	Accept	0.13157	176.29	0.12079	0.12079	trainlm	3	5	tansig	logsig	-	-
34	Accept	0.14267	183.74	0.12079	0.12079	trainlm	7	5	tansig	logsig	-	-
35	Accept	0.1375	179.69	0.12079	0.12079	trainlm	5	4	purelin	purelin	-	-
36	Accept	0.1364	178.2	0.12079	0.12079	trainlm	6	4	tansig	purelin	-	-
37	Accept	0.13399	180.11	0.12079	0.12079	trainlm	7	4	logsig	purelin	-	-
38	Accept	0.13547	175.37	0.12079	0.12079	trainscg	6	4	logsig	purelin	-	-
39	Accept	0.14588	177.31	0.12079	0.12079	trainscg	4	4	tansig	tansig	-	-
40	Accept	0.15803	174.43	0.12079	0.12079	trainscg	4	4	logsig	purelin	-	-
41	Accept	0.14429	183.45	0.12079	0.12079	trainlm	4	5	logsig	logsig	-	-
42	Accept	0.1413	175.21	0.12079	0.12079	trainlm	3	4	purelin	purelin	-	-
43	Best	0.11782	181.07	0.11782	0.11782	trainlm	7	4	purelin	purelin	-	-
44	Accept	0.13051	185.72	0.11782	0.11782	trainlm	6	4	purelin	purelin	-	-
45	Accept	0.129	179.24	0.11782	0.11782	trainscg	7	4	purelin	purelin	-	-
46	Best	0.11655	180.45	0.11655	0.11655	trainscg	6	4	purelin	purelin	-	-
47	Accept	0.12819	179.68	0.11655	0.11655	trainscg	5	4	purelin	purelin	-	-
48	Accept	0.12746	182.96	0.11655	0.11655	trainscg	3	4	purelin	purelin	-	-
49	Accept	0.13364	188.44	0.11655	0.11655	trainscg	4	4	purelin	purelin	-	-
50	Accept	0.37457	185.47	0.11655	0.11655	trainindx	6	4	purelin	purelin	7	2
51	Accept	0.14053	184.62	0.11655	0.11655	trainlm	3	6	logsig	purelin	-	-
52	Accept	0.14109	184.74	0.11655	0.11655	trainlm	4	3	logsig	logsig	-	-
53	Accept	0.12607	179.74	0.11655	0.11655	trainscg	6	6	purelin	purelin	-	-
54	Accept	0.12907	188.67	0.11655	0.11655	trainscg	3	6	purelin	purelin	-	-
55	Accept	0.13318	184.9	0.11655	0.11655	trainlm	7	3	purelin	purelin	-	-
56	Accept	0.1262	181.99	0.11655	0.11655	trainlm	7	6	purelin	purelin	-	-
57	Accept	0.14521	180.41	0.11655	0.11655	trainscg	7	7	logsig	purelin	-	-
58	Accept	0.13028	209.31	0.11655	0.11655	trainlm	5	7	tansig	purelin	-	-
59	Accept	0.11789	184.44	0.11655	0.11655	trainlm	3	3	logsig	purelin	-	-
60	Accept	0.13266	188.35	0.11655	0.11655	trainlm	7	7	logsig	purelin	-	-
61	Accept	0.13341	192.66	0.11655	0.11655	trainlm	4	7	logsig	tansig	-	-
62	Accept	0.13156	194.59	0.11655	0.11655	trainlm	7	7	purelin	logsig	-	-
63	Accept	0.14538	185.44	0.11655	0.11655	trainlm	3	7	purelin	logsig	-	-
64	Accept	0.13521	200.57	0.11655	0.11655	trainlm	7	4	purelin	logsig	-	-
65	Best	0.11153	184.93	0.11153	0.11153	trainscg	7	4	tansig	purelin	-	-
66	Accept	0.1279	187.33	0.11153	0.11153	trainscg	5	3	tansig	purelin	-	-
67	Accept	0.1279	192.34	0.11153	0.11153	trainscg	7	6	tansig	purelin	-	-
68	Accept	0.1243	193.77	0.11153	0.11153	trainscg	7	3	tansig	purelin	-	-
69	Accept	0.12054	192.88	0.11153	0.11153	trainscg	6	5	tansig	purelin	-	-
70	Accept	0.13025	190.34	0.11153	0.11153	trainscg	7	6	tansig	logsig	-	-

EK 10 BO-2 ve BO-3 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler
(Tek Gizli Katmanlı)

