T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

SİNCAP ARAMA ALGORİTMASI İLE YAPAY SİNİR AĞLARININ EĞİTİMİ

Hazırlayan
Betül KARAKAŞ

Danışman
Prof. Dr. Sinem KULLUK

Yüksek Lisans Tezi

Ekim 2020
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

SİNCAP ARAMA ALGORİTMASI İLE YAPAY SİNİR AĞLARININ EĞİTİMİ

(Yüksek Lisans Tezi)

Hazırlayan
Betül KARAKAŞ

Danışman
Prof. Dr. Sinem KULLUK

Ekim 2020
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Betül KARAKAŞ

YÖNERGEYE UYGUNLUK

"Sincap Arama Algoritması ile Yapay Sinir Ağlarının Eğitimi" adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Hazırlayan
Betül KARAKAŞ

Danışman
Prof. Dr. Sinem KULLUK

Endüstri Mühendisliği ABD Başkanı

Prof. Dr. Lale ÖZBAKIR

KABUL VE ONAY

Prof. Dr. Sinem KULLUK danışmanlığında **Betül KARAKAŞ** tarafından hazırlanan **"Sincap Arama Algoritması ile Yapay Sinir Ağlarının Eğitimi"** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Endüstri Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

05/10/2020

JÜRİ:

Danışman : Prof. Dr. Sinem KULLUK .....

Üye : Prof. Dr. Lale ÖZBAKIR .....

Üye : Doç. Dr. Burcu ORALHAN .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 20/10/2020 tarih ve 2020/51-10 sayılı kararı ile onaylanmıştır.

.....

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Bana çalışmalarım süresince her türlü yardımı ve fedakârlığı sağlayan, öneri ve yardımları ile çalışmamı yönlendirerek katkıda bulunan değerli danışman hocam sayın Prof. Dr. Sinem KULLUK'a en derin duygularımla teşekkür ederim.

Sayın Doç. Dr. Esra KAHYA ÖZYİRMİDOKUZ'a çalışmamı yürütmem için gerekli tüm kolaylıkları sağladığı, bilgi ve birikimleriyle bana yol gösterdiği ve proje kapsamında toplanan verileri benimle paylaştığı için teşekkür ederim.

Çalışmalarım süresince maddi, manevi pek çok fedakârlık gösteren, destekleri ile her anımda yanımda olan, bana güç veren sevgili eşim Mehmet Fatih KARAKAŞ'a ve üzerimde sonsuz emekleri olan, çalışmalarım boyunca beni destekleyen, motive eden, çalışma ortamımı oluşturmak için her türlü desteği sağlayan başta kıymetli annem Nesrin AKPINAR olmak üzere tüm aileme minnettarlığımı belirterek teşekkür ederim.

Bu çalışma 116E676 nolu TÜBİTAK 1003 Öncelikli Alanlar Ar-Ge Projesi olan "Mutlu Proje: Sosyal Mutluluk Ölçen Bir Sosyo-teknik Madenleme Sistemi" tarafından desteklendiği için TÜBİTAK-ARDEB' e teşekkür ederim.

Betül KARAKAŞ

Ekim 2020, KAYSERİ

SİNCAP ARAMA ALGORİTMASI İLE YAPAY SİNİR AĞLARININ EĞİTİMİ

Betül KARAKAŞ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ekim 2020
Danışman: Prof. Dr. Sinem KULLUK

ÖZET

Yapay Sinir Ağları (YSA) karmaşık problemlerin çözümünde oldukça başarılı sonuçlar elde etmektedir. Yapay Sinir Ağlarının ağırlık değerlerinin belirlenmesi ve güncellenmesi ile ağın eğitilmesi mümkündür.

Bu tez çalışmasında YSA eğitilmesinde son yıllarda geliştirilen Sincap Arama Algoritması kullanılmıştır (SSA). Uçan sincapların dinamik arama davranışlarını ve süzülme hareketlerini taklit eden bu algoritmanın performansı UCI makine öğrenme deposundan (https://archive.ics.uci.edu/ml/index.php) alınan 6 sınıflandırma veri kümesi ve bir gerçek hayat verisi olan Oxford Mutluluk Ölçeği veri kümesi üzerinde değerlendirilmiştir. MSE değerini minimize eden ağırlıkların bulunması ve test verileri için doğruluk değerlerinin maksimize edilmesi bu çalışmanın etkinlik ölçütleridir. En iyi sınıflandırma doğruluğunu veren parametrelerin belirlenmesi için çalışmada "Deney Tasarımı" gerçekleştirilmiştir. Test veri kümeleri üzerinde elde edilen sonuçlar ve karşılaştırmalar, SSA ile eğitilen YSA'nın yüksek doğruluklarla oldukça etkin sonuçlar verdiğini göstermektedir. Önerilen yöntemle diğer yöntemler arasında anlamlı bir farklılık olup olmadığını test etmek için "İstatistiksel Analiz" gerçekleştirilmiştir.

Yapılan deneysel çalışmalar neticesinde SSA kullanılarak eğitilen YSA'nın sınıflandırma problemlerinde oldukça etkin olduğu sonucuna varılmıştır. YSA'nın sınıflandırma performansı SSA ile eğitilerek arttırılmıştır.

Anahtar Kelimeler: Yapay Sinir Ağı, Metasezgisel Algoritmalar, Sincap Arama Algoritması, Sınıflandırma, Mutluluk, Oxford Mutluluk Ölçeği

TRAINING ARTIFICIAL NEURAL NETWORKS WITH SQUIRREL SEARCH ALGORITHM

Betül KARAKAŞ

Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, October 2020
Supervisor: Prof. Dr. Sinem KULLUK

ABSTRACT

Artificial Neural Networks (ANN) achieve very successful results in solving complex problems. It is possible to train the network by determining and updating the weight values of Artificial Neural Networks.

In this study, recently developed Squirrel Search Algorithm (SSA) was used in ANN training. The performance of this algorithm, which simulates the dynamic search behavior and gliding movements of flying squirrels, was evaluated using 6 classification datasets taken from the UCI machine learning repository (https://archive.ics.uci.edu/ml/index.php) and a real-life data from the Oxford Happiness Scale. Finding the weights that minimize the MSE value and maximizing the accuracy values for the test data are the efficiency criteria of this study. To determine the parameters that give the best classification accuracy, "Experiment Design" was carried out in the study. The results and comparisons obtained from the test data sets show that the ANN trained with SSA gives very effective results with high accuracy. "Statistical Analysis" was conducted to test whether there is a significant difference between the proposed method and other methods. As a result of the experimental studies, it has been concluded that ANN trained using SSA is way more effective in classification problems. Classification performance of ANN has been increased by training with SSA.

Keywords: Artificial Neural Network, Metaheuristic Algorithm, Squirrel Search Algorithm, Classification, Happiness, Oxford Happiness Scale

İÇİNDEKİLER

SİNCAP ARAMA ALGORİTMASI İLE YAPAY SİNİR AĞLARININ EĞİTİMİ

YÖNERGEYE UYGUNLUK....iii
KABUL VE ONAY....iv
TEŞEKKÜR....v
ÖZET....vi
ABSTRACT....vii
İÇİNDEKİLER....viii
TABLOLAR LİSTESİ....xii
ŞEKİLLER LİSTESİ....xiii
GİRİŞ....1

1. BÖLÜM
GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1 Literatür Araştırması....4
1.1.1 Yapay Sinir Ağları Literatürü....4
1.1.2. Metasezgisel Algoritmalar Literatürü....5
1.1.3. Yapay Sinir Ağlarının Metasezgisel Algoritmalar ile Eğitimi Literatürü....8
1.1.4. Mutluluk Literatürü....11
1.2 Araştırmanın Amacı ve Önemi....13

2. BÖLÜM
YÖNTEM VE MATERYAL

2.1 Yapay Sinir Ağları....14
2.1.1 Biyolojik Sinir Hücresi ve Yapay Sinir Ağları....14

2.1.2 Yapay Sinir Ağları Çalışma Yapısı, Özellikleri ve Kullanım Alanları 17

2.1.2.1 Durdurma Kriteri 20

2.1.2.2 Aktivasyon Fonksiyonu 22

2.1.2.3 Yapay Sinir Ağlarında Öğrenme 24

2.1.3 Yapılarına Göre Yapay Sinir Ağları 27

2.1.3.1 İleri Beslemeli Yapay Sinir Ağı Modeli 27

2.1.3.2 Geri Beslemeli Yapay Sinir Ağı Modeli 28

2.1.4 Kullanım Alanlarına Göre Yapay Sinir Ağı Modelleri 28

2.1.5 Tek Katmanlı Algılayıcılar 30

2.1.6 Çok Katmanlı Algılayıcılar 31

2.1.6.1 Momentum Katsayısı 32

2.1.7 Yapay Sinir Ağları Eğitimi 33

2.1.7.1 Yapay Sinir Ağı Eğitimi için Kullanılacak Ağın Özelliklerinin Belirlenmesi 34

2.2 Metasezgisel Algoritmalar 37

2.2.1 Sincap Arama Algoritması 40

2.2.1.1 Sincap Arama Algoritması Tarihçesi 40

2.2.1.2 Sincap Arama Algoritması Temeli 42

2.2.1.3 Sincap Arama Algoritması Aşamaları 43

2.2.1.4 Sincap Arama Algoritması Çalışma Yapısı 50

3. BÖLÜM

UYGULAMA ve BULGULAR

3.1 Veri Kümeleri 53

3.2 Algoritmanın Çalıştırılması ve Etkinlik Ölçütleri 57

3.3 Deney Tasarımı 63

3.3.1 UCI Referans Sınıflandırma Veri Kümeleri Deney Tasarımı 65

3.3.2 Gerçek Hayat Verisi Deney Tasarımı 80

4. BÖLÜM
SONUÇ VE DEĞERLENDİRME

4.1 Sonuç ve Değerlendirme 90

5 KAYNAKÇA 92

6 ÖZGEÇMİŞ 105

KISALTMALAR

YSA: Yapay Sinir Ağları

SSA: Sincap Arama Algoritması

ISSA: Geliştirilmiş Sincap Arama Algoritması

RTS: Reaktif Tabu Arama Algoritması

DE: Diferansiyel Evrim Algoritması

ACO: Karınca Koloni Algoritması

APSO: Adaptif Parçacık Sürüsü Optimizasyon Algoritması

GSA: Yerçekimi Arama Algoritması

PSO: Parçacık Sürü Optimizasyon Algoritması

GSA: Yerçekimi Arama Algoritması

ABC: Yapay Arı Koloni Algoritması

HS: Harmoni Arama Algoritması

IWO: Yabani Ot Algoritması

CS: Guguk Kuşu Arama Algoritması

GA: Genetik Algoritma

BA: Yarasa Algoritması

FF: Ateşböceği Algoritması

MLP: Çok Katmanlı Algılayıcı

SLP: Tek Katmanlı Algılayıcı

MSE: Hata Kareler Ortalaması

ME: Ortalama Hata

RMS: Hata Karelerinin Ortalamasının Karekökü

MAE: Ortalama Mutlak Hata

MR: Manyetik Rezonans

MATLAB: Matrix Laboratory

WDBC: Wisconsin Diagnostic Breast Cancer

STD: Standart Sapma

TABLOLAR LİSTESİ

Tablo 1.1 Sürü Tabanlı Metasezgisel Optimizasyon Algoritmaları.....7
Tablo 2.1 Biyolojik Sinir Sistemine ait Elemanlar ve Yapay Sinir Ağındaki Karşılığı .16
Tablo 2.2 Geleneksel algoritmalar ve YSA'nın Karşılaştırılması 17
Tablo 2.3 Kullanım Alanlarına ve Amaçlarına Göre Kullanılan YSA Topolojileri 29
Tablo 2.4 Sincap Arama Algoritması Sözde Kodu.....51
Tablo 3.1 UCI Makine Öğrenmesi Deposundan Alınan Veri Kümelerinin Özellikleri .53
Tablo 3.2 Oxford Mutluluk Veri Kümesi Özellikleri 56
Tablo 3.3 Mutluluk Veri Kümesi Örneği.....57
Tablo 3.4 Veri Kümelerinin Eğitim ve Test Kümelerine Bölümlenmesi 58
Tablo 3.5 SSA ile Eğitilmiş YSA Modelinin Sözde Kodu 59
Tablo 3.6 Deney Tasarımı Parametreleri 65
Tablo 3.7 Her Bir Veri Kümesi için En İyi Deney Tasarım Parametreleri.....66
Tablo 3.8 Her Bir Veri Kümesinin Ortalama, Minimum, Maksimum MSE, Standart Sapma ve Doğruluk Değerleri.....73
Tablo 3.9 Her Bir Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması 75
Tablo 3.10 Metasezgisel Algoritmalar ile YSA Eğitimi Yapılan Yöntemlerin Sınıflandırma Performansları Karşılaştırma Tablosu.....76
Tablo 3.11 Tek Örneklem-t testi p Değerleri 78
Tablo 3.12 Eşleştirilmiş-t testi p Değerleri 80
Tablo 3.13 Oxford Mutluluk Veri Kümesi En İyi Deney Tasarım Parametreleri 81
Tablo 3.14 Oxford Mutluluk Veri Kümesi MSE ve Doğruluk Değerleri.....82
Tablo 3.15 Sınıflandırıcı Yöntemlerin Sınıflandırma Performansları Karşılaştırma Tablosu.....87
Tablo 3.16 Oxford Mutluluk Veri Kümesi için Eşleştirilmiş-t testi p Değerleri 88
Tablo 3.17 Oxford Mutluluk Veri Kümesi için Tek Örneklem-t testi p Değerleri.....89

ŞEKİLLER LİSTESİ

Şekil 2.1 Biyolojik Sinir Sistemi Blok Gösterimi 15
Şekil 2.2 Biyolojik Sinir Sistemi Matematiksel Gösterimi 15
Şekil 2.3 Biyolojik Sinir Hücresi Genel Yapısı 16
Şekil 2.4 Hata Bazlı ve İterasyon Sayısı Bazlı Durdurma Kriterleri Gösterimi 20
Şekil 2.5 YSA Öğrenme İşleminin Durdurulması 20
Şekil 2.6 Optimum Modelin Seçilmesi 21
Şekil 2.7 Hatayı Minimize Edecek Optimum Ağırlıkların Gösterimi 22
Şekil 2.8 YSA'da Sık Kullanılan Aktivasyon Fonksiyonları ve Özellikleri 23
Şekil 2.9 YSA ile Öngörü Modellenmesi 26
Şekil 2.10 İleri Beslemeli YSA Modeli 28
Şekil 2.11 Çok Katmanlı Algılayıcı Modeli 31
Şekil 2.12 Küresel Çözüm Gösterimi 32
Şekil 2.13 a: Düşük Öğrenme Oranı b: Büyük Öğrenme Oranı c: Momentum Terimi Eklenmiş Büyük Öğrenme Oranı 33
Şekil 2.14 Sınıflandırma İşleminde, Gizli Katman Sayısının İki Boyutlu Örnek Uzayındaki Rolünün Geometrik Gösterimi 35
Şekil 2.15 Metasezgisel Algoritmaların Sınıflandırılması 38
Şekil 2.16 Uçan Sincapların Süzülme Hareketinin Kavramsal Modeli 46
Şekil 2.17 Uçan Sincapların Doğrusal Süzülme Yolunun Gösterimi 46
Şekil 2.18 sf=18 Olduğu Varsayılarak Uçan Sincapların Yol Uzunluğunun Ölçeklenmesi 48
Şekil 3.1 Mutluluk Verilerine Ait YSA Yapısı 62
Şekil 3.2 Wine Veri Kümesi için Eğitim ve Test Doğrulukları 67
Şekil 3.3 Ionosphere Veri Kümesi için Eğitim ve Test Doğrulukları 68
Şekil 3.4 WDBC Veri Kümesi için Eğitim ve Test Doğrulukları 69
Şekil 3.5 Zoo Veri Kümesi için Eğitim ve Test Doğrulukları 70
Şekil 3.6 Glass Veri Kümesi için Eğitim ve Test Doğrulukları 70
Şekil 3.7 Nursery Kümesi için Eğitim ve Test Doğrulukları 71
Şekil 3.8 Her Bir Veri Kümesine Ait MSE Değerleri 72

Şekil 3.9 UCI Veri Ambarından Alınan 6 Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması Histogram Grafiği 74
Şekil 3.10 Oxford Mutluluk Veri Kümesi için Eğitim ve Test Doğrulukları 82
Şekil 3.11 Oxford Mutluluk Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması Histogram Grafiği 83
Şekil 3.12 Oxford Mutluluk Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin MSE Karşılaştırması Histogram Grafiği 84

GİRİŞ

Teknolojinin gelişmesiyle birlikte günümüzde vazgeçilmez olan bilgisayarın, insan gibi karar verme ve öğrenme yeteneklerini kazanması ile kullanım alanları bir hayli artmıştır. Yapay zekâ yöntemlerinin en belirgin özelliği, problemler için çözümler üretebilmek amacıyla örneklerden, benzetmelerden öğrenebilme, genelleme yapabilme, yorumlayabilme ve karar verebilme yeteneklerine sahip olmasıdır. Bu sayede insanlar tarafından çözülmesi zor olan problemleri çözmek amacıyla yapay zekâ yöntemleri kullanılmaya başlanmıştır.

Günümüzde gelişmiş Yapay Zekâ tekniklerinden bir tanesi olan Yapay Sinir Ağları (YSA) karmaşık problemlerin çözümünde oldukça başarılı sonuçlar elde etmektedir ve sınıflandırma problemlerinde sıklıkla kullanılmaktadır. Yapay Sinir Ağları (YSA) insan beyin yapısını temel alınarak ortaya çıkarılmış bir yapıdır. YSA'nın, insan sinir sisteminin bilgisayarlara aktarılarak anlamlı sonuçlar üretme çabalarının bir ürünü olduğu söylenebilir. Yapay Sinir Ağlarının insan beyni gibi öğrenme kabiliyetleri bulunmaktadır ve bu bilgileri genelleyebilirler yani yorumlama kabiliyetleri kazanabilirler.

Yapay sinir ağları bilgi kaynağından öğrenebilme yeteneğine sahiptir. YSA modeli öğrenme sırasında elde ettiği bilgileri, sinir hücreleri arasında bağlantı ağırlıkları olarak saklamaktadır. Bu ağırlıkların güncellenme mekanizması sınıflandırma başarısını doğrudan etkilemektedir. Yapay Sinir Ağlarının ağırlık değerlerinin belirlenmesi ve güncellenmesi ile ağın eğitilmesi mümkündür. Modelin eğitilebilmesi için bu ağırlık değerlerinin kurallar ile dinamik olarak değiştirilebilmesi gerekmektedir. İyi bir öğrenme algoritması kullanılarak eğitilen yapay sinir ağları ise YSA'nın daha iyi performans sergilemesini sağlamaktadır. Bu yüzden YSA ağırlık güncelleme işlemi iyi bir öğrenme algoritması ile gerçekleştirilmelidir.

Metasezgisel algoritmaların birçok farklı alanda başarılı bir şekilde uygulanması ile, yapay sinir ağlarının eğitilmesi işleminde de sıklıkla kullanılmaya başlanmıştır. Literatürde YSA eğitiminin birçok metasezgisel algoritma ile gerçekleştirildiği ve başarılı sonuçların elde edildiği çalışmalar mevcuttur. Metasezgisel algoritmalar, hesaplama gücü, uygulama kolaylığı ve birden çok probleme uygulanabilir olması sebebiyle günümüzde geniş alanda mühendislik problemlerinin çözülmesinde kullanılmaktadır ve geleneksel algoritmalara oranla daha iyi sonuçlar vermektedir.

