YAPILARIN MEKANİK DAVRANIŞININ YAPAY SİNİR AĞLARI İLE MODELLENMESİ VE OPTİMİZASYONU

Burak AYDOĞDU

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPILARIN MEKANİK DAVRANIŞININ YAPAY SİNİR AĞLARI İLE MODELLENMESİ VE OPTİMİZASYONU

Burak AYDOĞDU
0000-0001-9580-878X

Prof. Dr. Necmettin KAYA
0000-0002-8297-0777
(Danışman)

YÜKSEK LİSANS
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2023
Her Hakkı Saklıdır

ÖZET

Yüksek Lisans Tezi

YAPILARIN MEKANİK DAVRANIŞININ YAPAY SİNİR AĞLARI İLE MODELLENMESİ VE OPTİMİZASYONU

Burak AYDOĞDU

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Makine Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Necmettin KAYA

Kullanılan malzemeye ve gerçekleştirilen analiz türüne göre ürün geliştirme çalışmalarında kullanılan sonlu elemanlar yöntemi uzun sürelere ihtiyaç duymaktadır. Yüksek iterasyon sayıları gerektirebilen sonlu elemanlar yöntemi optimizasyon çalışmaları ile birleştirildiğinde gerekli süreler daha da artmaktadır. Özellikle farklı çözümleri inceleyen popülasyon tabanlı evrimsel algoritmaların yüksek işlem gücü gereksinimleri ürün geliştirme çalışmalarını maliyetli hale getirmektedir.

Yapay zekâ algoritmalarından birisi olan, yüksek başarı oranlarına ve hızlı sonuçlar üretme kapasitesine sahip makine öğrenmesi algoritmaları, farklı parametreler ile sonlu elemanlar modelinin tekrar tekrar kurulup çözülmesine kıyasla optimizasyon yöntemleri için büyük avantajlar sunmaktadır. Bu tez çalışmasında, yapıların mekanik davranışını modellenmek için sonlu elemanlar analizine dayanan bir makine öğrenmesi modeli geliştirilmiş ve model, yapıların optimizasyonu için ihtiyaç duyulan amaç ve kısıt fonksiyonlarını belirlemede kullanılmıştır. Makine öğrenmesi modeli için veri seti eldesi gerçekleştirilmiş, model eğitilmiş ve doğrulaması yapılmıştır. Eğitilen model, doğrusal ve doğrusal olmayan yapıların optimizasyon çalışmalarında genetik ve diferansiyel gelişim algoritmaları ile kullanılarak sunulan metodolojinin ürün geliştirme çalışmalarına olan katkıları incelenmiştir.

Eğitilen makine öğrenmesi modeli hem doğrusal hem de doğrusal olmayan yapıların davranışını %98-99 civarlarındaki yüksek bir başarı oranı ile modelleyebilmektedir. Sunulan yöntem ile gerçekleştirilen optimizasyon çalışmaları optimum sonuçlarına saniyeler içerisinde ulaşmaktadır. Sunulan yöntem deneme-yanılma ve tecrübeye dayanarak tasarlanan, yüksek işlem gücü ve uzun süreler gerektiren yapıların tasarımı için yüksek başarı oranına sahip alternatif bir çözüm yolu olmaktadır.

Anahtar Kelimeler: Sonlu elemanlar analizi, makine öğrenmesi, yapay sinir ağı, optimizasyon, genetik algoritma, diferansiyel gelişim algoritması
2023, xi + 84 sayfa.

ABSTRACT

MSc Thesis

MODELLING AND OPTIMIZATION OF MECHANICAL BEHAVIOR OF STRUCTURES WITH ARTIFICIAL NEURAL NETWORKS

Burak AYDOĞDU

Bursa Uludag University
Graduate School of Natural and Applied Sciences
Department of Mechanical Engineering

Supervisor: Prof. Dr. Necmettin KAYA

Depending on the material used and the type of analysis performed, the implementation of the finite element method used in product development studies requires long periods of time. When the finite element method, which may require high iteration numbers, is combined with optimization studies, the required time increases even more. Especially the high processing power requirements of population-based evolutionary algorithms that examine different solutions make product development studies costly.

Machine learning algorithms, which are one of the artificial intelligence algorithms, have high success rates and the capacity to produce fast results, offer excellent advantages in optimization methods compared to repeatedly building and solving a finite element model with different parameters. In this thesis, a machine learning model based on the finite element analysis was developed to model the mechanical behavior of structures, and the model was used to determine the required objective and constraint functions for the optimization of structures. Data acquisition for the machine learning model was performed, the model was trained and validated. The trained model was used with a genetic and differential evolution algorithm for the optimization studies of linear and nonlinear structures, and the contributions of the presented methodology to product development studies were examined.

The trained machine learning model can model the behavior of both linear and nonlinear structures with a high success rate of around 98-99%. Optimization studies which carried out with the presented method reach optimal results within seconds. The presented method is an alternative solution with a high success rate for designing structures that are designed based on trial-and-error and experience, requiring high processing power and long periods.

Keywords: Finite element analysis, machine learning, artificial neural network, optimization, genetic algorithm, differential evolution algorithm
2023, xi + 84 pages.

ÖNSÖZ ve/veya TEŞEKKÜR

Tez çalışmamız için gereken tüm akademik bilgi ve tecrübelerini samimi bir duygu ile aktaran, karşılaştığım problemler ile yakından ilgilenen ve rehber olan değerli ve saygı değer hocam Prof. Dr. Necmettin KAYA'ya en derin saygılarımı ve sonsuz teşekkürlerimi sunarım.

Hayatım boyunca maddi ve manevi desteklerini üzerimden hiçbir zaman esirgemeyen aileme, akademik ve kültürel gelişimimde üzerimde emeği olan tüm öğretmenlerime ve üniversite hocalarıma teşekkürlerimi sunarım.

Burak AYDOĞDU
08/05/2023

İÇİNDEKİLER

Sayfa

ÖZET iii
ABSTRACT iv
ÖNSÖZ ve/veya TEŞEKKÜR v
İÇİNDEKİLER vi
SİMGELER ve KISALTMALAR DİZİNİ viii
ŞEKİLLER DİZİNİ ix
ÇİZELGELER DİZİNİ xi
1. GİRİŞ 1
1.1. Tezin Amacı 2
1.2. Tezin Katkısı ve Yenilikleri 2
2. KAYNAK ARAŞTIRMASI 3
2.1. Şekil Optimizasyonu Çalışmaları 3
2.2. Evrimsel Algoritmalar Kullanılan Çalışmalar 5
2.3. Makine Öğrenmesi İçeren Çalışmalar 7
2.4. Bütünleşik Yaklaşım İçeren Çalışmalar 9
3. MATERYAL VE YÖNTEM 12
3.1. Optimizasyon 12
3.1.1. Tasarım Değişkenleri 13
3.1.2. Amaç Fonksiyonları 13
3.1.3. Kısıt Fonksiyonları 13
3.1.4. Standart Bir Optimizasyon Probleminin Tanımlanması 14
3.1.5. Çok Amaçlı Optimizasyon 15
3.2. Evrimsel Algoritmalar 17
3.2.1. Çalışma Prensibi 19
3.2.1.1. Üreme (Initialization) 19
3.2.1.2. Çaprazlama (Crossover) 20
3.2.1.3. Mutasyon (Mutation) 20
3.2.1.4. Seçim (Selection) 21
3.2.2. Evrimsel Algoritma Türleri 21
3.2.2.1. Genetik Algoritma 22
3.2.2.2. Diferensiyel Gelişim Algoritması 22
3.2.3. Performansı Etkileyen Faktörler 23
3.3. Makine Öğrenmesi 23
3.3.1. Çalışma Prensibi 25
3.3.1.1. Problemin Tanımlanması 25
3.3.1.2. Veri Setinin Hazırlanması 26
3.3.1.3. Modelin Eğitilmesi 27
3.3.1.4. Modelin Değerlendirilmesi 28
3.3.2. Makine Öğrenmesinin Üç Dalı 30
3.3.2.1. Denetimli Öğrenme 30
3.3.2.2. Denetimsiz Öğrenme 31
3.3.2.3. Takviyeli Öğrenme 33
3.3.3. Aşırı Öğrenme ve Öğrenememe 33
3.3.4. Derin Öğrenme 36
3.3.4.1. Çok Katmanlı Perceptron 38
3.3.4.2. Kayıp ve Optimizasyon Fonksiyonları 39

3.3.4.3. Geri Besleme 40
3.4. Yöntem 41
3.4.1. Geliştirilen Yöntem 42
3.4.2. Tasarımda Yapay Zekâ Uygulamasının Örnek Çalışmaları 43
3.4.2.1. Disk Yay Optimizasyonu 44
3.4.2.2. Kauçuk Burç Optimizasyonu 49
3.4.2.3. Elektrikli Araç Batarya Taşıyıcısı Optimizasyonu 55
4. BULGULAR VE TARTIŞMA 62
4.1. Disk Yay 62
4.1.1. Disk Yayların YSA ile Modellenmesi 62
4.1.2. Disk Yay Rijitlik Optimizasyonu 64
4.2. Kauçuk Burç 67
4.2.1. Kauçuk Burcun YSA ile Modellenmesi 67
4.2.2. Kauçuk Burcun Rijitlik Optimizasyonu 69
4.3. Elektrikli Araç Batarya Taşıyıcısı 71
4.3.1. Batarya Taşıyıcısının YSA ile Modellenmesi 71
4.3.2. Batarya Taşıyıcısı Kütle Minimizasyonu 75
5. SONUÇLAR 77
KAYNAKLAR 78
ÖZGEÇMİŞ 84

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	**Açıklama**
f	Amaç fonksiyonu
h	Eşitlik kısıt fonksiyonu
g	Eşitsizlik kısıt fonksiyonu
n	Tasarım değişkeni sayısı
P	Eşitlik kısıt fonksiyonu sayısı
R^2	Belirlilik katsayısı
ω	Sinir ağının ağırlıklandırmaları
X	Sinir ağı girdileri
x	Tasarım değişkeni ya da özelliğin bir veri için gerçek değeri
y_i	Gerçek değerler
$\bar{y}$	Gerçek değerlerin ortalaması
$\hat{y}_i$	Makine öğrenmesinin tahmin ettiği değerler
z	x'in ölçeklendirilmiş değeri
σ	Belirtilen özelliğin standart sapması
μ	Belirtilen özelliğe ait ortalama değer

Kısaltmalar	**Açıklama**
BT	Benzetimli Tavlama
BTO	Biyocoğrafya Tabanlı Optimizasyon
CAD	Computer Aided Design
CNN	Convolutional Neural Network
CoDE	Composite Differential Evolution
CSV	Comma Separated Value
DE	Differential Evolution
DGA	Diferensiyel Gelişim Algoritması
EA	Evrimsel Algoritma
EP	Evrimsel Programlama
FADE	Fuzzy Adaptive Differential Evolution
GA	Genetik Algoritma
GDE3	Generalized Differential Evolution
JADE	Adaptive Differential Evolution with Optional External Archive
LDA	Linear Discriminant Analysis
MAE	Mean Absolute Error
MLP	Multi-layer Perceptron
MSE	Mean Square Error
NDE	Neighborhood Based Differential Evolution
NSGA-II	Nondominated Sorting Genetic Algorithm 2
ODE	Opposition Based Differential Evolution
PCA	Principle Component Analysis
PSO	Parçacık Sürü Optimizasyonu
ReLU	Rectified Linear Unit
SaDE	Self-Adaptive Differential Evolution
SEY	Sonlu Elemanlar Yöntemi
YSA	Yapay Sinir Ağı

ŞEKİLLER DİZİNİ

Sayfa

Şekil 1. Optimum tasarım süreci (Yıldız, 2006). 12
Şekil 2. Örnek bir Pareto diyagramı 16
Şekil 3. Bir evrimsel algoritmanın genel akış diyagramı 18
Şekil 4. Mutasyon işlemi 20
Şekil 5. Klasik programlama ve makine öğrenmesi (Chollet, 2018). 24
Şekil 6. OneHotEncoder örneği 26
Şekil 7. Şematik olarak K-Fold gösterimi 27
Şekil 8. Örnek bir karmaşıklık matrisi 28
Şekil 9. Spam filtreleme örneği (Géron, 2022). 30
Şekil 10. Bir regresyon örneği (Géron, 2022). 30
Şekil 11. Anomalilik tespiti (Géron, 2022). 31
Şekil 12. Kümeleme algoritmaları (Scikit-Learn, 2023). 32
Şekil 13. Örnek bir takviyeli öğrenme 33
Şekil 14. Öğrenememe ve aşırı öğrenme (Scikit-Learn, 2023). 34
Şekil 15. Bir dropout örneği (Chollet, 2018). 35
Şekil 16. Yapay zekâ, makine öğrenmesi ve derin öğrenme (Chollet, 2018). 36
Şekil 17. Bir perceptron örneği 37
Şekil 18. Katman, kayıp ve optimizasyon fonksiyonları (Chollet, 2018). 38
Şekil 19. Çok katmanlı bir perceptron 38
Şekil 20. Bazı aktivasyon fonksiyonları ve türevleri (Géron, 2022) 41
Şekil 21. Önerilen optimizasyon algoritmasının akış diyagramı 43
Şekil 22. Disk yay tasarım parametreleri (Kaya & Öztürk, 2002). 44
Şekil 23. Disk yay sınır şartları 45
Şekil 24. Ağdan bağımsızlık çalışması 46
Şekil 25. Yay rijitlikleri 46
Şekil 26. Disk yay için oluşturulan sinir ağı yapısı 48
Şekil 27. Test verileri için Ogden N=3 malzeme modeli 50
Şekil 28. Kauçuk burç tasarım parametreleri ve sınır şartları 51
Şekil 29. Kauçuk burç sınır şartları 51
Şekil 30. Kauçuk burç mesh yapısı 52
Şekil 31. Kauçuk burç ağdan bağımsızlık çalışması 52
Şekil 32. Kauçuk burç Latin Hypercube Örneklemesi 53
Şekil 33. Çalışma için kurulan YSA modeli 54
Şekil 34. Hedef rijitlik eğrileri 55
Şekil 35. Batarya taşıyıcısı katı modeli 56
Şekil 36. Orta yüzeyler 56
Şekil 37. Taşıyıcının sonlu elemanlar modeli 57
Şekil 38. Eğilme sınır şartları 57
Şekil 39. Burulma sınır şartları 58
Şekil 40. Batarya taşıyıcısı tasarım prametreleri 59
Şekil 41. Batarya taşıyıcısı için kurulan sinir ağı yapısı 60
Şekil 42. 1500 epoch için test ve tahmin değerlerinin karşılaştırılması 63
Şekil 43. Yay 6 için Pareto diyagramı 66
Şekil 44. Yay 6 için tasarım uzayı 66
Şekil 45. 25 mm deplasman için test ve tahmin değerlerinin karşılaştırılması 68
Şekil 46. Sinir ağı ile SEA sonuçlarının karşılaştırılması 69

Şekil 47. Birinci vaka çalışmasının optimizasyon sonucu 69
Şekil 48. İkinci optimizasyon çalışması sonucu 70
Şekil 49. Üçüncü optimizasyon çalışması sonucu 71
Şekil 50. Kütle için test ve tahmin değerlerinin karşılaştırması..... 73
Şekil 51. Eğilme deformasyonu için test ve tahmin değerlerinin karşılaştırması ... 73
Şekil 52. Eğilme gerilmesi için test ve tahmin değerlerinin karşılaştırması 74
Şekil 53. Burulma açısı için test ve tahmin değerlerinin karşılaştırması 74
Şekil 54. Burulma gerilmesi için test ve tahmin değerlerinin karşılaştırması..... 75

ÇİZELGELER DİZİNİ

		Sayfa
Çizelge 1.	Farklı DGA varyantları (Bilal vd., 2020).	23
Çizelge 2.	Alüminyum alaşımı mekanik özellikleri	58
Çizelge 3.	R2 belirlilik katsayısı	62
Çizelge 4.	Ortalama mutlak hata (MAE)	62
Çizelge 5.	Ortalama kareler hatası (MSE)	63
Çizelge 6.	Sinir ağı eğitimi için gerekli olan zaman	63
Çizelge 7.	Tek amaçlı optimizasyon sonuçları	64
Çizelge 8.	Çok amaçlı optimizasyon sonuçları	65
Çizelge 9.	Yay 6 için elde edilen rijitliklerden bazıları	67
Çizelge 10.	Kauçuk burç için sinir ağı başarı metrikleri	68
Çizelge 11.	Sinir ağı eğitimi için gerekli süre	68
Çizelge 12.	R2 belirlilik katsayısı	72
Çizelge 13.	Ortalama mutlak hata	72
Çizelge 14.	Ortalama kareler hatası	72
Çizelge 15.	İlk tasarım ile optimum tasarımın karşılaştırması	76
Çizelge 16.	Optimum tasarım rijitlikleri	76

1. GİRİŞ

Sonlu elemanlar yöntemi (SEY), mühendislikte ve matematiksel modellerde sıklıkla kullanılan bir sayısal analiz yöntemidir. Temel prensibi ürünün sonlu sayıda elemandan oluşan modelinin çıkartılarak sayısal işlemlerin bu model üzerinden gerçekleştirilmesi olan sonlu elemanlar yönteminde bu elemanların sayısına bağlı olarak çözüm hassasiyeti ile işlem gücü gerekliliği de artmaktadır. Parçaların veya montajın karmaşıklığına, yapılacak analizin türüne, seçilen sonlu eleman sayısına ve kullanılan bilgisayarın işlem gücüne bağlı olarak sonlu elemanlar analizinin sonuçlanması saatler hatta günler sürebilir.

En uygun çözüme deneme yanılma yerine bilimsel bir yaklaşım ile ulaşılmasını sağlayan optimizasyon yöntemleri gradyan tabanlı ve gradyan tabanlı olmayan algoritmalar olarak ikiye ayrılabilir. En büyük dezavantajının lokal optimum noktaya takılması olan gradyan tabanlı algoritmalara kıyasla yüksek işlem gücü ve maliyetler gerektirmesine rağmen gradyan tabanlı olmayan algoritmalar global optimum noktanın bulunmasında büyük avantaja sahiptirler.

Optimizasyon yöntemleri için gerekli değerlerin elde edilmesinde modelin tekrardan kurulması ve çözülmesini gerektiren SEY kullanımı ile optimizasyon algoritmasının yakınsaması için gereken süreler daha da artmaktadır. Özellikle doğrusal olmayan bir malzeme ve gerçekleştirilecek analizin türüne bağlı olarak kurulan sonlu eleman modelinin çözümü için gereken yüksek iterasyon sayıları ürün geliştirme çalışmalarını maliyetli hale getirmektedir.

Bu tez kapsamında, kısıt içeren tek ve çok amaçlı mühendislik problemlerinin çözülmesi ve elemanların optimum tasarlanabilmesi için sonlu elemanlar yöntemine dayanan bir makine öğrenmesi modeli ile evrimsel algoritmalar entegre edilerek bir optimizasyon metodu geliştirilmiştir. Geliştirilen metot ile doğrusal ve doğrual olmayan davranış sergileyen malzemeye sahip tek ve çok amaçlı optimizasyon problemleri çözülmüştür. Özellikle analitik olarak hesaplanması mümkün olmayan ve nümerik olarak uzun süreler gerektiren problemlerde sunulan yöntemin ürün geliştirme çalışmalarına sağladığı katkı yönteminin gerekliliğini ortaya koymaktadır.

1.1. Tezin Amacı

Analizlerin bir kez gerçekleştirilerek elde edilen veri seti ile eğitilen makine öğrenmesi algoritmalarının istenilen çıktıları yüksek başarı ile ortaya koyabilmelerinin yanı sıra, sonuçlara ucuz ve hızlı bir şekilde ulaşmamıza da imkân sağlamaktadırlar. Bu avantajlara sahip makine öğrenmesi, karmaşık ve doğrusal olmayan mühendislik problemlerinin çözümünde alternatif bir çözüm olarak karşımıza çıkmaktadır.

Ürün geliştirme maliyetlerinin yüksek olması nedeniyle rekabet etmede zorluklarla karşılaşılması, işletmeleri kısa sürede düşük maliyetli ürünler geliştirmeye yöneltmekte ve optimizasyon yöntemlerini zaruri hale getirmektedir. Bu doğrultuda gerçekleştirilen tez çalışmasında, mühendislik problemlerinin çözümünde mevcut çözümlere kıyasla global optimuma hızlı ve düşük maliyetler ile ulaşılması için makine öğrenmesine dayanan yeni bir hibrit yöntemin geliştirilmesi ve geliştirilen metodun örnek problemlere uygulanarak kullanılabilirliğinin gösterilmesi amaçlanmaktadır.

1.2. Tezin Katkısı ve Yenilikleri

Gerçekleştirilen tez çalışması, literatürde ürün geliştirme aşamalarında karşılaşılan problemlerin (özellikle halen deneysel olarak tasarlanan kauçuk gibi doğrusal davranış sergilemeyen malzemeler ve yüksek iterasyon sayıları gerektiren çarpışma analizleri) çözümünde global optimuma hızlı ve yüksek doğruluk ile ulaşılmasında etkin bir çözüm olarak kullanılabileceğini göstermiştir.

Sunulan yöntem ile gerçekleştirilen optimizasyon çalışmasında yüksek işlem gücü gerektiren, popülasyon tabanlı evrimsel algoritmaların kullanılmasına rağmen optimum sonuca saniyeler içerisinde ulaşılabilmekte ve ürün geliştirme çalışmalarına harcanan maliyetler azaltılabilmektedir.

2. KAYNAK ARAŞTIRMASI

Bu bölümde, ortaya konulan optimizasyon yöntemini oluşturan evrimsel algoritmalar ve makine öğrenmesi ile ilgili literatürde gerçekleştirilen çalışmalar incelenmiştir.

2.1. Şekil Optimizasyonu Çalışmaları

Kim & Kim (1997), kauçuk motor takozları için şekil optimizasyonu gerçekleştirmişlerdir. Gerçekleştirdikleri parametrik şekil optimizasyonu motor takozlarının, arzu edilen rijitliğe ulaşması adına, optimum tasarlanmasında başarılı olmuştur.

Woon vd. (2001), bir genetik algoritma kullanarak şekil optimizasyonu çalışması gerçekleştirmişlerdir. Somun anahtarı ve flanş için gerçekleştirdikleri şekil optimizasyonu neticesinde gerçekleştirilen çalışmanın, minimal kütle arttırımları ile optimum sonuca ulaşmada başarılı olduğu görülmüştür.

Yıldız vd. (2003), mevcut topoloji ve şekil optimizasyonlarının eksiklerini gidermek için entegre bir prosedür geliştirmeyi amaçlamışlardır. Üç aşamadan oluşan yöntemin (topoloji optimizasyonu, topoloji optimizasyon sonuçlarının yorumlanması ve şekil optimizasyonu) iki örnek ile doğrulaması gerçekleştirilmiş ve geliştirilen yöntemin topoloji ve şekil optimizasyonu için etkili bir yöntem olduğu görülmüştür.

Montrone vd. (2004), gerçekleştirdikleri çalışmada türbin kanatlarının verimliliğini optimize etmek için gradyan tabanlı bir algoritma kullanarak şekil optimizasyonu gerçekleştirmişlerdir. Optimizasyon neticesinde ulaşılan optimum kanat profilindeki basınç düşüşünün ilk tasarıma göre uniform olduğu görülmüştür.

Zhou vd. (2016), 3D yazdırılabilir nesnelerin dayanımını arttırmak için bir şekil optimizasyonu yöntemi sunmuşlardır. 7 farklı model ile gerçekleştirdikleri çalışmada fiziksel doğrulama testlerini gerçekleştirmişler ve yöntemin avantajlarının yanı sıra yavaş yakınsaması gibi eksiklerini de ortaya koymuşlardır.

Hilbert vd. (2006), ısı eşanjörlerinin çok amaçlı şekil optimizasyonunu gerçekleştirmek için genetik algoritmaya dayanan bir metodoloji sunmuşlardır. Basınç kaybı ile giriş ve çıkış sıcaklıkları arasındaki farklar için Pareto çözümleri incelenmiş ve altı farklı optimum tasarım elde edilmiştir.

Xudong vd. (2009), rüzgâr türbininin enerji maliyetlerini minimize etmeyi amaçlayan bir tasarım aracı sunmuşlardır. Farklı boyutlardaki üç rüzgâr türbini için kanat şekil parametrelerini kullanarak gerçekleştirdikleri optimizasyon çalışması sonucunda sunulan optimizasyon modelinin enerji üretim maliyetlerini azaltmada başarılı olduğu görülmüştür.

Park vd. (2012), otomotiv soğutma fanlarında kullanılan kauçuk izolasyonu için şekil optimizasyonu gerçekleştirmişlerdir. Titreşim izolasyonu ve yorulma ömrünü maksimize etmeyi amaçlamışlardır. 9 farklı tasarım parametresi ile gerçekleştirdikleri çalışmada S tipindeki kauçuk burç için elde ettikleri parametre değerlerinin ilk tasarıma göre yaklaşık iki kat daha fazla yorulma ömrü sağladığı görülmektedir.

Mankovits vd. (2014), arzu edilen rijitliğe sahip olması beklenen kauçuk parçalar için sonlu elemanlar ve destek vektör regresyon modellerine dayanan bir şekil optimizasyonu sunmuşlardır. Gerçekleştirilen iki optimizasyon çalışması neticesinde sunulan şekil optimizasyonunun optimum tasarım parametrelerini belirlemede doğru ve hızlı olduğu görülmüştür.

Yan vd. (2019), makine öğrenmesine dayanan bir şekil optimizasyonu metodu sunmuşlar ve füzelerin seyir halindeki kaldırma-sürükleme oranı katsayısını maksimize edecek şekil parametrelerini bulmayı hedeflemişlerdir. Sunulan yöntem karşılaştırılan diğer evrimsel algoritmalara kıyasla hem daha hızlı hem de ulaşılan optimum değerin daha iyi olduğu görülmüştür.

2.2. Evrimsel Algoritmalar Kullanılan Çalışmalar

Annicchiarico & Cerrolaza (2001), gerçekleştirdikleri çalışmada Genetik Algoritma (GA) kullanımının geleneksel optimizasyon algoritmalarına kıyasla büyük avantajlar getirdiğini ve performans arttırımı sağladığını belirtmişlerdir.

Deb vd. (2002), çok amaçlı bir evrimsel algoritma olan NSGA-II (Nondominated Sorting Genetic Algorithm) algoritmasını geliştirmişlerdir. Literatürde yer alan dokuz farklı test problemi ile değerlendirilen algoritmanın çalışmada değerlendirilen diğer iki elitist algoritmaya karşı daha iyi sonuçlara ulaştığı ve yakınsadığı görülmüştür.

Yıldız vd. (2003), çok amaçlı şekil optimizasyonu problemlerinin çözümü için genetik algoritma kullanarak bir hibrit çözüm yöntemi geliştirmişlerdir. Sunulan hibrit algoritmanın hibrit olmayan çok amaçlı genetik algoritmalara kıyasla daha iyi sonuçlar sağladığı gözlemlenmiştir.

Vesterstram & Thomsen (2004), Diferensiyel Gelişim Algoritması (DGA), Parçacık Sürü Optimizasyonu (PSO) ve Evrimsel Algoritmaları (EA) sayısal değerlendirme problemleri kullanarak karşılaştırmışlardır. 34 farklı sayısal problem ile değerlendirilen algoritmalar içerisinden DGA'nın en iyi performansı sergilediği belirtilmiştir.

Wang & Tai (2005) yapısal topoloji optimizasyonu için genetik algoritma kullanarak bir bit-dizisi (bit-array) temsil yöntemi uygulaması yapmışlardır. Elde edilen sonuçlar sunulan yöntemin topoloji optimizasyonu problemlerinde kütle azaltımı için başarılı olduğunu göstermektedir.

Yoshida (2006), rüzgâr türbini kulelerindeki maliyetin yaklaşık %20-30'unun kuleden kaynaklandığını ancak tasarım parametrelerinin birbirleriyle olan ilişkileri sebebiyle optimizasyonun zorlu bir hal aldığını belirtmiştir. Bu doğrultuda gerçekleştirdiği çalışmada doğal frekansları, burkulmayı ve ekstrem şartları göz önüne alarak rüzgâr kulesi optimizasyonu için genetik algoritmaya dayanan bir optimizasyon programı geliştirmiştir.