Öğrenme Algoritması	Gizli Nöron Sayısı	Aktivasyon Fonksiyonu	Deneme No	R-değeri eğitim	R-değeri doğrulama	R-değeri test	MMRE eğitim	MMRE doğrulama	MMRE test	PRED(%25) eğitim	PRED(%25) doğrulama	PRED(%25) test	En iyi adım
trainscg	3	tansig	1	0.89	0.76	0.97	0.23	0.29	0.15	%67	%67	%91	15
trainscg	3	tansig	2	0.84	0.96	0.92	0.28	0.15	0.22	%54	%67	%64	16
trainscg	3	tansig	3	0.88	0.95	0.96	0.24	0.15	0.19	%59	%78	%91	17
trainscg	3	tansig	4	0.87	0.92	0.91	0.26	0.22	0.25	%53	%78	%64	12
trainscg	3	tansig	5	0.84	0.98	0.96	0.28	0.11	0.17	%51	%89	%73	13
trainscg	3	tansig	6	0.83	0.94	0.94	0.28	0.26	0.22	%53	%56	%73	9
trainscg	3	tansig	7	0.89	0.89	0.87	0.24	0.25	0.30	%59	%56	%64	12
trainscg	3	tansig	8	0.84	0.95	0.93	0.27	0.29	0.22	%57	%33	%73	12
trainscg	3	tansig	9	0.66	0.77	0.82	0.40	0.32	0.36	%40	%33	%55	3
trainscg	3	tansig	10	0.80	0.92	0.88	0.33	0.16	0.30	%47	%89	%64	8
trainscg	4	tansig	1	0.93	0.74	0.92	0.20	0.25	0.25	%75	%44	%73	40
trainscg	4	tansig	2	0.83	0.97	0.94	0.27	0.25	0.22	%54	%56	%64	17
trainscg	4	tansig	3	0.84	0.92	0.98	0.29	0.23	0.17	%56	%67	%82	5
trainscg	4	tansig	4	0.91	0.80	0.91	0.23	0.26	0.24	%64	%56	%73	22
trainscg	4	tansig	5	0.89	0.94	0.96	0.23	0.23	0.18	%63	%67	%82	21
trainscg	4	tansig	6	0.84	0.92	0.89	0.28	0.28	0.28	%54	%44	%64	17
trainscg	4	tansig	7	0.88	0.78	0.95	0.26	0.22	0.17	%63	%56	%82	9
trainscg	4	tansig	8	0.88	0.84	0.94	0.26	0.19	0.23	%57	%56	%64	13
trainscg	4	tansig	9	0.68	0.83	0.79	0.40	0.30	0.41	%35	%56	%18	3
trainscg	4	tansig	10	0.79	0.68	0.89	0.33	0.21	0.24	%48	%67	%55	4
trainscg	5	tansig	1	0.85	0.92	0.94	0.27	0.17	0.25	%62	%67	%64	13
trainscg	5	tansig	2	0.87	0.96	0.93	0.24	0.18	0.23	%64	%67	%73	20
trainscg	5	tansig	3	0.91	0.91	0.97	0.23	0.22	0.23	%68	%67	%73	7
trainscg	5	tansig	4	0.87	0.80	0.94	0.27	0.21	0.22	%64	%56	%82	8
trainscg	5	tansig	5	0.86	0.91	0.92	0.25	0.20	0.21	%58	%67	%64	10
trainscg	5	tansig	6	0.87	0.76	0.93	0.26	0.22	0.21	%62	%56	%73	18
trainscg	5	tansig	7	0.88	0.95	0.92	0.24	0.21	0.21	%59	%67	%73	22
trainscg	5	tansig	8	0.85	0.93	0.91	0.27	0.24	0.24	%58	%44	%64	7