Bu tez çalışmasında yapay sinir ağlarının eğitilmesinde son yıllarda geliştirilen Sincap Arama Algoritması kullanılmıştır. Sincap Arama Algoritması Jain ve arkadaşları tarafından 2019 yılında doğadan esinlenilerek geliştirilmiş yeni, basit ve güçlü bir küresel optimizasyon algoritmasıdır [1]. Algoritmanın performansı UCI makine öğrenme deposundan (https://archive.ics.uci.edu/ml/index.php) alınan 6 veri kümesi ve bir gerçek hayat verisi olan Oxford Mutluluk Ölçeği veri kümesi ile değerlendirilmiştir.

Mutluluk, son yıllarda birçok disiplin için önem kazanan çalışma alanlarındandır. Günümüzde tüm dünyada olduğu gibi, Türkiye'de de mutluluk ile ilgili araştırmalara artan bir ilgi bulunmaktadır. Özellikle 2000'li yıllardan sonra mutluluk ile ilgili araştırmalarda artış görülmektedir [2]. Konu ile alakalı en sık kullanılan araçlardan bir tanesi Oxford Mutluluk Ölçeğidir. Oxford Mutluluk Ölçeği Likert anketten oluşan bir ölçme aracıdır. Gerçek hayat verisi oluşturulurken bu ölçek kullanılmıştır.

Çalışmada kullanılan veri kümeleri, basit bölümleme ile eğitim ve test kümelerine ayrılmıştır. Verilerin %70'lik kısmı eğitim verilerini oluştururken %30'luk kısmı test verisi olarak kullanılmıştır. Karşılaştırmalarda ele alınan tüm veri kümeleri için bölümleme işlemi aynı şekilde yapılmıştır.

Modelin eğitiminin ardından, modelin hiç görmediği test verileri üzerinde performans analizleri gerçekleştirilmiştir. Bu çalışmada etkinlik ölçütü olarak Hata Kareleri Ortalaması (MSE) kullanılmıştır. MSE değeri kullanılarak hesaplanan ağırlık değerlerinin gerçek değerlere ne kadar yaklaştığı gözlemlenmektedir. MSE değerini minimize eden ağırlıkların bulunması ve eğitim ile test verileri için doğruluk oranlarının maksimize edilmesi bu çalışmanın etkinlik kriterleridir.

SSA ve YSA parametrelerinin farklı değerleri için veri kümelerinin sınıflandırma başarısının nasıl etkilendiği deney tasarımı ile gözlemlenmiş ve en iyi parametreler belirlenerek her veri kümesi için sınıflandırma başarıları sayısal olarak incelenmiştir. Çalışmada Sincap Arama Algoritması kullanılarak YSA eğitiminin yapılması ve en iyi parametrelerin belirlenmesi MATLAB ortamında gerçekleştirilmiştir. Tez çalışması ile sincap arama algoritmasının, sınıflandırma problemleri üzerinde yapay sinir ağlarının eğitimindeki başarısı ortaya konmuştur.

1. BÖLÜM

GENEL BİLGİLER VE LİTERATÜR ÇALIŞMASI

1.1 Literatür Araştırması

Bu bölümde çalışmada kullanılan yöntemlerin literatürü hakkında bilgi verilmektedir.

1.1.1 Yapay Sinir Ağları Literatürü

Yapay zekâ, çoğunlukla insana özgü olan görevlerin (anlama, anlam çıkarma, genelleme yapma ve geçmiş verileri kullanarak öğrenme yapabilmek gibi) görevlerin bir makine tarafından yapılabilmesi yeteneği olarak tanımlanmıştır [3]. Slage'nin tanımına göre yapay zekâ; sezgisel programlama temelli bir yaklaşımdır [4]. Popov'un tanımına göre yapay zeka; insanın yapabildiklerini bilgisayara kazandırabilmektir [5]. Axe'ın yapay zekâ tanımı ise; akıllı programları hedefleyen bir bilimdir [6]. Genesereth ve Nilsson ise yapay zekânın, akıllı davranışlar hakkında yapılmış bir çalışma olduğunu belirtmiştir. Asıl amacı, doğada var olan akıllı davranışlardan yapay bir şekilde üretmeyi hedefleyen bir kuramın oluşturulabilmesidir [7].

Turing makineleriyle ile birlikte temeli atılmış ve yapay zekâ alanında en fazla araştırmaya sahip konulardan bir tanesi "Yapay Sinir Ağları (YSA)"dır [8]. İlk yapay sinir ağının modellenmesi çalışması Warren McCulloch ve Walter Pitts tarafından 1943 yılında ortaya atılmıştır. McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden yola çıkarak, basit bir ağ modeli gerçekleştirmiştir. 1954 yılında B.G. Farley ve W.A. Clark bir ağda girişlere karşılık çıkış üreten ve aynı zamanda bu uyarılara uyum sağlayabilen bir model oluşturmuşlardır. 1960 yılında ilk sinirsel bilgisayarın ortaya çıkmasıyla beraber 1963 yılında fark edilen eksiklikler 1970'li ve 1980'li yıllarda doğrusal olmayan

modellerin geliştirilmesine kadar tamamlanamamıştır. Bu yüzden bu yıllarda YSA'ya olan ilgi azalmış ancak doğrusal olmayan modeller geliştirildikten sonra tekrar odak noktası haline gelmiştir. [9]

YSA, yapay zekâ uygulamalarından biri olmakla beraber, insan beyninin çalışma yapısını taklit etmektedir. Verilerin analiz edilip, algoritmalar aracılığıyla bu verilerden yeni bilgiler oluşturmak için geliştirilmiş mantıksal sonuçlar elde edilebilen yazılımlardır. YSA, biyolojik sinir hücrelerinden oluşan gerçek ağları taklit etme yeteneğine sahip yapay ağlardır [10]. YSA'nın en çok tanınan tanımlarından bir tanesi Teuvo Kohonen'e aittir: "Yapay sinir ağları, paralel bağlanmış birçok sayıda basit elemandan oluşan, gerçek dünya nesneleri ile biyolojik sinir sisteme benzer yolla etkileşim kuran hiyerarşik bir organizasyondur" [11].

YSA çalışma yapısı öncelikle verilerle ilgili bilgi toplayıp, genelleme yaparak daha önce karşılaşmadığı veriler hakkında karar verebilmek olarak özetlenebilir. Böylece geniş uygulama alanı mevcut olan YSA karmaşık problemleri çözebilme konusunda yüksek başarı ortaya koymaktadır [8].

Sonuç olarak Yapay Sinir Ağları, uzun yıllardır yapılan çalışmalarda başarılı sonuçlar elde etmiştir ve geliştirilen teknikler güncel olarak da kullanılmaya devam etmektedir.

1.1.2. Metasezgisel Algoritmalar Literatürü

Literatürde ilk defa 1980'li yıllarda Glover tarafından kullanılan "Metasezgisel" terimi, "Modern Sezgisel" olarak da bilinmektedir. Glover'ın metasezgisel terimini kullandığı bu çalışmada Tabu Arama algoritması sunulmuştur. Bu algoritma ile komşuluk arama yaklaşımına hafıza kullanımı eklenerek üst bir arama stratejisi oluşturulmuştur [12].

Doğadaki bir canlının davranışının modellenmesi ile oluşturulmuş ilk metasezgisel algoritma "Karınca Koloni Algoritması"dır. Karınca Koloni Algoritması 1990'lı yıllarda Dorigo tarafından, karınca sürü zekasından esinlenilerek geliştirilmiştir [13]. 2008 yılında ise Socha ve Dorigo tarafından bu metasezgisel algoritma sürekli optimizasyon problemleri için uyarlanmıştır [14]. Sürü zekasından esinlenilerek geliştirilen bir diğer algoritma ise "Parçacık Sürü Optimizasyonu"dur. James ve Eberhart tarafından 1995 yılında geliştirilen bu metasezgisel algoritma kuş ve balık sürülerinin sosyal

davranışlarından esinlenilerek geliştirilmiştir. Kolay, anlaşılır, başarı oranı yüksek olan bu algoritma başlangıç parametrelerinin de azlığı sebebiyle literatürde oldukça sık kullanılmaktadır [15].

Storn ve Price tarafından geliştirilen Diferansiyel Gelişim Algoritması popülasyon tabanlı evrimsel bir algoritmadır ve literatürde diferansiyel evrim algoritması tabanlı geliştirilmiş birçok yöntem önerilmiştir [16].

21. yüzyılın başlarında ilk olarak Geem ve arkadaşları su dağıtımı, taşıma modelleme ve zamanlama gibi çeşitli optimizasyon problemlerinin çözümünde yaygın olarak kullanılan harmoni arama (HS) algoritmasını geliştirmişlerdir [17]. Karaboğa (2005) çalışmasında arıların yiyecek arama davranışlarını modelleyerek yapay arı kolonisi (ABC) algoritmasını geliştirmiştir [18].

Yang (2008)'deki çalışmasında ateşböceklerinin yanıp-sönen ışıklarından esinlenerek ateşböceği algoritmasını geliştirmiştir. Ateşböcekleri biyoluminesans -canlı bir organizma tarafından kimyasal bir reaksiyon esnasında kimyasal enerjinin ışık enerjisine dönüştürülerek ışık üretilmesi ve ışık yayılması- ile bir sinyalizasyon sistemine sahiptir [19].

Rashedi ve arkadaşları ise, ağırlık ve kütle etkileşimleri kanununa dayalı yer çekimi arama algoritmasını geliştirmişlerdir [20].

2009'da Cambridge Üniversitesinden Xin She Yang ve India Raman Mühendislik Fakültesinden Suash Deb guguk kuşlarının yumurtalama ve yuva seçme davranışları ile 'levy'uçuşlarına dayalı verimli bir Guguk Kuşu Arama Algoritmasını tanıtmışlardır. Guguk kuşu arama bazı guguk kuşu türlerinin kuluçka asalaklığı üzerine dayandırılmıştır [21].

Bunun yanında, teknoloji ve mühendislik alanında optimizasyona duyulan ihtiyacın artmasından dolayı yaygın olarak üstünde çalışılan optimizasyon algoritmalarının yanında çok sayıda yeni algoritma örneğin; Karides Sürüsü Optimizasyon Algoritması [22], Göçmen Kuşlar Optimizasyon Algoritması [23], Meyve Sineği Optimizasyon Algoritması [24] ve Sosyal Örümcek Algoritması [25] önerilmiştir.

Son yirmi yılda farklı disiplinlerden ilham alınarak çok sayıda sezgisel metot geliştirilmiştir. Özellikle biyolojiden ilham alınan algoritmalar üzerine çalışmalar yoğunluk kazanmıştır. Bu ilginin ana nedeni, bu algoritmaların esnek ve çok yönlü olmalarıdır. Optimizasyon son yirmi yılda yapılan ilerlemeye rağmen, hala çok geniş bir araştırma konusudur. Çünkü optimizasyon birçok temel bilimde karşılaşılan problemlerin çözümünde sıklıkla kullanılmaktadır. Özellikle mühendislik ve endüstri uygulamalarının çoğunda kaynaklar (para, zaman, hammadde, işgücü ve enerji gibi) en verimli şekilde kullanılarak amaca (düşük maliyet, maksimum kâr, yüksek kapasite kullanımı ve verimlilik gibi) ulaşmak hedeflenir.

Doğa ilhamlı algoritmalar üzerindeki ilginin hala devam etmesinin sebebi ise optimizasyon alanındaki ihtiyaçların artarak devam etmesi ve şimdiye kadar araştırmacıların doğal sistemlerin sadece çok karakteristik olaylarından ve canlılarından ilham almış olmasıdır. Daha fazla algoritma geliştirmek için doğada ilham alınacak çok daha fazla sistem vardır. Çünkü doğal sistemler milyonlarca yıldır birçok zorlu problemi çözmektedir [26].

Metasezgisel Optimizasyon algoritmaları arasında, sürü tabanlı metasezgisel algoritmalar; hayvanların toplu hareketlerinin gözlemlenmesiyle geliştirilmiş algoritmalardır. Son 10 yılda geliştirilen sürü tabanlı metasezgisel algoritmalara örnek bazı metasezgisel algoritmalar Tablo 1.1'de verilmiştir.

Tablo 1.1 Sürü Tabanlı Metasezgisel Optimizasyon Algoritmaları

Algoritma Adı	**Yıl**
Kaotik Arı Kolonisi Algoritması [27]	2010
Yarasa Algoritması [28]	2010
Guguk Kuşu Arama Algoritması [29]	2011
Karides Sürüsü Algoritması [22]	2012
Yunus Balığı Algoritması [30]	2016
Sosyal Örümcek Algoritması [25]	2013
Gri Kurt Optimizasyon Algoritması	2014
Jaguar Algoritması [31]	2015

Virüs Koloni Arama Algoritması [32]	2016
Balina Optimizasyon Algoritması [33]	2016
Aslan Optimizasyon Algoritması [34]	2016
Çekirge Optimizasyon Algoritması [35]	2017
Karga Arama Algoritması [36]	2016
Baykuş Arama Algoritması [37]	2018
Sincap Arama Algoritması [1]	2019
Xerus Optimizasyon Algoritması [38]	2019
Renk Uyumu Algoritması [39]	2020

1.1.3. Yapay Sinir Ağlarının Metasezgisel Algoritmalar ile Eğitimi Literatürü

Metasezgisel algoritmaların birçok farklı alanda başarılı bir şekilde uygulanması ile, yapay sinir ağlarının eğitilmesi işleminde de sıklıkla kullanılmaya başlanmıştır. Yapay sinir ağları eğitiminin metasezgisel algoritmalar ile yapılması hakkındaki çalışmalar genel olarak bu bölümde belirtilmiştir.

Battiti ve Tecchiolli [40], tabu arama algoritmasını geliştirip Reaktif Tabu Arama (RTS) algoritmasını önererek yapay sinir ağları ile etkisini gözlemlemek için yapay sinir ağı eğitimini bu algoritmayı kullanarak yapmış ve belirli veri kümeleri ile performanslarını incelemişlerdir.

Sexton ve ark. [41], optimizasyonda küresel aramanın başarısını yükseltmek amacıyla yapay sinir ağının tavlama benzetimi algoritması ile eğitmişlerdir. Sonuç olarak YSA için küresel arama algoritmalarının yanı sıra tavlama benzetimi algoritmasının da yüksek başarı elde ettiğini ispatlamışlardır.

Mendes ve ark. [42] ise yaptığı çalışmada, parçacık sürü optimizasyonu ile YSA eğitimi yapmış ve farklı ağ mimarilerini birbiri ile karşılaştırarak analizini gerçekleştirmişlerdir.

Ilonen ve ark. [43], birçok hastalığın sınıflandırılma işlemi için diferansiyel gelişim (DE) algoritması ile yapay sinir ağı eğitimi üzerinde çalışmışlardır. Sonuçlar incelendiğinde önerilen modelin daha başarılı sonuç verdiği gözlemlenmektedir.

Blum ve Socha [44] çalışmalarında, sağlık alanında çalışmalarda en sık kullanılan 3 veri kümesi üzerinde sınıflandırma işlemini başarılı bir şekilde gerçekleştirebilmek amacıyla karınca koloni algoritması (ACO) kullanarak sinir ağını eğitmişlerdir. Bu çalışma ile karınca koloni algoritması optimizasyon tekniği olarak kullanılabilir ve etkin bir algoritma olarak belirlenmiştir.

Zhang ve ark. [45] adaptif parçacık sürüsü optimizasyon algoritması (APSO) ve geleneksel geri-yayılım algoritmasını kullanarak melez algoritma oluşturmuş ve bu melez algoritma ile YSA eğitimi yapmışlardır. Daha sonra oluşturdukları bu ağı, çeşitli veri kümeleri kullanarak test etmişlerdir. Elde edilen sonuçlara göre oluşturulmuş bu melez algoritma her iki algoritmaya göre yakınsama hızı ve başarısı konusunda daha yüksek başarı elde etmiştir.

Özbakir ve ark. [46], sınıflandırmada kural çıkarımı yapmak ve veriler arasındaki gizli bilgileri öğrenebilmek amacıyla karınca koloni optimizasyon algoritması tabanlı yeni bir metasezgisel algoritma ile YSA eğitimi gerçekleştirmişlerdir. Sınıflandırma veri kümeleri ile elde edilen deney sonuçları dikkate alındığında, önerilen bu yeni modelin doğru keşif yapabildiği sonucuna ulaşılmıştır.

Zamani ve Sadeghian [47], sınıflandırma yapmak amacıyla oluşturdukları sinir ağının eğitimini parçacık sürü optimizasyon algoritması kullanarak gerçekleştirmişler ve farklı veri kümeleri üzerinde başarı oranlarını incelemişlerdir. Bu çalışma ile yapılmış YSA eğitiminin yüksek başarı elde ettiği gözlemlenmiştir.

Kulluk ve ark. [48] ise, 2 farklı veri kümesi üzerinde küresel harmoni arama algoritması ile yapılmış olan YSA eğitimi ile elde edilen yeni sinir ağının analizini gerçekleştirmişlerdir. Sonuçlarda önerilen bu sinir ağı modelinin eğitim ve test verileri üzerinde yüksek oranda başarı sağladığı gözlemlenmiştir.

Zanchettin ve ark. [49], çalışmalarında genetik algoritma, tabu arama algoritması ve tavlama benzetimi algoritmalarının birleşiminden meydana gelmiş melez algoritma ile YSA eğitimini elde etmişlerdir. Bu sayede diğer algoritmalar teker teker bu veri kümelerine uygulandığında elde edilen başarı oranının geliştirilmiş sinir ağının başarısından daha düşük olduğu sonucu elde edilmişlerdir.

Vasquez [50] yaptığı çalışmada, guguk kuşu arama algoritması kullanarak YSA eğitimini gerçekleştirmiş ve 5 farklı veri kümesi üzerinde test etmiştir. Bu 5 veri kümesinden 1 tanesinde istenilen başarı oranı yakalanamamış ancak diğer veri kümeleri üzerinde istenilen seviyede doğruluk elde edilmiştir.

Mirjalili ve ark. [51], yerçekimi arama algoritması (GSA) ve parçacık sürü optimizasyon (PSO) algoritmasının birleşiminden oluşan bir melez algoritma elde edip bu yeni melez metasezgisel algoritma ile YSA eğitimi yapmışlardır. Bu çalışma sonucunda ise yakınsama hızında ve doğruluk oranlarında daha yüksek başarı elde edildiği belirlenmiştir.

Kulluk [52] yaptığı çalışmada, yeni melez bir algoritma olarak harmoni arama (HS) ve av arama (HuS) algoritmalarının birleşiminden oluşan yeni bir melez algoritma ile sinir ağının eğitimi işlemi gerçekleştirmiş ve 8 farklı veri kümesi kullanarak diğer metasezgisel algoritmalar ile karşılaştırmıştır. Çalışmada oluşturulmuş melez metasezgisel algoritma geleneksel algoritmaların birçoğundan yüksek başarı sağlamıştır.

Pereira ve ark. [53] ise, sosyal örümcek algoritması kullanarak sinir ağı eğitimi yapmışlardır ve bu melez algoritmayı sağlık verilerinin sınıflandırma amacıyla kullanmışlardır. Oluşturulan bu algoritma diğer tekniklerle karşılaştırıldığında genel olarak ortalama başarı sağlamıştır.