Cheng (2007), yapıların güvenliğini incelemek için yapının hata oranını hesaplayan ve hibrit olarak çalışan bir genetik algoritmaya ve yapay sinir ağına başvurmuştur.

Li vd. (2009) kauçuk takozlar için sonlu elemanlar analizleri ve genetik sinir ağına dayanan bir çalışma ortaya koymuşlar ve tasarım isterlerini sağlayacak kauçuk takoz parametreleri optimize edilmiştir. İstenilen rijitlik ile optimizasyon ile ulaşılan rijitlik arasındaki hatanın %5,8 olduğunu belirtmiş ve sunulan optimizasyon metodunun ürün geliştirme çalışmalarına ayrılan zamanı kısaltmasının yanı sıra tasarım ve ürün geliştirme harcamalarını da düşüreceğini belirtmiştir.

Na vd. (2011), geleneksel hasar tespit metotlarında kullanılan rijitlik matrisinin veri eksikliğine bağlı olarak kaçınılmaz hatalara sebep olduğunu belirtmişler ve bu sorunu aşmak için yapının esneklik matrisini kullanarak genetik algoritmaya dayanan bir yöntem geliştirmişlerdir.

Kaya (2014), kauçuk burçların rijitlik optimizasyonu için gerçekleştirdiği çalışmasında sonlu elemanlar analizi ile bütünleşik çalışan bir DGA geliştirmiştir.

Gentils vd. (2017), deniz aşırı rüzgâr türbinlerinde yer alan destek yapılarının yapısal optimizasyonu için sonlu elemanlar analizinin ve genetik algoritmanın birlikte çalıştığı bir yöntem uygulamışlardır. Sunulan optimizasyon modeli, destek yapılarının kütlesinin %19,8 azaltılmasını sağlamıştır.

Shui vd. (2018), elektrikli araç batarya taşıyıcısının yapısal optimizasyonunu gerçekleştirmişlerdir. Kütleyi minimize edip, birinci doğal frekansı maksimize etmeyi amaçlayarak gerçekleştirdikleri çalışmada bir genetik algoritma varyantı olan NSGA-II algoritmasına başvurmuşlar ve elde ettikleri tasarım parametreleri ile imalatını gerçekleştirdikleri batarya taşıyıcısının fiziksel testlerini gerçekleştirmişlerdir.

Blank & Deb (2020) Python programlama dilini kullanarak oluşturdukları pymoo isimli modül ile tek ve çok amaçlı optimizasyon algoritmalarına, literatürde yer alan test

fonksiyonlarına ve çok boyutlu uzayda görselleştirmeye yarayan fonksiyonlara ulaşımı kolay hale getirmişlerdir.

Joshi vd. (2021) belirledikleri 10 farklı test problemini kullanarak GA, PSO ve Benzetimli Tavlama (BT) algoritmalarını karşılaştırmışlar ve üç algoritma içerisinden en başarılı olanının PSO algoritması olduğunu belirtmişlerdir.

Xiong vd. (2021) gerçekleştikleri optimizasyon çalışmasında elektrikli araç batarya taşıyıcısının ağırlık azaltımını hedeflemişlerdir. Sonlu eleman analizleri ve deney tasarımı ile bir yanıt yüzey metodu (response surface method) geliştirmişler ve bir genetik algoritma kullanarak çarpışma durumlarında istenilen dayanımı sağlayacak optimum batarya taşıyıcısını elde etmişlerdir. Optimum tasarıma sahip taşıyıcının ilk tasarıma göre %11,73 daha hafif olduğunu belirtmişlerdir.

Zhang vd. (2023), elektrikli araç batarya taşıyıcısının titreşim yorulma ömrünü ve çarpışma stresini optimize etmeyi amaçlamışlardır. Bir genetik algoritma varyantı olan NSGA-II algoritmasını kullanarak amaç fonksiyonlarını karşılayan uyun tasarım parametrelerine ulaşmışlar ve Pareto çözümlerini ortaya koymuşlardır.

2.3. Makine Öğrenmesi İçeren Çalışmalar

Öztürk & Öztürk (2001), gerçekleştirdikleri çalışmada tasarımlardaki standart (delikler ve slotlar vb.) ve standart olmayan özelliklerin tanımlanması için sinir ağı kullanımını incelemiş ve tasarımda bulunan özellikleri sınıflandırmışlardır. Sinir ağı kullanımının geleneksel yöntemlere göre standart olmayan özelliklerin sınıflandırılmasında avantaj sağladığını belirtmişlerdir.

Karina vd. (2017) yaptıkları çalışmada, sonlu elemanlar analizi ve deney sonuçlarından elde ettikleri korozyona uğramış çelik levhaların yüzey ve malzeme özelliklerini kullanarak uygulaması basit, yüksek doğruluk oranına sahip ve maliyetsiz olan bir yapay sinir ağı (YSA) modeli geliştirmişler ve eğitilmiş YSA modelini kullanarak çelik levhanın çekme dayanımını tahmin etmeyi amaçlamışlardır.

Müller vd. (2018), kaçınılması mümkün olmayan otomobil kazalarında doğru kararı vermenin insanoğlu için neredeyse imkânsız olduğunu belirterek birbirine paralel olarak çalışacak iki yol (makine öğrenmesi ve fiziksel olasılık) ile çözüm önerisinde bulunmuşlardır. İlk yol, sonlu elemanlar analizlerinin verileri ile eğitilmiş bir makine öğrenmesi modeli, ikinci yol ise kaza kuvvetlerini ve ivmelerini tahmin eden bir iki-boyutlu kütle-yay-damper modelini içermektedir.

Liang vd. (2018), yaptıkları çalışmada makine öğrenmesi tekniklerini kullanarak aorttaki (şahdamarı) gerilme dağılımını hesaplayabilecek bir derin öğrenme modeli geliştirmişlerdir. Eğitilmiş modelin gerilme dağılımı %0,492 ile %0,891 hata ile tahmin edebilme yeteneğine sahip olduğu görülmüştür.

Qi vd. (2019) yaptıkları çalışmada, SEY ve makine öğrenmesine dayanarak karbon fiberin mekanik özelliklerinin tahminini amaçlamışlardır. Çalışmada karar ağacı algoritması kullanımı uygun görülmüş ve aşırı öğrenmeden kaçınmak için uygun ağaç derinliği aranmıştır.

Sampaio vd. (2019), titreşim ölçümlerinden elde ettikleri veri seti ile eğittikleri YSA'nın motorların bozulma zamanlarını tahmin etmesini hedeflemişlerdir.

Francis vd. (2020) on altı adet üç fazlı motoru dört farklı çalışma şartları altında incelemişler ve motorlardan altıkları titreşim sinyallerini YSA eğitiminde veri seti olarak kullanmışlardır. Eğitilen sinir ağını üç fazlı motorların bozulma durumlarının sınıflandırılmasında kullanmışlardır.

Jung & Kim (2020) kauçuk burçlar için gerçekleştirdikleri çalışmada, davranışlarının doğrusal olmaması sebebiyle mekanik özelliklerini belirlemek için kullanılan sonlu elemanlar analizlerinin yüksek çözüm süreleri gerektirdiğini belirterek burçların rijitliklerini tahmin etmek amacıyla yapay zekâya başvurmuşlardır. Kauçuğun davranışını tasarım değişkenlerini kullanarak tahmin eden çok katmanlı YSA ve burcun şeklini kullanarak tahmin eden bir CNN eğitimi gerçekleştirmişlerdir.

Liu vd. (2020) mühendislik problemlerinde kullanılan analitik ve ampirik çözümlerin mümkün olmadığı karmaşık problemlerde makine öğrenmesine dayanan modellerin kullanımına değinmişlerdir. Karar ağacı ve yapay sinir ağı algoritmalarını mekanik kırılma problemlerine uygulanabilirliğini ve başarılarını incelemişlerdir.

Ma vd. (2021), doğrusal olmayan (nonlinear) mafsallı yapılar için giderek artan verimli ve doğru modelleme gereksinimlerini karşılamak için sonlu elemanlar modeline ve derin öğrenme tekniklerine dayanan yeni bir hibrit modelleme yöntemi sunmuşlardır.

Paszyński vd. (2021) çalışmalarında SEY için sayısal çözümün doğruluğunun hesaplanan ağ kalitesine bağlı olduğunu belirterek, yapay sinir ağı kullanarak optimum ağ detayını üreten bir yöntem geliştirmişlerdir.

Kalay vd. (2023) gerçekleştirdikleri çalışmada, 1 boyutlu CNN'e dayanan bir teşhis metodu (sınıflandırma algoritması) geliştirmişlerdir. Simetrik ve simetrik olmayan dişlilerin titreşim tepkilerini ve diş simetrisinin arıza teşhisini kolaylaştırıp kolaylaştırmadığını incelemişlerdir.

2.4. Bütünleşik Yaklaşım İçeren Çalışmalar

Cook vd. (2000), işlem çalışma parametrelerine ve durumlarına bağlı olarak bir üretim prosesindeki bir kritik güç parametresinin tahmini için sinir ağı geliştirmişlerdir. Geliştirilen sinir ağı ile bir genetik algoritma birlikte kullanılarak hangi proses parametresinin istenilen ürün özelliklerini sağlayacağını belirlemek için optimizasyon çalışması yapmışlardır.

Kurtaran vd. (2005), sonlu eleman analizlerinden elde ettikleri veri seti ile eğittikleri sinir ağı ve bir genetik algoritmayı birlikte kullanarak plastik enjeksiyon prosesindeki çarpılmayı minimize etmeyi hedeflemişlerdir. Elde ettikleri sonuçlara dayanarak çarpılmanın döküm sıcaklığı, erime sıcaklığı, paketleme basıncı ve zamanı ile ters orantılı, soğuma sıcaklığı ile doğru orantılı olduğunu belirtmişlerdir.

Shi vd. (2010), plastik enjeksiyon prosesinde çarpılmayı minimize etmek için bir optimizasyon çalışması gerçekleştirmişlerdir. Optimizasyon iterasyon süreçlerinde pahalı hale gelen sonlu eleman analizlerini döküm sıcaklığı, erime sıcaklığı, enjeksiyon süresi, paketleme basıncı, paketleme sıcaklığı ve soğuma süresine parametrelerine bağlı olarak çarpılma değerini tahmin eden bir sinir ağı ile değiştirmişlerdir. Kullanılan yöntemin çarpılma değerini azaltmaya ve optimizasyon sonuçlarına hızlıca ulaşmaya imkân tanığı görülmüştür.

Yin vd. (2011), plastik enjeksiyon prosesinde çarpılmanın plastiğin kalitesini etkilediğini belirtmişler ve döküm sıcaklığı, erime sıcaklığı, paketleme basıncı, paketleme zamanı ve soğuma zamanına bağlı olarak çarpılma değerini tahmin edecek bir sinir ağı geliştirmişlerdir. Çarpılma değerini minimize etmek için gerçekleştirdikleri optimizasyon çalışmasında sinir ağı kullanarak 2,358 mm olan çarpılma değerini %32,99 azaltarak 1,58 mm'ye indirmişlerdir.

Pfrommer vd. (2018), CoFRP (Continuous Fiber Reinforced Plastic) üretim prosesindeki parametrelerin optimizasyonu için kayma açısını tahmin edecek bir sinir ağı geliştirmişlerdir. Optimizasyon çalışması ile elde edilen sonuçlar SEY ile doğrulanarak sinir ağı modelinin doğruluğunun arttırılması için eğitim için kullanılan veri setine dahil edilmiştir.

Sun vd. (2020), rüzgâr türbinlerinin modellenmesi için bir yapay sinir ağı geliştirmişler ve rüzgâr türbininin güç oranını maksimize etmek için genetik algoritmaya başvurarak bir optimizasyon çalışması gerçekleştirmişlerdir.

Guo vd. (2020), plastik enjeksiyon prosesinde döküm ve erime sıcaklıklarının yanı sıra soğuma sıcaklığı, soğuma Reynolds sayısı ve gaz konsantrasyonlarını da araştırmalarına dahil ederek çarpılma değerini incelemiştir. Çarpılma değerini tahmini için bir sinir ağı geliştirmişler ve genetik algoritma kullanarak çarpılma değerini minimize etmişlerdir.

Bulut vd. (2022), elektrikli araç modüllerinin soğutulması ve enerji tüketiminin azaltılması için soğutma kanalının boyutlarını ve kütle akış hızını tahmin eden bir sinir

ağı geliştirmişlerdir. Sinir ağı kullanımı ile gerçekleştirdikleri optimizasyon çalışmasında soğutma performansından ödün vermeyerek enerji tüketiminde %22,4 azalmaya ulaşılmıştır.

Gao vd. (2021), rotor-rulman sitemlerindeki arıza teşhisinin önemi vurgulayıp deneysel olarak elde ettikleri verilerin yanı sıra toplayamadıkları veriler için sonlu elemanlar yöntemini kullanmışlardır. Çekişmeli üretken ağlar ile elde ettikleri sentetik verileri de deneysel ve sonlu elemanlar verileri ile birleştirerek rulmanların arıza teşhis sınıfını belirleyebilecek bir yapay zekâ modeli geliştirmişlerdir.

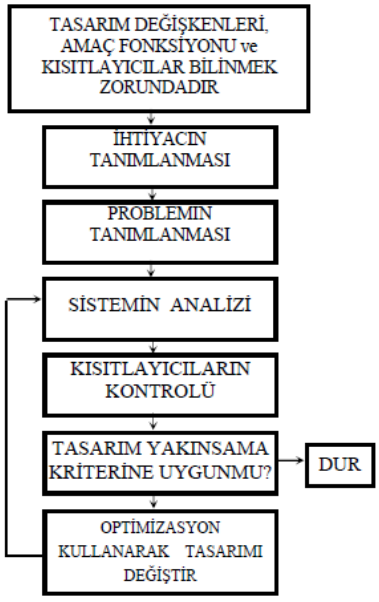
Literatürde sinir ağı kullanımına korozyona uğramış levhaların dayanımını belirlemek, insan tepkilerinin yeterince hızlı olamayacağı otomobil çarpışmalarında araç güvenliğini sağlamak amacıyla süspansiyon bileşenlerini uygun konuma getirmek, rotorlar için bakım zamanlarını belirlemek ve bazı parçaların hasar durumlarını belirlemek için gerçekleştirilen çalışmalarda rastlanmaktadır. Literatürde ürün tasarımında sinir ağı kullanımına Jung & Kim'in (2020) gerçekleştirdiği çalışmada rastlansa da yazarlar, optimizasyon çalışması gerçekleştirmemişler, kauçuk burcun rijitliğini belirleme üzerine çalışmışlardır. Makine öğrenmesi kullanılarak gerçekleştirilen optimizasyon çalışmalarının özellikle hammaddesi plastik olan ürünlerin üretim proseslerinde yer aldığı görülmekle birlikte ürün tasarımında, vekil model yerine makine öğrenmesi ile gerçekleştirilen optimizasyon çalışmaları literatürde yer almamaktadır.

3. MATERYAL VE YÖNTEM

Bu bölümde, tez kapsamında kullanılan optimizasyon, evrimsel algoritmalar ve makine öğrenmesi hakkında temel bilgilere yer verilmiş ve optimizasyon çalışmasında kullanılan yöntem açıklanmıştır.

3.1. Optimizasyon

Kökeni Latince *optimus* olan optimizasyon, kaynakların en verimli şekilde kullanılarak en yüksek bilgiye ulaşılması (Andradbttir vd., 1997) ya da en iyinin elde edilmesi olarak tanımlanabilir. Kavramsal değişiklikler yapmak ve tasarım sürecine ilave şartlar koymak geleneksel tasarım sürecinin temel avantajıdır (Yıldız, 2006). Optimum tasarım süreci ise kısıtlayıcı fonksiyonları, minimize edilecek fonksiyonları ve tasarım değişkenlerini açıkça tanımlamaya zorlar (Yıldız, 2006). Şekil 1'de bir optimum tasarım süreci belirtilmiştir.



Şekil 1. Optimum tasarım süreci (Yıldız, 2006).

Bir optimizasyon problemi minimize ya da maksimize edilecek olan *amaç fonksiyonunu*, her bir tasarımın sağlaması gereken *kısıt fonksiyonlarını* ve *tasarım değişkenlerini* içerir.

3.1.1. Tasarım Değişkenleri

Bir optimizasyon çalışmasını tanımlanmasında seçilen parametreler *tasarım değişkenleri* olarak adlandırılmaktadır. Tasarım değişkenleri aynı zamanda problemin tasarım uzayını belirtmektedir. Tanımlanan tüm tasarım değişkenleri birbirinden bağımsız olmalı ve alt ile üst sınırlar içerisinde tanımlanmalıdır. Belirtilen alt ve üst sınırlar içerisinde amaç fonksiyonunun minimum ya da maksimum değerinin bulunması amaçlanır.

$$x_{a_i} < x_i < x_{ü_i} \quad (1)$$

$$x_{a_j} < x_j < x_{ü_j} \quad (2)$$

3.1.2. Amaç Fonksiyonları

Bir optimizasyon probleminde optimum tasarımı aramak tasarım değişkenlerinin bir fonksiyonu olan bir kriter, *amaç fonksiyonu*, ile gerçekleşir (Yıldız, 2006). Amaç fonksiyonu sayısal değeri olan bir fonksiyon olarak tanımlanır ve tasarım değişkenlerine bağlı olarak minimum ya da maksimum değeri aranır. Literatürde, rijitlik optimizasyonu ve kütle minimizasyonu gibi farklı amaç fonksiyonları kullanılır.

$$f(X) = f(x_1, x_2, \dots, x_n) \quad (3)$$

Birçok problemde amaç fonksiyonu ile tasarım değişkenleri arasındaki ilişkiyi analitik olarak ifade etmek mümkün değildir. Bu durumda uygun amaç fonksiyonunu belirlemek için ya doğrudan sonlu elemanlar analizi ya da elde edilmiş veriler ile bir vekil model kurularak farklı tasarım değişkenlerine karşın amaç fonksiyonu hesaplanır.

3.1.3. Kısıt Fonksiyonları

Her kısıt fonksiyonu tasarım uzayındaki çözümlerin bir bölümünün kullanılamayacağını belirtir ve kısıt fonksiyonları bir bütün olarak tasarım uzayındaki uygun çözümleri belirler (Kochenderfer & Wheeler, 2019). Kısıt fonksiyonları bir mühendislik problemi için tasarımın şeklini bozacak, arzu edilen fonksiyonelliği sağlayamayacak tasarımları

ortadan kaldırabileceği gibi bir yük altında çalışan parçanın belirli bir maksimum gerilme ya da deformasyon değerini de kısıtlayabilir.

Bir optimizasyon problemi eşitlik kısıtlayıcılarına sahip olabileceği gibi eşitsizlik kısıtları da içerebilir. Optimizasyon çalışmasında elde edilecek olan uygun tasarımlar bu kısıt fonksiyonlarını sağlamak zorundadır. Denklem 4'te bir eşitlik kısıtlayıcı, Denklem 5'te ise bir eşitsizlik kısıtlayıcısı belirtilmiştir.

$$h(X) = 0 \quad (4)$$

$$g(X) \leq 0 \quad (5)$$

Standart bir optimizasyon çalışmasında aşağıdaki hususlara dikkat edilmelidir (Yıldız, 2006):

1. Bağımsız eşitlik kısıtlarının sayısı tasarım değişkenlerinin sayısından az veya eşit olmalıdır ($P \leq n$). Eğer $P > n$ olursa gereğinden fazla denklem sistemine sahip olunur ve bu durumda ya gereksiz eşitlik kısıtlayıcıları vardır ya da formülasyon çelişkilidir. $P = n$ olması sistemin optimizasyonunu gereksiz kılar çünkü eşitlik kısıtlayıcılarının çözümü sadece optimum çözüme uygundur.
2. Bağımsız eşitlik kısıtlarının sayısında sınırlama olmasına karşın eşitsizlik kısıtlarının sayısında sınırlama yoktur.
3. Bir optimizasyon problemi herhangi bir kısıtlamaya sahip olmak zorunda değildir.
4. Eğer tüm fonksiyonlar X tasarım değişkeninde doğrusal ise problem doğrusal programlama problemi olarak adlandırılır.

3.1.4. Standart Bir Optimizasyon Probleminin Tanımlanması

Bir optimizasyon probleminin çözümü için tasarım değişkenleri ile kısıt ve amaç fonksiyonları tanımlanmış olmalıdır. Ürün tasarımını belirleyen parametreler olan tasarım değişkenlerinin sınırları belirlenmelidir. Bir topoloji veya topografya optimizasyonu gerçekleştiriliyor ise tasarım değişkenleri CAD (Computer Aided Design) model üzerinde tanımlanır. Kütle ve hacim kısıt ve amaç fonksiyonlarında kullanılacak ise bu durumda da CAD modeli kullanılır. Ancak gerilme veya doğal

frekans gibi çıktılar gerekli ise Sonlu Elemanlar Analizlerine başvurulur. Standart bir optimizasyon problemi aşağıda belirtildiği gibi tanımlanabilir:

1. Tasarım değişkenleri:

$$X = (x_1, x_2, \dots, x_n), (x_i)_{min} \leq x_i \leq (x_i)_{maks} \quad (6)$$

2. Amaç:

$$\text{min veya maks } f_i(X) = f_i(x_1, x_2, \dots, x_n) \quad (7)$$

3. Kısıt:

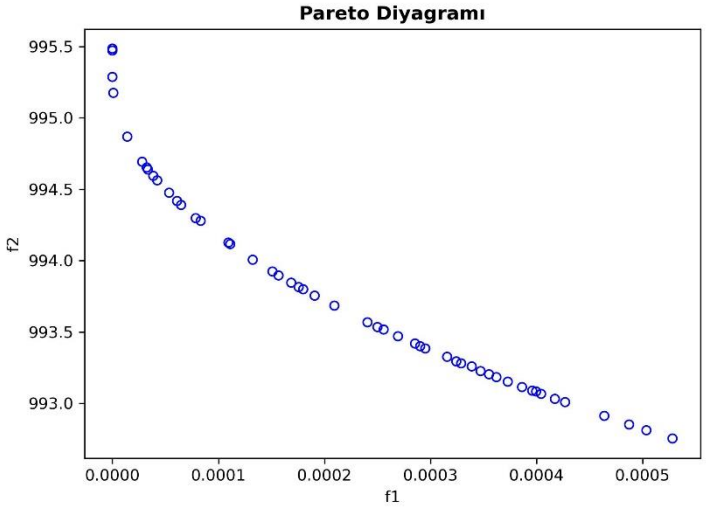
$$h_i(X) = h_i(x_1, x_2, \dots, x_n) = 0; 1 \leq i \leq m \quad (8)$$

$$g_j(X) = g_j(x_1, x_2, \dots, x_n) \leq 0;\ 1 \leq j \leq p \quad (9)$$

3.1.5. Çok Amaçlı Optimizasyon

Bir optimizasyon problemi bir amaç fonksiyonu içeriyorsa *tek amaçlı optimizasyon*, birden fazla amaç fonksiyonu içeriyorsa *çok amaçlı (multi-objective) optimizasyon* olarak adlandırılır. Birçok mühendislik problemi maliyeti azaltmak ve performansı, dayanımı arttırmak gibi birden fazla amacı içerisinde barındırabilir. Birden fazla amaç fonksiyonunu minimize edilmesi ağırlıklandırılma yöntemleri ile gerçekleşebileceği gibi Pareto çözümleri de incelenebilir.

Pareto çözümleri ağırlıklandırılma yöntemleri gibi tek bir sonucun olduğu çözümlere kıyasla daha çok tercih edilir (Konak vd., 2006). Pareto çözümlerinin olduğu bir çoklu amaç fonksiyonunda tek bir çözüm yerine her biri kabul edilebilir seviyede olan birden çok çözüm veya bir çözüm seti elde etmek amaçlanır (Deb vd., 2002; Konak vd., 2006; Marseguerra vd., 2006). Bir amaç fonksiyonundan feragat edilerek diğerine ağırlık verilebileceği düşünüldüğünde Pareto çözümleri, aranan sonucun amaç fonksiyonları arasında ödün vererek ulaşılmasına olanak sağlar (Konak vd., 2006; Marseguerra vd., 2006). Örnek bir Pareto diyagramı Şekil 2'de gösterilmektedir.



Şekil 2. Örnek bir Pareto diyagramı

Ağırlıklandırılma yöntemleri kullanıcının hedeflerine (amaç fonksiyonlarına) göreceli önemler atfederek birden fazla amaç fonksiyonunun tek bir fonksiyona indirgenmesine olanak sağlar. Ağırlıklandırılmış yöntemlerin çoğu kullanıcının tercihlerini yansıtacak şekilde ayarlanabilen ve değişime açık olan katsayılar ve üsler gibi parametreleri içerir (Marler & Arora, 2004). Ağırlıklandırma ile birden fazla amaç fonksiyonu içeren optimizasyon problemlerinin tek bir fonksiyona indirgenmesi için kullanılan bazı yöntemler aşağıda verilmiştir (Marler & Arora, 2004).

$$Ağırlıklandırılmış\ Global\ Kriter: U = \sum_{i=1}^{k} \omega_i [F_i(X)]^p \quad (10)$$

$$Ağırlıklandırılmış\ Toplam: U = \sum_{i=1}^{k} \omega_i F_i(X) \quad (11)$$

$$Ağırlıklandırılmış\ Eksponensiyel: U = \sum_{i=1}^{k} (e^{p\omega_i} - 1) e^{pF_i(X)} \quad (12)$$

$$Ağırlıklandırılmış\ Çarpım: U = \prod_{i=1}^{k} [F_i(X)]^{\omega_i} \quad (13)$$

Burada:

U: Ağırlıklandırılmış amaç fonksiyonu,

ω: Kullanıcı tarafından belirlenen ağırlıklandırma. Her bir ağırlıklandırma sıfırdan büyük olmalı ve ağırlıklandırmaların toplamları bir olmalıdır. ($\sum_{i=1}^{k} \omega_i = 1$ ve $\omega > 0$)

F: Amaç fonksiyonları,

p: Çoğunlukla sabit olarak seçilen bir üstel kuvvet.

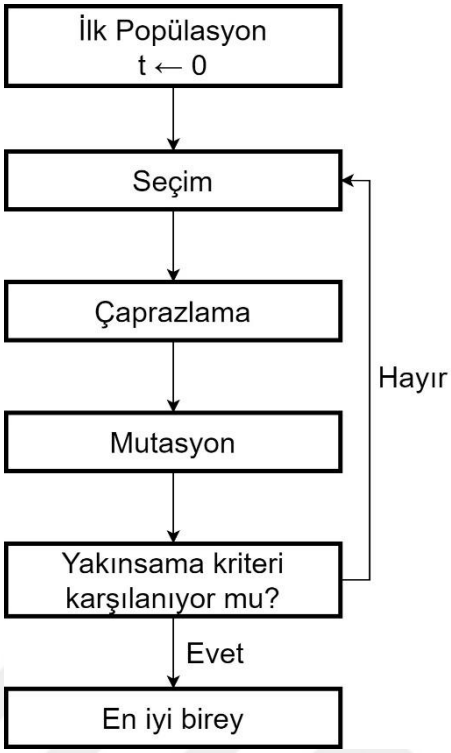
3.2. Evrimsel Algoritmalar

Bremermann (Bremermann, 1962) ve Friedberg (Friedberg, 1958) gibi bilim insanlarından etkilenerek başlayan evrimsel algoritmaların kökeni 1950'lere kadar dayanmasına rağmen neredeyse otuz yıl boyunca (Holland ve diğer bilim insanlarının 1970 sonları ve 1980'lerde gerçekleştirdikleri çalışmalara kadar) bilim dünyasında bir bilinmeyen olarak kalmıştır (Back vd., 1997). Bunun asıl sebebi evrimsel algoritmaların yüksek işlem gücü ihtiyacı duyması ve piyasadaki işlem gücü yüksek bilgisayar eksikliğidir (Back vd., 1997).