trainscg	5	tansig	9	0.83	0.92	0.92	0.28	0.28	0.24	%52	%44	%73	7
trainscg	5	tansig	10	0.85	0.84	0.88	0.27	0.28	0.27	%53	%44	%73	11
trainscg	6	tansig	1	0.90	0.90	0.94	0.23	0.26	0.20	%64	%56	%64	24
trainscg	6	tansig	2	0.85	0.96	0.96	0.28	0.18	0.22	%56	%67	%64	11
trainscg	6	tansig	3	0.90	0.48	0.83	0.23	0.23	0.27	%62	%78	%64	15
trainscg	6	tansig	4	0.88	0.97	0.91	0.24	0.16	0.27	%62	%78	%64	27
trainscg	6	tansig	5	0.90	0.95	0.90	0.22	0.21	0.27	%68	%78	%64	17
trainscg	6	tansig	6	0.90	0.97	0.96	0.23	0.12	0.19	%60	%89	%82	21
trainscg	6	tansig	7	0.90	0.98	0.93	0.23	0.10	0.27	%64	%89	%73	15
trainscg	6	tansig	8	0.89	0.96	0.88	0.24	0.17	0.27	%63	%78	%73	20
trainscg	6	tansig	9	0.90	0.44	0.86	0.24	0.28	0.29	%63	%67	%64	13
trainscg	6	tansig	10	0.87	0.82	0.90	0.26	0.39	0.28	%60	%33	%45	12
trainscg	7	tansig	1	0.88	0.98	0.95	0.24	0.13	0.17	%65	%100	%73	16
trainscg	7	tansig	2	0.92	0.97	0.98	0.19	0.16	0.14	%73	%78	%91	34
trainscg	7	tansig	3	0.91	0.97	0.94	0.21	0.12	0.20	%70	%100	%73	27
trainscg	7	tansig	4	0.93	0.98	0.91	0.19	0.10	0.23	%74	%100	%82	28
trainscg	7	tansig	5	0.92	0.92	0.97	0.20	0.24	0.15	%72	%78	%82	26
trainscg	7	tansig	6	0.93	0.95	0.95	0.20	0.12	0.17	%72	%89	%82	33
trainscg	7	tansig	7	0.92	0.95	0.95	0.21	0.20	0.19	%72	%89	%82	20
trainscg	7	tansig	8	0.93	0.95	0.98	0.18	0.16	0.15	%74	%78	%91	28
trainscg	7	tansig	9	0.95	0.90	0.94	0.16	0.21	0.18	%81	%78	%73	37
trainscg	7	tansig	10	0.93	0.97	0.96	0.19	0.14	0.18	%74	%78	%82	32
trainscg	8	tansig	1	0.92	0.95	0.97	0.20	0.17	0.13	%77	%78	%82	33
trainscg	8	tansig	2	0.90	0.89	0.96	0.24	0.25	0.20	%58	%56	%82	12
trainscg	8	tansig	3	0.87	0.96	0.96	0.27	0.15	0.20	%52	%89	%73	12
trainscg	8	tansig	4	0.90	0.83	0.95	0.23	0.23	0.21	%65	%67	%73	15
trainscg	8	tansig	5	0.85	0.82	0.95	0.29	0.22	0.25	%56	%67	%64	9
trainscg	8	tansig	6	0.91	0.90	0.92	0.22	0.20	0.24	%63	%67	%73	15
trainscg	8	tansig	7	0.90	0.98	0.93	0.21	0.14	0.20	%67	%78	%73	13
trainscg	8	tansig	8	0.91	0.77	0.95	0.21	0.29	0.21	%70	%56	%64	22
trainscg	8	tansig	9	0.90	0.93	0.90	0.24	0.20	0.27	%62	%67	%73	22
trainscg	8	tansig	10	0.88	0.96	0.87	0.23	0.20	0.28	%63	%67	%64	25