Sarangi ve ark. [54] yapay arı kolonisi algoritması kullanarak sinir ağı eğitimini gerçekleştirerek 5 veri kümesi üzerinde test etmişlerdir. Bu çalışmada önerilen algoritmanın sinir ağı eğitimi diğer algoritmalara kıyasla daha başarılı sonuçlar elde ettiği sonucu elde edilmiştir.

Chen ve ark. [55] yaptıkları çalışmada parçacık sürü optimizasyonu algoritması ve guguk kuşu arama algoritmasını başarılı bir şekilde birleştirerek sinir ağı eğitimi yapmışlardır. Her iki algoritmanın başarılı yönleri birleştirilerek geliştirilmiş bu yöntem fonksiyon tahmini ve sınıflandırma durumlarında kullanılmıştır. Bu iki algoritma ile yapılan sinir ağı eğitimi teker teker bu iki algoritmaya göre başarı oranı yükselmiştir.

Jaddi ve ark. [56] ait çalışmada ise, guguk kuşu arama algoritması kullanarak yapay sinir ağlarının eğitimi yapılmıştır. Algoritma yakınsama kabiliyetini artırmak için

modifikasyonlar gerçekleştirilmiştir. Eğitimi yapılmış ağ ile zaman serisi verileri üzerinde test edilmiştir. Önerilen bu modelde elde edilen sonuç başarı oranlarının yükseldiği şeklindedir.

Nuaimi ve Abdullah [57], parçacık sürü algoritması ve sürü zekâsına dayalı yapay arı kolonisi algoritması kullanarak bu iki algoritma ile yeni bir melez algoritma önermişlerdir. Oluşturdukları bu melez metasezgisel algoritma ile sinir ağının eğitimi gerçekleştirmişlerdir. Elde edilen sonuçlarda önerilen bu modelin daha başarılı olduğu sonucuna varılmıştır.

Dash [58] yapmış olduğu çalışmada, geliştirilmiş sıçrayan kurbağa algoritması ile sinir ağı eğitimi gerçekleştirerek, üç farklı veri kümesinde uygulamışlardır. Bu çalışmada elde edilen sonuçlar ortalamanın üstünde başarı oranını yakalamıştır.

Bütün bu çalışmalar dikkate alındığında, metasezgisel algoritmalar kullanarak yapay sinir ağı eğitiminin ve melez metasezgisel algoritma oluşturma ile alakalı çalışmaların oldukça fazla olduğu görülmektedir. Geleneksel yöntemlere kıyasla oluşturulmuş bu melez algoritmalar genel olarak daha başarılı sonuçlar elde etmiştir. Başarıyı arttırmak için bazı çalışmalarda gerekli modifikasyonlar yapılmıştır. Birçok veri kümesinde test edilen bu yeni metasezgisel algoritmalar, geleneksel yöntemlere alternatif olarak konunun literatürünün şekillenmesinde büyük rol oynamıştır.

1.1.4. Mutluluk Literatürü

Psikoloji alanında en önemli konulardan bir tanesi olan “Mutluluk” günümüzde araştırmacıların ilgisini oldukça çeken konulardan bir tanesidir. Doğan ve Sapmaz’ın yorumuna [59] göre mutluluk, olumlu olarak sınıflandırılabilen duyguların daha fazla, olumsuz olarak sınıflandırılabilen duyguların ise olumlu duygulara oranda daha az yaşanması ve bununla birlikte yaşamdan yüksek doyum alınmasıdır. Hills ve Argle’e [60][61] göre ise mutluluk, var olma ve bir bütünün parçası olma ile alakalıdır. Diener’e [62] göre mutluluk, anlık olumlu ve olumsuz durumlardan etkilenip mutlu veya mutsuz olmakla alakalı olmamakla beraber yaşamın gidişatını belirleyen uzun süreli duygularla ilgilidir. Bu yüzden mutluluk ile alakalı araştırmalar kısa süren veya anlık duygulardan çok uzun süren ve bu süre zarfında hissedilen duyguları konu edinmelidir.

Bentham'ın kısa ve öz mutluluk tanımı ise: Hayattan alınan keyif, zevk, hüzün ve acıların toplamıdır [63]. Mutluluğun başka bir tanımı ise, bir bireyin yaşamının tamamını genel olarak değerlendirdiğinde, ne düzeyde kaliteli, olumlu duygulardan oluşan, başarılı ve verimli bir hayat sürdüğü ile ilgilidir [63].

Mutluluk ve iyi olma konusu psikolojide uzun süredir çalışılmaktadır. Ekonomiden sosyolojiye, bilgisayar bilimlerinden pazarlamaya kadar mutluluk kavramı üzerine çok farklı disiplinlerde, araştırmacılar çalışmalar yapmaya devam etmektedirler [64]. Birçok araştırmacı, mutluluğu anket yöntemi ile ölçmüştür. Literatürdeki bazı önemli ölçekler şunlardır: Mutluluk Tek Madde Ölçeği (Single-Item Measurement of Happiness) [65]; Öznel Mutluluk Ölçeği (Subjective Happiness Scale) [66]; Yaşam Memnuniyeti Ölçeği (Satisfaction with Life Scale, SWLS) [62]; Oxford Mutluluk Ölçeği (Oxford Happiness Questionnaire) [60]; ve Gayri Safi Milli Mutluluk Endeksi (Bhutan's Gross National Happiness Index) [67]. Bu anketlerde katılımcıların mutluluklarını ve yaşam memnuniyetlerini anlamaya yönelik olarak sorular sorulmaktadır. Cevaplar ise çalışmalarda belirlenen ölçeklere göre, üç ölçekliden on ölçekliye kadar değişebilmektedir.

Küresel çalışmaların yanı sıra son zamanlarda, Türkiye'de de mutluluk araştırmalarına artmakta olan bir ilgi bulunmaktadır. Özellikle 2000'li yıllardan itibaren mutluluk alanında araştırmaların sayısında oldukça fazla artış görülmektedir [2]. Konu ile alakalı en sık kullanılan araçlardan bir tanesi Oxford Mutluluk Ölçeğidir. Oxford Mutluluk Ölçeği Likert anketten oluşan bir ölçme aracıdır. Önceki yıllarda yapılan araştırmalarda bu ölçeğin Türkçe formunun yüksek derecede güvenilirliğe sahip olduğu kanıtlanmıştır [59].

Oxford Mutluluk Ölçeği, tüm dünya genelinde mutluluk alanında yapılan araştırmalarda çok yaygın bir şekilde kullanılmaktadır. Ölçek Çince'ye [68], Japonca'ya [69], İbranice'ye [70], Farsça'ya [71] ve Almanca'ya [72] çevrilmiş ve psikometrik özellikleri incelenmiştir. Oxford mutluluk ölçeği Hills ve Argyle [60] tarafından geliştirilmiş 29 maddelik bir ölçme aracıdır.

Jalloh, [73] mutluluk konusunun literatürdeki yerini özetlerken, mutluluk ölçümünde en yaygın kullanılan ölçekleri ve araçları da dahil ederek mutluluk kavramını açıklamaktadır. En temel tanımlama ile mutluluk, pozitif duygusal iyi olmadır ve birinin

subjektif iyi olmasını açıklamak için kullanılmaktadır. Birinin hayatı ile ilgili hisleri ve duyguları içeren subjektif bir değerlendirmeye dayanır.

1.2 Araştırmanın Amacı ve Önemi

Gelişmiş Yapay Zekâ tekniklerinden bir tanesi olan Yapay Sinir Ağları (YSA) karmaşık problemlerin çözümünde oldukça başarılı sonuçlar elde etmektedir ve sınıflandırma problemlerinde sıklıkla kullanılmaktadır. YSA kullanarak başarılı sonuçlar elde etmek için ağ eğitilmelidir. Yapay Sinir Ağlarının ağırlık değerlerinin belirlenmesi ve güncellenmesi ile ağın eğitilmesi mümkündür. İyi bir öğrenme algoritması kullanılarak eğitilen yapay sinir ağları ise YSA'nın daha iyi performans sergilemesini sağlamaktadır. Metasezgisel algoritmalar, geleneksel algoritmalara oranla daha iyi sonuç vermektedir. Literatürde YSA eğitiminin birçok metasezgisel algoritma ile gerçekleştirildiği ve başarılı sonuçlar elde edilen çalışmalar mevcuttur.

Bu tez çalışmasında yapay sinir ağlarının eğitilmesinde son yıllarda geliştirilen Sincap Arama Algoritması kullanılmıştır. Algoritmanın performansı UCI makine öğrenme deposundan alınan 6 veri kümesi ve bir gerçek hayat verisi olan Oxford Mutluluk Ölçeği veri kümesi ile değerlendirilmiştir.

Mutluluk, son yıllarda birçok disiplin için önem kazanan çalışma alanlarındandır. Günümüzde tüm dünyada olduğu gibi, Türkiye'de de mutluluk ile ilgili araştırmalara artan bir ilgi bulunmaktadır. Özellikle 2000'li yıllardan sonra mutluluk ile ilgili araştırmalarda artış görülmektedir [2]. Konu ile alakalı en sık kullanılan araçlardan bir tanesi Oxford Mutluluk Ölçeğidir. Oxford Mutluluk Ölçeği Likert anketten oluşan bir ölçme aracıdır. Önceki yıllarda yapılan araştırmalarda bu ölçeğin Türkçe formunun yüksek derecede güvenilirliğe sahip olduğu kanıtlanmıştır [59]. Gerçek hayat verisi oluşturulurken bu ölçek kullanılmıştır.

Modelin eğitiminin ardından, modelin hiç görmediği test verileri üzerinde performans analizleri gerçekleştirilmiştir. Çalışma sonucunda Sincap Arama Algoritma ile eğitilen YSA'nın en başarılı sınıflandırma doğruluğunu oluşturan ağırlıklar elde edilmiştir. SSA kullanarak, en iyi ağırlıkların elde edilmesi ile YSA'nın sınıflandırma başarısı arttırılmıştır. Çalışmada yeni ve güncel olan Sincap Arama Algoritması kullanılarak Yapay Sinir Ağı eğitimi ve Deney Tasarımı aşamaları MATLAB kullanılarak kodlanmıştır.

2. BÖLÜM

YÖNTEM VE MATERYAL

Bu bölümde Yapay Sinir Ağları, Metasezgisel Algoritmalar ve bu çalışmada kullanılan metasezgisel algoritma olan Sincap Arama Algoritmasının detayları hakkında bilgi verilmektedir.

2.1 Yapay Sinir Ağları

2.1.1 Biyolojik Sinir Hücresi ve Yapay Sinir Ağları

Biyolojik Sinir Sistemi; beyin tarafından yönetilen ve sinirlerden gelen bilgileri yorumlayarak uygun kararlar veren bir sistemdir. Biyolojik Sinir sisteminde yer alan ana elemanlar Nöron olarak adlandırılan sinir hücreleridir. Biyolojik sinir hücresi Dentrit'ler (Dendrite), Hücre Gövdesi (Soma), Akson'lar (Axon) ve Sinaps'lardan (Synapse) oluşmaktadır [74].

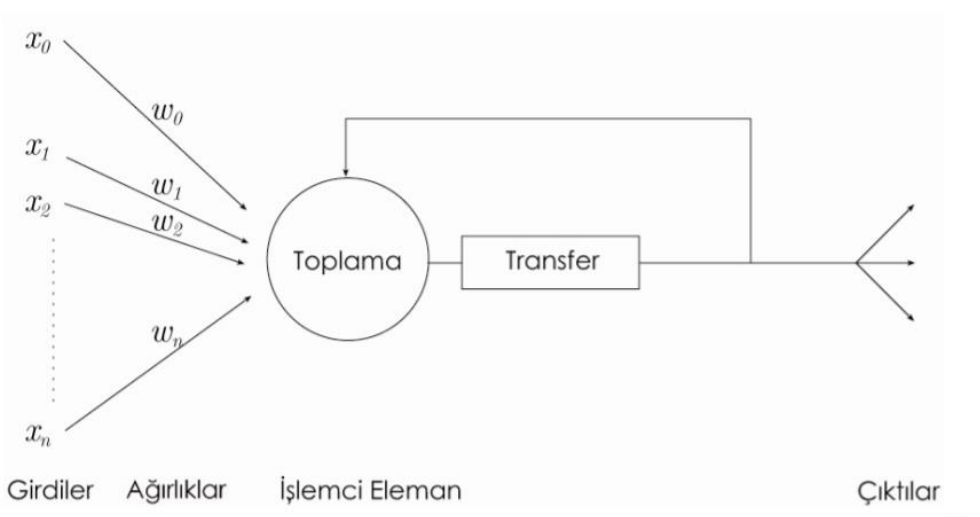
Biyolojik sinir sisteminde öğrenme, nöronlar arasındaki sinaptik bağlantıların düzenlenmesiyle meydana gelmektedir. Nöronlar, çok sayıda bulunan dentritler aracılığıyla girdi sinyalini almaktadır. Girdiler toplanıp, dendritler aracılığıyla nöronun gövde kısmına ulaştırılmaktadır. Bu girdi, belirli bir eşik değerini aştığı zaman, hücre içi kararlılık durumu bozulmaktadır, hücre diğer hücrelere aksonu vasıtasıyla bu sinyali iletmektedir. Biyolojik sinir sistemine ait bu yapı, yapay sinir ağlarını oluşturan yapay nöronlar için bir model oluşturmaktadır [75]. Sinir hücrelerine iletilen bu sinyaller ile uyumlu olan hücreler arasındaki ilişkilerin değiştirilmesi ya da hücreler arasında yeni ilişkiler kurulması yapay sinir ağlarında "öğrenme" işlemine karşılık gelmektedir [76]. Bir nöron, gelen girdiler toplayan ve yalnızca girdilerin toplamı eşik değerini geçtiğinde

bir çıktı üreten işlem elemanlarıdır. Sıklıkla kullanılmakta olan beş adet eşik fonksiyonu bulunmaktadır. Bu eşik fonksiyonları; doğrusal, rampa, basamak, sigmoid ve tanh(x) fonksiyonlarıdır [75]. Biyolojik Sinir Sistemi blok gösterimi Şekil 2.1'de verildiği gibidir.

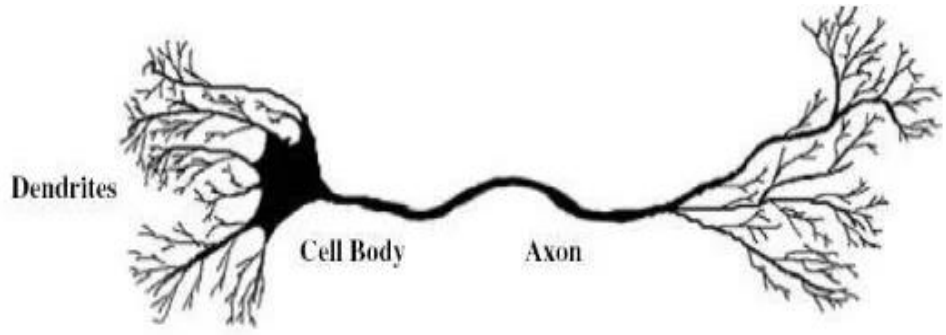


Şekil 2.1 Biyolojik Sinir Sistemi Blok Gösterimi

Yapay sinir ağlarının ana elemanları, biyolojik sinir sisteminde var olan nöronların işleyişlerini gerçekleştiren Yapay Nöronlardır. Bu nöronlar, aralarında bağlantılar oluşturarak ve katmanlar halinde gruplandırılarak YSA'yı oluştururlar. Şekil 2.2'de gösterilen biyolojik sinir sisteminin matematiksel gösterimi McCullogh ve Pitts [77] tarafından tanımlanmıştır. Yapay sinir ağları bu mantıkla çalışmakta olan nöronların birbirleriyle bağlantısı sayesinde ağa dönüştürülerek oluşturulmaktadır. Şekil 2.3'te ise biyolojik sinir hücresinin genel yapısı gösterilmektedir.



Şekil 2.2 Biyolojik Sinir Sistemi Matematiksel Gösterimi



Şekil 2.3 Biyolojik Sinir Hücresi Genel Yapısı

Mimari yapı olarak değerlendirildiğinde YSA genel olarak girdi katmanı, gizli katmanları ve çıktı katmanından oluşur.

Biyolojik Sinir sistemi elemanları ve yapay sinir ağlarına uyarlanması Tablo 2.1'de verildiği gibidir. Biyolojik sinir sistemi bölümlere parçalanmış ve yapay sinir ağında bir karşılığı verilmiştir [76].

Tablo 2.1 Biyolojik Sinir Sistemine ait Elemanlar ve Yapay Sinir Ağındaki Karşılığı

Sinir Sistemi	**Yapay Sinir Ağı**
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

2.1.2 Yapay Sinir Ağları Çalışma Yapısı, Özellikleri ve Kullanım Alanları

Yapay Sinir Ağları genel olarak 3 farklı katman yapısına sahiptir. Bu katmanlar ve özellikleri aşağıda belirtildiği gibidir;

- ***Giriş Katmanı (Input Layer);*** Bilgi girişinin yapıldığı katmandır. Bu katmanda hiçbir işlem yapılmadan katmana gelen bilgi gizli katmanlara iletilir. Her işlem elemanından yalnızca bir tane girdisi ve bir tane çıktısı vardır.

- ***Gizli Katmanlar (Hidden Layers);*** Girdilere karşılık çıktıları üretecek matematiksel işlemlerin gerçekleştiği katmanlardır. Giriş katmanından çıkış katmanına bilgi aktarırlar. Bir YSA içerisinde birçok gizli katman bulunabilir.

- ***Çıkış Katmanı (Output Layer);*** Gizli katmanlarda üretilen sonucu dışarı aktaran ve çıktıyı oluşturan katmandır.

Geleneksel algoritmalar ve YSA'nın karşılaştırılması Tablo 2.2'de verildiği gibidir [78].

Tablo 2.2 Geleneksel algoritmalar ve YSA'nın Karşılaştırılması

Geleneksel Algoritmalar	**Yapay Sinir Ağları**
Çıkışlar, koyulan kurallara girişlerin uygulanması ile elde edilir.	Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur.
Bilgiler ve algoritmalar kesindir.	Deneyimden yararlanır.
Hesaplama; merkezi, eş zamanlı ve ardışıktır.	Hesaplama; toplu, eş zamansız ve öğrenmeden sonra paraleldir.
Bellek paketlenmiş ve hazır bilgi depolanmıştır.	Bellek ayrılmış ve ağa yayılmıştır.
Hata toleransı yoktur.	Hata toleransı vardır.
Nispeten hızlıdır.	Yavaş ve donanıma bağımlıdır.

Yapay sinir ağlarının genel avantajları aşağıdaki gibi sıralanabilir [75].

- Lineer olmayan yapısı
- Girdiler ve çıktıların eşleşmesi ile oluşan tasarımlar
- Uyum sağlayabilmesi ve öğrenebilmesi
- Hataya karşı toleranslı olması

YSA mühendislikten, ekonomiye birçok farklı alanda kullanılmaktadır. Yapay sinir ağlarının gerçek hayatta sıklıkla kullanılan uygulama alanlarına örnekler aşağıdaki gibidir [79, .