Geleneksel optimizasyon algoritmalarında karşılaşılan ana sorunlardan biri algoritmanın global optimum yerine lokal optimuma yakınsamasıdır. Geleneksel algoritmalar ile doğrusal ve az sayıda değişken içeren problemlerin çözümü mümkün olsa da karmaşık problemler için lokal optimuma takılma problemi ile karşılaşılabilir. Karmaşık problemlerin çözümü ve lokal optimuma takılma sorunu evrimsel algoritmalar gibi yenilikçi optimizasyon algoritmaları ile mümkün olmaktadır.

Optimizasyon yöntemleri temel olarak gradyan tabanlı ve gradyan tabanlı olmayan olmak üzere iki grupta toplanabilir. Yakınsamanın yavaş gerçekleşmesi ve global optimum sonuca ulaşmanın garanti edilmemesi gradyan tabanlı optimizasyondaki iki dezavantajdır (Ahmadianfar vd., 2020). Başlangıç noktalarının rastgele olarak dağıtıldığı gradyan tabanlı olmayan optimizasyonda (diferensiyel gelişim algoritması, genetik algoritmalar vb.) süreç, yakınsama kriteri karşılanana kadar her bir adımda yeni popülasyonun önceki popülasyondaki örneklerin amaç fonksiyonu ile olan uyumuna bakılarak elde edinilen bilgiler doğrultusunda güncellenmesiyle devam eder (Ahmadianfar vd., 2020; Marseguerra vd., 2006; Yıldız vd., 2003).

Birçoğu popülasyon tabanlı olan evrimsel algoritmalar iteratif optimizasyon metotlarıdır. Her iterasyonda tek bir çözümün incelenmesi (Benzetimli Tavlama) işlem gücü avantajı sağlayabilirken popülasyon tabanlı algoritmalar (GA, DGA vb.) birden çok çözümün incelenmesine olanak sağlar ve çözüm uzayı daha iyi incelenir. Bir evrimsel algoritmanın genel akış diyagramı Şekil 3 ile gösterilmiştir.



Şekil 3. Bir evrimsel algoritmanın genel akış diyagramı

Rastgele dağıtılan noktalar ile farklı alanlardaki çözümleri incelemek yüksek işlem gücü gerektirmesine rağmen yerel optimum yerine global optimumun ulaşılabilmesine olanak sağladığından bu gerekliliğe değen bir harcamadır (Marler & Arora, 2004). Lokal optimuma yakalanmadan global optimumu bulabilme imkânı sağlayan gradyan tabanlı olmayan algoritmalara uygulamalarda sıklıkta başvurulmaktadır.

Bir optimizasyon çalışmasında farklı koşullar altında farklı sonlandırma/yakınsama kriteri kullanılabilir. Yaygın olarak kullanılan sonlandırma kriterleri aşağıda listelenmiştir.

1. *Aranan tasarım değerlerinin yakalanması:* Bir optimizasyon probleminde amaç fonksiyonunun minimum değeri biliniyor ise optimizasyon çalışması aranan tasarım değerleri kabul edilebilir bir hata değeri ile yakalanana kadar devam ettirilebilir.
2. *Maksimum iterasyon:* Birçok uygulamada amaç fonksiyonunun optimum değeri bilinmez veya optimizasyon algoritması optimum sonuca ulaşmak için yeterli

süreye ya da kaynağa sahip olmayabilir. Bu durumda optimizasyon algoritması arama işlemini belirlenen bir iterasyon sayısına ulaşınca durdurur.

3. *Popülasyon istatistikleri:* Bir optimizasyon çalışması belirlenen bir popülasyon istatistiğine bağlı olarak da sonlandırılabilir. Örneğin popülasyondaki amaç fonksiyonları içerisinde yer alan en iyi ve en kötü sonuçlar arasındaki fark önceden belirlenmiş olan bir limit değerinin altına düştüğünde algoritmanın sonlandırma kriterini karşılanmış olur (Price vd., 2006). Bir diğer istatistik ise popülasyon çeşitliliğindeki kısıtlama olabilir (Qing, 2009). Popülasyondaki bireylerin birbirlerinden çok az farklılık göstermesi algoritmanın optimum noktaya yakınsadığının bir göstergesi olarak kabul edilebilir. Dolayısıyla bu noktadan sonra algoritmayı sürdürmenin bir anlamı olmaz.

3.2.1. Çalışma Prensibi

Bir evrimsel algoritmayı tanımlamak için bir dizi operatöre ihtiyaç duyulmaktadır. Üreme işleminden sonraki her bir iterasyon için mevcut popülasyondaki bireyler üzerinde üç temel işlem gerçekleştirilir:

1. Çaprazlama
2. Mutasyon
3. Seçim

3.2.1.1. Üreme (Initialization)

Evrimsel algoritma, belirlenen tasarım değişkenleri ile tanımlanan tasarım uzayı içerisinde rastgele oluşturulan başlangıç popülasyonu ile başlar. Oluşturulan gerçek dünya popülasyon bireylerinin evrimsel algoritma dünyasında tanımlanması için ilk adım olarak temsil (representation) adı verilen ve fenotip (phenotype) ile genotip (genotype) olarak adlandırılan işlemin gerçekleşmesi gerekmektedir (Eiben & Smith, 2015).

Orijinal problemdeki olası çözümler fenotipler, fenotiplerin kodlamaları yani EA içerisindeki bireylere genotipler denir (Eiben & Smith, 2015). Örneğin 38 sayısı bir

fenotip ve ikili sistemde karşılığı olan 100110 EA sistemindeki genotipi olarak adlandırılmaktadır.

3.2.1.2. Çaprazlama (Crossover)

Yeniden kombinleme (recombination) ya da çaprazlama ikili varyasyon operatörü olarak adlandırılır (Eiben & Smith, 2015). İsminden de anlaşılacağı üzere çaprazlama iki genotipteki bilgilerin sonraki jenerasyon için bir ya da iki farklı genotipte birleştirilmesidir.

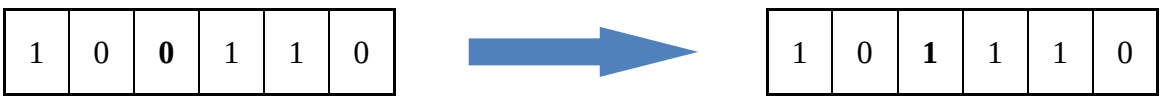
Çaprazlama da diğer evrimsel algoritma komponentleri gibi stokastiktir. Örneğin her bir genotipteki hangi bölümlerin seçileceği ve nasıl birleştirileceği rastgele gerçekleştirilir. Bireylerin sonraki jenerasyon için uygun olan bölümleri birleştirilebileceği gibi uygun olmayanları da birleştirilebilir.

İkili sistem için literatürde bilinen ve sıklıkla kullanılan 3 adet çaprazlama işlemi bulunmaktadır (Yıldız, 2006). Bunlar:

1. Tek Noktalı Çaprazlama
2. Çift Noktalı Çaprazlama
3. Uniform Çaprazlama

3.2.1.3. Mutasyon (Mutation)

Üreme ve çaprazlama işlemlerinin tamamlayıcısı olan mutasyon komponenti, bit değerinin tersine çevrilmesidir (Yıldız, 2006). Bir genotipe uygulanan mutasyon sonucunda çok az değiştirilmiş yeni bireyler oluşturulmuş olur. Mutasyon operasyonunda değeri değiştirilen bit rastgele seçilir yani mutasyon operasyonu da çaprazlama gibi stokastiktir. Mutasyon işleminin bir şematik gösterimi Şekil 4'teki gibidir.



Şekil 4. Mutasyon işlemi

Varyasyon operatörleri olarak adlandırılabilen çaprazlama ve mutasyon işlemlerinin amaçları eski bireylerden yeni bireyler oluşturmaktır. Mutasyon, bit ile işlem yapması sebebiyle tekil varyasyon operatörü iken çaprazlama çoğul varyasyon operatörüdür (Eiben & Smith, 2015).

3.2.1.4. Seçim (Selection)

Ebeveyn seçimi, bireyler arasında kalitelerine (uygunluklarına) göre ayrım yaparak özellikle daha iyi bireylerin bir sonraki nesilde ebeveyn olmasını sağlamaktır. Yüksek uygunluğa sahip bireylerin bir sonraki nesilde ebeveyn olma olasılığı daha yüksektir ancak, düşük kalitedeki bireylerin de ebeveyn olmasına küçük de olsa izin verilmelidir. Aksi taktirde arama algoritması yeni sonuçlar elde edemez ve lokal optimum noktaya takılır (Eiben & Smith, 2015).

Literatürde farklı karakteristiklere sahip birçok seçim yöntemi vardır. Farklı mekanizmaların en ideali en basit şekilde kodlanabilen ve en verimli olanıdır(Miller vd., 1995). Sıklıkla kullanılan seçim mekanizmaları 4 ana sınıfta toplanabilir. Bunlar:

1. Orantılı Seçim
2. Sıralı Seçim
3. Turnuva Seçim
4. Denge Durumu Seçim

3.2.2. Evrimsel Algoritma Türleri

Literatürde arklı türlerde evrimsel algoritmalar bulunsa da algoritmaların temelinde yatan fikir aynıdır: sınırlı kaynaklara sahip bireylerden oluşan bir popülasyonda bu kaynaklar popülasyon için rekabete, doğal seçilime (en uygun olanın hayatta kalması) neden olur (Eiben & Smith, 2015). Literatürde yer alan evrimsel algoritmaların bazıları aşağıda listelenmiştir.

1. Genetik Algoritma (Goldberg & Holland, 1988)
2. Diferensiyel Gelişim Algoritması (Storn & Price, 1996)
3. Evrimsel Programlama (EP) (Yao vd., 1999)

4. Biyocoğrafya Tabanlı Optimizasyon (BTO) (Simon, 2008)

3.2.2.1. Genetik Algoritma

Adından anlaşılacağı üzere Genetik Algoritmalar (GA), Darwin'in evrim teorisinden esinlenilerek geliştirilmiştir (Goldberg & Holland, 1988). Genetik bilimdeki evrim prensiplerini ve kromozomal prosesleri bilgisayar ortamında taklit ederek çalışarak bir optimizasyon metodu olan genetik algoritmalar popülasyon tabanlıdır (Yıldız, 2006).

Genetik algoritmalar optimizasyon problemi için uygun çözümlerin incelendiği ve biyolojik özelliklerin taklit edildiği bir bölüm operatörlerden (çaprazlama, mutasyon) meydana gelir. Algoritma içerisinde bireyler diziler ile temsil edilirken bireylerin oluşturduğu gruba da popülasyon adı verilir.

Genetik algoritmaların direkt kodlama ile kullanılması, tek bir noktayı değil bir popülasyonu incelemesi, gradyan teknikleri gibi yardımcı bilgiye ihtiyaç duymaması ve çaprazlama ve mutasyon gibi operatörlerinin rastgele olması ile diğer algoritmalardan farklılık göstermektedir (Yıldız, 2006).

3.2.2.2. Diferensiyel Gelişim Algoritması

Görece yeni bir evrimsel optimizasyon algoritması sayılan DGA, çok yönlü ve kullanımı kolay bir global optimizasyon algoritmasıdır (Kaya, 2014). Diğer evrimsel algoritmalar gibi popülasyon tabanlı olan DGA konsept olarak basit ve kullanımı kolaydır (Storn & Price, 1996). Genetik algoritmada da olduğu gibi DGA algoritması mutasyon, çaprazlama ve seçim operatörlerini kullanır.

Storn ve Price'ın 1996 yılındaki makalesinde yer alan iki varyantın (DE/rand/1 ve DE/best/2) yanı sıra araştırmacılar tarafından farklı varyantlar da geliştirilmiştir. 2005 yılından bu yana farklı DGA varyantları IEEE International Conference on Evolutionary Computation (CEC) yarışmasında 2013 yılındaki dördüncülük hariç en başarılı ilk üç algoritma olmayı başarmışlardır (Bilal vd., 2020). Google Scholar verilerine göre 500'den fazla atıf alan DGA varyantları Çizelge 1'de listelenmiştir.

Çizelge 1. Farklı DGA varyantları (Bilal vd., 2020).

Varyant	Yıl	Atıf
Basic Differential Evolution (DE)	1996	20366
Self-Adaptive Differential Evolution (SaDE)	2005	2410
Adaptive Differential Evolution with Optional External Archive (JADE)	2009	1888
Opposition Based Differential Evolution (ODE)	2008	1296
Neighborhood Based Differential Evolution (NDE)	2009	960
Composite Differential Evolution (CoDE)	2011	898
Fuzzy Adaptive Differential Evolution (FADE)	2005	857
Generalized Differential Evolution (GDE3)	2005	523

3.2.3. Performansı Etkileyen Faktörler

Algoritmanın popülasyon boyunun tasarım uzayının incelenmesinde büyük önemi vardır. Özellikle global optimumun arandığı göz önüne alınırsa arama uzayındaki çözümleri içeren popülasyonun düşük sayıda seçilmesi ile arama uzayı yeterince incelenemeyecektir ve global optimum noktaya ulaşılmayabilir. Popülasyonun büyük seçilmesi ile aramanın etkinliği artar ancak bir dezavantaj olarak yüksek işlem gücü gereksinimi ile yakınsama yavaş gerçekleşir.

Popülasyon değişkenliğini gerçekleştiren çaprazlama oranı performansa etki eden diğer bir faktördür. Düşük çaprazlama oranı aramanın yavaş gerçekleşmesine neden olur (Yıldız, 2006).

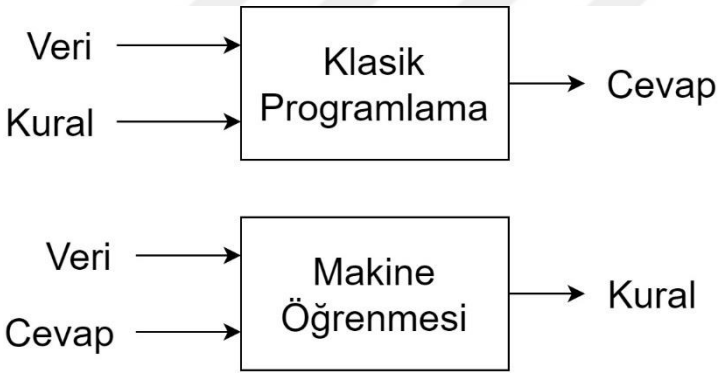
Birbirine benzeyen diziler ile algoritmanın sıkıştığı yerden kurtulmasının en iyi yolu mutasyondur. Yüksek mutasyon oranı algoritmada kararsızlığa neden olacağı için düşük mutasyon oranı tavsiye edilmektedir (Yıldız, 2006).

3.3. Makine Öğrenmesi

Yapay zekânın bir alt dalı olan makine öğrenmesi, matematiksel ve istatistiksel işlemler ile veriler üzerinden çıkarımlar yaparak doğru tahminleri yapabilecek olan sistemler elde etme olarak tanımlanabilir. Burada amaç, veri setini tamamen doğru bir şekilde

tanımlayamasak da veriyi en iyi şekilde ifade edebilecek bir genelleştirme yapmak, modelin eğitim aşamasında bilgi sahibi olmadığı verilerde de yeterince doğru sonuçlar üretmesini sağlamaktır (Alpaydin, 2020).

Klasik programlamada insanoğlu, algoritmaya girdi olarak kurallar ve bu kurallara (programa) bağlı olarak işlenecek veri setini vererek bir çıktı elde etmeye çalışır (Şekil 5). Makine öğrenmesinde ise algoritmaya veri setindeki girdi ve çıktı değerleri verilerek makinenin girdi ve çıktılar arasındaki bağlantıyı sağlayan kuralları belirlemesi beklenir (Chollet, 2018). Dolayısıyla bir makine öğrenmesi sistemi programlamadan ziyade eğitilir.



Şekil 5. Klasik programlama ve makine öğrenmesi (Chollet, 2018).

Makine öğrenmesi modelinden elde ettiğimiz çıktı kategorik ise sınıflandırma (classification), nümerik ise regresyon (regression) olarak adlandırılır. Açıklayıcı bir modelleme olan kümeleme (clustering) ise benzer gözlemleri aynı kümelere atama işlemidir (Uzun, 2023).

Sınıflandırma ve regresyon işlemleri yapı olarak birbirine benzerdir, mevcut verilerden elde edinilen bilgiler ile yeni verileri tahmin etme işlemidir. Elde edilen veri seti kullanılarak etiketlenmiş (labeled) veriler ile öğrenme işlemi gerçekleştirilir. Sınıflandırmada sonuç bilgisi 0-1 gibi bir kategorik veri iken regresyon işlemi için çıktı 8,26 gibi bir sayısal değerdir.

Kümelemede ise, sınıflandırma ve regresyondan farklı olarak, veri setindeki veriler etiketlenmemiştir (unlabeled), veri setinde çıktı bilgisi yoktur. Ancak burada amaç veri setinde kümeleme yaparak verileri kümelere ayırmaktır. Bu sebeple, sınıflandırma ve regresyon denetimli öğrenme (supervised learning) algoritmaları olarak adlandırılırken, kümeleme ise denetimsiz öğrenme (unsupervised learning) algoritması olarak adlandırılır.

3.3.1. Çalışma Prensibi

Bir makine öğrenmesi modeli geliştirme süreci temelde 4 adımdan oluşmaktadır (Liu vd., 2020). Bunlar;

1. Problemin tanımlanması
2. Veri setinin hazırlanması
3. Modelin eğitilmesi
4. Modelin değerlendirilmesi

3.3.1.1. Problemin Tanımlanması

Bir makine öğrenmesi modeli için en önemli adımı veri seti oluşturmaktadır. Bir mühendislik problemi için; o problemin başlangıç ve sınır şartları, malzeme özellikleri ve mesh yapısı gibi değişkenlerin tanımlanarak bir sonlu elemanlar programındaki çözümler ile veri setine ulaşılabileceği gibi doğrudan fiziksel testler ile de bir veri seti elde edilebilir.

Veri seti tüm anormalliklerde arındırılmış olabileceği gibi günümüz bilgisayarlarının anlayamayacağı durumda da olabilir. Veri setini anlamlı hale getirmek ise mühendisin tecrübe ve becerilerine bağlıdır. Bu yüzden bir model eğitimine bağlanılmadan önce verinin setinin incelenmesi önem arz etmektedir. Verinin görselleştirilmesi yani x ve y-eksenlerinde birer özelliğin (feature) ifade edilmesi veri setinin incelenmesinin en iyi yoludur (A. C. Müller & Guido, 2016).

Veri seti hakkında elde edilmesi gereken en önemli bilgilerden biri de hangi türdeki makine öğrenmesi metodunun (denetimli, denetimsiz vb.) veri setine uygun olduğudur.

Veri setinde çıktıların olup olmadığı; bir çıktı var ise sınıflandırma mı yoksa regresyon problemi mi olduğuna karar verilmelidir.

3.3.1.2. Veri Setinin Hazırlanması

Hem veri hem de kullanılacak olan makine öğrenmesi modeli veri setinde düzenlemeler gerektirebilir. Yüksek girdiye sahip veri setleri için özellikler arasındaki korelasyonun bilinmesi hem eğitim için gereken işlem gücünün azaltılması hem de aşırı öğrenmeden (overfitting) kaçınılması için önemlidir. PCA (Principal Component Analysis) ve LDA (Linear Discriminant Analysis) gibi yöntemler boyut indirgeme (dimension reduction) metotları olarak karşımıza çıkmaktadır.

Bilgisayarın veri seti içerisinde yer alan kategorik verileri işleyebilmesi için bu veriler sayısal hale getirilmelidir. En çok kullanılan yöntemlerden biri OneHotEncoder() olarak karşımıza çıkmaktadır. Bu yöntem temelde bir kategorik özelliğin birden fazla özelliğe ayrılarak 0 ve 1 ile ifade edilmesidir. OneHotEncoder için bir örnek Şekil 6'de belirtilmiştir.

Renk
Kırmızı
Sarı
Kırmızı
Mavi

OneHotEncoding

Kırmızı	**Sarı**	**Mavi**
1	0	0
0	1	0
1	0	0
0	0	1

Şekil 6. OneHotEncoder örneği

Bir diğer gereksinim ise birçok makine öğrenmesi modeli için ortak gereksinim olan verilerin ölçeklendirilmesidir. Birden fazla özelliğin bulunduğu veri setlerinde özellikler çoğunlukla sayısal olarak farklı aralıklarda değişmektedir. Sayısal olarak yüksek değerlere sahip olan özelliğin model eğitiminde baskın olacağı ve çıktılara olan etkisinin daha yüksek olacağı göz önüne alınırsa her bir özelliğin kendi içinde ölçeklendirilmesi ihtiyacı ortaya çıkmaktadır. StandardScaler() sıklıkla kullanılan ölçeklendirme yöntemlerinden biridir.

StandardScaler(), her bir özelliği kendi içerisinde olmak üzere, veri setini -1 ile 1 arasında ölçeklemektedir. Bu standardizasyon işlemi Denklem 14'te belirtilmiştir.

$$z = (x - \mu)/\sigma \quad (14)$$

Burada;

z: x'in ölçeklendirilmiş değeri,

x: Özelliğin bir veri için gerçek değeri,

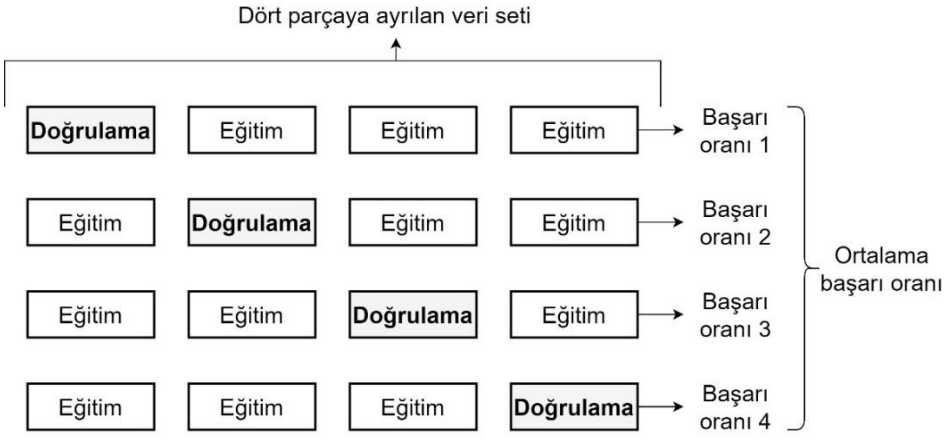
μ: Özelliğin ortalaması,

σ: Özelliğe ait standart sapma değeridir (Scikit-Learn, 2023).

3.3.1.3. Modelin Eğitilmesi

Daha önce de bahsedildiği üzere amaç veri setini ifade edecek en genel modele ulaşmaktır. Veri setindeki tüm verilerin eğitim için kullanılmasındansa genel olarak veri setinin %80'inin eğitim , %20'sinin ise test için kullanılması tercih edilir (Géron, 2022). Farklı uygulamalarda test için ayrılan veri %33'e kadar çıkabilmektedir.

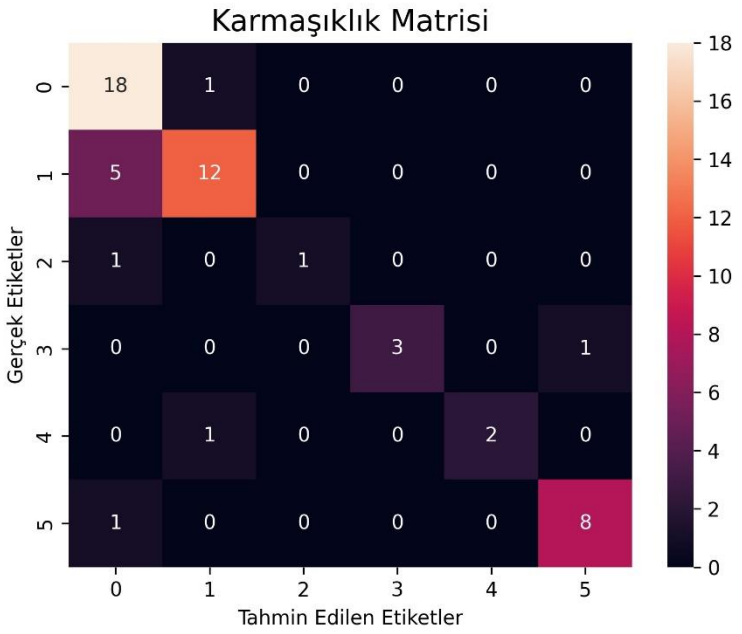
Eğitim aşamasındaki bir diğer yaklaşım olan K-Fold cross-validation, veri setinin eşit sayıda veri içeren K farklı bölüme ayrılarak eğitimin gerçekleştirilmesidir (Chollet, 2018). Modelin aynı veri setini kullanmasına rağmen farklı sıralamalarda işlediği veriler ile başarı oranlarında değişiklikler gözlemlenebilmektedir. Eğitimdeki genel başarı oranı K farklı bölümünün başarı oranlarının ortalaması olarak hesaplanabilir. Şematik gösterimi Şekil 7'de verilmiştir.



Şekil 7. Şematik olarak K-Fold gösterimi

3.3.1.4. Modelin Değerlendirilmesi

Eğitilen modelin tahmin ettiği çıktı değerlerinin gerçek değerler ile ne kadar uyumlu olduğu eğitimin başarısını belirlemektedir. Gerçekleştirilen makine öğrenmesinin sınıflandırma ya da regresyon problemi olup olmadığı başarının değerlendirilmesinde kullanılan yöntemi de değiştirmektedir. Sınıflandırma problemi için çıktıların gerçek değerler ile olan uyumunu belirlemede karmaşıklık matrisi (confusion matrix) kullanılmaktadır. Şekil 8'de iris veri setinin sınıflandırılması için belirtilen matrisin köşegeni tahmin edilen değerler ile gerçek değerlerin birbiri ile uyuştuğunu göstermektedir. Diyagonal dışarısında kalan değerler ise yanlış sınıflandırılmış olan verileri göstermektedir.



Şekil 8. Örnek bir karmaşıklık matrisi

Sınıflandırmadan farklı olarak regresyon problemlerinde nümerik değerler söz konusu olduğundan tahmin edilen değerlerin gerçek değerlere ne kadar yakın olduğu belirlilik katsayısı (R^2) adı verilen bir hesaplama yöntemi ile belirlenir. Belirlilik katsayısının

mümkün olan en iyi sonucu, tahmin edilen değerler ile gerçek değerlerin birbirine eşit olması, 1'dir. R^2 aşağıda verilen Denklem 15 ile hesaplanabilmektedir.

$$R^2 = 1 - \sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \,/\, \sum_{i=0}^{n}(y_i - \bar{y})^2 \quad (15)$$

Burada;

y_i: Gerçek değerler,

$\hat{y}_i$: Tahmin edilen değerler

$\bar{y}$: Gerçek değerlerin ortalaması,

R^2: Belirlilik katsayısıdır.

Literatürde çok sayıda algoritmanın mevcut olması farklı makine öğrenmesi algoritmalarının da değerlendirilerek veri setine uygun olan modelin belirlenmesini gerektirmektedir. Derin öğrenme gibi yüksek sayıdaki düğüm ve gizli katman sayısına sahip olabilen bir modelde iyi sonuçlar elde edilebilmektedir. Ancak yüksek işlem gücü ve eğitim için gerekli olan uzun süreler ile modelin kullanımı maliyetli hale gelmektedir. Benzer başarı oranlarının yakalanabildiği iki farklı algoritma göz önüne alındığında veri setini en genel hali ile ifade edebilecek en basit model tercih edilmelidir.