EK 10 - devam BO-2 ve BO-3 Sonucu Seçilen Ağ Yapıları ile Yapılan Denemeler
(İki Gizli Katmanlı)

Öğrenme Algoritması	Gizli Nöron Sayısı	Aktivasyon Fonksiyonu	Deneme No	R-değeri eğitim	R-değeri doğrulama	R-değeri test	MMRE eğitim	MMRE doğrulama	MMRE test	PRED(%25) eğitim	PRED(%25) doğrulama	PRED(%25) test	En iyi adım
trainscg	6-4	purelin-purelin	1	0.86	0.81	0.90	0.26	0.25	0.27	%62	%67	%64	24
trainscg	6-4	purelin-purelin	2	0.84	0.96	0.90	0.27	0.26	0.26	%64	%78	%64	32
trainscg	6-4	purelin-purelin	3	0.85	0.88	0.92	0.27	0.28	0.26	%60	%56	%73	13
trainscg	6-4	purelin-purelin	4	0.86	0.87	0.90	0.27	0.18	0.28	%59	%78	%64	22
trainscg	6-4	purelin-purelin	5	0.85	0.92	0.90	0.27	0.19	0.25	%60	%67	%64	21
trainscg	6-4	purelin-purelin	6	0.84	0.95	0.91	0.27	0.16	0.25	%59	%78	%73	12
trainscg	6-4	purelin-purelin	7	0.83	0.80	0.93	0.28	0.29	0.24	%56	%78	%64	12
trainscg	6-4	purelin-purelin	8	0.84	0.96	0.93	0.28	0.11	0.22	%57	%89	%64	26
trainscg	6-4	purelin-purelin	9	0.85	0.86	0.91	0.28	0.18	0.28	%54	%89	%45	18
trainscg	6-4	purelin-purelin	10	0.84	0.53	0.90	0.28	0.38	0.31	%52	%22	%55	17
trainscg	7-4	tansig-purelin	1	0.92	0.99	0.95	0.21	0.09	0.21	%72	%100	%82	28
trainscg	7-4	tansig-purelin	2	0.85	0.94	0.99	0.26	0.22	0.14	%64	%56	%73	16
trainscg	7-4	tansig-purelin	3	0.89	0.92	0.96	0.24	0.23	0.21	%67	%56	%73	45
trainscg	7-4	tansig-purelin	4	0.88	0.96	0.94	0.25	0.16	0.24	%59	%78	%73	12
trainscg	7-4	tansig-purelin	5	0.89	0.92	0.91	0.24	0.26	0.27	%65	%56	%73	13
trainscg	7-4	tansig-purelin	6	0.86	0.90	0.91	0.27	0.21	0.25	%58	%56	%64	22
trainscg	7-4	tansig-purelin	7	0.87	0.98	0.94	0.25	0.11	0.24	%58	%100	%82	17
trainscg	7-4	tansig-purelin	8	0.89	0.72	0.93	0.24	0.20	0.24	%59	%78	%73	26
trainscg	7-4	tansig-purelin	9	0.87	0.82	0.93	0.25	0.24	0.24	%68	%56	%73	21
trainscg	7-4	tansig-purelin	10	0.89	0.96	0.94	0.24	0.16	0.21	%60	%78	%82	16
trainlm	7-4	purelin-purelin	1	0.85	0.97	0.89	0.27	0.15	0.27	%60	%100	%64	3
trainlm	7-4	purelin-purelin	2	0.85	0.92	0.90	0.27	0.23	0.28	%60	%67	%64	3
trainlm	7-4	purelin-purelin	3	0.84	0.97	0.91	0.27	0.24	0.26	%63	%67	%64	3
trainlm	7-4	purelin-purelin	4	0.86	0.94	0.88	0.26	0.20	0.27	%62	%78	%64	4
trainlm	7-4	purelin-purelin	5	0.85	0.95	0.90	0.27	0.16	0.27	%60	%78	%64	6
trainlm	7-4	purelin-purelin	6	0.86	0.83	0.88	0.26	0.27	0.27	%65	%67	%64	2
trainlm	7-4	purelin-purelin	7	0.89	0.65	0.90	0.24	0.29	0.25	%64	%56	%64	4
trainlm	7-4	purelin-purelin	8	0.86	0.90	0.90	0.26	0.24	0.27	%64	%56	%55	3
trainlm	7-4	purelin-purelin	9	0.84	0.93	0.88	0.29	0.21	0.29	%58	%67	%55	3
trainlm	7-4	purelin-purelin	10	0.84	0.85	0.94	0.25	0.43	0.19	%64	%33	%73	2

ÖZGEÇMİŞ

Adı Soyadı : Burcu Şengüneş
Doğum Yeri ve Tarihi : Bursa, 1988
Yabancı Dil : İngilizce

Eğitim Durumu

Lise : Bursa Şükrü Şankaya Anadolu Lisesi
Lisans : Endüstri Mühendisliği / Başkent Üniversitesi
Yüksek Lisans : Endüstri Mühendisliği / Eskişehir Osmangazi Üniversitesi

Çalıştığı Kurum/Kurumlar :

2022- - Scantinel Photonics
2019-2022 Ficontec Service GmbH
2016-2019 Ermaksan Optoelektronik Ar-Ge Merkezi
2012-2014 Atılım Üniversitesi

İletişim (e-posta) : burcusenguenes@gmail.com

Yayımları :

Şengüneş, B., & Öztürk, N. (2023). An artificial neural network model for project effort estimation. *Systems* 2023, 11, 91. <https://doi.org/10.3390/systems11020091>

Kaya, B., & Öztürk, N. (2017). An artificial neural network approach for the logistics center location selection. *Journal of Management Marketing and Logistics*, 4(2), 107–115.

Kaya, B., & Öztürk, N. (2017). “Lojistik Merkez Yer Seçimi için bir Yapay Sinir Ağı Yaklaşımı”, 6. Ulusal Lojistik ve Tedarik Zinciri Kongresi, ULZK 2017, 17-19 Mayıs, Antalya, 2017.

Kaya, B., & Türen, U. (2017). Kurumsal kaynak planlaması projelerinin nihai başarı faktörlerinin firma performansı üzerindeki etkileri. *Gümüşhane Üniversitesi Sosyal Bilimler Elektronik Dergisi*, 8(19). <https://doi.org/10.17823/GUSB.381>

Türen, U., Kaya, B., Akkocaoğlu, H. (2013). Basit reaksiyon zamanını etkileyen faktörler üzerine bir deney, *Journal of Human Sciences*, 10(2), 637-657.