- Tahminleme, Optimizasyon, Kümeleme, Sınıflandırma, Örüntü Tanımlama
- Zaman Serileri, Regresyon
- Veri Madenciliği, Veri ilişkilendirme ve yorumlama
- Filtreleme, Sinyal İşleme, Derecelendirme, Fiziksel Sistem Modelleme ve Yönetimi
- Hastalık Teşhisi, Doku Analizi, Koku Analizi ve Tanımı
- Yazılım Hataları, Üç Boyutlu Nesne Tanıma
- İflas Tahmini, Ekonomik Gösterge Tahminleme, Dolandırıcılık Algılama

Yapay sinir ağının özellikleri ise aşağıda listelenmektedir;

- Doğrusal Olmama
- Paralel çalışabilme ve buna bağlı hız
- Öğrenme ve genelleme
- Hatayı tolere edebilme ve esneklik
- Çok sayıda parametre ve değişken ile çalışabilme
- Analiz ve tasarım kolaylığı

Örneklerin Toplanması: Ağın girdi değerlerine karşılık çıktı değerlerini bulması için eğitim verilerinin ve test verilerinin toplanması gerekmektedir.

Ağın Topolojik Yapısının Belirlenmesi: Ağın doğru tahmin yapabilmesi için ağın yapısının belirlenmesi gerekmektedir. Kaç adet girdiden, ara katmandan oluşacak ve ara

katmanlarda kaç hücre elemanı bulunacak ve kaç çıktı elde edilecek bu parametreler olukça önemlidir. Ağın topolojik yapısı kullanılacak verilere göre seçilmelidir.

Öğrenme Parametrelerinin Belirlenmesi: YSA'nın öğrenme katsayısı, aktivasyon fonksiyonu gibi önemli parametreler bu aşamada belirlenmektedir.

Ağın Başlangıç Değerlerinin Atanması: Ağırlık değerleri ve eşik değerlerinin başlangıç değerine atanması işlemi bu adımda gerçekleştirilmektedir.

Eğitim Verilerinden Örneklerin Seçilmesi ve Ağa Gösterilmesi: Ağın öğrenmeye başlaması için gerekli olan adımdır.

Öğrenme Sırasında İleri Hesaplama Yapılması: Ağa verilen girdi verileri için çıktı değerlerinin hesaplanması işlemi bu adımda gerçekleştirilmektedir.

Elde Edilen Çıktı Değerleri ile Gerçek Çıktı Değerlerinin Karşılaştırılması: Bu adımda hata oranı elde edilerek ağın başarılı bir şekilde çalışıp çalışmadığı gözlenmelidir.

Ağırlıkların Değiştirilmesi: Elde edilen çıktı değerleri ile beklenen çıktı değerleri karşılaştırılarak elde edilen hata oranını azaltmak için ağın ağırlık değerlerinin değiştirileceği adımdır. Ağırlıkların değiştirilip gerçeğe daha yakın sonuçlar elde edilmesi ağın başarısını yükseltmek için oldukça önemlidir ve ağın performansını arttırarak verilerin daha doğru tahmin edilmesine olanak sağlamaktadır.

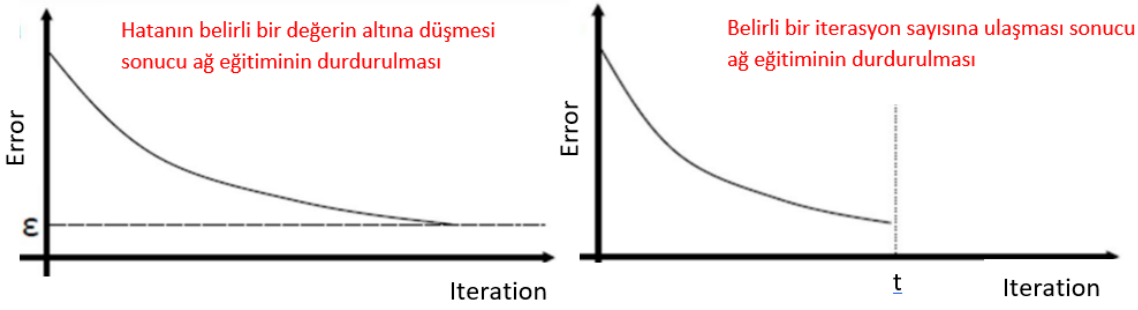
YSA'nın doğru çalışması için bazı gereksinimlerin karşılanması gerekir. Bu gereksinimler aşağıda belirtildiği gibidir;

- Problemi tanımlayabilecek bilgiyi içeren veri kümesi olmalıdır.
- Eğitim ve test aşamalarında kullanılacak yeterli büyüklükte veri kümeleri olmalıdır.
- Problemin iyi anlaşılması gerekir. Problem iyi anlaşılamazsa ağ yapısının kurulması, topoloji seçimi, fonksiyon ve öğrenme kurallarının seçimi doğru olmayabilir.

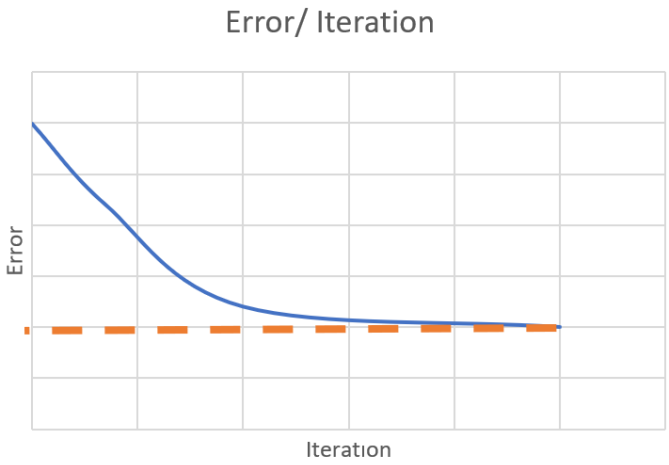
Bu gereksinimler karşılandıktan sonra YSA geleneksel yöntemler ile çözüme ulaşamayan birçok problemin çözümüne olanak sağlamaktadır [80].

2.1.2.1 Durdurma Kriteri

Öğrenme sürecini durdurmak için bazı kriterler vardır. Hatanın istenen belirli bir değerin altına düşmesi sonucu ağ eğitim işlemi durdurulabilir veya durdurma kriteri olarak iterasyon sayısı belirlenebilir. Ağ performansı kontrol edilerek durdurma kriterinin belirlenmesi, öğrenme için önemli bir adımdır. Durdurma kriteri hatanın belirli bir değerin altına düşmesi veya iterasyon sayısı olarak belirlenebilir. Durdurma kriterinin belirlenmesinde problemin yapısı önemli bir etkendir. Optimum durdurma kriteri seçimi hatanın minimum olduğu ve daha fazla azalmadığı durum olarak belirlenmelidir. Şekil 2.6'da durdurma kriterinin optimum seçilmesi için hatanın izlenmesi ve eğitimin durdurulması durumuna ait grafikler verilmiştir.



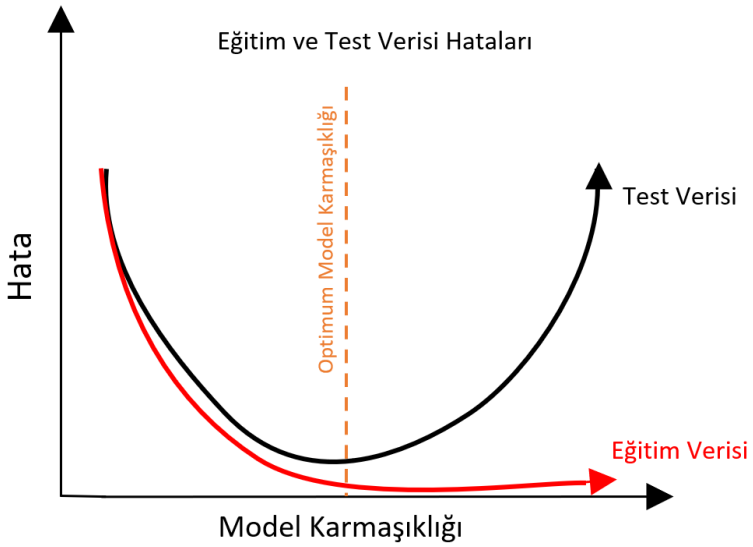
Şekil 2.4 Hata Bazlı ve İterasyon Sayısı Bazlı Durdurma Kriterleri Gösterimi



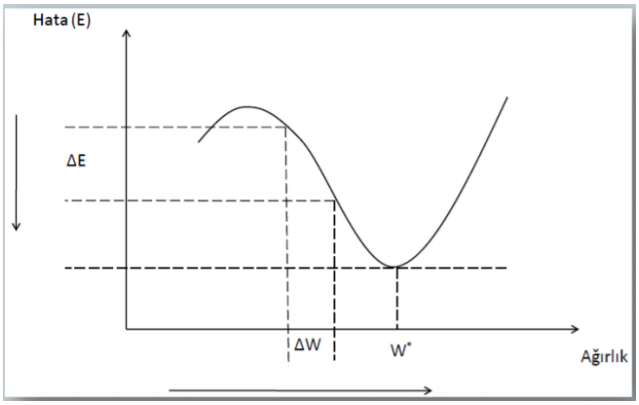
Şekil 2.5 YSA Öğrenme İşleminin Durdurulması

Her öğrenme döngüsü tamamlandığında ağın genelleme yeteneği kazanıp kazanmadığı kontrol edilir ve eğer ağ yeteri kadar öğrendiyse ve yeni gelecek veriyi genelleme yaparak tahmin etme yeteneğine ulaştıysa öğrenme işleminin sona ermesi gerekir. Döngü tekrarlanıyor ancak ilerleyen iterasyonlarda ağın performansı artmıyorsa ağın öğrenme işleminin durdurulması gerekir.

Bazı durumlarda ağ eğitim verileri ile %100 performans göstermesine rağmen yeni veriler üzerinde tahmin yeteneği çok düşük olabilir. Bunun gibi durumlarda ağ artık öğrenmeyi tamamlamıştır daha fazla eğitildiğinde ağ verileri ezberlemektedir. Bu duruma "aşırı öğrenme" denir ve ağ artık veriler için genelleme yapamaz, bu yüzden hatalı sonuçlar üretir ve bu durum istenmeyen bir durumdur.



Şekil 2.6 Optimum Modelin Seçilmesi



Şekil 2.7 Hatayı Minimize Edecek Optimum Ağırlıkların Gösterimi

Problemin çözümü için optimum ağırlık vektörü W* hata oranının en düşük olduğu ağırlık vektörüdür. Teorik olarak beklenen W* değerine pratikte ulaşmak mümkün olmayabilir. Bu nedenle ağdaki ϵ kadar hata kabul edilir. Bu noktanın altındaki herhangi bir noktaya ulaşıldığında ağ öğrenmiş kabul edilir.

Optimum nokta bulunamamasının bazı sebepleri aşağıda sıralandığı gibidir;

- Ağ eğitiminde kullanılan eğitim veri kümesi problem uzayının tamamını temsil etmiyor olabilir.
- Ağ modellenirken doğru parametreler seçilmemiş olabilir.
- Ağ ağırlıkları doğru şekilde belirlenmemiş olabilir.
- Ağın topolojisi veri kümesine uygun olmayabilir.
- Ağ yerel sonuçlara takılabilir.

Bunun gibi durumlarla karşılaşıldığında parametreleri değiştirmek, başlangıç değerlerini değiştirmek, topolojide değişiklik yapmak, eğitim veri kümesinin sayısını arttırmak veya azaltmak gibi işlemler yapılarak ağın yeniden eğitilmesi durumunda optimum noktaya ulaşılabilir.

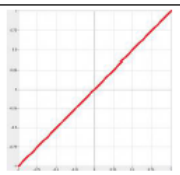
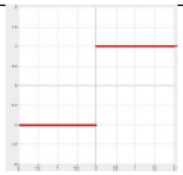
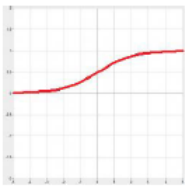
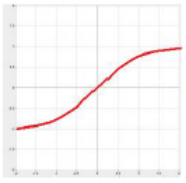
2.1.2.2 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu sayesinde girdiler ve çıktılar birbirleri ile eşleşmektedir. Ağın performansı yüksek bir şekilde çalışabilmesi için aktivasyon fonksiyonunun doğru seçilmesi son derece önem arz etmektedir. Aktivasyon fonksiyonu sayesinde ağ doğrusal olmayan yapıyı öğrenme işlemini gerçekleştirmektedir.

Literatürde çok fazla aktivasyon fonksiyonu bulunmaktadır. Yapay Sinir ağlarında kullanılan aktivasyon fonksiyonlarından bazıları aşağıda verildiği gibidir.

- Doğrusal (Lineer) Aktivasyon Fonksiyonu
- Adım (Step) Aktivasyon Fonksiyonu
- Sigmoid Aktivasyon Fonksiyonu
- Tanjant Hiperbolik Aktivasyon Fonksiyonu
- Eşik Değer Fonksiyonu
- Sinüs Aktivasyon Fonksiyonu

Bazı aktivasyon fonksiyonları ve bu fonksiyonların matematiksel, sözel ifadeleri Şekil 2.10'da verildiği gibidir [82].

Doğrusal (Lineer) Aktivasyon Fonksiyonu		F(NET)=A* NET (A sabit bir sayı)	Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon olarak seçilebilir. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step) Aktivasyon Fonksiyonu		F(Net)= { 1 if Net>Eşik Değer 0 if Net<=Eşik Değer	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(Net)=\frac{1}{1+e^{-Net}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.
Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(Net)=\frac{e^{Net}+e^{-Net}}{e^{Net}-e^{-Net}}$	Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer Fonksiyonu		F(Net)= { 0 if Net<=0 Net if 0<Net < 1 1 if Net>= 1	Gelen bilgilerin 0 dan küçük-eşit olduğunda 0 çıktısı, 1 den büyük-eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretilebilir.
Sinüs Aktivasyon Fonksiyonu		F(Net) = Sin(Net)	Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.

Şekil 2.8 YSA'da Sık Kullanılan Aktivasyon Fonksiyonları ve Özellikleri

Daha önce de belirtildiği gibi aktivasyon fonksiyonu seçimi, ağ performansı için büyük önem taşımaktadır. Her fonksiyonun birbirine göre avantajları ve dezavantajları mevcuttur. Problem iyi tanımlanmalı ve probleme uygun aktivasyon seçimi sağlanması ile sağlıklı analiz yapmak mümkün hale gelmektedir.

2.1.2.3 Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağları bilgi kaynağından öğrenebilme yeteneğine sahiptir. YSA modeli öğrenme sırasında elde ettiği bilgileri, sinir hücreleri arasında bağlantı ağırlıkları olarak saklamaktadır. Bu ağırlık değerleri YSA'nın verileri başarılı bir şekilde işleyebilmesi için gerekli olan bilgileri içerir [83]. Verilere ait bilgiler tüm ağda saklanır bu yüzden tüm ağdaki ağırlıklar en iyi değerleri almalıdır. Modelin eğitilebilmesi için bu ağırlık değerlerinin kurallar ile dinamik olarak değiştirilebilmesi gerekmektedir. YSA'nın öğrenmesinin amacı YSA'ya verilen örnekler için doğru çıktıları üreten ağırlık değerlerini bulmaktır, bu sayede ağ bundan sonra gelecek örnekler hakkında bilgi sahibi olup genelleme yapma yeteneğine sahip olur. Bu işlemlerin tamamına "Öğrenme" adı verilmektedir [84].

Temelde öğrenme yöntemleri danışmanlı (supervised learning) ve danışmansız (unsupervised learning) olmak üzere iki grupta incelenmektedir. Danışmanlı öğrenme yönteminde, YSA kullanılmadan önce eğitim veri kümesinin girdi ve çıktı bilgileri kullanılarak YSA eğitilir. Bu sayede YSA girdi bilgisine karşılık üretilen çıktı değerini, beklenen çıktı değeriyle karşılaştırarak ağırlıklarının değiştirilmesi için kullanılacak bilgiyi oluşturmaktadır. Elde edilen değer ile beklenen çıktı değeri arasındaki fark, hata değeri olarak önceden belirlenmiş olan değerden daha küçük olana kadar YSA eğitimine devam edilerek azaltılmaya çalışılır. Hata değeri, daha önceden belirlenen hata değerinde daha küçük olduğunda ise tüm ağırlıklar sabitlenerek eğitim işlemi tamamlanmaktadır. Danışmansız öğrenme yönteminde ise sistem beklenen çıktı değerleri hakkında bilgi sahibi değildir ve girdi değerlerine göre kendisi örnekler. Danışmansız olarak eğitilebilen ağlar ise, beklenen çıktı değerleri ağa gösterilmeden girdi bilgilerini inceleyerek bu girdilerin özelliklerine göre ağırlık değerlerini ayarlamaya çalışmaktadır [85].

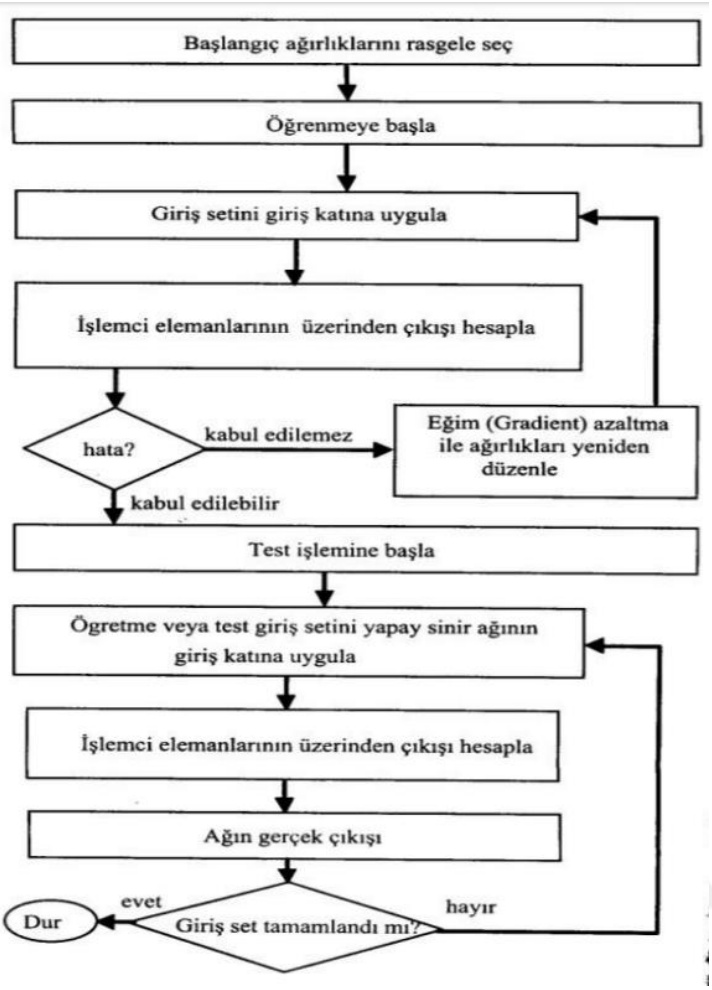
Öğrenme sürecinde gerçek çıktılar ve elde edilen çıktılar karşılaştırılırken elde edilen hata oranının makul seviyede olması beklenmektedir. YSA'nın asıl amacı girdi-çıktı çiftleri

için genelleme yapabilmektir. Bu yüzden hata oranının çok düşük olması her zaman YSA'nın genelleme yapabilme yeteneğine kavuştuğu anlamına gelmez.

YSA'nın çalışması sırasında kullanılan veriler ikiye ayrılır.