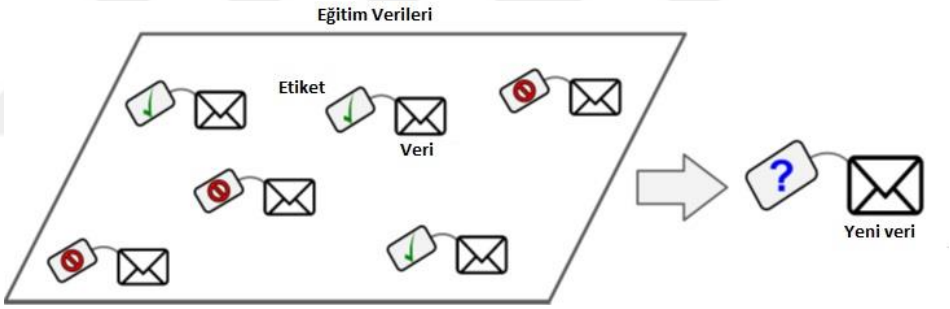
Bir diğer değerlendirme ise modeldeki parametreler ile gerçekleştirilmektedir. Hypertuning de denilen bu işlem bir algoritmanın farklı parametreler (rassal ağaçlar için ağaç sayısı, polinom regresyon için polinomun derecesi vb.) için gösterdiği başarı oranlarının incelenerek optimum konfigürasyonun bulunmasıdır (Chollet, 2018). Örneğin rassal ağaçlar kullanılarak gerçekleştirilen ve 50 ile 25 ağaç sayısı içeren bir çalışma için başarı oranlarının birbirine yakın olduğu kabul edilirse burada tercih edilmesi gereken model, daha basit bir yapıda olması, daha az işlem gücü gerektirmesi sebebiyle 25 ağaç içeren model olmaktadır.

3.3.2. Makine Öğrenmesinin Üç Dalı

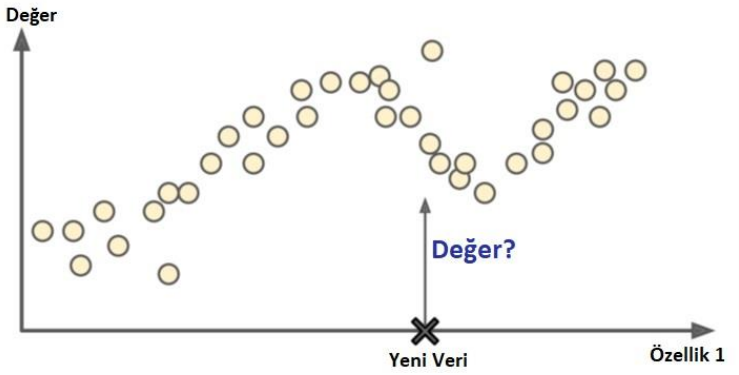
Makine öğrenmesini genel olarak denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli ya da takviyeli öğrenme (reinforcement learning) olarak üçe ayırabiliriz (Fahle vd., 2020).

3.3.2.1. Denetimli Öğrenme

Denetimli öğrenme en yaygın olan makine öğrenmesi yöntemidir (Chollet, 2018). Girdilere karşılık çıktı değerlerinin (genellikle etiketlenmiş veriler denir) bilindiği ve bu bilgiler doğrultusunda algoritmanın eğitildiği yöntemdir. Spam filtreleme gibi sınıflandırma algoritmalarının yanı sıra nümerik bir sayısının tahmin edildiği regresyon algoritmaları da birer denetimli öğrenme algoritmalarıdır (Géron, 2022). Şekil 9'de bir spam filtreleme örneği, Şekil 10'de ise bir regresyon örneği gösterilmiştir.



Şekil 9. Spam filtreleme örneği (Géron, 2022).



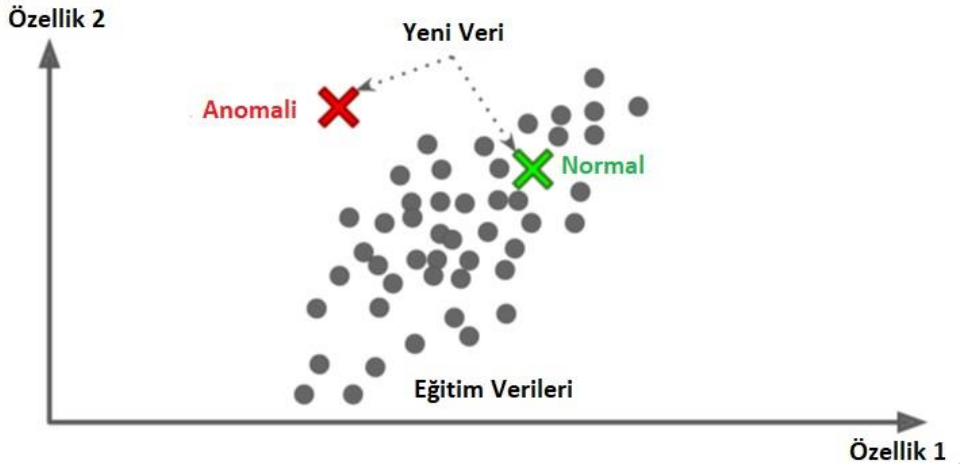
Şekil 10. Bir regresyon örneği (Géron, 2022).

En çok bilinen denetimli öğrenme algoritmalarından birkaçı aşağıda listelenmiştir.

- Doğrusal Regresyon
- Polinom Regresyon
- Karar Ağacı
- Rassal Ağaçlar
- Lojistik Regresyon
- Destek Vektör Makineleri
- Naive Bayes
- Yapay Sinir Ağları

3.3.2.2. Denetimsiz Öğrenme

Etiketlenmiş verilerin olmadığı denetimsiz öğrenmede ya verilerin görselleştirilmesi ya da verilerin birbiri ile olan korelasyonunun daha iyi anlaşılması amaçlanmaktadır (Chollet, 2018). Verilerin kümelenmesi (clustering), anormalliklerin tespiti (anomaly detection) ya da boyut indirgeme (örneğin PCA) gibi algoritmalar denetimsiz öğrenme algoritmalarıdır. Örnek bir anormallik tespiti Şekil 11'de gösterilmiştir.



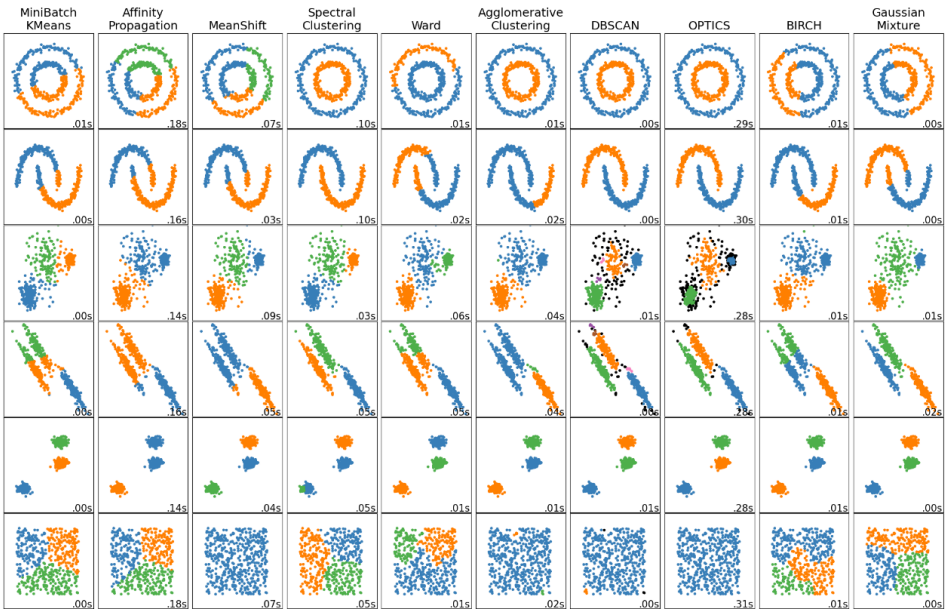
Şekil 11. Anomalilik tespiti (Géron, 2022).

Sıklıkla kullanılan denetimsiz öğrenme algoritmaları aşağıda listelenmiştir (Géron, 2022).

- Kümeleme

  - K-Means
  - Ward
  - DBSCAN
  - OPTICS
  - BIRCH
  - Gaussian Mixture
- Anormallik Tespiti
  - Isolation Forest
- Görselleştirme ve Boyut İndirgeme
  - PCA
  - Locally-Linear Embedding
  - t-distributed Stochastic Neighbor Embedding.

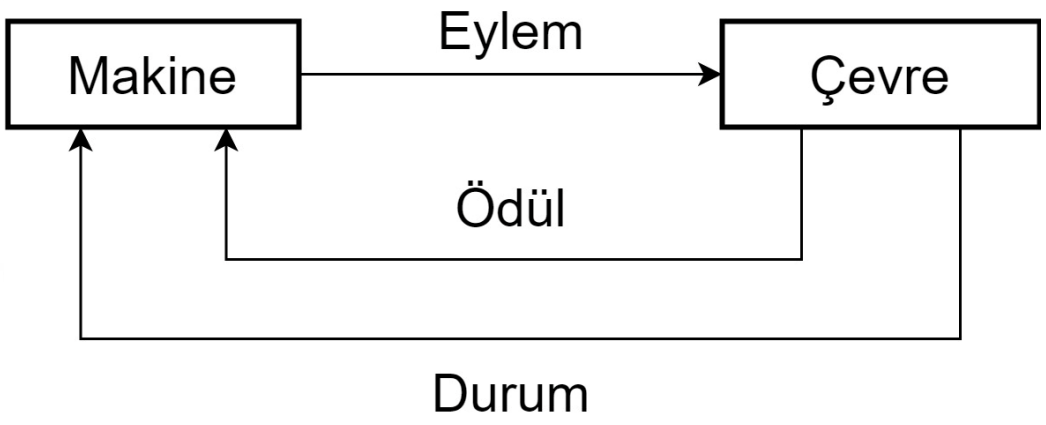
Literatürde çok sayıda kümeleme algoritması bulunmakta ve verilerin dağılıma göre gösterdikleri performanslar değişiklik göstermektedir. Bir veri seti üzerinde aynı başarıyı yakalayan iki algoritmadan hangisinin daha hızlı çalıştığı da önem arz etmektedir. Şekil 12'de on farklı kümeleme algoritmasının altı farklı veri seti için gösterdikleri performanslar belirtilmiştir.



Şekil 12. Kümeleme algoritmaları (Scikit-Learn, 2023).

3.3.2.3. Takviyeli Öğrenme

Takviyeli öğrenme, denetimli ve denetimsiz öğrenmeden farklı olarak alınan bir eyleme karşılık olarak ödül ve ceza sistemi ile çalışan bir yöntemdir. Zaman içerisinde en fazla ödülü almak için politika adı da verilebilen eylemleri seçmeyi makinenin kendisinin öğrenmesidir (Chollet, 2018; Géron, 2022). Oyunlar dışında pek önemli kullanım alanı olmamasına karşın gelecekte robotik, kaynak yönetimi ve eğitim gibi gerçek dünya uygulamalarında sıklıkla kullanılması beklenilmektedir (Chollet, 2018). Hatalardan ders çıkararak öğrenmeye dayalı olan takviyeli öğrenme en genel hali ile Şekil 13'te belirtilmiştir.

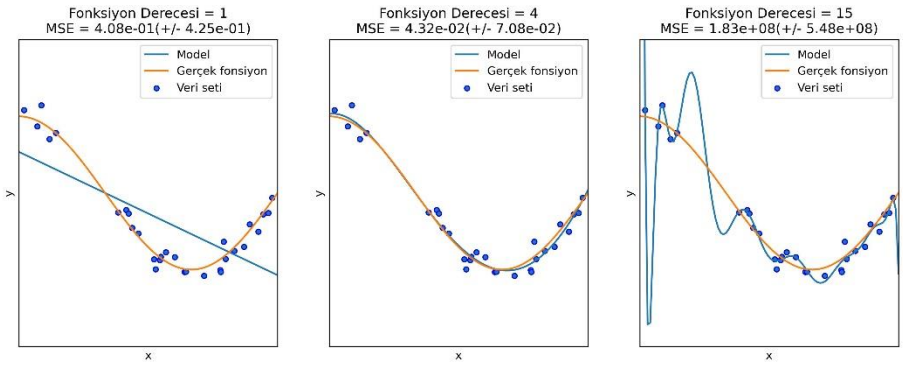


Şekil 13. Örnek bir takviyeli öğrenme

3.3.3. Aşırı Öğrenme ve Öğrenememe

Aşırı öğrenme kurulan modelin eğitim verileri için çok iyi çalışması ancak test verilerinde ise kötü sonuçlar üretmesidir. Yani bir öğrenme işleminden ziyade, modelin eğitim verileri için ezberleme yapmasıdır. Öğrenememe ise aşırı öğrenmenin aksine modelin eğitim verilerinde bile hatalı sonuçlar üretmesidir. Bu, kurulan model çok basit olduğunda karşılaşılan bir durumdur.

Birçok makine öğrenmesindeki asıl problem optimizasyon (makine öğrenmesindeki öğrenme işlemi) ve genelleştirme (modelin daha önce görmediği verilerdeki performansı) arasındaki ikilemdir (Chollet, 2018). Çoğunlukla da bias/varyans ikilemi olarak adlandırılır (Alpaydin, 2020). Eğitim verilerindeki hatayı belirten bias modelin karmaşıklığı, kompleksliği arttıkça azalır ancak bu durum modelde aşırı öğrenmeye neden olabileceğinden test verilerindeki hatanın bir ölçüsü olan varyans artar. Şekil 14'te bir polinom regresyon üzerinden aşırı öğrenme ve öğrenmeme problemi gösterilmiştir.



Şekil 14. Öğrenememe ve aşırı öğrenme (Scikit-Learn, 2023).

Şekil 14'te görüleceği üzere birinci dereceden bir polinom için modelin verileri yeterince öğrenmediği söylenebilir. Tam aksine on beşinci dereceden bir polinom ile gerçekleştirilen makine öğrenmesi işleminde kurulan modelin eğitim verilerini tahmin etmede son derece başarılı olduğu, bias değerinin çok küçük olduğu söylenebilir. Ancak herhangi bir test verisi için varyans değerinin yüksek olacağı yani modelin aşırı öğrenme problemi yaşayacağı görülmektedir. Optimum model, bias ve varyans değerleri arasında bir dengenin olduğu (Alpaydin, 2020) ve veri setini yeterince iyi genelleyebilecek en basit modeldir (Géron, 2022).

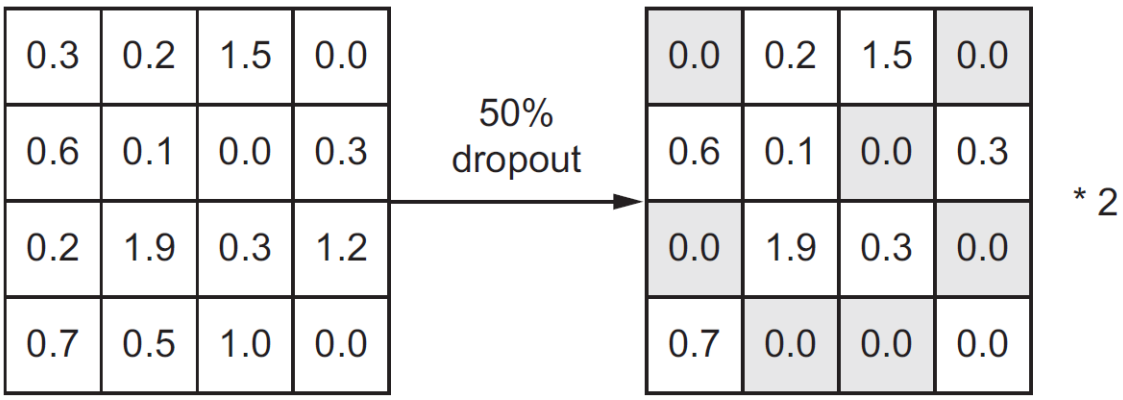
Modeli hatalı öğrenme (aşırı öğrenme ve öğrenememe) işlemlerinden korumanın en kolay yolu daha çok veriye sahip olmaktır (Chollet, 2018). Modelin daha çok veri üzerinde çalışması o veri setini daha iyi genelleştirmesini sağlar. Ancak daha fazla veriye ulaşmak maliyetli ya da mümkün olmayabilir. Bu durumda bir sonraki çözüm ise

modelin depolayabileceği bilgi miktarını (ağaç sayısı, gizli katmanlar veya düğüm sayıları vb.) değiştirmektir (Chollet, 2018). Modelin depolayabileceği bilgi miktarının azaltılması aşırı öğrenmenin önüne geçerek veriyi en genel hali ile ifade etmeye olanak sağlarken yetersiz depolama kapasitesi (çok basit model) eğitim verilerinin öğrenilememesine yol açabilir.

Aşırı öğrenmeden kaçınma yollarından bir diğeri ise ağırlık düzenlemesidir (weight regularization). Modeldeki aşırı öğrenmesi azaltmanın yaygın bir yolu olan ağırlık düzenlemesi, sinir ağının ağırlıklarının küçük değerler almaya zorlayarak ağın karmaşıklığına kısıtlamalar getirir (Chollet, 2018). Ağırlık düzenlemesi sinir ağının ağırlıklandırmalarına sadece eğitim aşamasında uygulanır. Yaygın olarak kullanılan ağırlık düzenlemeleri aşağıda listelenmiştir (Keras, 2023).

- L1
- L2
- L1L2
- Orthogonal Regularizer

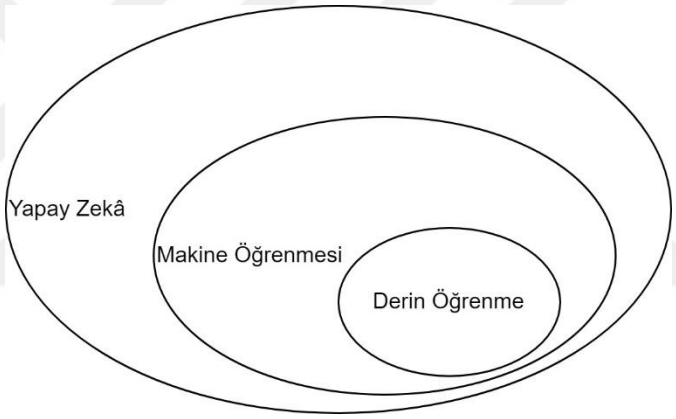
Sinir ağları için en efektif ve en yaygın kullanılan tekniklerden biri *dropout* olarak adlandırılan yöntemdir. Dropout, eğitim sırasında rastgele bir çıktı özelliğinin sıfıra ayarlanarak hesaplamadan çıkarılmasına dayanmaktadır (Chollet, 2018). Bu rastgelelik, sinir ağı düzeninde beklenmedik bir durum oluşturarak ağın ezberden sapmasını sağlar. Bir dropout örneği Şekil 15 ile gösterilmiştir.



Şekil 15. Bir dropout örneği (Chollet, 2018).

3.3.4. Derin Öğrenme

Makine öğrenmesinin özel bir alt dalı olan derin öğrenme (Şekil 16), girdiler ile çıktılar arasındaki ilişkiyi belirlemek için birbiri ardına gelen katmanları kullanır. Derin öğrenmedeki bu katmanlar sinir ağı denilen modeller ile öğrenme işlemini gerçekleştirir (Chollet, 2018). İnsanoğlunun geliştirdiği birçok algoritmada (evrimsel algoritmalar, sürü algoritmaları) ve icatta doğadan ve doğadaki canlılardan ilham alınması gibi sinir ağlarında da canlılardaki biyolojik sinir ağlarından esinlenilmiştir. Ancak derin öğrenme modelleri beynin bir modellenmesi değil verilerden öğrenme işleminin gerçekleştirilmesi için kurulan matematiksek modellemelerdir (Chollet, 2018).



Şekil 16. Yapay zekâ, makine öğrenmesi ve derin öğrenme (Chollet, 2018).

Yapay sinir ağları bir çıktı değerine ulaşmak için birden fazla doğrusal modelin genellemesi olarak tanımlanabilir (A. C. Müller & Guido, 2016).

$$\hat{y} = w_0 * x_0 + w_1 * x_1 + \cdots + w_n * x_n + b \quad (16)$$

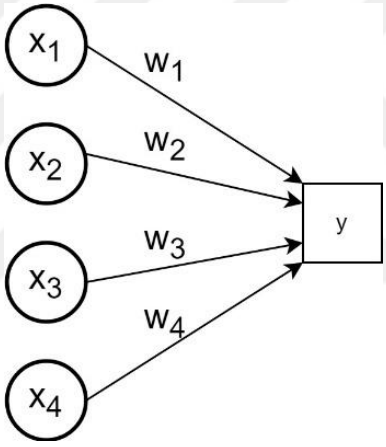
Denklem 16 ile ifade edilen eşitlik bir çoklu doğrusal regresyon denkleminin benzeri olmakla birlikte burada;

X: Girdiler,

ω: Geri besleme (backpropagation) ile güncellenen ağırlıklandırma,

$\hat{y}$: Girdilerin ağırlıklandırılmış toplamı, yapay sinir ağının tahmin değeridir (A. C. Müller & Guido, 2016).

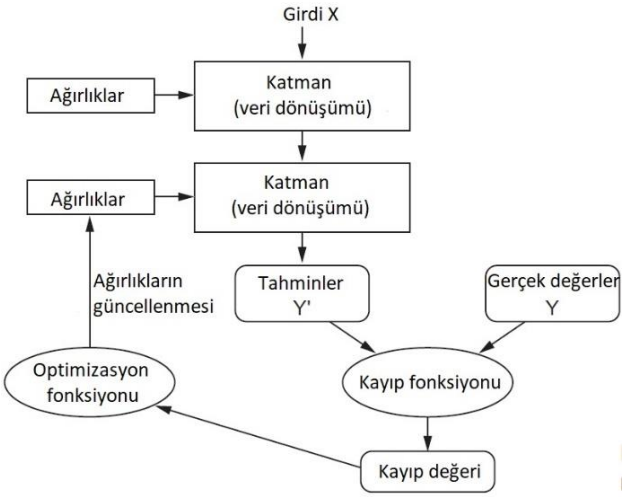
1957 yılında Frank Rosenblatt tarafından bulunan perceptron en basit yapay sinir ağı yapılarından biridir (Géron, 2022). En temel işlem elemanı olan perceptronlar, veri setinden ya da diğer perceptronlardan gelen girdilere sahiptirler (Alpaydin, 2020). Şekil 17'de görüldüğü üzere birbirine bağlanan düğümlerden (node) meydana gelen perceptronlar girdiler ile çıktılar arasındaki ilişkiyi ortaya koyarlar. Bir çıktıya ulaşmak için girdiler ağırlıklandırılarak toplanır ve sinir ağı eğitimi dediğimiz olay, bu ağırlıklandırmaların çıktıları en iyi ifade edecek şekilde optimizasyon fonksiyonu tarafından güncellenmesidir.



Şekil 17. Bir perceptron örneği

Bir sinir ağı eğitimi aşağıda belirtilen nesneler ile gerçekleşmektedir ve bu nesnelerin arasındaki ilişki Şekil 18'de belirtilmiştir.

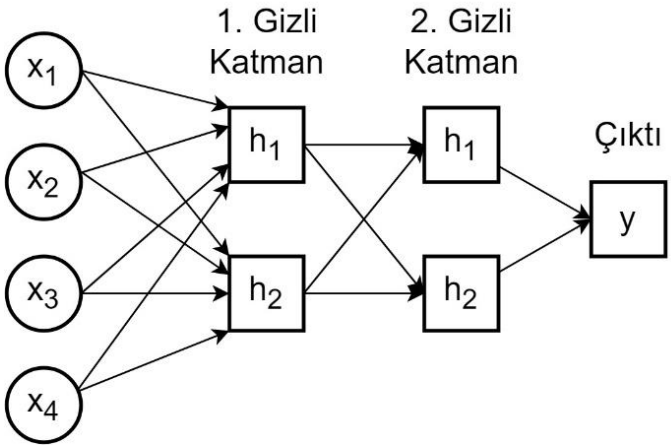
- Birleşerek model veya ağ haline gelen katmanlar
- Girdiler ve bunlara karşılık gelen çıktılar
- Eğitim işlemindeki geri besleme sinyalini oluşturan kayıp fonksiyonu
- Eğitimin nasıl ilerleyeceğini belirleyen optimizasyon fonksiyonu (Chollet, 2018).



Şekil 18. Katman, kayıp ve optimizasyon fonksiyonları (Chollet, 2018).

3.3.4.1. Çok Katmanlı Perceptron

Birden fazla perceptronun birbirine bağlanması ile meydana gelen (Şekil 19) çok katmanlı perceptronlar (multilayer perceptron - MLP), bir girdi ve çıktı katmanı ile bir ya da birden fazla gizli katman içerirler (Gardner & Dorling, 1998). 1990'larda ikiden daha fazla gizli katmana sahip sinir ağları derin ağlar olarak adlandırılırken bugün onlarca hatta yüzlerde gizli katmana sahip sinir ağı ile karşılaşmak mümkündür (Géron, 2022).



Şekil 19. Çok katmanlı bir perceptron

3.3.4.2. Kayıp ve Optimizasyon Fonksiyonları

Yapay sinir ağlarının öğrenme aşamasındaki en önemli nesnelerden biri kayıp fonksiyonudur. Bu fonksiyon sinir ağının eğitiminde doğru yönde hareket edilip edilmediğinin ya da bir başka deyişle belirlenen ağırlıklandırmaların eğitim veri setine uyup uymadığının anlaşılmasını sağlar. Kayıp fonksiyonu, tahminleri hedef (gerçek) değerler ile karşılaştırır ve tahminlerin beklenenle ne kadar uyumlu olduğunun bir ölçütü olan kayıp değerini, eğitim sırasında minimize edilecek olan niceliği (Chollet, 2018), üretir. Birden fazla çıktı olması durumunda kayıp değerleri ortalamaları alınarak birleştirilir ve tek bir nicelik olarak değerlendirilir.

Ele alınan problemdeki çıktı tipine göre (nümerik ya da kategorik) doğru kayıp fonksiyonunun seçilmesi, ağın kayıp değerini minimize etmesi açısından önemlidir (Chollet, 2018). Örneğin, ikili sınıflandırma problemleri için binary cross-entropy, regresyon problemi için ise genellikle ortalama kareler hatası (mean square error) tercih edilir. Sıklıkla kullanılan kayıp fonksiyonlarından bazıları aşağıda listelenmiştir. Fonksiyonlardan bazılarının matematiksel eşitlikleri Denklem 17, 18 ve 19'da verilmiştir.

1. Regresyon
   - Ortalama Kareler Hatası (Mean Square Error)
   - Ortalama Mutlak Hata (Mean Absolute Error)
   - Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error)
   - Ortalama Kareler Logaritmik Hata (Mean Square Logarithmic Error)
2. Sınıflandırma
   - Binary Cross-Entropy
   - Categorical Cross-Entropy

$$Ortalama\ Kareler\ Hatası = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \quad (17)$$

$$Ortalama\ Mutlak\ Hata = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \quad (18)$$

$$Binary\ Cross - Entropy = -[y_i * \log \widehat{y_i} + (1 - y_i) * \log(1 - \widehat{y_i})] \quad (19)$$

Eğitim işleminin en önemli parçalarından biri olan optimizasyon fonksiyonu, ağın kayıp fonksiyonuna bağlı olarak nasıl güncelleneceğini belirler (Chollet, 2018). Doğru kayıp ve optimizasyon fonksiyonunun seçilmesi ağırlıklandırmaların kısa sürede ve yüksek başarı oranıyla güncellenmesini sağlar. RMSProp (Tieleman vd., 2012) ve AdaGrad (Duchi vd., 2011) gibi optimizasyon fonksiyonlarının yanı sıra, Adam (Kingma & Ba, 2014) gibi farklı optimizasyon fonksiyonlarının avantajlarını birleştiren hibrit çözümler de araştırmacılar tarafından geliştirilmiştir.

3.3.4.3. Geri Besleme

Bugün de sinir ağlarının eğitimi için yaygın olarak kullanılan geri besleme (backpropagation) algoritması (Rumelhart vd. 1985) 1985 yılında David Rumelhart, Geoffrey Hinton ve Ronald Williams tarafından geliştirilmiştir. Geri besleme temelde bir ileri ve bir geri olmak üzere iki adımdan oluşmaktadır. Geri besleme algoritması aşağıda belirtildiği üzere kayıp fonksiyonu ve optimizasyon fonksiyonu ile kullanılmaktadır (Géron, 2022).

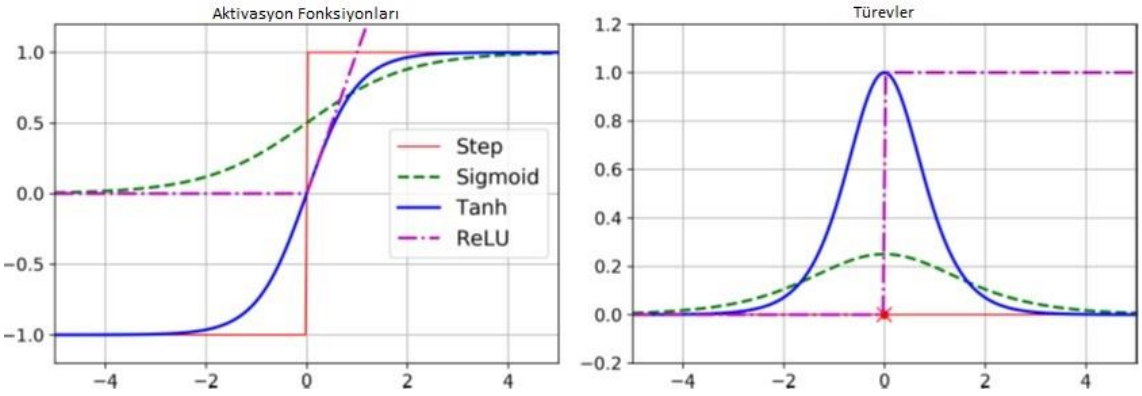
1. Ağın ağırlıklandırmaları rastgele dağıtılır.
2. Ağ eğitim verileri ile beslenir ve son katmana ulaşılana kadar her katmandaki nöronun tahmin değerleri hesaplanır. Bu adım temelde sinir ağı kullanılarak bir tahmin değeri üretmekle aynıdır ancak tüm sonuçlara geri beslemede ihtiyaç duyulacağı için sonuçlar saklanır.
3. Kayıp fonksiyonu, ağın çıktı değerini gerçek değer ile karşılaştırarak bir hata değeri üretir.
4. Her bir nöronun hataya ne kadar katkı sağladığı zincir kuralı kullanılarak hesaplanır.
5. Optimizasyon fonksiyonu geri besleme ile sinir ağının ağırlıklarını günceller.
6. Yeni ağırlıklandırmalar kullanılarak ağ tekrardan eğitim verileri ile beslenir ve yeni çıktılar hesaplanır.
7. Belirtilen epoch sayısına ya da kabul edilen hata miktarına ulaşılana kadar 2-6 arasındaki adımlar tekrarlanır.

Geri beslemedeki önemli noktalardan biri başlangıçta gizli katman bağlantılarındaki ağırlıklandırmaların rastgele olarak dağıtılmasıdır. Ağırlıklandırmalara belirli bir değer

atanması durumunda onlarca nöron ve katmana rağmen katmanlar birebir aynı olacak ve ağ, tek bir nörona sahipmiş gibi davranacaktır (Géron, 2022).

Gradyan tabanlı çalışan geri besleme algoritması için katmanların aktivasyon fonksiyonunun step fonksiyonu ya da benzeri olması durumunda hesaplanabilecek bir gradyan olmadığından çoğunlukla lojistik, hiperbolik veya sigmoid gibi (Şekil 20) fonksiyonlar tercih edilir (Géron, 2022). Sıklıkla kullanılan aktivasyon fonksiyonları Denklem 20'de belirtilmiştir (Keras, 2023).

$$
\begin{aligned}
&Rectified\ Linear\ Unit: ReLU(x) = \max(0, x) \\
&Sigmoid: sigmoid(x) = \frac{1}{1 + \exp(-x)} \\
&Softplus: softplus(x) = \log(exp(x) + 1) \\
&Hiberbolik\ Tanjant: \tanh(x) = \frac{\sinh(\mathrm{x})}{\cosh(x)} = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \\
&Exponensiyel: \exp\,(x)
\end{aligned}
\tag{20}
$$



Şekil 20. Bazı aktivasyon fonksiyonları ve türevleri (Géron, 2022).

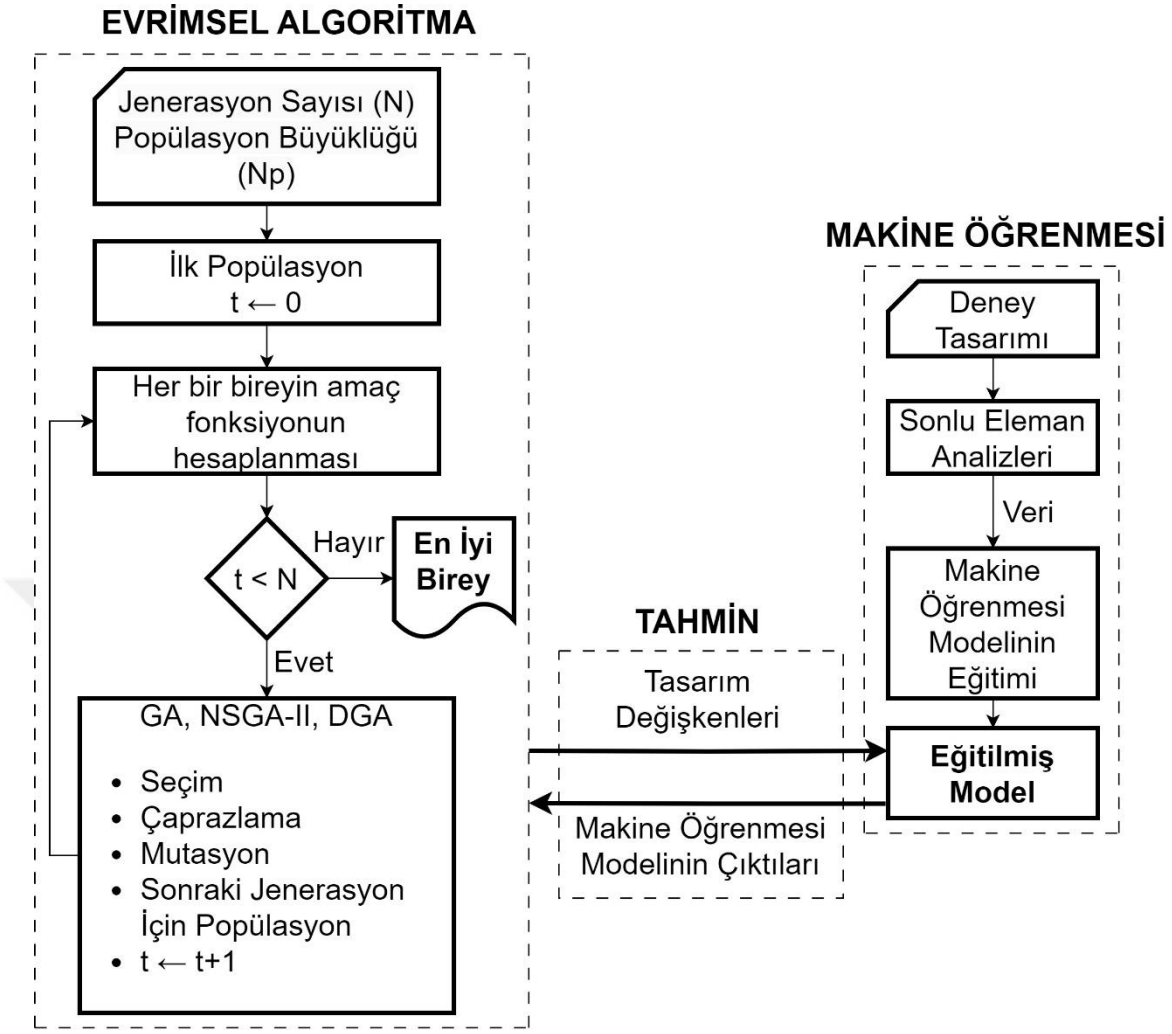
3.4. Yöntem

Bu tezde, tek ve çok amaçlı mühendislik problemleri ile yapıların optimum tasarlanabilmesi ve ürün geliştirme süreçlerine ayrılan kaynakların (deneme yanılma, tecrübeye dayanarak yapılan tasarımlar ve sonlu eleman analizlerine ayrılan süreler vb.) minimize edilmesi için global optimumu bulmayı amaçlayan evrimsel algoritmalar ile

makine öğrenmesi entegre edilerek bir bütünleşik (hibrit) optimizasyon yöntemi geliştirilmiştir. Optimizasyon çalışmalarında doğrudan sonlu elemanlar modeli kullanımı yerine makine öğrenmesinin ürün geliştirme çalışmalarına sağlayacağı faydalar incelenmiştir. Tez çalışmasında optimizasyon algoritması için pymoo (Blank & Deb, 2020) modülünde yer alan GA, NSGA-II ve DGA algoritmaları tercih edilmiştir.

3.4.1. Geliştirilen Yöntem

Önerilen optimizasyon yönteminde ilk olarak Latin Hypercube örneklemi ile alt ve üst limitleri belirli olan tasarım değişkenlerinin homojen dağılmasını sağlanarak tasarım parametreleri belirlenmiştir. Belirlenen parametreler ile sonlu elemanlar analizleri koşturulmuş ve makine öğrenmesinin eğitimi için veri seti eldesi gerçekleştirilmiştir. Elde edilen veri seti makine öğrenmesi modelinin eğitiminde kullanılmış ve modelin test verilerindeki başarısı incelenerek veri setini en genel hali ile ifade edecek en basit model aranmıştır. Veri setini temsil etmede yeterli seviyede başarılı bulunan makine öğrenmesi modeli, optimizasyon çalışmasında amaç ve kısıt fonksiyonlarının belirlenmesi için kullanılmıştır. Önerilen optimizasyon algoritmasına ait akış diyagramı Şekil 21'de görselleştirilmiştir.



Şekil 21. Önerilen optimizasyon algoritmasının akış diyagramı

3.4.2. Tasarımda Yapay Zekâ Uygulamasının Örnek Çalışmaları

Önerilen optimizasyon algoritması önce disk yayların rijitlik optimizasyonunda tek ve çok amaçlı olarak gerçekleştirilmiş, daha sonra kauçuk burçların rijitlik optimizasyonunda kullanılmıştır. Kauçuk malzemenin doğrusal olmayan bir davranış sergilemesi nedeniyle sonlu elemanlar metoduna kıyasla makine öğrenmesi ürün geliştir çalışmalarında büyük avantajlar barındırmaktadır.

Bir diğer uygulama ise elektrikli araç batarya taşıyıcısının kütle minimizasyonu olmuştur. Tasarım hedefi olan eğilme ve burulma rijitlikleri de makine öğrenmesi ile tahmin edilmiş ve bataryanın modülleri güvenli taşımasını sağlayacak tasarım

parametreleri aranmıştır. Tez çalışması kapsamında amaç ve kısıt fonksiyonları için yapay sinir ağlarının kullanıldığı uygulamalar aşağıda listelenmiştir.

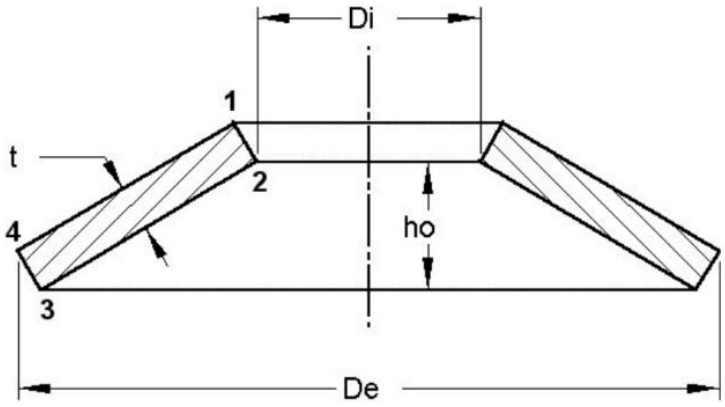
1. Disk yayların tek ve çok amaçlı rijitlik optimizasyonu
2. Kauçuk burçların rijitlik optimizasyonu
3. Elektrikli araç batarya taşıyıcısının kütle minimizasyonu

3.4.2.1. Disk Yay Optimizasyonu

Disk yaylar Şekil 22'de belirtildiği üzere kesitin bir eksen etrafında döndürülmesi ile elde edilebilir. Disk yayların hacminin hesaplanması için Puppes-Gulden teoremi kullanılır ise hacim Denklem 21'de belirtildiği üzere yazılabilir (Kaya & Öztürk, 2002).

$$V = 0{,}7854\sqrt{(D_e + D_i)^2(D_e^2 - 2D_eD_i + D_i^2 + 4h_0^2)t^2} \quad (21)$$

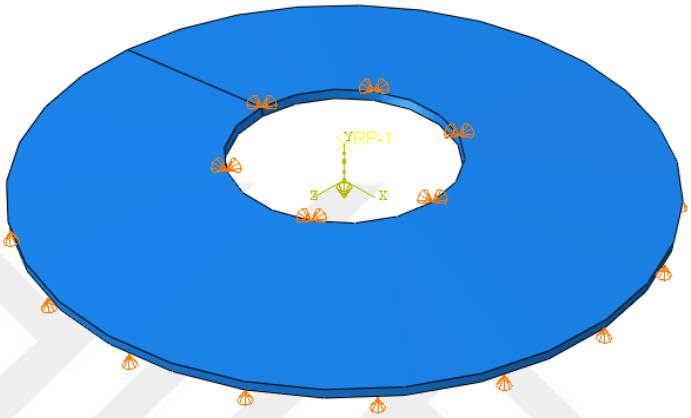
Disk yayların sonlu eleman analizleri için Python programlama dilinde yazılan bir makro ile tasarım değişkenlerinin manuel olarak değiştirilmesi yerine bu işi otomatik olarak gerçekleştirilecek bir kod geliştirilmiştir. Disk yay tasarımı için gerekli olan dört parametre (iç (D_i) ve dış (D_e) çap, kalınlık (t) ve yükseklik (h_0)) yine Şekil 22'de gösterilmiştir.



Şekil 22. Disk yay tasarım parametreleri (Kaya & Öztürk, 2002).

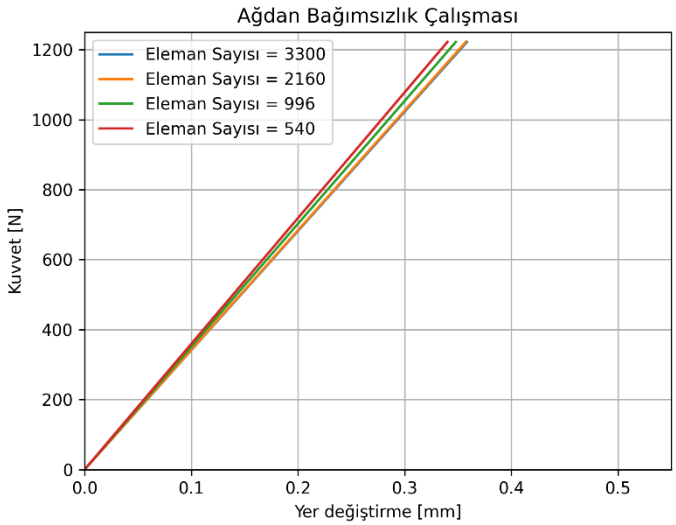
Abaqus/CAE yazılımı ile gerçekleştirilen sonlu elemanlar analizlerinde ele alınan disk yayların malzeme özellikleri için elastisite modülü, Poisson oranı ve akma gerilmesi

sırasıyla 210 GPa, 0.3 ve 450 MPa olarak kabul edilmiştir. Kurulan sonlu elemanlar modelinin sınır şartları Şekil 23'te belirtilmiştir. Burada, disk yayın alt yüzeyin serbestlik derecesi düzey doğrultuda (y), üst yüzeydeki dairesel bölgenin serbestlik dereceleri yatay doğrultuda (x ve z) sıfırlanmıştır. Yayın üst yüzeyinde yer alan dairesel bölge ile referans noktası (RP-1) arasında coupling ilişkisi kurulmuş ve 1222N kuvvet bu referans noktasına uygulanmıştır.



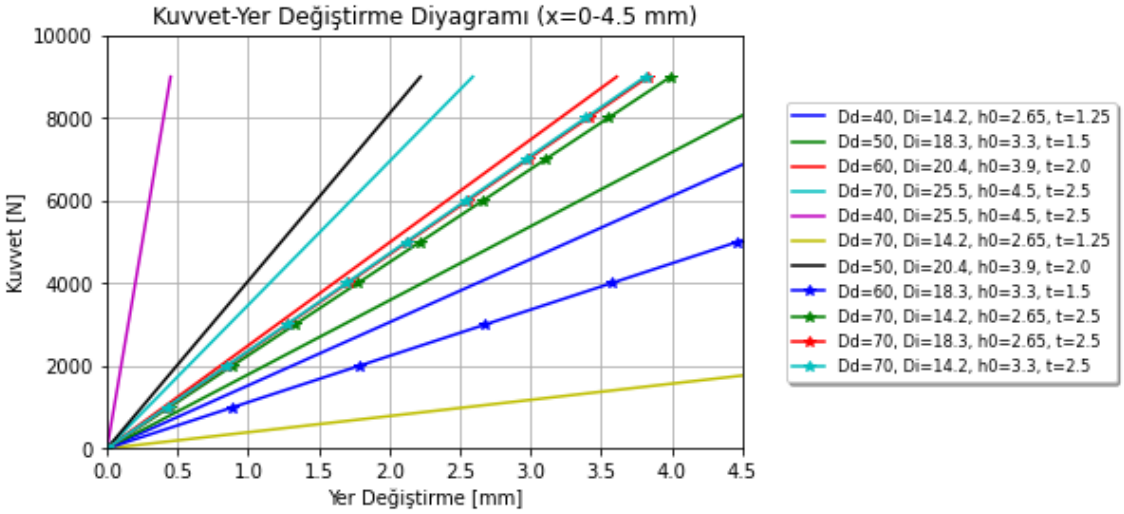
Şekil 23. Disk yay sınır şartları

Disk yayların sonlu elemanlar analizi için kullanılan Abaqus yazılımın öğrenci versiyonu olması nedeni ile düğüm ve eleman sayıları 1000 ile sınırlandırılmıştır. Analizler için eleman tipi hexahedron olarak seçilmiş ve gerçekleştirilen ağdan bağımsızlık çalışması Şekil 24'te gösterilmiştir.



Şekil 24. Ağdan bağımsızlık çalışması

Tez çalışmasında yer alan disk yayların rijitlikleri incelendiğinde (Şekil 25) yayların doğrusal davranış sergilediği görülmektedir. Dolayısıyla Denklem 22 ve 23 ile verilen denklemler yayların herhangi bir yük altındaki deplasmanın ve açığa çıkacak olan gerilmenin hesaplanmasında kullanılabilir.



Şekil 25. Yay rijitlikleri

$$y_y = \frac{\hat{y}}{1222} F_y \tag{22}$$

$$\sigma_y = \frac{\hat{\sigma}}{1222} F_y \quad (23)$$

Burada:

$\hat{y}$: Yapay sinir ağının tahmin ettiği yer değiştirme değeri [mm],

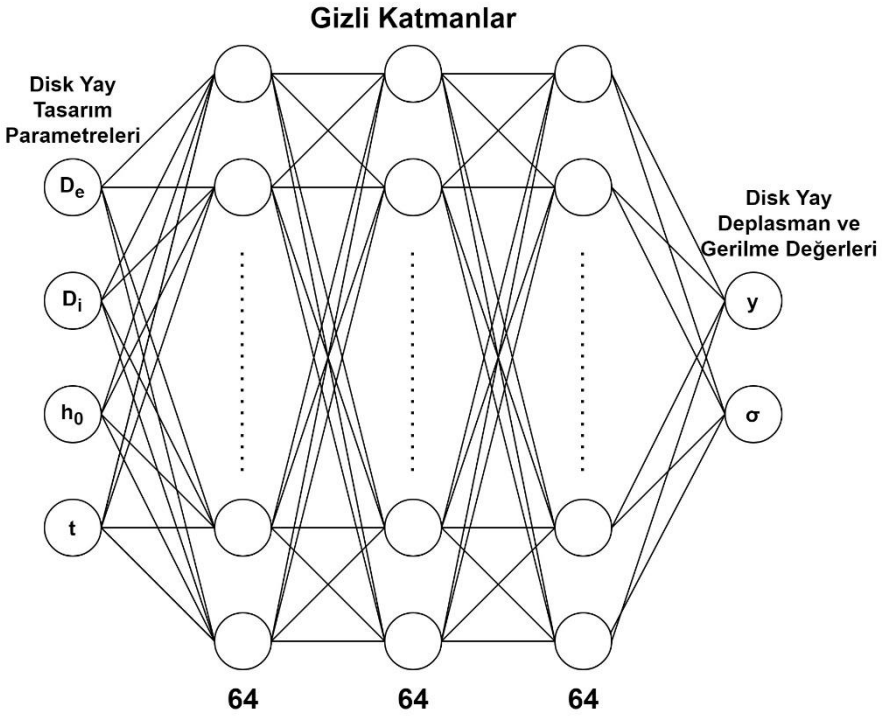
F_y: Tasarım parametreleri aranan yayın yükü [N],

y_y: Hesaplanan yaylanma değeri [mm],

$\hat{\sigma}$: Yapay sinir ağının tahmin ettiği gerilme değeri [N/mm^2],

σ_y: Hesaplanan gerilme değeri [N/mm^2].

Belirlenen dört değişkenin her biri dört kademeye sahip olmak üzere sonlu elemanlar analizleri için toplam 256 adet deney tasarımı oluşturulmuştur. Elde edilen 256 adet verinin %75'i makine öğrenmesi eğitimi için ayrılmıştır. Modelin girdileri tasarım parametreleri (D_e, D_i, t ve h_0), çıktıları ise gerilme ve yer değiştirme değerleri olarak belirlenmiştir. Girdi ve çıktı katmanlarının yanı sıra her biri 64 adet düğüm noktasına sahip üç adet gizli katmanı bulunan olan sinir ağının (Şekil 26) farklı epoch sayılarındaki başarı oranları incelenmiştir. Katmanların aktivasyon fonksiyonu olarak ReLU, optimizasyon fonksiyonu olarak Adam, kayıp fonksiyonu olarak MSE ve başarı metriği olarak MAE seçilmiştir.



Şekil 26. Disk yay için oluşturulan sinir ağı yapısı

Gerçekleştirilen optimizasyon algoritması için pymoo modülünde yer alan GA ve NSGA-II algoritması tercih edilmiştir. Belirtilen algoritmalar için gerekli olan amaç ve kısıt fonksiyonları ile tasarım değişkenleri aşağıdaki denklemlerde belirtilmiştir.

1. Amaç Fonksiyonları:

$$min\, f_1 = 0{,}5\left(y_y - y\right)^2 \tag{24}$$

$$min\, f_2 = V \tag{25}$$

2. Kısıt Fonksiyonları:

$$g_1 = h_0 - y_y \geq 0 \tag{26}$$

$$g_2 = \sigma_{Akma} - \sigma_y \geq 0 \tag{27}$$

3. Tasarım Değişkenleri:

$$40 \leq D_e \leq 70\ [mm] \tag{28}$$

$$14{,}2 \leq D_i \leq 25{,}5\ [mm] \tag{29}$$

$$2{,}65 \leq h_0 \leq 4{,}5\ [mm] \tag{30}$$

$$1{,}25 \leq t \leq 2{,}5\ [mm] \tag{31}$$

Burada:

y: Tasarım parametreleri aranan yayın yaylanma miktarı [mm],

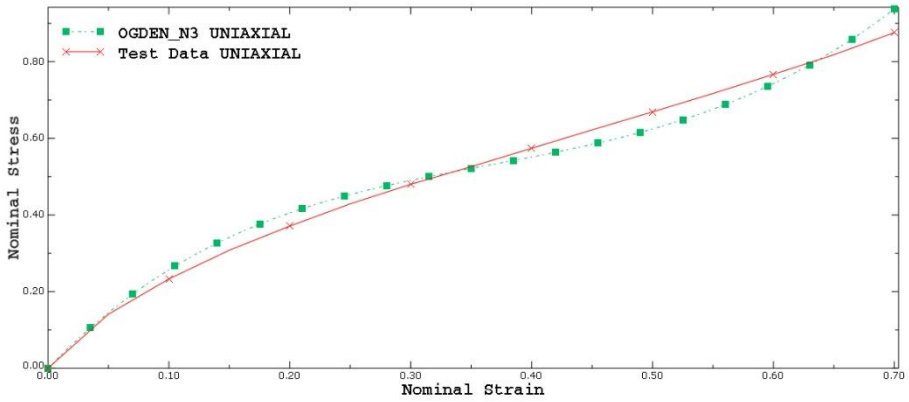
f_1: Yapay sinir ağının tahmin ettiği yaylanma miktarı ile istenilen yaylanma miktarı arasındaki uyum, minimize edilecek olan rijitlik fonksiyonu [mm],
f_2: Denklem 21 ile hesaplanıp minimize edilecek olan hacim fonksiyonudur [mm].

3.4.2.2. Kauçuk Burç Optimizasyonu

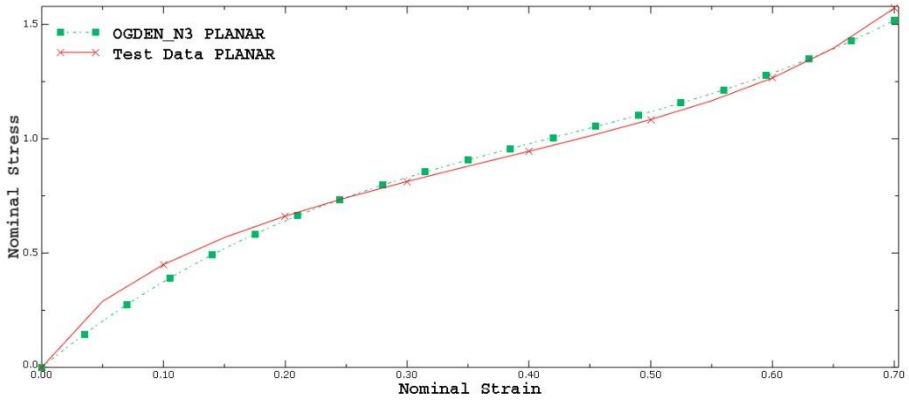
Otomotiv endüstrisinde konfor gereksinimlerini karşılamak için kullanılan kauçuk burçların (Kaya, 2014) ana görevi yoldan ve motordan gelen titreşimleri sönümlemek ve ses izolasyonunu sağlamaktır (Blundell, 1998; Kadlowec vd., 2003). Otomobillerin sürüş performansları süspansiyon sitemine bağlı olduğundan kauçuk burçların arzu edilen rijitlikle tasarlanmalı üreticiler için önem arz etmektedir (Kaldas vd., 2014).

Kauçuk, doğrusal davranış göstermeyen bir hiper elastik malzeme olarak ele alınabilir ve gerilme ve deformasyon arasındaki ilişkinin bilinmesi sonlu eleman analizlerinin gerçekleştirilmesi için gereklidir. Bu sebeple tek eksenli (uniaxial) ve düzlem (planar) test düzenekleri ile ele alınan kauçuk malzemenin davranışı incelenmiştir. Elde edilen veriler Abaqus yazılımı kullanılarak hiper elastik malzeme modelleri ile karşılaştırılmıştır. Malzeme modelleri içerisinden stabil olan ve deneysel veriler ile en iyi uyumu sağlayan üçüncü dereceden Ogden malzeme modeli çalışma için uygun bulunmuştur (Şekil 27).

a) Tek Eksenli Çekme

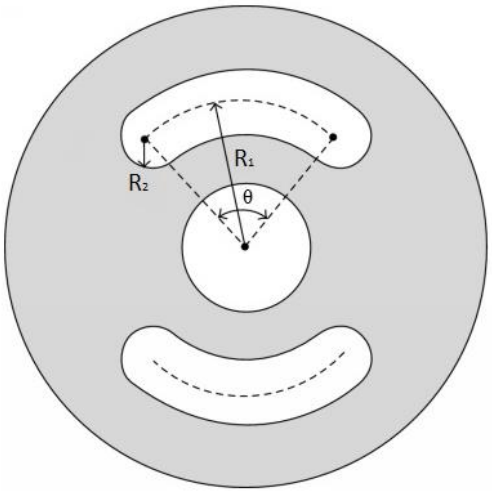


b) Düzlem Çekme



Şekil 27. Test verileri için Ogden N=3 malzeme modeli

Gerçekleştirilecek olan sonlu elemanlar analizi için kauçuk burcun tasarım parametreleri ve parametre limitleri Şekil 28'de verilmiştir. Şekil parametreleri otomatik olarak değiştirilen kauçuk burcun tasarımını koruması için belirtilen parametre limitlerine ihtiyaç duyulmuştur. Kauçuk burç için dış yarıçap, iç yarıçap ve kalınlık sırasıyla 90, 15 ve 30 mm olmak üzere çalışma boyunca sabit kabul edilen değerlerdir.