- Eğitim Verisi
- Test Verisi

Eğitim verisi ağın öğrenme sürecinin geliştirilmesi için kullanılırken, öğrenme başarısı test verisi ile kontrol edilmektedir. Test verisi sayesinde ağın performansı ölçülmektedir. YSA'nın öğrenme sürecinin sağlıklı olabilmesi ve doğru sonuçlar üretebilmesi için eğitim ve test verisinin miktarlarının yeterli olması ve verinin ne kadarıyla eğitim ne kadarıyla test yapılacağının iyi belirlenmesi önem arz etmektedir. Eğitim verisinin ne kadar olacağı; problemin karmaşıklığı, ağın modeli gibi parametrelere bağlı olarak değişmektedir. Eğitim verisinin optimum seçilip seçilmediğini anlamanın bir yolu olarak, eğitim verisi arttırıldığında performansın artıp artmadığı kontrol edilebilir.



Şekil 2.9 YSA ile Öngörü Modellenmesi

Ağın test edilmesi işlemi, eğitim kümesinde kullanılmamış verilerin kullanılması ile mümkündür. Test verilerindeki girdiler, ağa girdi olarak verilip, ağda işlenmesi sonucunda oluşan çıktılar istenilen çıktı değerleri ile karşılaştırılmaktadır. Eğitim ve test aşamalarında eğer gerçek verilere yakın sonuçlar elde edilirse YSA modeli kullanıma uygundur. YSA ile öngörü modellenmesi işlemine ait akış şeması Şekil 2.11'de verilmiştir [86].

Ağa girdi olarak verilen eğitim verilerinden çıktı değerlerini üretmesi işlemi ağırlık değerleri ile sağlanmaktadır. Başlangıçta bu ağırlık değerleri rastgele olarak atanır ve örneklerin hata oranları gözlemlenerek bu ağırlık değerleri değiştirilir. Bu işlem daha doğru sonuçlar elde edilmesini sağlamaktadır.

Öğrenme oranının belirlenmesinde beklenen çözüm hızı, YSA sisteminin karmaşıklığı, mimarisi, ağda kullanılan öğrenme kuralı ve ağdan beklenen başarı oranı gibi birçok faktör rol oynamaktadır. Birçok öğrenme fonksiyonu, öğrenme katsayısı için belirli standartlara sahiptir. Ağın ağırlıkları öğrenme katsayıları aracılığıyla arttırılabilir. Öğrenme katsayısının küçük olması, öğrenme sürecinin küçük adımlar halinde olmasını sağlar, bu sayede yavaş bir öğrenme süreci oluşturmaktadır. Ağ, yavaş öğrenme süreciyle beraber maksimum doğruluk derecesine yakınsama işlemini gerçekleştirmektedir [87].

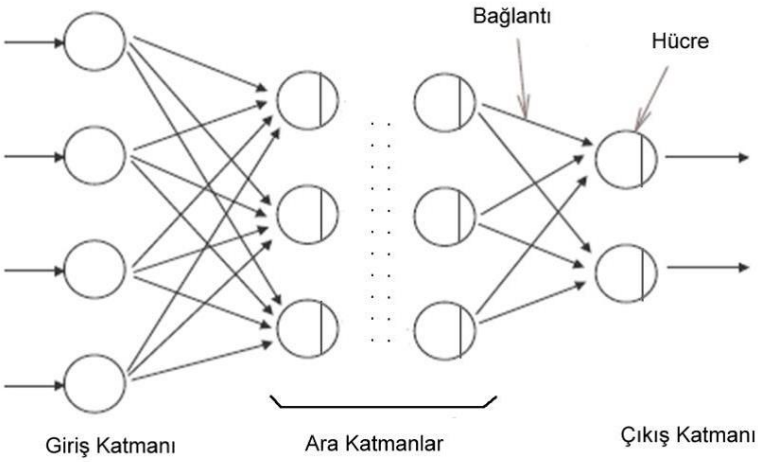
Öğrenme hızını arttırmak için öğrenme katsayısını büyütmek gerekir. Büyük adımlar halinde öğrenme sürecinde kullanılan büyük öğrenme katsayıları ağın küresel minimuma ulaşmasını engelleyebilir. Bu duruma salınım denmektedir. Salınımı engellemenin bir yolu momentum terimini kullanmaktır. Yerel çözümlere takılan ağların bir sıçrama ile yerel çözümlerden kurtulmasını sağlamaktadır.

2.1.3 Yapılarına Göre Yapay Sinir Ağları

Yapay Sinir Ağları çalışma yapılarına göre, İleri Beslemeli (Feedforward) YSA Modeli ve Geri Beslemeli (Feedback) YSA Modeli olarak iki şekilde sınıflandırılmaktadır.

2.1.3.1 İleri Beslemeli Yapay Sinir Ağı Modeli

İleri beslemeli sinir ağı YSA'nın en basit modelidir. Bilgi akışı yönü; Girdi katmanı, ara katmanlar ve çıktı katmanı şeklindedir. Geri yönlü besleme söz konusu değildir. Çıktılar yalnızca girdilerin ara katmanlarda işlenmesi ile elde edilmektedir. Daha önceki girdilere bağlı değildir.



Şekil 2.10 İleri Beslemeli YSA Modeli

İleri beslemeli YSA modelinin gösterimi Şekil 2.12'de belirtildiği gibidir [88]. Çok Katmanlı Algılayıcılar (Multi Layer Perceptron-MLP) ve Öğrenen Vektör Kuantizasyonu (Learning Vector Quantization- LVQ) ağları ileri beslemeli ağlara örnektir.

2.1.3.2 Geri Beslemeli Yapay Sinir Ağı Modeli

Yalnızca ileri yönlü bilgi akışı yoktur. Geri yönlü de bilgi akışlı söz konusudur. Böylece model dinamik bir yapı oluşturur ve çıktılar hem o andaki girdilere ve önceki girdilere bağlı oluşmaktadır. Girdiler hem ileri yönde hem de geri yönde aktarılarak önceki ara katmanlara veya giriş birimlerine geri beslenmektedir.

Tahminleme uygulamaları için başarı oranı yüksektir. Geri beslemeli YSA modeli zaman serileri tahminlemesinde oldukça yüksek başarı sağlamaktadır. Geri beslemeli ağlara örnek olarak Hopfield, Kendi kendine organize olan haritalar (SOM- Self Organizing Map), Elman ve Jordan ağları verilebilir [89].

2.1.4 Kullanım Alanlarına Göre Yapay Sinir Ağı Modelleri

Yapay sinir ağlarının kullanım alanları ve amaçlarına göre kullanılan topolojiler, aşağıda listelendiği gibi, genel olarak 8 maddede toplanabilir.

- Tahmin ve Öngörü
- Fonksiyon Yaklaştırma
- Desen (Örüntü) Sınıflandırma

- Veri İlişkilendirme
- Kavramsallaştırma ve Kümeleme
- Veri Filtreleme
- Optimizasyon
- Kontrol

Aşağıda Tablo 2.3'te amaçlarına göre kullanılabilecek bazı YSA topolojileri verilmiştir [87].

Tablo 2.3 Kullanım Alanlarına ve Amaçlarına Göre Kullanılan YSA Topolojileri

<table>
<tr>
<td rowspan="3">Tahmin-Öngörü
• Geri Yayılım
• Yönlendirilmiş Rastsal Tarama (Directed
• Random Search)
• Yüksek Dereceli Sinir Ağları
• Geri Yayılım içinde SOM
• Radyal Tabanlı Fonksiyon
• Elman Ağı
• Jordan Ağı</td>
<td>Fonksiyon Yaklaştırma
• Geri Yayılım
• RBF</td>
</tr>
<tr>
<td>Kavramlaştırma / Kümeleme
• LVQ
• SOM
• ART</td>
</tr>
<tr>
<td>Veri Filtreleme
• Yeniden Dolaşım (Recirculation)</td>
</tr>
<tr>
<td>Desen (Örüntü) Sınıflandırma
• LVQ
• ART
• Olasılık Tabanlı Sinir Ağları
• Tek veya Çok Katmanlı Algılayıcı
• Boltzmann Makinesi
• RBF</td>
<td>Veri İlişkilendirme
• Kohonen's SOM
• Hopfield Ağları
• Boltzmann Makinesi
• Hamming Ağları
• Çift Yönlü İlişkili Hafıza
• Spatio-Temporal Desen Tanıma
• ART</td>
</tr>
<tr>
<td>Optimizasyon
• Geri Yayılım
• Olasılık Tabanlı Sinir Ağları
• RBF</td>
<td>Kontrol
• Geri Yayılım
• LVQ
• RBF</td>
</tr>
</table>

Tablo 2.3'te görüldüğü gibi bazı ağ topolojileri, birden çok problem tipi için kullanılabilmektedir. İleri Beslemeli Geri Yayılım (Feedforward Back-Propagation) ağları hemen hemen tüm problemlerin çözümünde başarılı sonuçlar elde etmektedir ve bu yüzden en çok kullanılan ağ topolojisidir [90, 88].

2.1.5 Tek Katmanlı Algılayıcılar

Tek Katmanlı Algılayıcı (Single Layer Perceptron), F. Rosenblatt tarafından 1950'li yılların sonlarında ortaya atılmıştır [91]. İlk YSA modellerinden biri olan tek katmanlı algılayıcılar, oldukça sınırlı olmalarına rağmen ilerleyen zamanlarda geliştirilen modellere temel olmuşlardır ve mevcut algoritmalarla yakın ilişki içinde olmaları nedeniyle oldukça önemlidir.

Bu ağlarda eğitilebilen ağlar, tek bir işlemci elemandan oluşmaktadır. Burada eğitilebilirlik demek wi ağırlık değerlerinin değiştirilebilir olması demektir. İşlemci elemana girdi değerleri ve bu değerlere karşılık gelen çıktı değerleri gösterilerek öğrenme kuralına göre ağın çıktı değeri hesaplanmaktadır. Beklenen çıktı değerine ulaşılana kadar ağırlıklar değiştirilmektedir. Bu noktada çıktı değeri ile beklenen çıktı değerinin farkı Hata (Residual) olarak adlandırılmaktadır. Tek Katmanlı Algılayıcılarda, anlaşılacağı gibi ileri beslemeli ve danışmanlı öğrenme uygulanmaktadır.

Ağ çıktısı Eşitlik 2.1'de gösterilmiştir.

$$Y = f(\sum_{i=1}^{n} w_i . x_i + \theta) \tag{2.1}$$

Eşitlik 2.1'de i=1,2,…n olmak üzere; girdi katmanındaki toplam girdi sayısını, w_i girdilere karşılık gelen ağırlık değerlerini, x_i ağ girdilerinin değerlerini, Θ eşik değerini temsil etmektedir. F ise aktivasyon fonksiyonunu temsil eder.

$$f(g) = \begin{cases} 1 & Ç > 0 \\ -1 & Ç \leq 0 \end{cases} \tag{2.2}$$

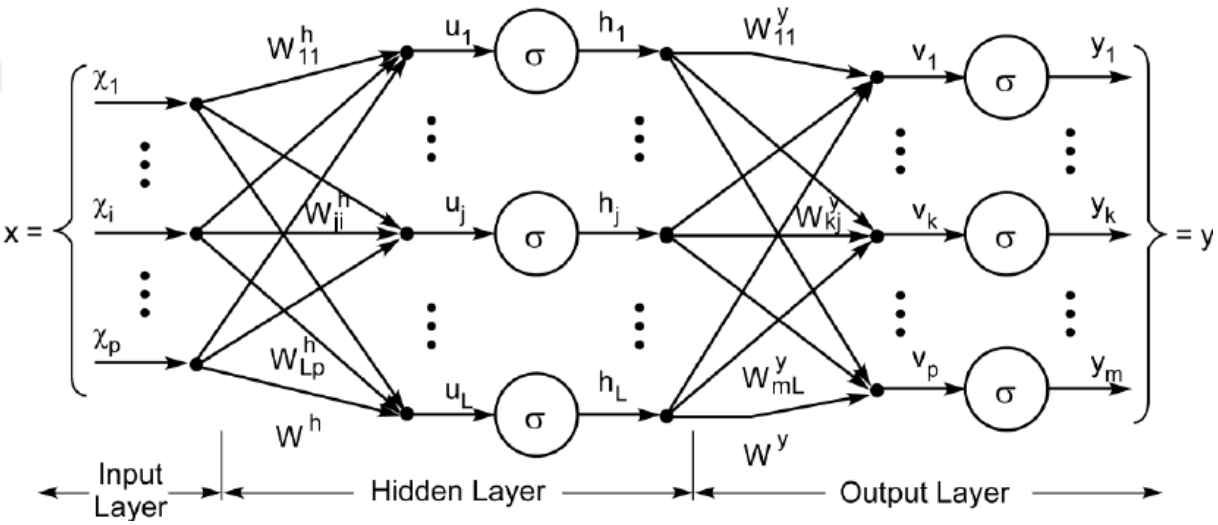
Doğrusal olmayan veriler sınıflandırma yeteneğine sahip değildir. Doğrusal olmayan verilerin sınıflandırılması işlemleri için çok katmanlı algılayıcılar geliştirilmiştir.

Minsky ve Papert Algılayıcının sadece Doğrusal Ayrılabilir (Linearly Seperable) desenleri sınıflandırabildiğini, ancak pratikte karşılaşılan birçok problemin doğrusal

ayrılabilir olmadığını ve bu yüzden de tek katmanlı algılayıcıların önemli sınırlamalara neden olduğunu vurgulamışlardır [92]. Özellikle mantıksal XOR problemlerinin çözümünün olmaması nedeniyle algılayıcı, günümüzde doğrusal olmayan problemlerin çözümünde çok kullanılmamaktadır.

2.1.6 Çok Katmanlı Algılayıcılar

Çok katmanlı ağ modeli doğrusal olmayan aktivasyon fonksiyonuna ve bir veya birden çok ara katmana sahiptir.



Şekil 2.11 Çok Katmanlı Algılayıcı Modeli

Her bir katmanın hiyerarşik bir şekilde bağlanması ile çok katmanlı model oluşmaktadır. Biyolojik nöronun çalışma mantığına tek katmanlı modele göre daha yakındır. Mühendislik alanındaki problem çözümlerinde oldukça sık kullanılmaktadır ve etkin sonuçlar üretmektedir. Çok katmanlı algılayıcılar temel olarak 3 katmandan meydana gelir bunlar: Girdi katmanı, gizli katman(lar) ve çıktı katmanıdır.

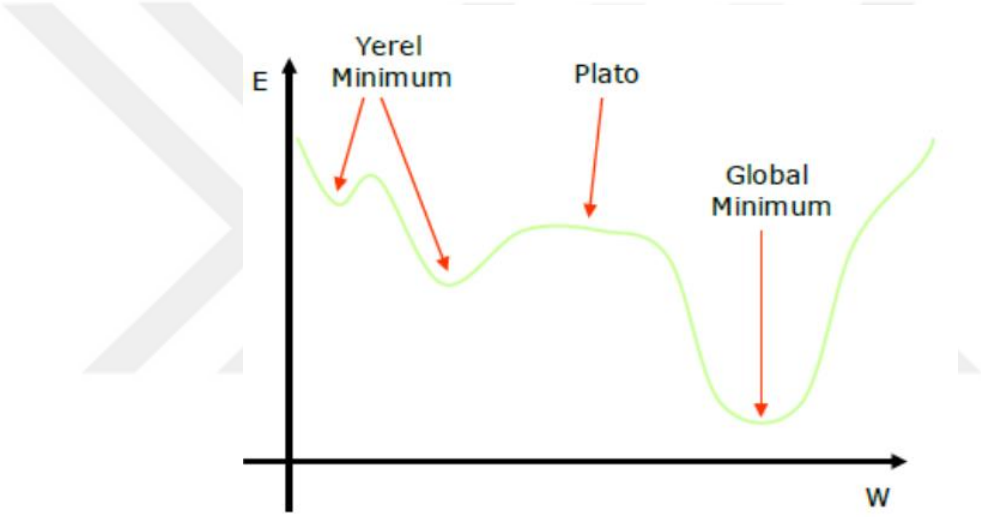
Birçok katmanlı algılayıcı örneklere bakarak çözüm üretir ve bu çözümden elde ettiği genelleme yeteneği sayesinde daha sonra gelecek veriler için çözüm üretebilmektedir. Çok Katmanlı Algılayıcı modeli Şekil 2.13'te verildiği gibidir [93].

Gizli katmanlar girdi ile çıktı arasındaki ilişkiyi anlamlı hale getiren katmanlardır. Ancak bu anlamlandırma işlemini nasıl yaptığı hakkında bilgi vermez. Bu durumda dışarıdan

müdahale edilmesini imkânsız kılar. Bu yüzden bu katmanlara gizli katman adı verilmektedir.

2.1.6.1 Momentum Katsayısı

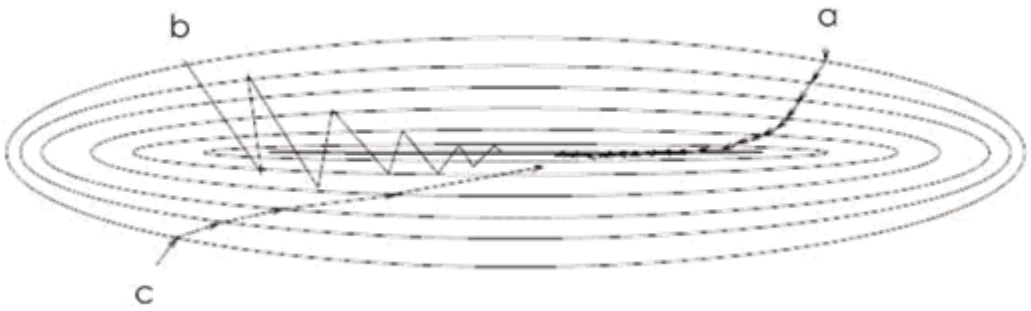
Momentum katsayısı çok katmanlı ağların yerel minimumlara takılıp, küresel çözümleri elde edememesi durumunu önlemek amacıyla geliştirilmiştir. Yerel minimumlara takılmadan küresel en iyi çözümün elde edilmesi işleminin gösterimi Şekil 2.4'te gösterildiği gibidir. Momentum katsayısının kullanılması ile elde edilen yerel çözümler belirlenmiş hata düzeyinin altına çekilebilir.



Şekil 2.12 Küresel Çözüm Gösterimi

Ağın belirlenen ağırlık değerleri başlangıçta çok büyük değerler olarak seçilirse ağ yerel çözümlere takılarak bir yerel çözümden diğerine sıçrama işlemi gerçekleştirmektedir. Ancak ağırlıklar küçük aralıkta seçilirse o zamanda en iyi çözümün elde edilmesi oldukça uzun zaman alabilir.

Bu yüzden momentum katsayısı oluşturulan ağın performansı açısından oldukça önemli bir özelliktir. Şekil 2.5'te Momentum Katsayısı kullanımının ağ üzerindeki etkisi gösterilmektedir. Eğer momentum katsayısı kullanılmazsa ağ düşük öğrenme katsayısı ile küresel minimuma ulaşması çok uzun zaman almaktadır. Büyük öğrenme katsayısı ile ise salınım durumu söz konusu olup küresel minimum değerine ağ hiç ulaşmayabilir. Momentum katsayısı kullanımı ile küresel minimum değerine hızlı bir şekilde ulaşması mümkündür [81].



Şekil 2.13 a: Düşük Öğrenme Oranı b: Büyük Öğrenme Oranı c: Momentum Terimi Eklenmiş Büyük Öğrenme Oranı

Momentum katsayısı, ağın yerel minimum değerlerine takılma durumunda sıçrama ile daha iyi çözümler elde etmesini sağlamaktadır.