Tasarım parametreleri:

$30 \leq R_1 \leq 65 \ [mm]$

$5 \leq R_2 \leq 15 \ [mm]$

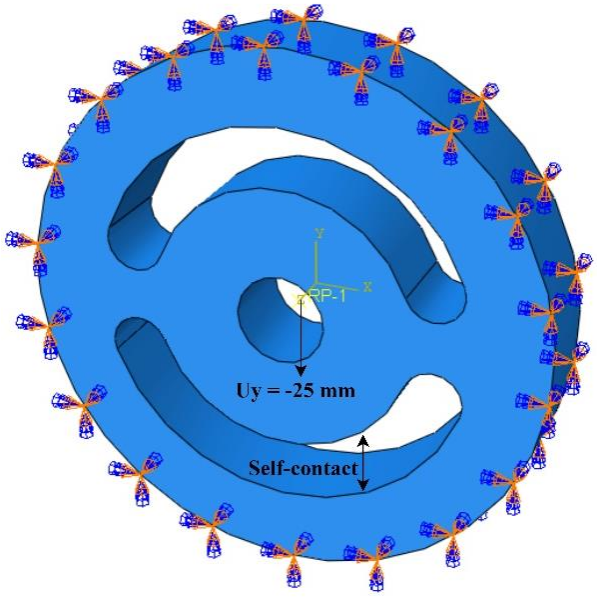
$10 \leq \theta \leq 140 \ [°]$

$R_1 + R_2 \leq 70 \ [mm]$

$R_1 - R_2 \geq 25 \ [mm]$

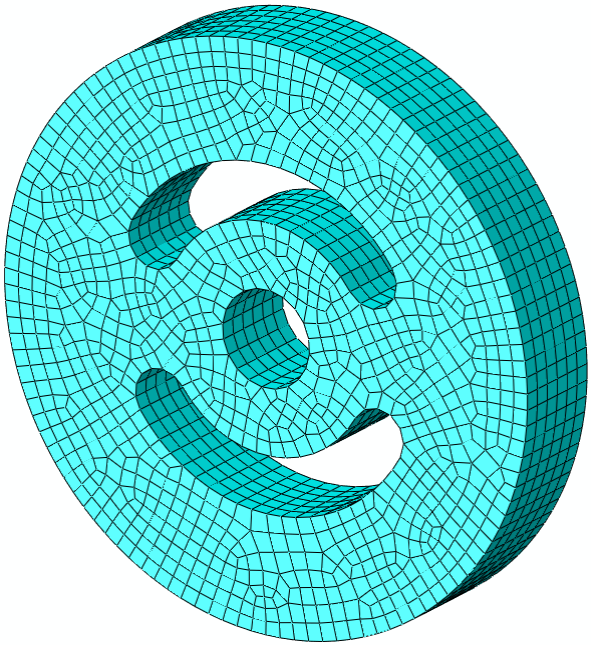
Şekil 28. Kauçuk burç tasarım parametreleri ve sınır şartları

Şekil 29'da de görüleceği üzere burcun dış yüzeyi tüm serbestlik dereceleri için sabitlenmiş ve burcun merkezinden 25 mm yer değiştirme düşey doğrultuda uygulanmıştır. Gerçekleştirilen analizin yüksek deformasyon içermesi nedeniyle burcun alt yüzeyinde bir temas tanımlaması yapılmıştır.

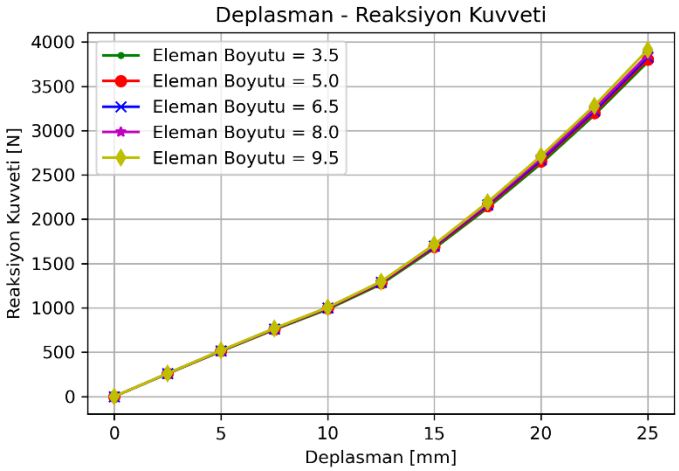


Şekil 29. Kauçuk burç sınır şartları

Sonlu eleman analizleri için kurulan ağ yapısı yaklaşık 9000 adet üç boyutlu eleman (C3D8RH) içermektedir ve örnek bir burcun ağ yapısı Şekil 30'da gösterilmiştir. Gerçekleştirilen ağdan bağımsızlık çalışması neticesinde uygun eleman boyutu 5 mm olarak seçilmiştir. Şekil 31 incelendiğinde bu eleman boyutunun kabul edilebilir doğruluğa sahip olduğu görülmektedir.

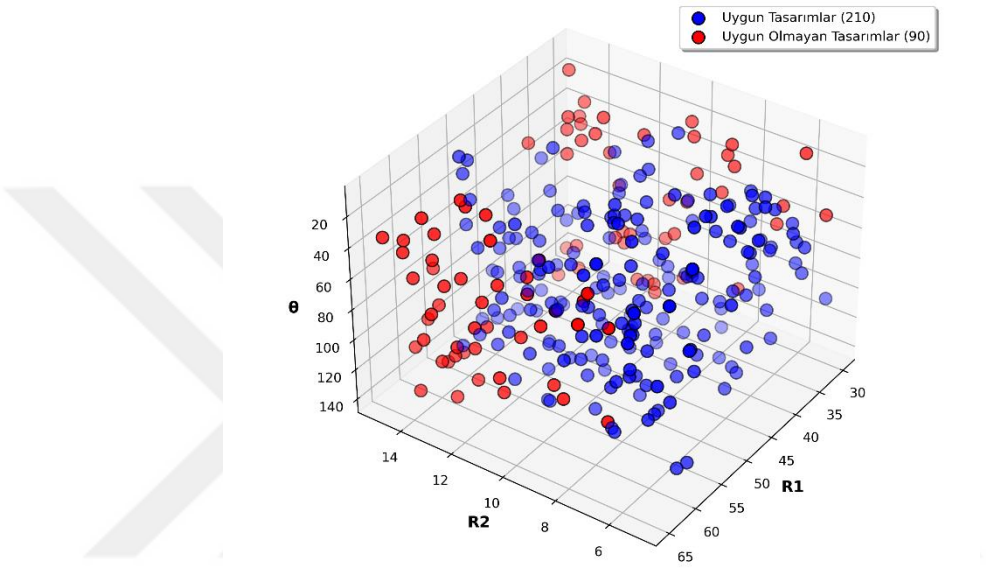


Şekil 30. Kauçuk burç mesh yapısı



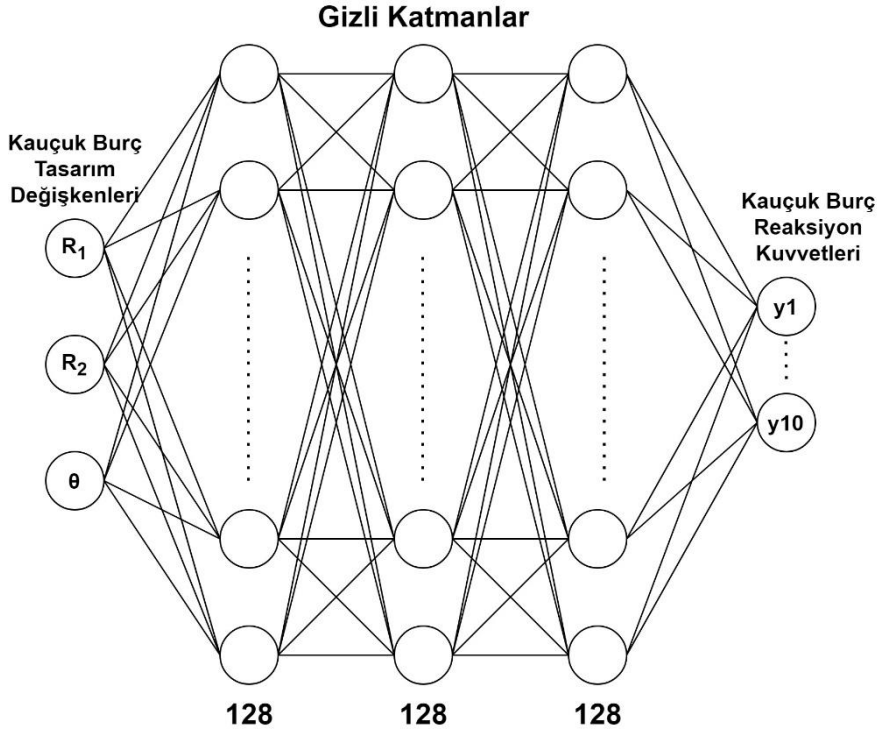
Şekil 31. Kauçuk burç ağdan bağımsızlık çalışması

Sinir ağı eğitimi için gerekli olan veriler sonlu elemanlar analizleri ile elde edilmiştir. Python programlama dili kullanılarak, pyDOE modülü içerisinde yer alan, Latin Hypercube Örneklemesi ile 300 adet tasarım oluşturulmuş ve tasarım kısıtlarını sağlayan 210 analiz doğrusal olmayan sonlu eleman analizleri için uygun bulunmuştur (Şekil 32). Gerçekleştirilen 210 analiz tasarım değişkenlerini ve reaksiyon kuvvetlerini içeren bir CSV (comma-separated value) dosyasına kaydedilmiştir.



Şekil 32. Kauçuk burç Latin Hypercube Örneklemesi

Yapay sinir ağı eğitimi için sonlu eleman analizleri ile elde edilen veri setinin üçte ikisi kullanılmıştır. Burcun tasarım parametreleri (R_1, R_2, θ) sinir ağının girdileri ve on adet reaksiyon kuvveti çıktıları olarak belirlenmiştir. Her biri 128 adet düğüm noktasına sahip üç gizli katmanı bulunan sinir ağı farklı epoch sayılarınca eğitilmiş ve çalışma için en uygun epoch sayısı aranmıştır. Çalışma için kurulan sinir ağı Şekil 33'te gösterilmiştir. Çalışmada, katmanların aktivasyon fonksiyonu ReLU (rectified linear unit), optimizasyon fonksiyonu Adam, kayıp fonksiyonu MSE ve başarı metriği ise MAE olarak belirlenmiştir.



Şekil 33. Çalışma için kurulan YSA modeli

Çalışmada sinir ağının tahmin ettiği reaksiyon kuvvetleri (F_{pred}) ile tasarım parametreleri aranan hedef reaksiyon kuvvetleri (F_{target}) arasındaki farkın ölçümü istatistiksel bir ifade olan Ki-Kare ile ölçülmüştür (Denklem 32). Arzu edilen rijitliğe ulaşmak için ki-kare değerinin mümkün olduğunca (en iyi durumda 0) olması beklenmektedir.

$$Ki - kare = \sum_{i=1}^{n} \frac{\left(F_{pred_i} - F_{target_i}\right)^2}{F_{target_i}} \tag{32}$$

Optimizasyon problemi için pymoo modülü içerisinde yer alan genetik algoritma tercih edilmiştir. Popülasyon büyüklüğü ve jenerasyon sayısı sırasıyla 40 ve 80 olarak belirlenmiş ve mutasyon, çaprazlama ve seçim parametreleri fonksiyon için varsayılan değerleri ile kullanılmıştır. Çalışma için gerekli olan amaç ve kısıt fonksiyonları ile tasarım parametreleri aşağıdaki denklemler ile belirtilmiştir.

1. Amaç Fonksiyonu:

   $$min\, f = Ki - kare \tag{33}$$

2. Kısıt Fonksiyonları:

   $$g_1 = 70 - R_1 + R_2 \geq 0 \tag{34}$$

$$g_2 = R_1 - R_2 - 25 \geq 0 \quad (35)$$

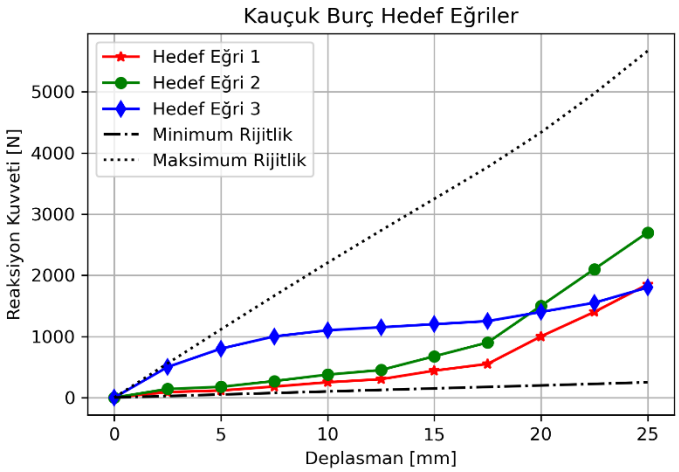
3. Tasarım Değişkenleri:

$$30 \leq R_1 \leq 65 \ [mm] \quad (36)$$

$$5 \leq R_2 \leq 15 \ [mm] \quad (37)$$

$$10 \leq \theta \leq 140 \ [°] \quad (38)$$

Kauçuk burç optimizasyon çalışmasında rijitlik limitleri içerisinde yer alan üç farklı durum için hedef rijitlikler belirlenmiştir (Şekil 34). Her bir çalışma için arzu edilen rijitleri sağlayan tasarım parametreleri aranmıştır.



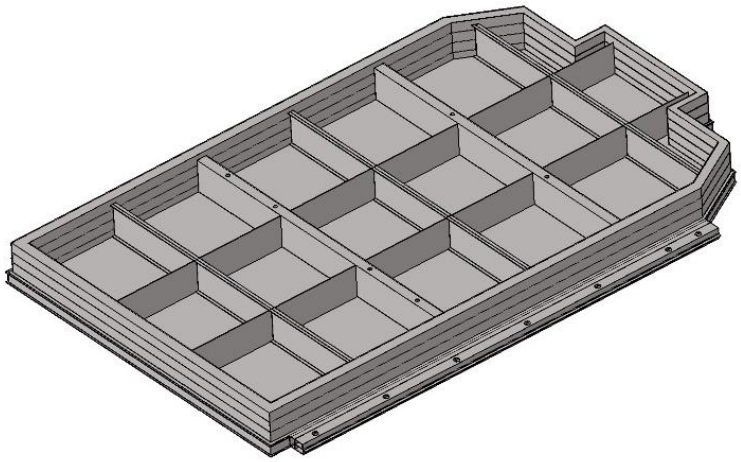
Şekil 34. Hedef rijitlik eğrileri

3.4.2.3. Elektrikli Araç Batarya Taşıyıcısı Optimizasyonu

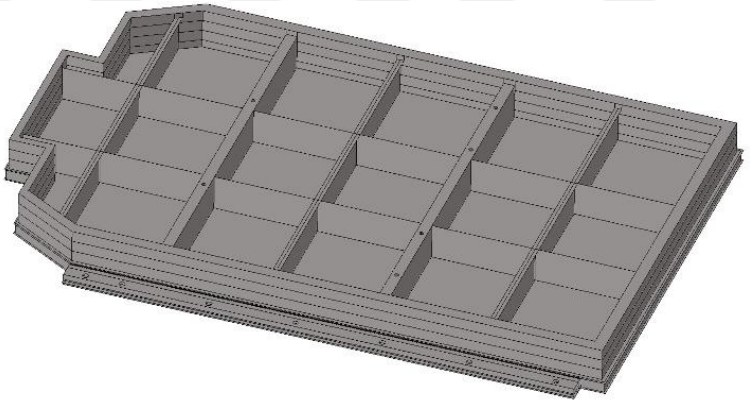
Elektrikli araçlarda kullanılan batarya taşıyıcı tasarımının, eğilme ve burulma rijitliği ile çarpma dayanıklılığı gibi belirli sınır koşulları altında istenen işlevleri yerine getirmesi beklenir. Günümüzde, sürdürülebilir ulaşım çözümlerine olan talebin yanı sıra otomobillerin hafif olması hem yakıt hem de emisyon açısından önem arz etmektedir. Bu çalışmada ise bir batarya taşıyıcısının eğilme ve burkulma kısıtlamaları altında yapısal optimizasyonu gerçekleştirilmiştir.

Taşıyıcı, modüllerin yapıya yerleştirilmesi ve herhangi bir dış yük altında modülleri hasar görmekten koruması için destekleyici yapılar kullanılarak kutu şeklinde

tasarlanmıştır. Şekil 35'te gösterilen batarya taşıyıcısı tasarımı için SolidWorks yazılımı kullanılmış ve orta yüzeyler (mid surface) Hypermesh programı kullanılarak elde edilmiştir (Şekil 36).

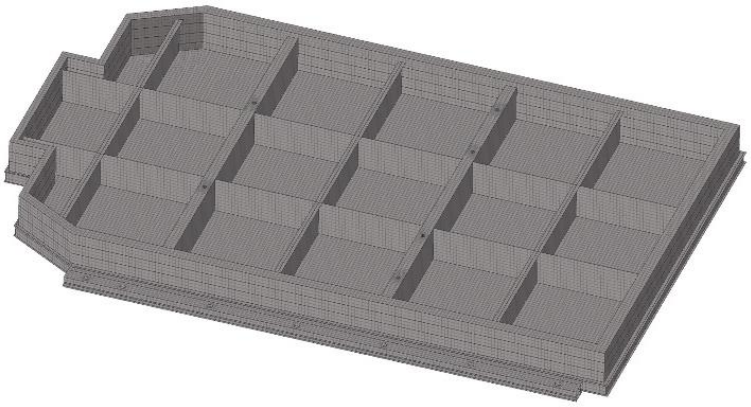


Şekil 35. Batarya taşıyıcısı katı modeli



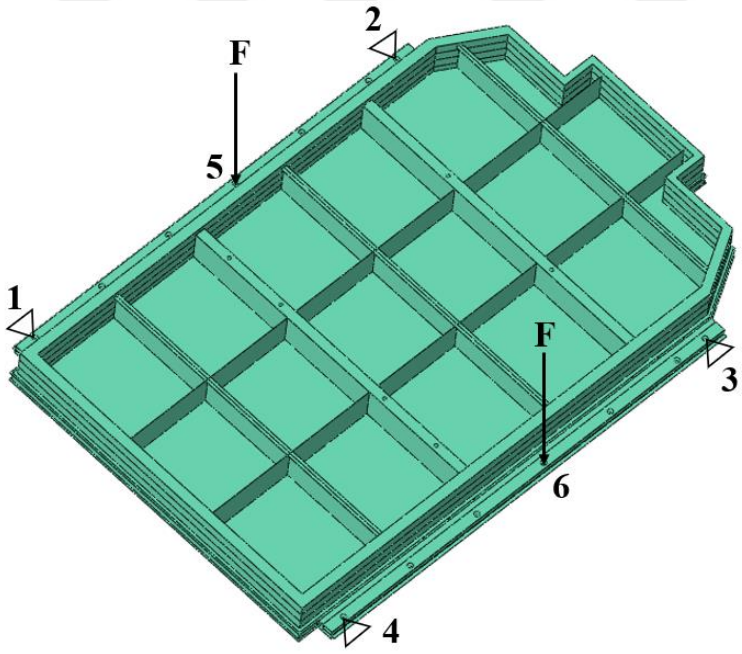
Şekil 36. Orta yüzeyler

Ağ yapısının oluşturulması için yine Hypermesh yazılımı kullanılmıştır. Model, yapının izin verdiği ölçüde iki boyutlu yüzey elemanlara bölünmüştür. Eleman boyutu 10 mm olarak seçilmiş ve 70362 eleman ile 65895 düğüm içermektedir (Şekil 37). Model Abaqus yazılımda sonlu elemanlar analizlerini gerçekleştirmek üzere Nastran dosyası (.bdf) olarak kaydedilmiştir.



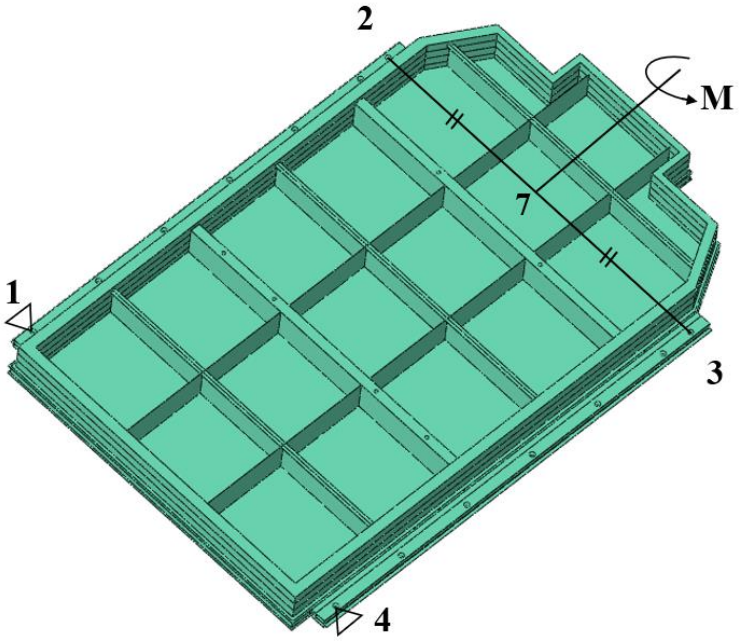
Şekil 37. Taşıyıcının sonlu elemanlar modeli

Nastran dosyası Abaqus yazılımına aktarılmış ve tüm sınır şartları ile malzeme de burada tanımlanmıştır. Şekil 38'de gösterildiği üzere 1, 2, 3 ve 4 noktalarından tüm serbestlik dereceli sabitlenen ve 5 ile 6 noktalarından 2500 N, toplamda 5000 N, yük uygulanan bataryanın taşıyıcısının eğilme analizleri gerçekleştirilmiştir.



Şekil 38. Eğilme sınır şartları

Burulma analizi için ise taşıyıcı, 1 ve 4 noktalarından hareketi kısıtlanmış (DOF 123) ve 2 ile 3 noktaları arasında rijit çubuk tanımı kullanılarak 7 noktasından 600 Nm moment uygulanmıştır (Şekil 39).



Şekil 39. Burulma sınır şartları

Alüminyum alaşımlarının hafif olmaları sebebiyle birçok çalışmada kullanıldığı gözlemlenmiştir. Bu çalışmada da malzeme olarak 6063-T6 alüminyum alaşımı tercih edilmiştir. Malzemenin elastisite modülü ve Poisson oranı sırasıyla 70 GPa ve 0.3 olarak kabul edilmiştir (Çizelge 2).

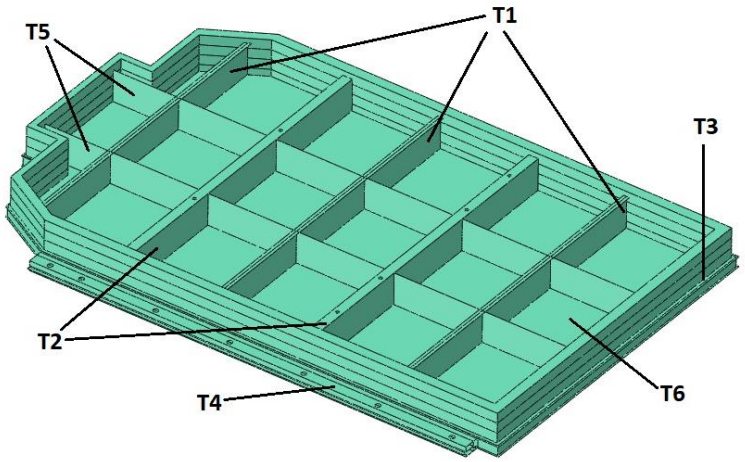
Çizelge 2. Alüminyum alaşımı mekanik özellikleri

6063-T6	
Yoğunluk	2.7 gr/cm^3
Elastisite Modülü	70000 MPa
Poisson Oranı	0.35
Akma Gerilmesi	250 MPa

Yapay sinir ağı eğitimi için gerekli olan veri seti sonlu eleman analizleri ile elde edilmiştir. Bu çalışmada pyDOE modülünde yer alan Latin Hiper küp Örneklemesi

kullanılmış ve Python programlama dili kullanılarak 300 tasarım belirlenmiştir. Kalınlık değerlerinin orta yüzeyler kullanılarak kontrol edilmesi daha kolay olduğundan, sonlu elemanlar analizi için Abaqus/CAE'nin kodlama altyapısı kullanılarak Python programlama dili ile kalınlık değerleri değiştirilerek analizler gerçekleştirilmiştir.

Hem eğilme hem de burulma analizleri için kullanılan 300 tasarım ile toplamda 600 analiz yapılmıştır. Sonuçlar ile kalınlıklar bir CSV dosyasına yazdırılmıştır. CSV dosyası çalışmada tasarım parametreleri olan 6 kalınlık değerini, minimize edilecek olan kütleyi ve eğilme ve burkulma analizi sonuçlarını içermektedir. Elektrikli araç batarya taşıyıcısının tasarım parametreleri Şekil 40'ta belirtilmiştir.

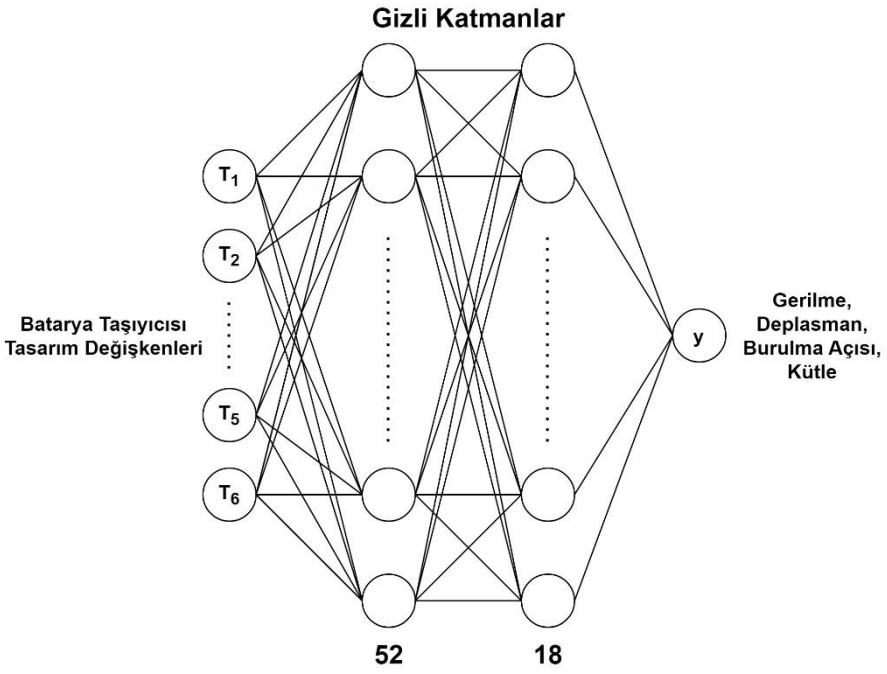


Şekil 40. Batarya taşıyıcısı tasarım prametreleri

Çalışmada, kütle, eğilme ve burulma sonuçlarını tahmin etmek için vekil model yerine makine öğrenmesinin bir alt dalı olan yapay sinir ağları tercih edilmiştir. 6 girdi, sırasıyla 52 ve 18 adet düğüme sahip iki gizli katman ve bir çıktıya sahip sinir ağı (Şekil 41) eğitimi için elde edilen veri setinin üçte ikisi kullanılmıştır. Batarya taşıyıcısının kalınlıkları sinir ağının girdileri, analiz sonuçları ise çıktılar olarak belirlenmiştir.

Makine öğrenmesinde yaygın bir problem olan aşırı öğrenmeden kaçınmak için iki gizli katmandan sonra L2 ağırlık düzenlemesi (gizli katmanın ağırlık matrisindeki her bir değişken 0,001 ile çarpılarak ağırlıklandırmalar küçültülür) uygulanmıştır. Katmanların

aktivasyon fonksiyonu için ReLU, optimizasyon fonksiyonu için Adam, kayıp fonksiyonu için MSE ve başarı metriği için ise MAE tercih edilmiştir.



Şekil 41. Batarya taşıyıcısı için kurulan sinir ağı yapısı

Optimizasyon çalışması için pymoo modülünde yer alan ve popülasyon tabanlı bir evrimsel algoritma olan DGA tercih edilmiştir. DGA algoritması stratejileri içerisinden DE/rand/1/bin tercih edilmiş ve popülasyon büyüklüğü, jenerasyon sayısı ve çaprazlama oranı sırasıyla 60, 40 ve 0.3 seçilmiştir. Diğer parametreler için modülde tanımlanan varsayılan değerler kullanılmıştır. Gerçekleştirilen boyut optimizasyonunun amacı arzu edilen rijitlikler altında (Denklem 40 ve 41) taşıyıcı için minimum kütleyi bulmaktır.

Elektrikli araç batarya taşıyıcısının kütle minimizasyonu için boyut optimizasyonu aşağıda belirtildiği gibi tanımlanmıştır:

1. Amaç Fonksiyonu:

$$min\,kütle \tag{39}$$

2. Kısıt Fonksiyonları:

$$Eğilme\ Rijitliği\ \geq 5000\ N/mm \tag{40}$$

$$Burulma\ Rijitliği\ \geq 600\ Nm/° \tag{41}$$

3. Tasarım Değişkenleri:

$$1 \leq T_1 \leq 5\ [mm] \quad (42)$$

$$1 \leq T_2 \leq 5\ [mm] \quad (43)$$

$$1 \leq T_3 \leq 5\ [mm] \quad (44)$$

$$1 \leq T_4 \leq 5\ [mm] \quad (45)$$

$$1 \leq T_5 \leq 5\ [mm] \quad (46)$$

$$1 \leq T_6 \leq 5\ [mm] \quad (37)$$

4. BULGULAR VE TARTIŞMA

Bu bölümde Şekil 21 ile akış diyagramı verilen yöntem kullanılarak disk yayın, kauçuk burcun ve elektrikli araç batarya taşıyıcısının optimizasyon çalışmaları gerçekleştirilmiş, optimizasyon sonuçları sunulmuş ve sunulan yöntemin avantajları incelenmiştir.

4.1. Disk Yay

4.1.1. Disk Yayların YSA ile Modellenmesi

Yapay sinir ağının farklı epoch sayılarındaki başarıları R^2, MAE ve MSE metriklerini içeren Çizelge 3-5'te belirtilmiştir. Çizelgede 1500 epoch eğitilen sinir ağının test verileri için %97,67 R^2 başarı oranı ile öne çıktığı görülmektedir. 2000 epoch eğitilen sinir ağında ise aşırı öğrenme problemi ile karşılaşılmıştır. Çizelge 6'da ise yapay sinir ağlarının eğitimi için gerekli olan zaman saniye biriminden belirtilmiştir.

Çizelge 3. R^2 belirlilik katsayısı

Epoch	Eğitim Verileri		Test Verileri	
	Gerilme	Yer Değiştirme	Gerilme	Yer Değiştirme
500	%99,74	%67,93	%99,53	%54,86
1000	%99,85	%94,46	%99,80	%90,37
1500	%99,95	%98,39	%99,86	%97,67
2000	%99,93	%86,88	%99,78	%78,18

Çizelge 4. Ortalama mutlak hata (MAE)

Epoch	Eğitim Verileri		Test Verileri	
	Gerilme	Yer Değiştirme	Gerilme	Yer Değiştirme
500	3,3538	0,5838	4,3357	0,4924
1000	2,6545	0,1929	3,0746	0,2191
1500	1,2667	0,0882	2,3387	0,0901
2000	1,3201	0,2882	3,1812	0,2993

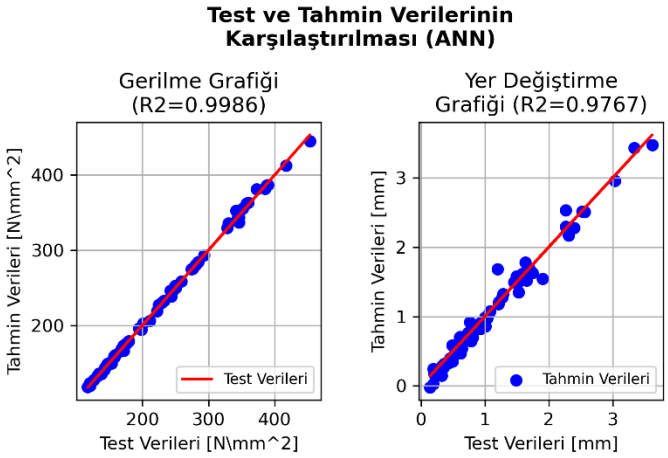
Çizelge 5. Ortalama kareler hatası (MSE)

Epoch	Eğitim Verileri		Test Verileri	
	Gerilme	Yer Değiştirme	Gerilme	Yer Değiştirme
500	3,3538	0,5838	4,3357	0,4924
1000	2,6545	0,1929	3,0746	0,2191
1500	1,2667	0,0882	2,3387	0,0901
2000	1,3201	0,2882	3,1812	0,2993

Çizelge 6. Sinir ağı eğitimi için gerekli olan zaman

Epoch	Zaman [s]
500	68,53
1000	138,98
1500	196,62
2000	271,91

1500 epoch eğitilen sinir ağı için hem eğitim hem de test verilerinde yüksek ve birbirine yakın başarı oranlarının yakalanmıştır. Bu, eğitilen modelde bir aşırı öğrenme problemi ile karşılaşılmadığını göstermektedir. Yapay sinir ağının tahmin ettiği değerler ile gerçek değerlerin karşılaştırması Şekil 42'de gösterilmiştir.



Şekil 42. 1500 epoch için test ve tahmin değerlerinin karşılaştırılması

Belirtilen başarı metrikleri ile Şekil 42 incelendiğinde 1500 epoch eğitilmiş ağın, optimizasyon çalışmasında gerilme ve yer değiştirme değerlerinin tahmininde kullanılması için yeterli seviyede başarıya sahip olduğu görülmektedir.

4.1.2. Disk Yay Rijitlik Optimizasyonu

4 farklı yay tasarımı için GA kullanılarak gerçekleştirilen tek amaçlı optimizasyon çalışmasının sonuçları Çizelge 7'de belirtilmiştir. 3. yay hariç diğer yayların amaç fonksiyonlarının hataları 3E-08 mm'den daha küçük olmak üzere optimizasyon çalışması başarılı sonuçlar üretmektedir. 3. yay ise gerilme sınırına takılarak 449,613 MPa ile optimum sonucuna ulaşmıştır. Dört çalışma için de 100 popülasyon 20 jenerasyon ile gerçekleştirilen çalışma optimum sonucuna 4,5 saniye içerisinde ulaşmaktadır.

Çizelge 7. Tek amaçlı optimizasyon sonuçları

		Yay 1	Yay 2	Yay 3	Yay 4
İstenilen Yay Rijitliği	F [N]	1200	3500	4000	4000
	y [mm]	1,2	2,8	3,8	3,8
Algoritma		GA	GA	GA	GA
Amaç Fonksiyonları		f_1	f_1	f_1	f_1
Kısıt Fonksiyonları		g_1, g_2	g_1, g_2	g_1, g_2	g_1
Amaç Fonksiyonu	f_1 [mm]	2,4233E-11	2,2980E-08	0,36378	7,9595E-09
y [mm]		1,200007	2,79975	2,94702	3,79987
Gerilme [N/mm^2]		316,4189	439,45	449,613	803,135
Hacim [mm^3]		2707,3306	7924,42	9296,67	4569,799
Popülasyon		100	100	100	100
Jenerasyon		20	20	20	20
Tasarım Parametreleri	D_e [mm]	53,3488	68,51	69,9791	65,9168
	D_i [mm]	20,0546	24,2796	14,3105	24,697
	h_0 [mm]	3,0362	3,16489	4,2397	3,8258
	t [mm]	1,38758	2,4336	2,4938	1,5316
Optimizasyon Süresi [s]		4,312	4,244	4,318	2,307

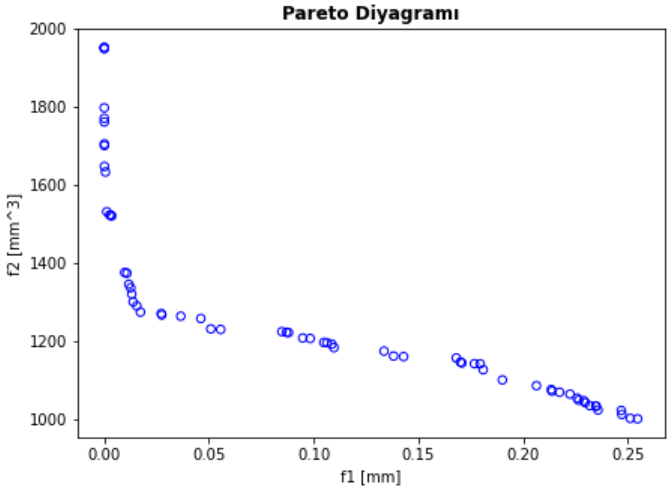
Çizelge 7 ile verilen tek amaçlı optimizasyon çalışmalarında ulaşılan 1E-9 seviyelerindeki hata değerlerinden ödün verilerek tasarımcı için kabul edilebilir hata seviyeleri ile daha hafif tasarımlara ulaşılabilir. Bu durumda hacmi ifade edecek ikinci bir amaç fonksiyonu ile gerçekleştirilecek çok amaçlı optimizasyon çalışması ihtiyacı ortaya çıkmaktadır.

Çizelge 8'de çok amaçlı optimizasyon algoritması olan NSGA-II ile gerçekleştirilen optimizasyon çalışmasının sonuçları verilmiştir. 4 farklı yay rijitliği için gerçekleştirilen çalışmada f_1 amaç fonksiyonun minimum olduğu tasarımlar listelenmiştir. Tek amaçlı optimizasyonda olduğu gibi çok amaçlı optimizasyonun da yakınsaması için gerekli süreler ortalama 4,5 saniyedir.

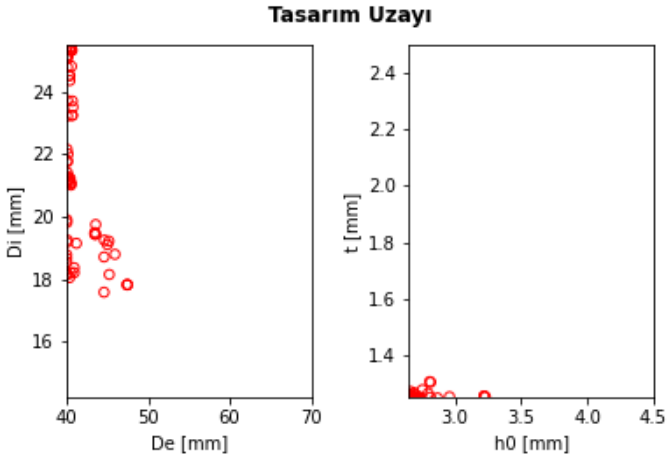
Çizelge 8. Çok amaçlı optimizasyon sonuçları

		Yay 5	Yay 6	Yay 7	Yay 8
İstenilen Yay Rijitliği	F [N]	700	1500	2800	3000
	y [mm]	0,86	1,5	2,4	1,2
Algoritma		NSGA-II	NSGA-II	NSGA-II	NSGA-II
Amaç Fonksiyonları		f_1, f_2	f_1, f_2	f_1, f_2	f_1, f_2
Kısıt Fonksiyonları		g_1, g_2	g_1, g_2	g_1, g_2	g_1, g_2
Amaç Fonksiyonları	f_1 [mm]	4,90E-10	6,31E-10	2,19E-09	6,743E-08
	f_2 [mm^3]	3029,009	1953,028	6935,77	2661,916
y [mm]		0,86003	1,49996	2,400066	1,19963
Gerilme [N/mm^2]		191,611	439,224	380,3619	437,1704
Hacim [mm^3]		3029,009	1953,03	6935,77	2661,916
Popülasyon		100	100	100	100
Jenerasyon		20	20	20	20
Tasarım Parametreleri	D_e [mm]	56,5897	47,445	63,4485	46,6885
	D_i [mm]	15,3433	17,825	18,221	24,1731
	h_0 [mm]	3,9282	3,2192	3,1424	2,681
	t [mm]	1,2769	1,2568	2,368	2,0665
Optimizasyon Süresi [s]		4,336	4,3127	4,371	4,375

f_1 amaç fonksiyonun hata değerinin arttığı ancak hacmin beklenildiği üzere azaldığı Pareto çözümleri ve tasarım uzayları Yay 6 için Şekil 43 ve 44'te, çok amaçlı optimizasyon çalışmasında elde edilen çözümlerden birkaçı ise Çizelge 9'da verilmiştir.



Şekil 43. Yay 6 için Pareto diyagramı



Şekil 44. Yay 6 için tasarım uzayı

Çizelge 9 incelendiğinde f_1 fonksiyonundan ödün verilerek hacmin azaltılabileceği gözlemlenmektedir. Çözüm 1 ve 2 karşılaştırıldığında, 5,33E-06 mm hataya sahip tasarımın seçilmesi ile hacmin 181,638 mm^3 azalacağı ve malzemeden %9,3 tasarruf edilerek ağırlığın azaltılabileceği gözlemlenmektedir.

Çizelge 9. Yay 6 için elde edilen rijitliklerden bazıları

		Çözüm 1	Çözüm 2	Çözüm 3	Çözüm 4
Fonksiyon Değerleri	f_1 [mm]	6,31E-10	5,33E-06	4,62E-05	5,42E-04
	f_2 [mm^3]	1953,028	1771,394	1705,753	1633,66
y [mm]		1,49996	1,4967	1,4904	1,46708
Gerilme [N/mm^2]		439,224	443,668	444,412	445,047
Hacim [mm^3]		1953,028	1771,394	1705,75	1633,66
Tasarım Parametreleri	D_e [mm]	47,445	45,932	45,006	44,63
	D_i [mm]	17,825	18,7936	19,1	19,245
	h_0 [mm]	3,2192	2,9579	2,7529	2,8134
	t [mm]	1,2568	1,2545	1,2793	1,2521

100 popülasyon ve 20 jenerasyon ile gerçekleştirilen optimizasyon çalışmasında doğrudan sonlu elemanlar yöntemine başvurulması halinde, optimizasyon algoritması için gereken yaklaşık 5 saatlik süreler sinir ağı kullanımı ile 4,5 saniyeye inmektedir. Bu fark, doğrusal olmayan, karmaşık geometrili ve çok sayıda elemandan oluşan modellerde daha fazla olacaktır.

4.2. Kauçuk Burç

4.2.1. Kauçuk Burcun YSA ile Modellenmesi

Kauçuk burç için kurulan ve eğitilen sinir ağının farklı epoch sayılarındaki başarıları R^2, MAE ve MSE metrikleri ile Çizelge 10'da belirtilmiştir. Çizelgedeki başarı oranları incelendiğinde her üç epoch sayısı içinde başarılı sonuçlara ulaşılmasına rağmen 1000 epoch eğitilen sinir ağının MSE metriğinin daha az hataya sahip olduğu görülmektedir. Çizelge 11'de ise farklı sinir ağlarının eğitimi için gereken süre saniye cinsinden verilmiştir.

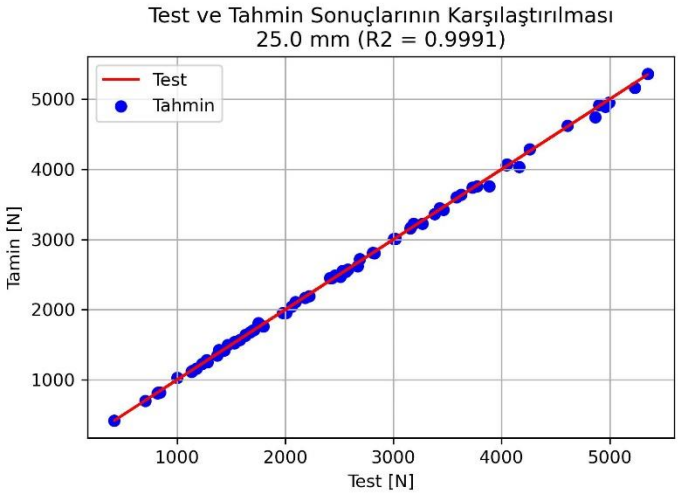
Çizelge 10. Kauçuk burç için sinir ağı başarı metrikleri

Epoch	R2		MAE		MSE	
	Train	Test	Train	Test	Train	Test
250	%99,92	%99,86	12,59	17,02	366,80	768,45
500	%99,90	%99,85	13,88	17,89	360,78	668,77
1000	%99,87	%99,90	12,65	15,36	326,21	528,29

Çizelge 11. Sinir ağı eğitimi için gerekli süre

Epoch	Time [s]
250	27,4
500	52,1
1000	100,2

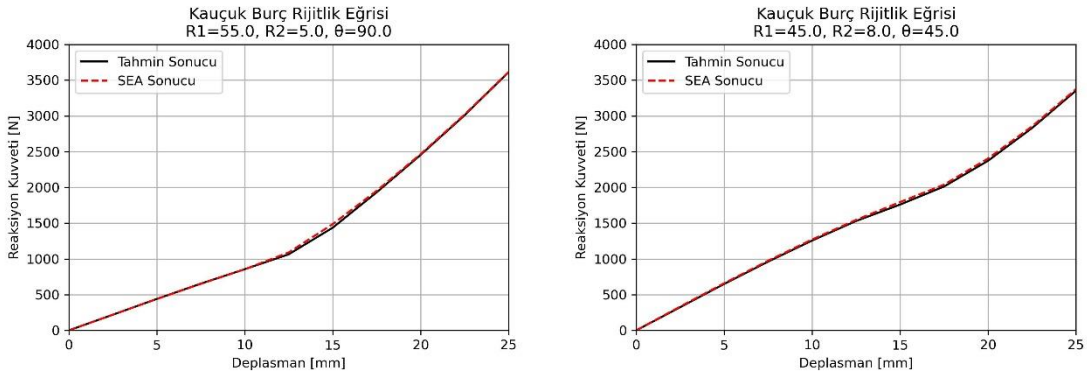
Sadece 25 mm yer değiştirme için yapay sinir ağının tahmin değerleri ile test verileri arasındaki karşılaştırma Şekil 45'te verilmiştir. Şekilde de görüleceği üzere tahmin ve test verileri arasındaki uyum, yapay sinir ağı modelinin girdilere karşılık çıktıları başarıyla tahmin ettiğini göstermektedir.



Şekil 45. 25 mm deplasman için test ve tahmin değerlerinin karşılaştırılması

Sinir ağının doğruluğunu göstermek için iki farklı kauçuk burç analizi gerçekleştirilmiştir. Şekil 46'da gösterilen sonuçlar, yapay sinir ağının sonlu eleman analizi sonuçlarını başarıyla tahmin ettiğini bir kez daha ortaya koymaktadır. Bu

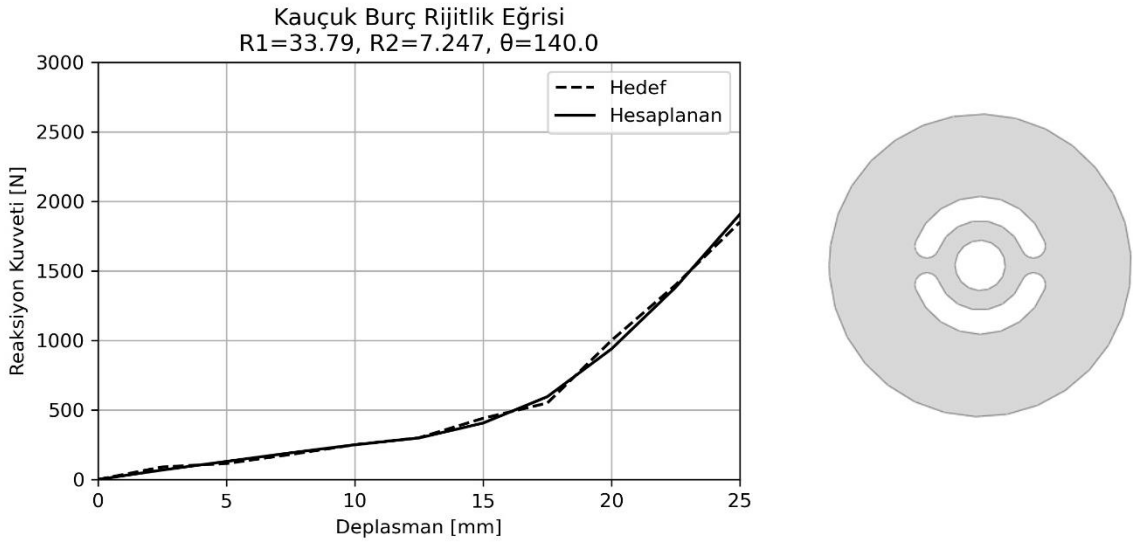
nedenle optimizasyon çalışmasında 1000 epoch eğitilmiş sinir ağı kullanımı tercih edilmiştir.



Şekil 46. Sinir ağı ile SEA sonuçlarının karşılaştırılması

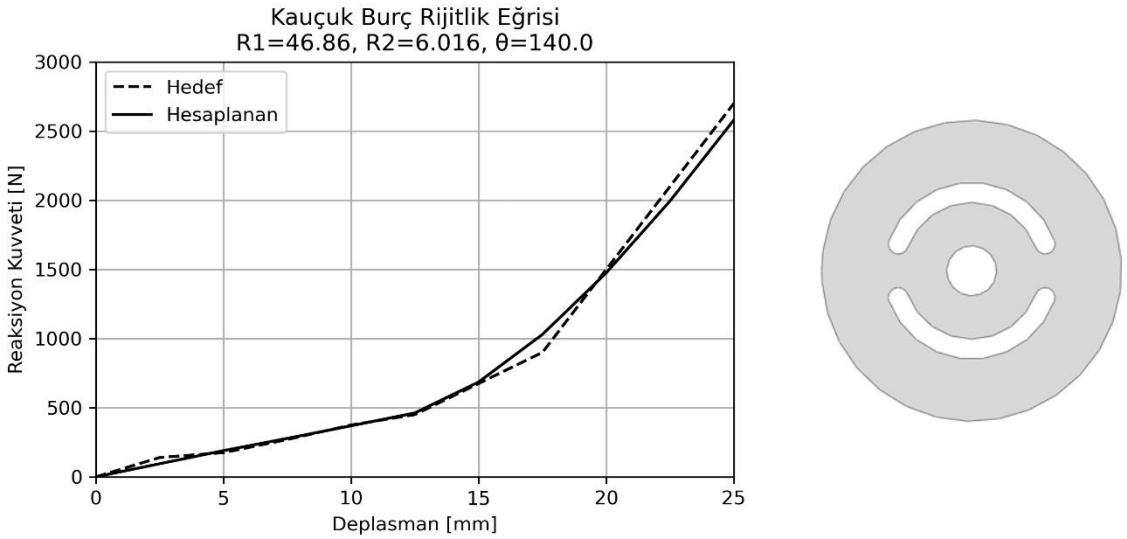
4.2.2. Kauçuk Burcun Rijitlik Optimizasyonu

Hedef Rijitlik 1: Optimizasyon çalışması ile elde edilen burç rijitliği ile hedef eğri arasındaki uyum Şekil 47'de gösterilmiştir. Algoritma optimum sonucuna R_1=33,79, R_2=7,247, and θ=140° tasarım parametreleri ile ulaşmıştır. Şekilde de görüleceği üzere hedef ile hesaplanan eğri arasındaki hata çok küçüktür.



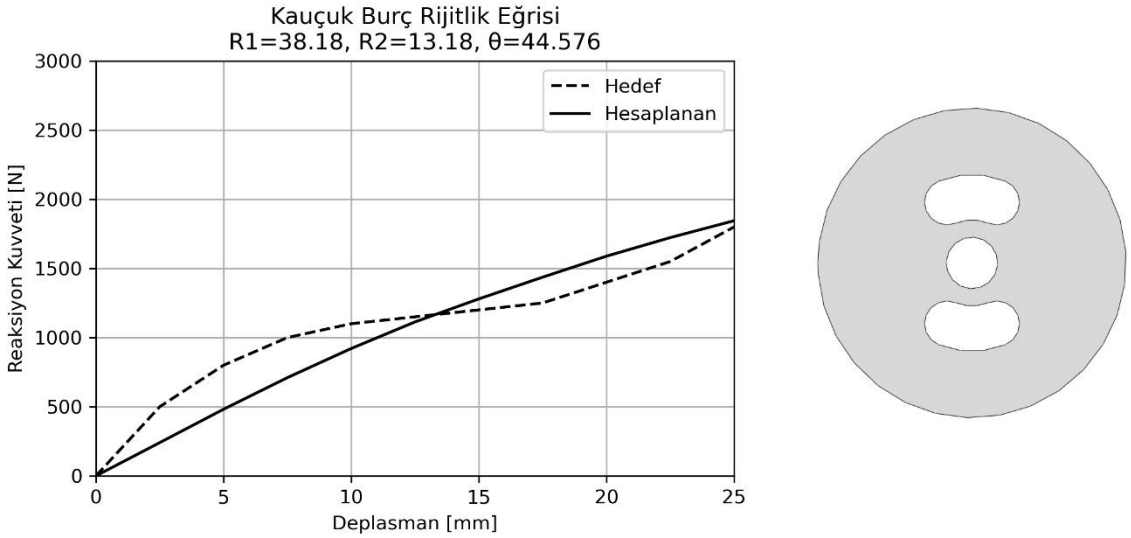
Şekil 47. Birinci optimizasyon çalışmasının optimizasyon sonucu

Hedef Rijitlik 2: İkinci optimizasyon çalışması için daha rijit bir burç ele alınmıştır. Elde edilen rijitlik ile hedef eğri arasındaki uyum Şekil 48'de belirtilmiştir. Algoritmanın optimum sonucuna $R_1=46{,}86$, $R_2=6{,}015$, and $\theta=140°$ tasarım parametreleri ile ulaştığı görülmektedir. İlk optimizasyon çalışması gibi burada da hedef ile hesaplanan eğri arasındaki hatanın çok küçük olduğu görülmektedir.



Şekil 48. İkinci optimizasyon çalışması sonucu

Hedef Rijitlik 3: Üçüncü optimizasyon çalışması için çalışmada belirlenen burç tasarımı ile ulaşılması beklenmeyen bir eğri incelenmiştir. Buradaki amaç sunulan yöntemin nasıl bir davranış sergileyeceğini görmektir. Optimizasyon algoritması optimum sonucuna $R_1=38{,}18$, $R_2=13{,}18$, and $\theta=44{,}781°$ tasarım parametreleri ile ulaşmıştır. Elde edilen rijitlik ile hedef eğri arasındaki uyum Şekil 49'de belirtilmiştir.



Şekil 49.Üçüncü optimizasyon çalışması sonucu

Her üç optimizasyon çalışması da optimum sonuçlarına 4 saniye içerisinde ulaşabilmektedir. Sinir ağı yerine doğrudan SEA kullanılması durumunda algoritma optimum sonucuna ulaşmak için 182 saate ihtiyaç duymaktadır. Bu durum sunulan metodolojinin, kauçuk benzeri malzemeler kullanılan ürün geliştirme çalışmalarına sağlayacağı katkıyı göstermektedir.

4.3. Elektrikli Araç Batarya Taşıyıcısı

4.3.1. Batarya Taşıyıcısının YSA ile Modellenmesi

Makine öğrenmesi analizinin sonuçları, veri kümesinin çıktılarını tahmin etmede beş sinir ağının başarı oranlarını Çizelge 12-14'te sunulmaktadır. Çizelgedeki yüksek R^2 ve düşük MAE ile MSE değerleri ile gösterildiği üzere beş sinir ağı da karmaşık verileri tahmin etmede önemli bir yeteneğe sahiptir.