2.1.7 Yapay Sinir Ağları Eğitimi

YSA, verilere ait örnekleri inceleyerek, bu bilgileri kullanarak problem ile ilgili genellemeler yaparak, veriler hakkında bilgiler edinmektedir. Daha sonra ise ağa daha önce hiç gösterilmeyen örnekler ile karşılaşınca, daha önce öğrendiği bilgilerden elde ettiği genellemelerden yola çıkarak, bu örnekler hakkında sonuç elde edebilip, karar verebilmektedir [88].

İşlem elemanlarının bağlantılarının ağırlık değerlerinin belirlenmesi işleminde farklı algoritmalar ve matematiksel işlemler kullanarak çıktıların elde edilmesi işlemine ağın eğitilmesi denir. Yapay sinir ağını birçok problem türüne yönelik kullanmak için ağın tasarlanmasına yani eğitilmesine ihtiyaç duyarız.

Literatürde yapay sinir ağlarının eğitimi amacıyla kullanılmış birçok algoritma mevcuttur. Geleneksel eğitim algoritmaları kullanılarak yapılan YSA eğitimleri ile yerel optimum noktasına takılma, karmaşık hesaplamaların yapılamaması gibi dezavantajlar söz konusudur. Problemin öğrenilmesi ve buna bağlı performans artışı sağlamak için yapay sinir ağı probleme uygun bir algoritma ile eğitime tabi tutulabilir. Ağ ne kadar iyi eğitilirse, problem çözme kabiliyeti o derece artar.

YSA'lar eğitirken kullanılan veriler girdi-çıktı çiftleri ağın eğitim kümesini oluşturmaktadır. Eğitim kümesinin kullanılmasındaki asıl amaç girdi verileri için doğru

çıktı değerlerini elde edecek ağırlık değerlerini elde etmektir. Eğitim verisinde kullanılan örnek çiftler sayesinde ağ genelleme yapabilme yeteneğini elde etmektedir.

Ağın ağırlıklarının doğru olarak belirlenmesi ve yüksek performans elde etmesi kullanılan algoritmanın işlem adımlarındaki fonksiyonlara ve güncelleme mekanizmalarına bağlıdır. Algoritmaların hemen hemen hepsi rastgele ağırlık başlangıç değeri ile işlemi başlatırken ağırlıkların değiştirilmesi işlemi algoritmadan algoritmaya farklılık göstermektedir.

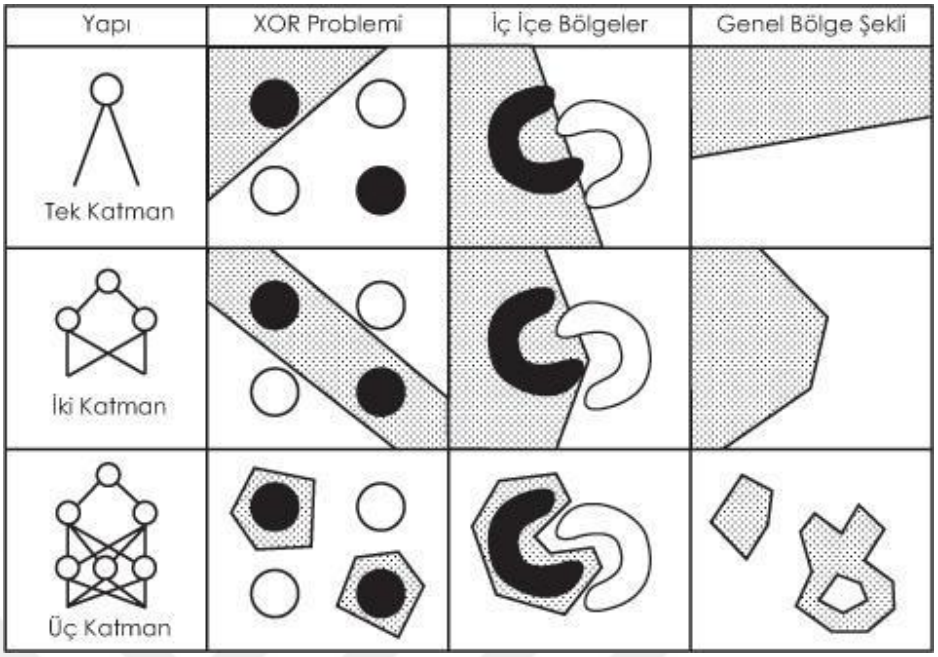
2.1.7.1 Yapay Sinir Ağı Eğitimi için Kullanılacak Ağın Özelliklerinin Belirlenmesi

Öğrenme Algoritması ve YSA Ağ Topolojisinin Belirlenmesi: Problemin çözülebilmesi için uygulanması gereken topoloji seçilirken problem yapısı anlaşılmalı ve bu yapıya göre karar verilmelidir. Ağ modellerinin ve bu ağ modellerinin kullanım alanlarının bilinmesi önemli bir adımdır.

Katman ve Nöron Sayısının Belirlenmesi: YSA'da kullanılacak katmanların sayısı ve bu katmanların nöron sayıları yapay sinir ağının karmaşıklığını belirlemektedir. Katmanların sayısının ve katmanlardaki nöron sayısının artması yapay sinir ağının öğrenme performansını arttırmakla beraber, ağın yakınsama işlemini daha uzun zamanda yapmasına da yol açar.

Yakınsama süresinin artmasının yanı sıra, YSA genelleme yeteneği azalmaktadır ve ağın örnekleri ezberlemesine sebep olmaktadır. Kullanılması gerekenden daha basit bir ağ yapısı oluşturarak, daha az katman ve nöron kullanılması, verilerin yeterince anlamlandırılamaması ve buna bağlı olarak ağın yeterince öğrenememesine yol açmaktadır. Birçok problemin çözümünde iki ya da üç katmana sahip bir YSA modeli başarılı sonuçlar elde etmektedir.

İşlemci eleman sayısı ve katman sayısını belirlemenin en iyi yolu, deneme yanılma yolu ile ağın başarı oranları gözlemlenerek performansı arttıracak en uygun katman sayısı ve bu katmanlarda bulunacak en uygun nöron sayısına karar vermektir. Performansı arttırmak için yapılması gereken; başlangıçta kullanılan gizli katmandaki nöron sayısının istenilen performansa ulaşıncaya kadar arttırılması veya azaltılması şeklindedir. Şekil 2.14'te ağ yapılarında bulunan katman sayıları ve buna bağlı olarak problemlerin öğrenme süreçlerinde elde edilen farklar gösterilmektedir [90].



Şekil 2.14 Sınıflandırma İşleminde, Gizli Katman Sayısının İki Boyutlu Örnek Uzayındaki Rolünün Geometrik Gösterimi

Aktivasyon Fonksiyonunun Belirlenmesi: İşlemci elemanlarının aktivasyon fonksiyonlarının belirlenmesi, işlemi yapay sinir ağının tasarım adımında alınması gereken kritik kararlardan biridir. Aktivasyon fonksiyonu seçme işlemi problemin verilerinin yapısına, türüne bağlıdır. Aktivasyon fonksiyonlarından en sık kullanılan fonksiyonlar: doğrusal, işaret, eşik, sigmoid, hiperbolik tanjant ve lojistik fonksiyonlarıdır.

Normalizasyon: Verilerin YSA'da öğrenme işlemine başlamadan önce normalizasyon ön işlemi yapmak gerekebilir. Yapılacak önişlemler verinin yapısına göre değişmektedir. Bu yüzden problemi çok iyi tanımlayabilmek ve veriyi incelemek, ağın başarı oranını yükseltmek için oldukça önemlidir. Çalışılan verilerde yerel optimumları bulmayı engelleyebilmek için öncelikle normalizasyon işlemi yapılmalıdır.

Bu işlem için logaritmik fonksiyonlar kullanılabilir veya veriler [0,1] veya [-1, +1] aralıklarında ölçeklendirilebilir. Ölçekleme işlemi yapılınca verilerin eksen sistemine sıkıştırılması işlemine tabi tutulduğundan dolayı bilgi kaybı da yaşanmaktadır; bu durumdan kaynaklı YSA modellerinin çalışmasını olumsuz yönde etkileyerek performansını düşürebilir. [74].

Etkinlik Ölçütünün Belirlenmesi: Etkinlik ölçütü hesaplanırken, beklenen çıktı değerleri ile ağda üretilen çıktı değerleri arasındaki farkın bazı değerleri hesaplamaktadır. Etkinlik ölçütleri sayesinde YSA'nın eğitim kümesinin tüm veri yapısına ne kadar yaklaştığı göz önünde tutularak, ağın bağlantılarının ağırlık değerlerinin değiştirilme işlemleri gerçekleştirilmektedir. Bu yüzden bu fonksiyonlar ağın öğrenme başarısını etkileyen önemli kriterlerdendir.

Literatürde sıklıkla kullanılan tipik etkinlik ölçütlerinden bazıları;

- Hata Kareleri Ortalaması (Mean Square Error, MSE),
- Ortalama Hata (Mean Error, ME),
- Hata Kareleri Ortalamasının Karekökü (Root Mean Square)
- Ortalama Mutlak Hata (Mean Absolute Error, MAE) şeklinde sıralanabilir.

Öğrenme Katsayısı ve Momentum Katsayısının Seçilmesi: Öğrenme sürecinde ağırlıklardaki değişim ve öğrenme doğru orantılıdır. Karmaşık bir ağda hata yüzeyi oldukça inişli çıkışlıdır. Bu durum, dereceli azaltmadan dolayı ağın yerel minimuma takılmasıyla sonuçlanmaktadır. Olasılık yöntemleriyle bu durumdan kurtulmak mümkün olmakla beraber, oldukça yavaş kalmaktadır. Bu iniş çıkışları azaltmak için bir başka yöntem ise, gizli katman veya gizli katmandaki nöron sayısının arttırılması olmalıdır. Bu yöntem işe yarasa da hata yüzeyinin boyut sayısını da arttırmaktadır.

Gizli nöron sayısının bir üst limiti her zaman olacaktır, çünkü araştırmalar gizli nöron sayısının fazla olduğu durumlarda ağın ezberlediğini ortaya koymuştur [80].

Büyük öğrenme oranlarında salınımı önlemenin bir yolu Momentum katsayısının eklenmesidir. Eğer momentum katsayısı kullanılmazsa ağın düşük öğrenme katsayısı ile yerel minimuma ulaşması çok uzun sürmekte, büyük öğrenme katsayısında ise salınımdan dolayı yerel minimuma asla ulaşamamaktadır. Momentum katsayısı kullanıldığında ise yerel minimuma hızlı bir şekilde ulaşılabilmektedir.

2.2 Metasezgisel Algoritmalar

Belirli sınırlar çerçevesinde bir probleme ait olan birçok parametrenin en iyi sonucu elde edecek şekilde belirlenmesi işlemine genel olarak optimizasyon adı verilmektedir. Bilim, mühendislik, ekonomi ve iş dünyası gibi birçok alanda karşılaşılan problemler oldukça karmaşık ve çözülmesi zor problemlerdir. Bu yüzden birbirinden farklı tipte olan bu problemleri çözebilmek için farklı problem çözme yöntemleri geliştirilmiştir [94].

Optimizasyon bir problem için en uygun çözümün bulunması işlemine verilen isimdir. Optimizasyondaki amaç, minimum kaynak ile kazancın maksimum olmasını sağlamaktır. Optimizasyon kelime anlamı ise, en iyilemedir. Problemin belirli sınırlar içerisinde bulunan bütün çözümleri arasından en iyi çözümü elde etmektir.

Optimizasyon problemi ise probleme ait sınırlar dahilinde, istenen parametrelerin değerlerinin bulunması işlemi olarak da tanımlanabilir [95]. Optimizasyon işleminin performansında rol oynayan değişkenlere ise karar değişkenleri denmektedir. Karar değişkenleri kullanarak, problemin amacına ait etkilerin gösterimi sayesinde amaç fonksiyonu oluşturulmaktadır.

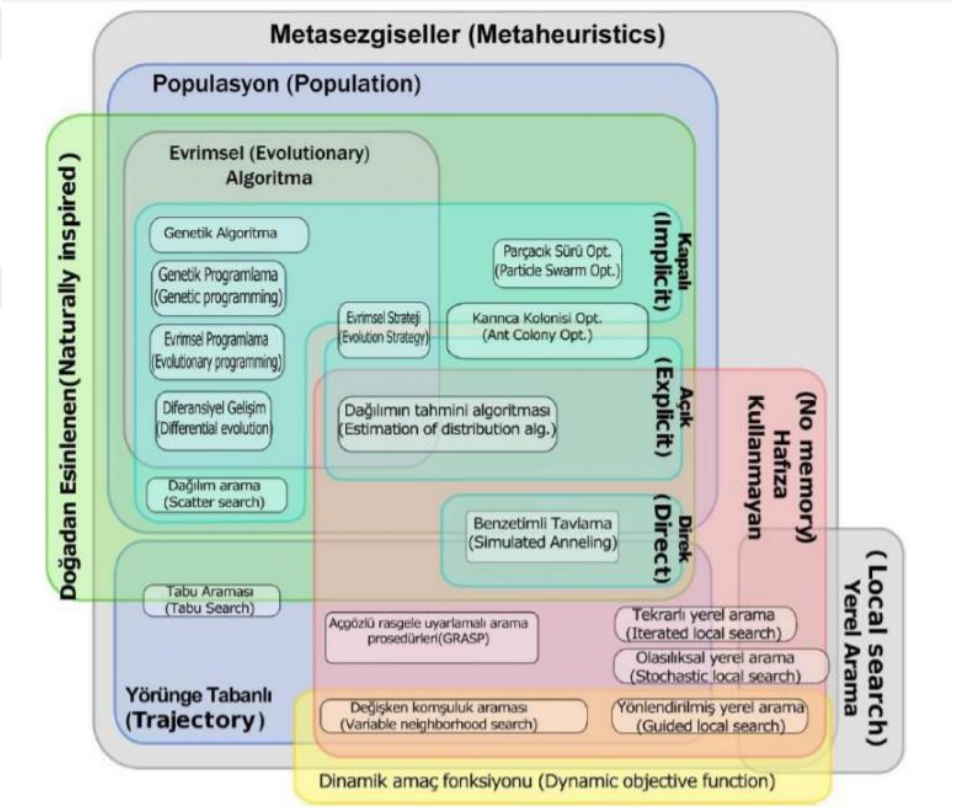
Optimizasyon algoritmalarının birçoğu, amaç fonksiyonunu oluşturmak ve sistemi modellemek için matematiksel modellere ihtiyaç duymaktadır. Ancak çok karmaşık verilerin veya gerçek hayat verilerinin çözümü için matematiksel model oluşturma işlemi bir hayli zor bir işlemdir veya oldukça zaman almaktadır. Bu yüzden klasik optimizasyon algoritmaları, gerçek hayat problemlerinde, doğrusal olmayan veya karmaşıklığı yüksek problemlerde oldukça yetersiz kalmaktadır. Klasik optimizasyon algoritmaları, verilen probleme uygun çözüm oluşturmada yeterince etkili olmadığından sezgisel veya metasezgisel algoritmalar bu problemlerin çözümü için önerilmektedir.

Metasezgiseller, özel problemlere uygulanabilen ancak genel amaçlı, en iyi çözümü elde edebilecek üst düzey stratejiye sahip algoritmalardır. Metasezgisel algoritmalar, hesaplama gücü, uygulama kolaylığı ve birden çok probleme uygulanabilir olması sebebiyle günümüzde geniş alanda mühendislik problemlerinin çözülmesinde kullanılmaktadır [96].

Metasezgisel algoritmaların temelde iki bileşeni mevcuttur. Bu bileşenler sayesinde en iyi veya en iyiye yakın çözümleri bulmak için arama uzayını etkin bir şekilde arama işlemini gerçekleştirmektedir. Bu iki bileşen aşağıdaki gibidir;

- Faydalanma (Exploitation): Bölgede bulunan mevcut en iyi çözüm bilgisini kullanarak yerel bölgede aramaya odaklanma.
- Keşif (Exploration): Küresel ölçekte arama uzayını keşfedene kadar çeşitli çözümler üretme.

Bu iki bileşenin doğru kombinasyonu sayesinde genellikle en iyi çözümler elde edilmektedir [26].



Şekil 2.15 Metasezgisel Algoritmaların Sınıflandırılması

Metasezgisel algoritmaları sınıflandırmanın birçok yolu bulunmaktadır. Metasezgisel algoritmalar, literatürde genellikle sürü zekasına dayalı, popülasyon tabanlı, olasılıksal, evrimsel, kimya tabanlı, müzik tabanlı, sosyal tabanlı ve fizik tabanlı şeklinde sınıflandırılmaktadır [97]. Bir metasezgisel algoritma birden fazla sınıfa dahil

edilebilmektedir. Genel olarak metasezgisel yöntemler Şekil 2.15'te belirtildiği gibi sınıflandırılmaktadır [98].

Metasezgisel algoritmalara ait önemli özellikler aşağıdaki gibi sıralanmaktadır [99];

• Optimum çözümü bulmayı kolaylaştıracak ve bu süreci yönlendirebilen stratejileri mevcuttur.
• Metasezgisel algoritmalardaki amaç, optimum çözümleri bulmak için arama uzayını verimli bir şekilde araştırmaktır.
• Metasezgisel algoritmaların teknikleri çok fazla olmakla beraber; basit yerel arama algoritmalarından karmaşık öğrenme süreçlerine kadar geniş bir alanı içermektedir.
• Arama alanlarında bulunan yerel optimum çözümlerine takılmamak için farklı mekanizmalar içermektedirler.
• Metasezgisel algoritmalar bir probleme özgü oluşturulmuş algoritmalar değil, birden çok probleme uygulanabilir algoritmalardır.
• Arama esnasında oluşan bilgi (hafızayı) kullanılarak daha sonraki aramalarda iyileştirmeye yardımcı olmaktadır.
• Metasezgisel algoritmalar, farklı yöntemler ile arama uzayı keşfedilen yüksek başarılı algoritmalardır.

Metasezgisel algoritmalar yaygın kullanım alanına sahiptir, genellikle doğadan ilham alınan ve başta mühendislik alanında olmak üzere, birçok alanda problemlerin çözümünde kullanılmaktadır. Doğada karşılaşılan problemlere yüksek başarılı üstün çözümler bulmayı başarmaktadır, bu yüzden geliştirilen birçok metasezgisel algoritma, problem çözme yeteneğini doğayı taklit ederek öğrenmektedir. Optimizasyon problemlerinin çözülmesinde seçilen optimizasyon tekniği, problemin özelliklerine bağlı olarak değişiklik göstermektedir [100].

Metasezgisel Algoritmalar etkili ve hızlı çözüm üretebilen algoritmalar olduğu için son yıllarda birçok alanda kullanımı oldukça yaygındır [96].

- Mühendislik tasarımı
- Aerodinamik ve akışkanlar dinamiği
- Telekominikasyon
- Otomotiv, robotik kodlama ve üretim

- Makine öğrenmesi ve veri madenciliği
- Finans, lojistik ve tedarik zinciri yönetimi
- Kontrol, rotalama ve sinyal işleme gibi birçok kullanım alanına sahiptir.