Çizelge 12. R^2 belirlilik katsayısı

Çıktılar	R^2		
	Epoch	Train	Test
Kütle	250	99,95%	99,86%
Eğilme Deformasyonu	354	98,28%	98,62%
Eğilme Gerilmesi	249	99,71%	98,52%
Burulma Açısı	92	99,66%	99,51%
Burulma Gerilmesi	267	99,09%	98,03%

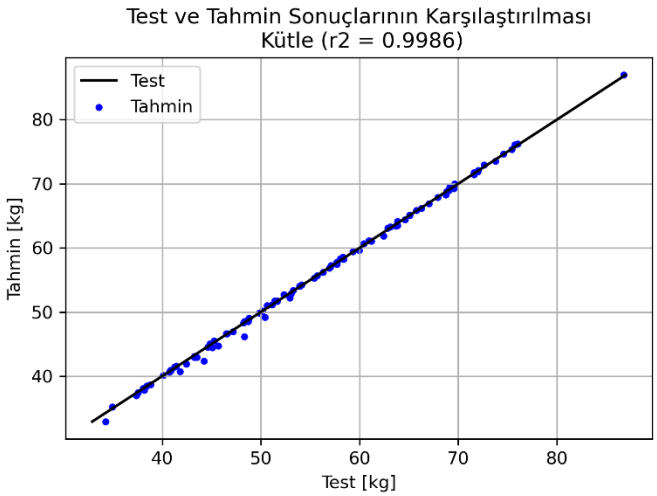
Çizelge 13. Ortalama mutlak hata

Çıktılar	MAE		
	Epoch	Train	Test
Kütle	250	0,1793	0,2624
Eğilme Deformasyonu	354	0,0058	0,0053
Eğilme Gerilmesi	249	0,6697	1,5574
Burulma Açısı	92	0,0162	0,0186
Burulma Gerilmesi	267	3,3452	16,910

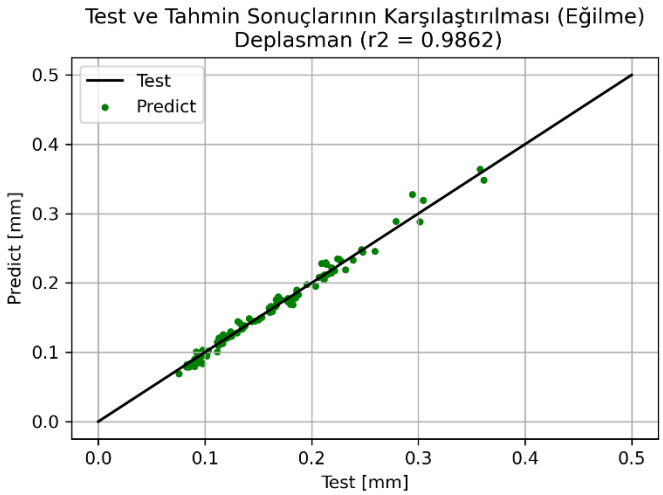
Çizelge 14. Ortalama kareler hatası

Çıktılar	MSE		
	Epoch	Train	Test
Kütle	250	0,058	0,1883
Eğilme Deformasyonu	354	9,53E-05	5,59E-05
Eğilme Gerilmesi	249	1,3262	4,8070
Burulma Açısı	92	4,87E-04	7,48E-04
Burulma Gerilmesi	267	219,532	477,862

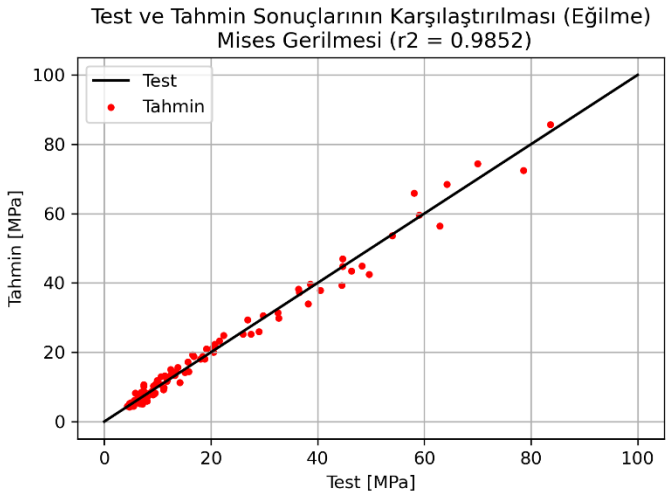
Şekil 50-54, tahmin edilen değerler ile gerçek test verileri arasındaki karşılaştırmayı göstermektedir. Kurulan sinir ağlarının, tahmin edilen değerler ile gerçek test verileri arasındaki uyumda da görüldüğü üzere, ağın daha önce görmediği test verilerini başarıyla tahmin ettiği görülmektedir.



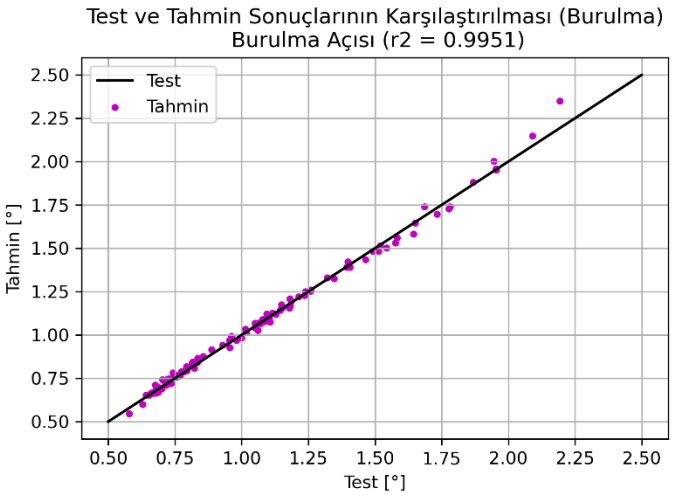
Şekil 50. Kütle için test ve tahmin değerlerinin karşılaştırması



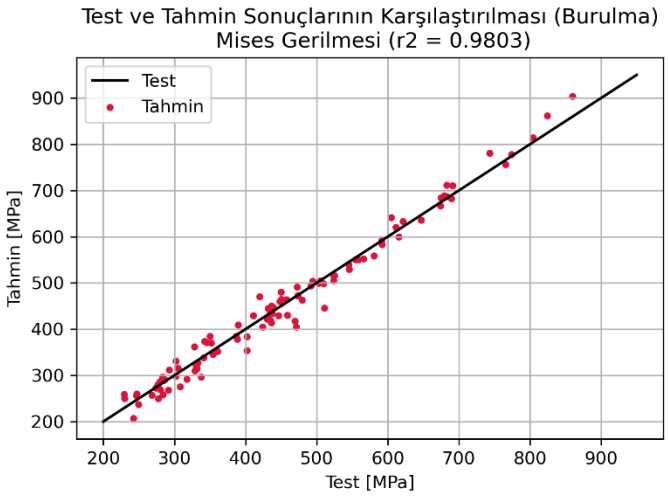
Şekil 51. Eğilme deformasyonu için test ve tahmin değerlerinin karşılaştırması



Şekil 52. Eğilme gerilmesi için test ve tahmin değerlerinin karşılaştırması



Şekil 53. Burulma açısı için test ve tahmin değerlerinin karşılaştırması



Şekil 54. Burulma gerilmesi için test ve tahmin değerlerinin karşılaştırması

4.3.2. Batarya Taşıyıcısı Kütle Minimizasyonu

Diferensiyel gelişim algoritması kullanılarak gerçekleştirilen optimizasyon çalışması sonuçları Çizelge 15 ile gösterilmekte ve optimizasyon algoritması belirlenen kısıtlar altında kütleyi başarıyla minimize etmiştir. Çizelge 15'te de görüleceği üzere batarya taşıyıcısının kütlesi 55.97 kilogramdan 32,31 kilograma inmiştir. Kütlede %42,27 azalma gerçekleşirken deformasyon ve burulma açısı sırası ile 0,999 mm ve 0,736°'ye çıkmıştır. Optimize edilmiş batarya taşıyıcısının eğilme ve burulma rijitlikleri Çizelge 16'da verilmiştir ve hedef rijitlikleri karşıladığı görülmektedir.

Çizelge 15. İlk tasarım ile optimum tasarımın karşılaştırması

	İlk Tasarım	Optimum Tasarım
T_1 [mm]	3,0	2,21
T_2 [mm]	3,0	1,00
T_3 [mm]	3,0	1,43
T_4 [mm]	3,0	4,50
T_5 [mm]	3,0	1,19
T_6 [mm]	3,0	1,01
Kütle [kg]	55,97	32,31
Eğilme Deformasyonu [mm]	0,628	0,999
Eğilme Gerilmesi [MPa]	61,63	42,12
Burulma Açısı [°]	0,379	0,736
Burulma Gerilmesi [MPa]	152,15	293,48

Çizelge 16. Optimum tasarım rijitlikleri

Eğilme	5005,01 N/mm
Burulma	815,22 Nm/°

Optimizasyon çalışması neticesinden kütlenin minimize edilmesinin yanı sıra 61,63 MPa olan eğilme gerilmesi 42,12 MPa'a düşmüştür. Çizelge 15, 4. kalınlığın hem eğilme hem de burulma rijitliği için tasarım hedeflerini karşılamada önemli bir role sahip olduğunu göstermektedir. 2., 5. ve 6. kalınlıklar ise alt tasarım sınırlarına yaklaşarak kütlenin azaltılmasına önemli ölçüde katkı sağlamıştır.

Günümüzde, sürdürülebilir ulaşım çözümlere yönelik talep hızla artmakta ve otomotiv sektöründe büyük bir rekabet haline gelmektedir. Batarya taşıyıcı tasarımının optimize edilmesi, bu talebi karşılayacak elektrikli araçların geliştirilmesinde önem arz etmektedir.

5. SONUÇLAR

Tez çalışmasında, sonlu eleman analizleri ile gerçekleştirilebilen optimizasyon çalışmasına kıyasla yapay sinir ağının kullanımının avantajları incelenmiştir. Bu doğrultuda disk yay, kauçuk burç ve batarya taşıyıcısı için sonlu elemanlar modeli oluşturulmuş ve yapay sinir ağı eğitimi için veri toplanmıştır. Eğitilen yapay sinir ağı modeli optimizasyon çalışmasında kullanılarak sonlu eleman analizlerinin sonuçları tahmin edilmiştir. Elde edilen bulgular neticesinde;

1. Önerilen metodolojinin uygun epoch sayısı ve sinir ağı yapısı ile sonlu eleman analizi sonuçlarını tahmin etmekte yüksek başarı oranlarının yakalanabildiği,
2. Yüksek işlem kapasitesi gerektiren popülasyon tabanlı bir evrimsel algoritma kullanımı durumunda sinir ağı kullanımı ile gerçekleştirilen optimizasyon çalışmalarının 5 saniye içerisinde tamamlanabildiği,
3. Doğrusal olmayan davranış sergileyen malzemeler, karmaşık geometrilere sahip tasarımlar veya çok sayıda eleman içeren modellerde sunulan yöntemin sağladığı zaman avantajının daha fazla olduğu gözlemlenmiştir.

Önerilen metodolojinin kauçuk burçların rijitlik optimizasyona başarılı bir şekilde uygulanabilmesi otomobillerdeki sürüş konforu ve güvenirliliğe olan katkısını ortaya koymaktadır. Aynı şekilde batarya taşıyıcısının arzu edilen rijitlikleri sağlayarak daha hafif tasarlanması elektrikli araçların hem performansına hem de yakıt tasarrufuna olan katkısını arttıracaktır.

Önerilen metodoloji ile gerçekleştirilen optimizasyon çalışmaları otomotiv parçalarının tasarımında önemli bir ilerlemeyi temsil etmektedir. Yöntemin etkinliği ve çok yönlülüğü, otomotiv ürünlerinin performansının ve verimliliğinin optimize edilmesi adına sunulan metodu önemli bir araç haline getirmektedir.

KAYNAKLAR

Ahmadianfar, I., Bozorg-Haddad, O., & Chu, X. (2020). Gradient-based optimizer: A new metaheuristic optimization algorithm. *Information Sciences*, *540*, 131–159. https://doi.org/10.1016/J.INS.2020.06.037

Alpaydin, E. (2020). *Introduction to machine learning* (2nd ed.). MIT press.

Andradbttir, E. S., Healy, K. J., Withers, D. H., Nelson, B. L., Carson, Y., & Maria, A. (1997). Simulation optimization. *Proceedings of the 29th Conference on Winter Simulation - WSC '97*, 118–126. https://doi.org/10.1145/268437.268460

Annicchiarico, W., & Cerrolaza, M. (2001). Structural shape optimization 3D finite-element models based on genetic algorithms and geometric modeling. *Finite Elements in Analysis and Design*, *37*(5), 403–415. https://doi.org/10.1016/S0168-874X(00)00041-X

Back, T., Hammel, U., & Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, *1*(1), 3–17. https://doi.org/10.1109/4235.585888

Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., & Abraham, A. (2020). Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, *90*. https://doi.org/10.1016/J.ENGAPPAI.2020.103479

Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Blundell, M. V. (1998). The influence of rubber bush compliance on vehicle suspension movement. *Materials and Design*, *19*(1–2), 29–37. https://doi.org/10.1016/S0261-3069(97)00101-5

Bremermann, H. J. (1962). Optimization through evolution and recombination. *Self-Organizing Systems*, *93*, 106.

Bulut, E., Albak, E. I., Sevilgen, G., & Öztürk, F. (2022). Prediction and optimization of the design decisions of liquid cooling systems of battery modules using artificial neural networks. *International Journal of Energy Research*, *46*(6), 7293–7308. https://doi.org/10.1002/er.7637

Cheng, J. (2007). Hybrid genetic algorithms for structural reliability analysis. *Computers & Structures*, *85*(19–20), 1524–1533. https://doi.org/10.1016/J.COMPSTRUC.2007.01.018

Chollet, F. (2018). *Deep Learning with Python*. Manning Publisher Co.

Cook, D. F., Ragsdale, C. T., & Major, R. L. (2000). Combining a neural network with a genetic algorithm for process parameter optimization. *Engineering Applications of Artificial Intelligence*, *13*(4), 391–396. https://doi.org/10.1016/S0952-1976(00)00021-X

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer-Verlag Berlin Heidelberg.

Fahle, S., Prinz, C., & Kuhlenkötter, B. (2020). Systematic review on machine learning (ML) methods for manufacturing processes – Identifying artificial intelligence (AI) methods for field application. *Procedia CIRP*, *93*, 413–418. https://doi.org/10.1016/J.PROCIR.2020.04.109

Francis, R., Junior, R., Alves De Almeida, F., Guilherme, •, & Gomes, F. (2020). Fault classification in three-phase motors based on vibration signal analysis and artificial neural networks. *Neural Computing and Applications*, *32*(18), 15171–15189. https://doi.org/10.1007/s00521-020-04868-w

Friedberg, R. M. (1958). A Learning Machine: Part I. *IBM Journal of Research and Development*, *2*(1), 2–13. https://doi.org/10.1147/RD.21.0002

Gao, Y., Liu, X., Huang, H., & Xiang, J. (2021). A hybrid of FEM simulations and generative adversarial networks to classify faults in rotor-bearing systems. *ISA Transactions*, *108*, 356–366. https://doi.org/10.1016/J.ISATRA.2020.08.012

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14–15), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Gentils, T., Wang, L., & Kolios, A. (2017). Integrated structural optimisation of offshore wind turbine support structures based on finite element analysis and genetic algorithm. *Applied Energy*, *199*, 187–204. https://doi.org/10.1016/J.APENERGY.2017.05.009

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media, Inc.

Goldberg, D. E., & Holland, J. H. (1988). Genetic Algorithms and Machine Learning. *Machine Learning*, *3*(2), 95–99. https://doi.org/10.1023/A:1022602019183/METRICS

Guo, W., Deng, F., Meng, Z., Hua, L., Mao, H., & Su, J. (2020). A hybrid back-propagation neural network and intelligent algorithm combined algorithm for optimizing microcellular foaming injection molding process parameters. *Journal of Manufacturing Processes*, *50*, 528–538. https://doi.org/10.1016/J.JMAPRO.2019.12.020

Hilbert, R., Janiga, G., Baron, R., & Thévenin, D. (2006). Multi-objective shape optimization of a heat exchanger using parallel genetic algorithms. *International Journal of Heat and Mass Transfer*, *49*(15–16), 2567–2577. https://doi.org/10.1016/J.IJHEATMASSTRANSFER.2005.12.015

Joshi, M., Gyanchandani, M., & Rajesh Wadhvani, D. (2021). Analysis of Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing on Benchmark Functions. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 1152–1157. https://doi.org/10.1109/ICCMC51019.2021.9418458

Jung, Y. W., & Kim, H. K. (2020). Prediction of Nonlinear Stiffness of Automotive Bushings by Artificial Neural Network Models Trained by Data from Finite Element Analysis. *International Journal of Automotive Technology 2020 21:6*, *21*(6), 1539–1551. https://doi.org/10.1007/S12239-020-0145-1

Kadlowec, J., Glassboro, N., Jersey, A., Wineman, G., Hulbert, A., & Arbor, M. (2003). Elastomer bushing response: experiments and finite element modeling. *Acta Mechanica*, *163*(1), 25–38. https://doi.org/10.1007/s00707-003-1018-1

Kalay, O. C., Karpat, E., Dirik, A. E., & Karpat, F. (2023). A One-Dimensional Convolutional Neural Network-Based Method for Diagnosis of Tooth Root Cracks in Asymmetric Spur Gear Pairs. *Machines*, *11*(4), 413. https://doi.org/10.3390/MACHINES11040413

Kaldas, M. M. S., Çaljgkan, K., Henze, R., & Küçükay, F. (2014). *Optimization of Damper Top Mount Characteristics to Improve Vehicle Ride Comfort and Harshness*. https://doi.org/10.1155/2014/248129

Karina, C. N. N., Chun, P., & Okubo, K. (2017). Tensile Strength Prediction of Corroded Steel Plates by Using Machine Learning Approach. *Steel and Composite Structures*, *24*(5), 635–641. https://doi.org/10.12989/SCS.2017.24.5.635

Kaya, N. (2014). Shape optimization of rubber bushing using differential evolution algorithm. *Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/379196

Kaya, N., & Öztürk, F. (2002). Dikdörtgen Kesitli Disk Yayların Optimum Kesit Tasarımı. *Makina Tasarım ve İmalat Dergisi*, *4*(3), 140–145.

Keras. (2023). *https://keras.io/*.

Kim, J. J., & Kim, H. Y. (1997). Shape design of an engine mount by a method of parametric shape optimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *211*(2), 155–159. https://doi.org/10.1243/0954407971526326

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://doi.org/10.48550/arxiv.1412.6980

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. MIT Press.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, *91*, 992–1007. https://doi.org/10.1016/j.ress.2005.11.018

Kurtaran, H., Ozcelik, B., & Erzurumlu, T. (2005). Warpage optimization of a bus ceiling lamp base using neural network model and genetic algorithm. *Journal of Materials Processing Technology*, *169*(2), 314–319. https://doi.org/10.1016/J.JMATPROTEC.2005.03.013

Li, Q., Zhao, J., Zhao, B., & Zhu, X. (2009). Parameter Optimization of Rubber Mounts Based on Finite Element Analysis and Genetic Neural Network. *Journal of Macromolecular Science, Part A*, *46*(2), 186–192. https://doi.org/10.1080/10601320802594923

Liang, L., Liu, M., Martin, C., & Sun, W. (2018). A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. *Journal of the Royal Society Interface*, *15*(138). https://doi.org/10.1098/rsif.2017.0844

Liu, X., Athanasiou, C. E., Padture, N. P., Sheldon, B. W., & Gao, H. (2020). A machine learning approach to fracture mechanics problems. *Acta Materialia*, *190*, 105–112. https://doi.org/10.1016/J.ACTAMAT.2020.03.016

Ma, Z. S., Ding, Q., & Zhai, Y. J. (2021). Hybrid Modeling of Nonlinear-Jointed Structures via Finite-Element Model Reduction and Deep Learning Techniques. *Journal of Vibration Engineering and Technologies*, *9*(4), 575–585. https://doi.org/10.1007/S42417-020-00249-8/FIGURES/13

Mankovits, T., Szabó, T., Kocsis, I., & Páczelt, I. (2014). Optimization of the Shape of Axi-Symmetric Rubber Bumpers. *Strojniški Vestnik-Journal of Mechanical Engineering*, *60*(1), 61–71. https://doi.org/10.5545/sv-jme.2013.1315

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395. https://doi.org/10.1007/S00158-003-0368-6

Marseguerra, M., Zio, E., & Martorell, S. (2006). Basics of genetic algorithms optimization for RAMS applications. *Reliability Engineering and System Safety*, *91*(9), 977–991. https://doi.org/10.1016/J.RESS.2005.11.046

Miller, B. L., Goldberg, D. E., & others. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, *9*(3), 193–212.

Montrone, F., Wever, U., & Zheng, Q. (2004). 2D shape optimization of turbine blades. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift Für Angewandte Mathematik Und Mechanik*, *84*(2), 86–95. https://doi.org/10.1002/ZAMM.200310087

Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly Media, Inc.

Müller, M., Long, X., Botsch, M., Böhmländer, D., & Utschick, W. (2018). Real-Time Crash Severity Estimation with Machine Learning and 2D Mass-Spring-Damper Model. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2018-November*, 2036–2043. https://doi.org/10.1109/ITSC.2018.8569471

Na, C., Kim, S. P., & Kwak, H. G. (2011). Structural damage evaluation using genetic algorithm. *Journal of Sound and Vibration*, *330*(12), 2772–2783. https://doi.org/10.1016/J.JSV.2011.01.007

Öztürk, N., & Öztürk, F. (2001). Neural network based non-standard feature recognition to integrate CAD and CAM. *Computers in Industry*, *45*(2), 123–135. https://doi.org/10.1016/S0166-3615(01)00090-2

Park, C. H., Shim, H. J., Choi, D. H., Kim, J. K., & Lee, S. M. (2012). Shape optimization of rubber isolators in automotive cooling modules for the maximization of vibration isolation and fatigue life. *International Journal of Automotive Technology*, *13*(1), 61–75. https://doi.org/10.1007/s12239-012-0006-7

Paszyński, M., Grzeszczuk, R., Pardo, D., & Demkowicz, L. (2021). Deep Learning Driven Self-adaptive Hp Finite Element Method. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12742 LNCS*, 114–121. https://doi.org/10.1007/978-3-030-77961-0_11

Pfrommer, J., Zimmerling, C., Liu, J., Kärger, L., Henning, F., & Beyerer, J. (2018). Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CIRP*, *72*, 426–431. https://doi.org/10.1016/J.PROCIR.2018.03.046

Price, K., Storn, R. M., & Lampinen, J. A. (2006). *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.

Qi, Z., Zhang, N., Liu, Y., & Chen, W. (2019). Prediction of mechanical properties of carbon fiber based on cross-scale FEM and machine learning. *Composite Structures*, *212*, 199–206. https://doi.org/10.1016/J.COMPSTRUCT.2019.01.042

Qing, A. (2009). *Differential evolution: fundamentals and applications in electrical engineering*. John Wiley & Sons.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*.

Sampaio, G. S., Rabello De Aguiar, A., Filho, V., Santos Da Silva, L., & Augusto Da Silva, L. (2019). *Prediction of Motor Failure Time Using An Artificial Neural Network*. https://doi.org/10.3390/s19194342

Scikit-Learn. (2023, March 13). *https://scikit-learn.org/*.

Shi, H., Gao, Y., & Wang, X. (2010). Optimization of injection molding process parameters using integrated artificial neural network model and expected improvement function method. *International Journal of Advanced Manufacturing Technology*, *48*(9–12), 955–962. https://doi.org/10.1007/S00170-009-2346-7/FIGURES/10

Shui, L., Chen, F., Garg, A., Peng, X., Bao, N., & Zhang, J. (2018). Design optimization of battery pack enclosure for electric vehicle. *Structural and Multidisciplinary Optimization*, *58*(1), 331–347. https://doi.org/10.1007/S00158-018-1901-Y/TABLES/12

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

Storn, R., & Price, K. (1996). Minimizing the real functions of the ICEC'96 contest by Differential Evolution. *In Proceedings of IEEE International Conference on Evolutionary Computation*, 842–844.

Sun, H., Qiu, C., Lu, L., Gao, X., Chen, J., & Yang, H. (2020). Wind turbine power modelling and optimization using artificial neural network with wind field experimental data. *Applied Energy*, *280*, 115880. https://doi.org/10.1016/J.APENERGY.2020.115880

Tieleman, T., Hinton, G., & others. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26–31.

Uzun, E. (2023, March 13). *Makine Öğrenmesi*. https://erdincuzun.com/makine_ogrenmesi/makine-ogrenmesi-metotlari/

Vesterstram, J., & Thomsen, R. (2004). A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, 1980–1987. https://doi.org/10.1109/CEC.2004.1331139

Wang, S. Y., & Tai, K. (2005). Structural topology design optimization using Genetic Algorithms with a bit-array representation. *Computer Methods in Applied Mechanics and Engineering*, *194*(36–38), 3749–3770. https://doi.org/10.1016/j.cma.2004.09.003

Woon, S. Y., Querin, O. M., & Steven, G. P. (2001). Structural application of a shape optimization method based on a genetic algorithm. *Structural and Multidisciplinary Optimization*, *22*(1), 57–64. https://doi.org/10.1007/S001580100124/METRICS

Xiong, Y., Pan, Y., Wu, L., & Liu, B. (2021). Effective weight-reduction- and crashworthiness-analysis of a vehicle's battery-pack system via orthogonal experimental design and response surface methodology. *Engineering Failure Analysis*, *128*, 105635. https://doi.org/10.1016/J.ENGFAILANAL.2021.105635

Xudong, W., Shen, W. Z., Zhu, W. J., Sørensen, J. N., & Jin, C. (2009). *Shape Optimization of Wind Turbine Blades*. https://doi.org/10.1002/we.335

Yan, X., Zhu, J., Kuang, M., & Wang, X. (2019). Aerodynamic shape optimization using a novel optimizer based on machine learning techniques. *Aerospace Science and Technology*, *86*, 826–835. https://doi.org/10.1016/J.AST.2019.02.003

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, *3*(2), 82–102. https://doi.org/10.1109/4235.771163

Yin, F., Mao, H., Hua, L., Guo, W., & Shu, M. (2011). Back Propagation neural network modeling for warpage prediction and optimization of plastic products during injection molding. *Materials & Design*, *32*(4), 1844–1850. https://doi.org/10.1016/J.MATDES.2010.12.022

Yıldız, A. R. (2006). *Tasarım optimizasyon problemlerinin çözümü için çok amaçlı arama tabanlı bir sistemin geliştirilmesi*. Bursa Uludag University (Turkey).

Yıldız, A. R., Öztürk, N., Kaya, N., & Öztürk, F. (2003). Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm. *Structural and Multidisciplinary Optimization*, *25*(4), 251–260. https://doi.org/10.1007/s00158-006-0079-x

Yıldız, A. R., Oztürk, N., Kaya, N., & Oztürk, F. (2003). Integrated optimal topology design and shape optimization using neural networks. *Struct Multidisc Optim*, *25*, 251–260. https://doi.org/10.1007/s00158-003-0300-0

Yoshida, S. (2006). Wind turbine tower optimization method using a genetic algorithm. *Wind Engineering*, *30*(6), 453–470. https://doi.org/10.1260/030952406779994150

Zhang, X., Xiong, Y., Pan, Y., Du, H., & Liu, B. (2023). Crushing stress and vibration fatigue-life optimization of a battery-pack system. *Structural and Multidisciplinary Optimization*, *66*(3), 1–16. https://doi.org/10.1007/S00158-023-03510-2/TABLES/10

Zhou, Y., Kalogerakis, E., Wang, R., & Grosse, I. R. (2016). Direct shape optimization for strengthening 3D printable objects. *Computer Graphics Forum*, *35*(7), 333–342. https://doi.org/10.1111/CGF.13030