Günümüzde biyoloji, fizik gibi disiplinlerden ilham alınarak genel amaçlı, esnek, probleme uyarlanabilir ve yüksek performanslı metasezgisel algoritmalar geliştirilmektedir. Son yıllarda karmaşıklığı yüksek problemlerin çözümü için geliştirilen metasezgisel algoritmalar genellikle biyoloji tabanlıdır. Biyoloji tabanlı metasezgisel algoritmalar, öncelikle doğadaki var olan olayları anlamak, davranışları incelemek ve taklit etmek ile modellenmektedir [101]. Canlılar referans alarak taklit edilen bu davranışlar, genellikle yiyecek arama davranışlarıdır. Sincap Arama Algoritması, son yıllarda geliştirilmiş, biyoloji tabanlı, başarılı metasezgisel algoritmalardan bir tanesidir.

Bu çalışmada Yapay Sinir Ağlarının eğitiminde kullanılacak yeni nesil metasezgisel algoritma Sincap Arama Algoritmasıdır.

2.2.1 Sincap Arama Algoritması

Bu bölümde Sincap Arama Algoritmasının tarihçesi, temeli, aşamaları ve çalışma yapısı hakkında bilgi verilmektedir.

2.2.1.1 Sincap Arama Algoritması Tarihçesi

Sincap arama algoritması, Jain ve arkadaşları tarafından 2019 yılında doğadan esinlenilerek geliştirilmiş; yeni, basit ve güçlü bir küresel optimizasyon algoritmasıdır. Uçan sincapların dinamik arama davranışlarını ve süzülme hareketlerini taklit etmektedir. Süzülme hareketi, küçük memelilerin uzun mesafelerde yol almak amacıyla kullandığı etkili bir harekettir. SSA, bu davranışı kullanarak problemlerin optimizasyonunu gerçekleştirmeyi hedeflemektedir [1].

Sincap arama algoritması yeni bir optimizasyon tekniği olduğu için konu ile alakalı çok fazla çalışma bulunmamaktadır. Bazı çalışmalarda sincap arama algoritması kullanılarak bir problemin çözümünde diğer algoritmalarla karşılaştırılarak başarı oranları incelenirken; bazı çalışmalarda ise başka metasezgisel algoritmalarla birlikte sincap arama algoritması melezleştirilmiştir.

Hu ve arkadaşları tarafından 2019 yılında yapılmış bir çalışmada, yabani ot algoritması (IWO) kullanılarak sincap arama algoritması tabanlı yeni bir melez algoritma çalışması yapılmıştır. Bu çalışmada algoritmanın, sınıflandırma ve tahmin için etkin bir algoritma olduğu gözlemlenmiştir [102].

SSA ve Guguk kuşu arama algoritması kullanarak melez bir algoritma, Agrawal ve arkadaşları (2019) tarafından Beyin Manyetik Rezonans (MR) görüntülerinin analizinde kullanılmak üzere önerilmiştir. Geliştirilen bu algoritmanın, Guguk Kuşu arama algoritması (CS) ve sincap arama algoritması ile tek tek karşılaştırıldığında daha hızlı ve daha üstün olduğu gözlemlenmiştir [103].

SSA'nın yerel optimumu yakınsama dezavantajını iyileştirmek için Zheng ve Luo (2019) tarafından "Optimizasyon için Geliştirilmiş Sincap Arama algoritması" çalışması yapılmıştır. Bu çalışmada kullanılan stratejiler ile birlikle çözümün kalitesi arttırılmaya çalışılmış ve önerilen geliştirilmiş sincap arama algoritmasının (ISSA) güçlü bir optimizasyon algoritması olduğu ve SSA ile karşılaştırıldığında daha iyi performans gösterdiği sonucu elde edilmiştir [104]. Bu alanda yapılan bir diğer çalışma ise; Wang ve Du tarafından yapılmış olup, küresel optimumu yakınsama performansını iyileştirmek için iki arama yöntemi geliştirilmiştir. Bu çalışmada doğruluk ve yakınsama hızı konusunda daha avantajlı bir algoritma elde edilmiştir [105].

Konu ile alakalı yapılan bir diğer çalışma ise; optik lensler ve kameralardaki görüntü füzyon tekniklerinin iyileştirilmesi için SSA kullanılarak gabor filtre parametre seçimleri çalışmasıdır. Böylece geleneksel yöntemlerin hesaplama karmaşıklığı problemi aşılmış ve önerilen melez filtreleme tekniği daha önce kullanılan en gelişmiş tekniklerle kıyaslanmış ve daha iyi performans sağlanmıştır [106].

Literatürde SSA, afet yönetiminde akıllı sensörlerin yönlendirilmesinde [107], güç sinyalleri kullanılarak hata tespitinin yapılmasında [108] kullanılmıştır. Swain ve arkadaşları tarafından yapılan bu çalışmada, büyük verilerden hızlı ve başarılı sonuçlar çıkarabilmek için denetimsiz öğrenme tekniği kullanılmıştır. Softmax regresyonu, sincap arama algoritması ve jaya optimizasyon algoritması kullanılarak özellik etiketi bulunmayan veriler ile insan tecrübelerine dayanan arıza tespitleri yapılmıştır. Sinyallerden gelen özelliklerin seçilmesi işlemi sincap arama algoritması ile

gerçekleştirilmiş ve bu sayede önerilen yöntem hata tespitinde yüksek başarı sağlamıştır. Önerilen yaklaşım basit ve uygulanabilirliği yüksektir.

Melez algoritma oluşturma çalışmalarının ve SSA'nın çözülmesi istenen probleme uygulanması çalışmalarının yanı sıra, Sincap Arama Algoritması literatürde performans değerlendirmek amacıyla da kullanılmıştır. Önerilen algoritmalar veri kümelerine uygulanmış ve başarı oranları güncel, performansı yüksek diğer algoritmalarla kıyaslanmıştır [109-112].

2.2.1.2 Sincap Arama Algoritması Temeli

Uçan sincaplar, süzülme hareketine istisnai bir şekilde uyum sağlamış bir çeşit gece kemirgen grubudur. Günümüzde, 15 cins ve 44 tür uçan sincap bulunmaktadır ve bunların çoğu, özellikle Güneydoğu Asya olmak üzere, Avrupa ve Asya'nın yaprak döken ormanlarında görülmektedir. Avrasya dışında bulunan ve üzerinde en çok çalışma yapılmış tür, güneydeki uçan sincaplardır [109]. Uçan sincaplar, paraşüt benzeri patagia denilen bir zara sahiptir. Bu zar sayesinde sincaplar bir ağaçtan diğerine süzülerek yol almaktadır [110]. Uçan sincaplar uçamamakla beraber, bu zarları sayesinde süzülme hareketiyle hızlı bir şekilde yırtıcı hayvanlardan kaçar ve sıcak havalarda yiyecek arama davranışlarında optimum enerji ile uzun mesafeler kat ederler [111].

Sonbaharda, günlük enerji ihtiyaçlarını yaprak döken ormanlarda iklim koşulları sıcak olduğunda bol miktarda bulunan meşe palamudu yiyerek karşılarlar. Günlük enerji ihtiyacını karşılayan sincaplar, kış için en uygun besin kaynağı olan yüksek enerjiye sahip Kuzey Amerika cevizini aramaya başlarlar ve bulduklarında ise yuvalarda, oyuklarda ve toprakta depolarlar [112]. Soğuk kış aylarında sincaplar kış uykusuna yatmazlar ancak daha pasif hale gelerek sıcak aylarda depoladıkları yiyecekleri tüketir, sıcak havalarda tekrar daha aktif hale gelerek soğuk aylar için yiyecek depolarlar. Bu döngü Sincap Arama Algoritmasının temelini oluşturmaktadır ve sincapların hayatı boyunca tekrarlanmaktadır [1].

Sincap arama algoritmasında, matematiksel modelin oluşturulabilmesi için aşağıdaki varsayımlar göz önünde bulundurularak yiyecek arama davranışları sadeleştirilmektedir;

1. Yaprak döken ormanda n adet sincap bulunmaktadır ve her sincap başlangıçta bir ağaçta konumlanmaktadır.

2. Her uçan sincap, ayrı ayrı yiyecek arama davranışı sergiler ve yiyecek kaynakları optimum şekilde kullanılmaktadır.
3. Ormanda, normal ağaç, meşe palamudu yiyecek kaynağı olan meşe ağacı ve Kuzey Amerika Cevizi yiyecek kaynağı olan ceviz ağacı olmak üzere yalnızca üç tür ağaç mevcuttur.
4. İncelenmekte olan orman bölgesinin üç tane meşe ağacı ve bir tane Kuzey Amerika cevizi ağacı içerdiği varsayılmaktadır.

2.2.1.3 Sincap Arama Algoritması Aşamaları

Algoritma, diğer popülasyon tabanlı algoritmalar gibi uçan sincapların rastgele başlangıç konumu belirlenmesi ile başlamaktadır. Uçan bir sincap, d boyutlu arama uzayında bir vektörle temsil edildiğinden, uçan sincaplar tek boyutlu, 2 boyutlu, 3 boyutlu veya çok boyutlu arama uzaylarında süzülerek yer değiştirebilmektedir [1].

I. Rastgele Başlangıç

Bir ormanda n sayıda uçan sincap (FS) (flying squirrels) vardır ve uçan sincap (i) konumu bir vektör ile belirtilebilir. Tüm uçan sincapların konumu Eşitlik 2.3'deki matris ile gösterilebilmektedir:

$$FS = \begin{bmatrix} FS_{1,1} & \cdots & FS_{1,d} \\ \vdots & & \vdots \\ FS_{n,1} & \cdots & FS_{n,d} \end{bmatrix} \tag{2.3}$$

Bu matriste kullanılan $FS_{i,j}$ i. sincabın j. konumunu göstermektedir. Ormanda bulunan her sincap için başlangıç konumu atama işlemi Eşitlik 2.4 baz alınarak yapılmaktadır.

$$FS_i = FS_L + \mathrm{U}(0, 1) \times (FS_U - FS_L) \tag{2.4}$$

U(0, 1), [0, 1] aralığında uniform dağılımdan rastgele bir sayıdır ve FS_L, FS_U uçan sincapların alt ve üst sınırlarını temsil etmektedir.

II. Uygunluk Değerlendirmesi

Her uçan sincabın konumunun uygun olup olmadığı, karar değişkeninin yani çözüm vektörünün değerleri kullanıcının tanımladığı bir uygunluk fonksiyonuna yerleştirilerek hesaplanmakta ve buna karşılık gelen değerler Eşitlik 2.5'teki dizide saklanmaktadır:

$$f = \begin{bmatrix} \begin{cases} f_1([FS_{1,1}, FS_{1,2} \dots FS_{1,d} \\ f_2([FS_{2,1}, FS_{2,2} \dots FS_{2,d} \\ . \\ . \\ f_n([FS_{n,1}, FS_{n,2} \dots FS_{n,d} \end{cases} \end{bmatrix} \quad (2.5)$$

Her uçan sincabın konumunun uygunluk değeri; aradığı gıda kaynağının kalitesini, yani optimal gıda kaynağı (ceviz ağacı), normal gıda kaynağı (meşe palamudu ağacı) ve hiçbir gıda kaynağı olmadığını (uçan sincap normal ağaç üzerindedir) ve hayatta kalma olasılıklarını göstermektedir.

III. Sıralama, Bildirim ve Rastgele Seçim

Uçan sincapların konumlarının uygunluk değerlerini kaydettikten sonra, dizi artan düzende sıralanmaktadır. Minimizasyon problemleri için, minimum uygunluk değerine sahip uçan sincap, ceviz ağacında kabul edilmektedir. Sonraki en iyi üç uçan sincap, meşe palamudu ağaçlarının üzerinde olduğu kabul edilip ve bunların ceviz ağacına doğru hareket ettiği varsayılmaktadır. Yiyecek kaynağı olan 4 ağaç uygunluk değerine göre 4 sincaba atandığına göre kalan uçan sincapların normal ağaçlarda olması gerekmektedir. Ayrıca rastgele seçim yoluyla, bazı sincapların, günlük enerji gereksinimlerini karşıladıklarını varsayarak, ceviz ağacına doğru hareket ettiği düşünülmektedir. Diğer sincaplar (günlük enerji ihtiyaçlarını karşılamak için) meşe palamudu ağaçlarına ilerleyecektir. Uçan sincapların bu yiyecek arama davranışı, yırtıcı hayvan avcıların varlığından daima etkilenmektedir.

Sincapların orman içindeki yiyecek arama davranışları, yukarıda anlatılan konum güncelleme mekanizması kullanılarak ve ormanda avcı bulunma olasılığı da göz önünde bulundurularak modellenmektedir.

IV. Yeni Konumların Oluşturulması

Uçan sincapların dinamik yiyecek araması davranışında üç durum bulunmaktadır. Bu durumların her birinde, yırtıcı hayvanlar yokken, sincaplar ormandaki en sevdiği yiyecek için etkin bir şekilde süzülerek arama yapar, yırtıcı hayvan varlığında ise daha temkinli davranmak zorundadır ve yakınlardaki saklanma yerini aramak için küçük rastgele bir gezinti yapar. Dinamik yiyecek arama davranışı, matematiksel olarak şu şekilde modellenmektedir:

Durum 1. Meşe palamudu ağaçlarında (FS_{at}) bulunan uçan sincaplar, ceviz ağacına doğru hareket edebilir. Bu durumda, yeni sincapların bulunduğu yer şu şekilde elde edilmektedir:

$$FS_{at}^{t+1} = \begin{cases} FS_{at}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{at}^{t}) & R_1 \geq P_{dp} \\ Rastgele\ Konum & Diğer \end{cases} \quad (2.6)$$

Burada, d_g Rastgele süzülme mesafesini temsil ederken, R_1 [0, 1] aralığında rastgele bir sayıdır. FS_{ht}, ceviz ağacına ulaşan ve t. mevcut iterasyonu ifade eden sincap yeridir. Arama ve kullanma arasındaki denge matematiksel modelde sabit G_c süzülme yardımı ile elde edilir. Bu değer, önerilen algoritmanın performansını önemli ölçüde etkiler.

Durum 2. Normal ağaçlarda (FS_{nt}) uçan sincaplar, günlük enerji ihtiyaçlarını karşılamak için meşe palamudu ağaçlarına doğru hareket edebilir. Bu durumda, yeni sincapların bulunduğu yer şu şekilde elde edilmektedir:

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^{t} + d_g * G_c * (FS_{at}^{t} - FS_{nt}^{t}) & R_2 \geq P_{dp} \\ Rastgele\ Konum & Diğer \end{cases} \quad (2.7)$$

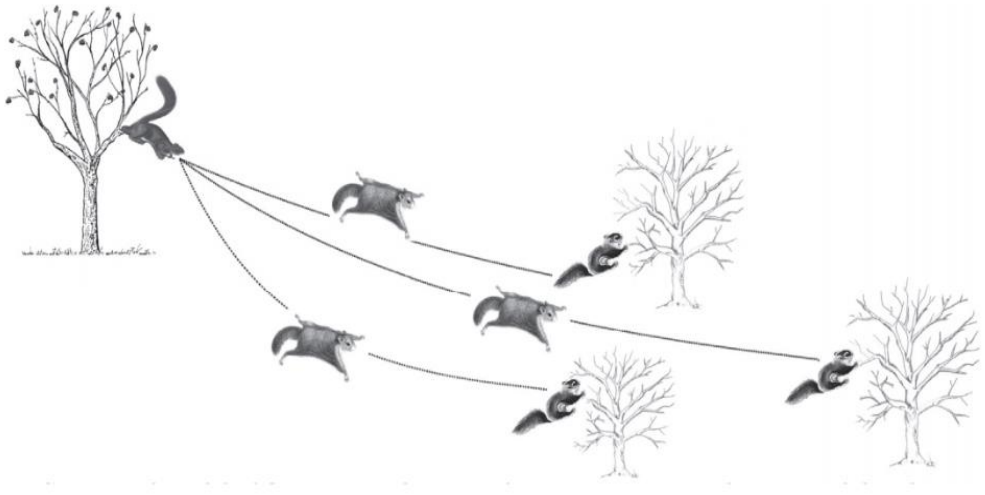
Burada, R_2 [0, 1] aralığında rastgele bir sayıdır.

Durum 3. Normal ağaçlar üzerinde olan ve meşe palamudu tüketip günlük ihtiyacını karşılamış olan bazı sincaplar, soğuk günlerde tüketecekleri cevizlerin depolanması için, ceviz ağacına doğru hareket edebilir. Bu durumda, yeni sincapların bulunduğu yer şu şekilde elde edilmektedir:

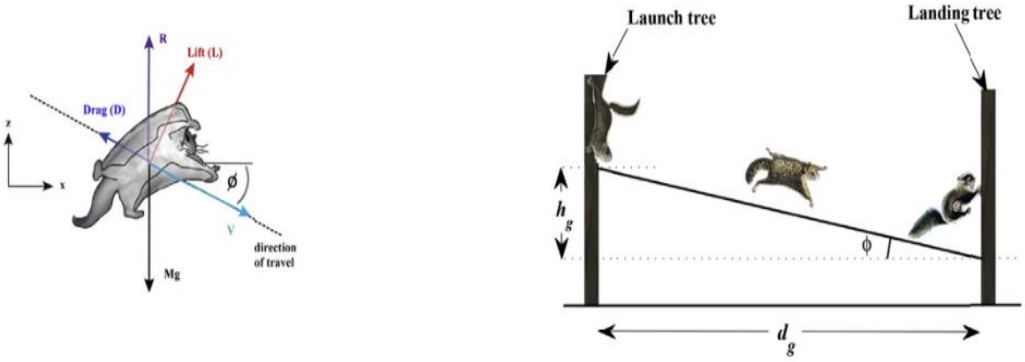
$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^{t} + d_g * G_c * (FS_{ht}^{t} - FS_{nt}^{t}) & R_3 \geq P_{dp} \\ Rastgele\ Konum & Diğer \end{cases} \quad (2.8)$$

Burada, R_3 [0, 1] aralığında rastgele bir sayıdır.

Şekil 2.16, gece boyunca yiyecek arama sırasında sincapların süzülme hareketlerin kavramsal modelini gösterirken, Şekil 2.17 ise uçan sincapların doğrusal süzülme yolunu göstermektedir.



Şekil 2.16 Uçan Sincapların Süzülme Hareketinin Kavramsal Modeli



Şekil 2.17 Uçan Sincapların Doğrusal Süzülme Yolunun Gösterimi

V. Süzülme Hareketinin Aerodinamiği

Uçan sincapların süzülme mekanizması, kaldırma (L) ve sürüklenme (D) kuvvetinin toplamının bileşkesi olan, (R) kuvveti olarak tarif edilmektedir. Bu R kuvvetinin büyüklüğü uçan sincap ağırlığı (M_g) kadar, ancak sincap ağırlığına ters yönlüdür. Böylece R, sabit hızla (V) uçan sincaplara Şekil 2.17'de gösterildiği gibi doğrusal bir süzülme yolu sağlar [113]. Bu çalışmada, optimizasyon algoritması tasarımında yaklaşık bir süzülme davranışı modeli kullanılmaktadır.

Sabit hızda kayan uçan bir sincap, her zaman φ yatay ve kaldırma-sürükleme oranı veya süzülme oranındaki bir açıda alçalmaktadır [114];

$$L/D = 1/tan\varphi \tag{2.9}$$

Uçan sincaplar, daha küçük süzülme açısı (φ) yaparak süzülme yolunun uzunluğunu arttırabilir ve böylece kaldırma-sürüklenme oranı artar. Burada kaldırma kuvveti, uçan sincapların kanatlarının üstünden geçen havanın aşağı doğru sapmasından kaynaklanmakta ve Eşitlik 2.10 ile tanımlanmaktadır:

$$L = 1/2\rho\ C_L V^2 S \quad (2.10)$$

Burada, ρ (= 1.204 kgm^{-3}) hava yoğunluğu, C_L Kaldırma katsayısı, V (= 5.25 ms^{-1}) hız ve S (= 154 cm^2) gövdenin yüzey alanıdır [114].

Sürtünme sürüklemesi Eşitlik 2.11'de belirtildiği gibidir;

$$D = 1/2\rho\ V^2 S\ C_D \quad (2.11)$$

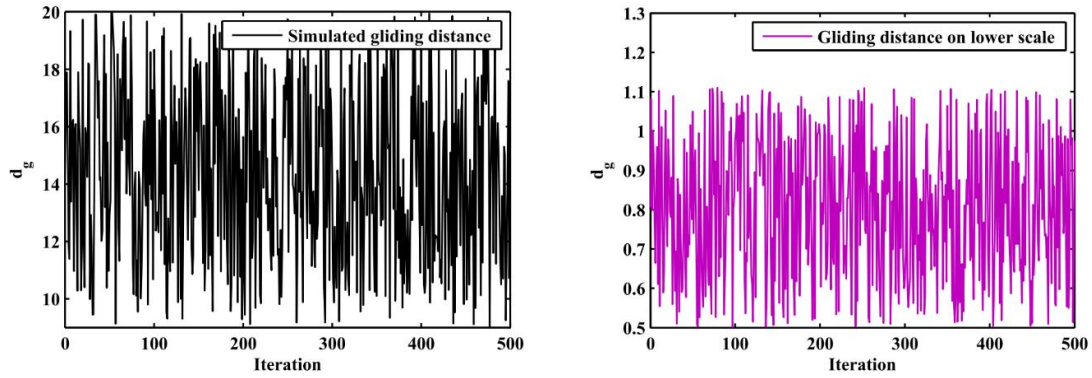
Eşitlikte, C_D Sürtünme sürükleme katsayısıdır. Yavaş hızda, bu sürükleme bileşeni çok yüksektir, yüksek hızda ise küçülür. Dolayısıyla, sabit durumda süzülme açısı Eşitlik 2.12 ile belirlenmektedir:

$$\Phi = \arctan (D/L) \quad (2.12)$$

Yaklaşık süzülme mesafesi (d_g) Eşitlik 2.13 ile hesaplanmaktadır;

$$d_g = \left(\frac{h_g}{tan\varphi}\right) \quad (2.13)$$

Eşitlik 2.13'de belirtilen h_g (= 8 m), süzülmeden sonra meydana gelen yükseklik kaybıdır. d_g'yi hesaplamak için gereken C_L ve C_D' de dahil olmak üzere tüm parametrik değerler gerçek veriler dikkate alınarak belirlenmiştir [113]. Bu nedenle uçan bir sincap, istenen iniş yerine göre kaldırma-sürüklenme oranını basitçe değiştirerek süzülme yolu uzunluğunu veya d_g'yi değiştirebilir. C_L, 0.675≤ CL ≤ 1.5 aralığında rasgele varyasyonlar dahil edilerek gerçekleştirilmektedir ve C_D'nin 0.60 olarak sabit olduğu kabul edilmektedir [1].



Şekil 2.18 sf=18 Olduğu Varsayılarak Uçan Sincapların Yol Uzunluğunun Ölçeklenmesi

Uçan sincaplar genellikle tek süzülmede yatayda 5 metre ila 25 metre arasında bir mesafe kat eder [110]. Önerilen modelde süzülme mesafesi Şekil 2.18'de gösterildiği gibi 9 -20 metre aralığında olduğu varsayılmıştır. d_g'nin değeri oldukça büyüktür, bu nedenle algoritmanın yetersiz performans göstermesine neden olabilir. Algoritmanın kabul edilebilir performansı için d_g değeri aşağı ölçeklenmelidir. d_g, referans fonksiyonları üzerinde deneyler yapılarak elde edilen ölçeklendirme faktörü (sf) olarak adlandırılan uygun bir değere bölünür. d_g'nin ölçeklenmesi işlemi Şekil 2.18'de gösterilmektedir. Bu çalışmada, sf = 18 olarak kabul edildiğinde d_g [0.5, 1.11] aralığında rastgele bir sayı olarak kabul edilmektedir. Böylece, sf, verimli bir metasezgisel tasarlamak için algoritmanın ideal sonuçlara yaklaşmasına yardımcı olmaktadır.

VI. Mevsimsel İzleme Koşulu

Mevsimsel değişiklikler, uçan sincapların yiyecek arama aktivitesini önemli ölçüde etkilemektedir [115]. Uçan sincaplar yüksek vücut sıcaklığına sahip oldukları için düşük sıcaklıklarda önemli ısı kaybı yaşarlar. Aktif yırtıcı hayvanların varlığı ve küçük boyutlara sahip olmaları yiyecek arama davranışının daha maliyetli ve riskli olmasına sebep olmaktadır. İklim koşulları sebebiyle soğuk kış aylarında daha pasiftirler [112]. Bu nedenle uçan sincapların hareketi hava değişimlerinden etkilenir ve bu tür davranışların dahil edilmesi optimizasyona daha gerçekçi bir yaklaşım sağlamaktadır.

Bu nedenle, kullanılan algoritmanın yerel optimum çözümlere takılmasını önleyen bir mevsimsel izleme koşulu kullanılmaktadır. Davranış modellemesinde aşağıdaki adımlar yer almaktadır [1]:

Öncelikle Eşitlik 2.14 kullanılarak mevsimsel sabit değer bulunmalıdır. (S_c)

$$S_c^t = \sqrt{\sum_{k=1}^{d}\left(FS_{at,k}^t - FS_{ht,k}\right)^2} \tag{2.14}$$

t = 1, 2, 3. olarak alınmaktadır.

Daha sonra mevsimsel izleme durumu kontrol edilmelidir. Mevsimsel izleme koşuluna göre S_c^t<S_{min} şartını sağlamalıdır. Mevcut iterasyonda hesaplanan S_c değeri mevsimsel sabit S_{min} değerinden küçük olmalıdır.

S_{min} , mevsimsel sabitin minimum değeri Eşitlik 2.15 kullanılarak hesaplanmaktadır:

$$S_{min} = \frac{10.E^{-6}}{(365)^{t/(t_m/2,5)}} \tag{2.15}$$

Eşitlik 2.15'de kullanılan t ve t_m sırasıyla mevcut ve maksimum iterasyon değerlerini ifade etmektedir. Etkili bir metasezgisel algoritma olması için, S_{min} değeri ideal olmalıdır [116]. Her ne kadar bu denge sabit G_c değeri ile korunsa da ancak iterasyonlar sırasında S_{min} değerinin kontrollü olarak değiştirilmesiyle geliştirilebilir.

Mevsimsel izleme koşuluna göre kış mevsimi bitmiş ise, en uygun kış besin kaynağı için ormanı keşfedemeyen uçan sincaplar rastgele konumlandırılmaktadır.

VII. Kış Mevsimi Sonunda Rastgele Yer Değiştirme

Kış mevsiminin bitmesiyle, düşük yiyecek arama maliyeti nedeniyle uçan sincaplar aktif hale gelecektir. Kışa en uygun yiyecek olan cevizi bulamayan, ormanı keşfedemeyen ve kış bitiminde hala hayatta olan uçan sincapların, daha iyi bir besin kaynağı bulabilmek için farklı yönlere hareket edeceği varsayılmaktadır. Bu davranışın modellemeye dahil edilmesi, önerilen algoritmanın keşif yeteneğini artırmaktadır [1].

Bu uçan sincapların yer değiştirmesi Eşitlik 2.16 ile modellenmiştir:

$$FS_{nt}^{new} = FS_L + L\text{évy(n)} * (FS_u - FS_L) \tag{2.16}$$

Lévy dağılımı çeşitli metasezgisel algoritmaların küresel arama yeteneğini daha iyi ve verimli hale getirmek için kullanılan güçlü bir matematiksel dağılımdır [117]. Lévy dağılımı matematiksel olarak Eşitlik 2.17'de verildiği gibi ifade edilmektedir;

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}\ exp}\left[-\frac{\gamma}{2(s-\mu)}\right] * \frac{1}{(s-\mu)^{\frac{3}{2}}} & 0 < \mu < s < \infty \\ 0 & Diğer \end{cases} \quad (2.17)$$

Eşitlik 2.17'de kullanılan μ, γ> 0 olmak üzere, y ölçek parametresini ve μ kaydırma parametresini ifade etmektedir.

Lévy uçuşu ise Eşitlik 2.18'de gösterildiği gibi hesaplanmaktadır:

$$\text{Lévy(x)} = 0{,}01 * \frac{r_a * \sigma}{|r_b|^{1/\beta}} \quad (2.18)$$

r_a ve r_b, normalde [0, 1] aralığında rastgele dağıtılmış iki sayı ise, β bu çalışmada 1.5 olarak kabul edilen bir sabittir ve σ, Eşitlik 2.19 kullanılarak hesaplanmaktadır:

$$\sigma = \left[\frac{\Gamma(1+\beta) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) * \beta * 2^{\left(\frac{\beta-1}{2}\right)}}\right]^{1/\beta} \quad (2.19)$$

$$\Gamma(\text{x}) = (\text{x} - 1)! \quad (2.20)$$

VIII. Durdurma Kriteri

Fonksiyon toleransı, son iki ardışık sonuç arasında izin verilen ancak küçük bir eşik değerinin tanımlandığı yaygın olarak kullanılan bir yakınsama kriteridir. Bazen maksimum çalışma süresi de durdurma kriteri olarak kullanılabilmektedir. Bu çalışmada maksimum iterasyon sayısı durdurma kriteri olarak kabul edilmektedir.

2.2.1.4 Sincap Arama Algoritması Çalışma Yapısı

Genel olarak, tüm doğadan esinlenen metasezgisel algoritmalar, doğanın iki farklı özelliğini, yani yüzeysel olarak onlara benzer bir görünüm veren uyarlanabilirliği ve en iyiyi bulma davranışını taklit etmektedir. Bu algoritmaların çoğu, incelenen optimizasyon probleminin rastgele oluşturulmuş çözümlerinden oluşan örüntü matris kavramını kullanmaktadır [118]. Örüntü matrisi, her iterasyonda uygun bir güncelleme mekanizması ile tekrar tekrar güncellenmektedir. Bu mekanizma, çözüm çeşitliliğini korurken, örüntü matrisine yeni özellikler veya örüntüler oluşturur. Metasezgisel algoritmalar, genellikle çözüm güncelleme stratejilerine dayanarak farklılaştırılmaktadır.

Sincap Arama Algoritmasının aşamaları ve çalışma yapısını özetleyen sözde kodu Tablo 2.4'te verildiği gibidir [1].

Tablo 2.4 Sincap Arama Algoritması Sözde Kodu

Başla: Giriş parametrelerinin tanımlanması
N adet uçan sincap için, rastgele başlangıç konumlarının belirlenmesi
Uçan sincapların konumlarının uygunluk değerlerinin hesaplanması
Uçan sincapların konumlarının uygunluk değerlerine göre artan sırada sıralanması, bu sıralamaya göre ceviz ağacı, meşe palamudu ve normal ağaçlara konumlandırılması
Normal ağaçlarda bulunan sincapların rastgele olarak, bazılarının ceviz ağacına, bazılarının meşe palamudu ağaçlarına yönlendirilmesi
While (durdurma kriteri)
For t=1 n1 (n1=meşe palamudu ağaçlarından ceviz ağacına doğru hareket eden toplam uçan sincap sayısı)
If (R1≥P_{dp})
Eşitlik 2.6
Else
Rastgele Konum
End
End
For t=1 n2 (n2=normal ağaçlardan meşe palamudu doğru hareket eden toplam uçan sincap sayısı)
If (R2≥P_{dp})
Eşitlik 2.7
Else
Rastgele Konum
End
End
For t=1 n3 (n3=normal ağaçlardan ceviz ağacına doğru hareket eden toplam uçan sincap sayısı)
If (R3≥P_{dp})
Eşitlik 2.8
Else
Rastgele Konum
End
End

Mevsimsel Sabiti Hesapla (S_c) Eşitlik 2.14
If ($S_c < S_{min}$)
Uçan Sincapların Rastgele Konumlandırılması Eşitlik 2.16
End
Minimum Mevsimsel Sabit Değerinin Güncellenmesi (S_{min}) Eşitlik 2.15
End
Ceviz ağacındaki uçan sincap optimum çözümdür.
End

SSA'nın performansı, klasik ve modern sayısal optimizasyon problemleri üzerinde analiz edilmiştir. Bu fonksiyonlar sürekli, süreksiz, doğrusal, doğrusal olmayan, tek biçimli, çok modlu, dışbükey, dışbükey olmayan, ayrılabilir ve ayrılamaz olarak tanımlanmaktadır. Arama alanı üstel olarak arttıkça, işlev boyutlarındaki artışla birlikte problem güçlüğünün arttığı düşünülmektedir [119].

SSA'nın performansını değerlendirmek için yapılan deneysel testler, algoritmanın faydalanma yeteneğini doğrularken, keşif özelliklerini kontrol etmektedir. Her deneysel çalışmada, SSA'nın performansı, GA, PSO, BA ve FF gibi doğadan ilham alan optimizasyon algoritmaları ile karşılaştırılmış, bu algoritmalara oranla daha başarılı sonuçlar elde edilmiştir [1].

Yapay sinir ağlarının daha önce birçok algoritma ile eğitildiği ve başarılı sonuçlar elde edildiği bilinmektedir. Bu bilgiler ışığında, Oxford Mutluluk Ölçeği gerçek verileri ve literatürdeki bazı diğer sınıflandırma veri kümeleri üzerinde eğitiminin SSA ile optimize edildiği YSA uygulanmış ve modelin hiç görmediği test verileri üzerinde performans analizleri gerçekleştirilmiştir.

3. BÖLÜM

UYGULAMA VE BULGULAR

3.1 Veri Kümeleri

Bu çalışmada kullanılan veri kümeleri UCI makine öğrenmesi veri deposundan (http://archive.ics.uci.edu/ml) alınmıştır. YSA modelinin farklı veri kümeleri için ne kadar iyi bir performans sergileyebileceğini incelemek amacıyla farklı özelliklere sahip olan veri kümeleri seçilmiştir. Bu veri kümeleri literatürde sınıflandırma problemleri için kullanılmaktadır, bu veri kümelerinin özellikleri kısaca Tablo 3.1'de belirtildiği gibidir [120];

Tablo 3.1 UCI Makine Öğrenmesi Deposundan Alınan Veri Kümelerinin Özellikleri

Veri Kümesi	**Nitelik Sayısı**	**Sınıf Sayısı**	**Örnek Sayısı**	**Nitelik Karakteristiği**
Wine	13	3	178	Tamsayı, Real
Ionosphere	34	2	351	Tamsayı, Real
WDBC	32	2	569	Real
Zoo	17	7	101	Tamsayı
Glass	10	7	214	Real
Nursery	8	5	12960	Kategorik

Tablo 3.1'de verilen veri kümeleri nitelik sayısı, çıktı sayısı ve örnek sayısı bakımından farklı özelliklere sahiptir.

Ionosphere ve Wisconsin Diagnostic Breast Cancer (WDBC) veri kümelerinde nitelik sayısı 30'un üzerindeyken, diğer veri kümelerinde daha az nitelik bulunmaktadır. Örnek sayısı Nursery veri kümesinde 12960 iken diğer veri kümelerinde örnek sayısı daha azdır. Ionosphere ve WDBC veri kümeleri iki sınıfa sahiptir. Zoo ve Glass veri kümelerinde ise 7 adet sınıf bulunmaktadır.

Literatürde bulunan bu veri kümelerine ek olarak, bir gerçek hayat verisi olan "Oxford Mutluluk Veri Kümesi" ile de sınıflandırma işlemi gerçekleştirilmiştir. Oxford Mutluluk veri kümesine ait veriler Oxford mutluluk ölçeği kullanılarak oluşturulmuştur.

Yapılan çalışmada; anket yoluyla Oxford mutluluk ölçeği verisi toplanmıştır. Araştırma katılımcılarını, 18-25 yaş arası Türk üniversite öğrencileri oluşturmaktadır. Toplanan veriler istatistiksel olarak değerlendirilip veri kümesi oluşturulmuştur.

Oxford mutluluk anketinde bulunan toplam 29 soru veri kümesinin girdilerini oluştururken, değerlendirilen veriler sonucunda alınan sonuçlar, veri kümesinin sınıflarını oluşturmaktadır. Bu çalışmada kullanılan gerçek hayat veri kümesinin toplam 469 örneği bulunmaktadır.

Oxford Mutluluk Ölçeğinde bulunan ve veri kümesinin girdilerini oluşturan toplam 29 soru aşağıda belirtildiği gibidir:

1. Genelde çok gülen biriyimdir.
2. Hayat güzeldir.
3. Genelde pek çok şeyi eğlenceli bulurum.
4. Neredeyse herkese karşı oldukça sıcak duygular besliyorum.
5. Hayatın oldukça ödüllendirici olduğunu hissediyorum.
6. Diğer insanlara karşı oldukça ilgiliyim.
7. Yaptığım şeylerle kararlı bir şekilde ilgilenirim.
8. Çok mutluyum.
9. Hayatımdaki her şeyden oldukça memnunum.
10. Çevremdeki güzellikleri fark ederim.
11. Özellikle gelecek hakkında pek iyimser değilim.
12. Nadiren dinlenmiş olarak uyanırım.
13. Diğer insanlar üzerinde daima neşeli bir etki bırakırım.

14. Her şeyi kontrol edebilecek güçte hissediyorum.
15. Kendimi oldukça enerjik hissediyorum.
16. Genellikle olaylar üzerinde olumlu etkim vardır.
17. Yapmak istediğim her şeyi yapabilirim.
18. Zihinsel olarak kendimi tamamen zinde hissediyorum.
19. Yaptıklarımla yapmak istediklerim arasında büyük fark var.
20. Çekici birisi olduğumu düşünmüyorum.
21. Genellikle neşeli birisiyim.
22. Dünyanın iyi bir yer olduğunu düşünmüyorum.
23. Herhangi bir konuda karar vermekte zorlanırım.
24. Hayatımdan memnun değilim.
25. Yaşamımın kontrolü elimde değilmiş gibi hissediyorum.
26. Yaşamımın belli bir amacı ve anlamı yok.
27. Diğer insanlarla birlikte olmaktan keyif almıyorum.
28. Özellikle kendimi çok sağlıklı hissetmiyorum.
29. Geçmişimle ilgili pek mutlu anılara sahip değilim.

Bu sorular katılımcılara 5'li likert ölçek kullanılarak uygulanmış ve sonuçlar da 5 sınıf oluşturulacak şekilde elde edilmiştir. Tüm girdiler ve çıktıların aynı ölçekte olması sebebiyle sistem tasarlanmadan önce herhangi bir normalizasyon işlemine tabi tutulmamıştır.

Bu ölçeğe göre girdiler aşağıda belirtilen değerleri almaktadır:

1. "Kesinlikle Katılmıyorum"
2. "Katılmıyorum"
3. "Ne katılıyorum Ne de Katılmıyorum"
4. "Katılıyorum"
5. "Kesinlikle Katılıyorum"

Çıktılar aşağıda belirtildiği gibi 5 sınıf oluşturmaktadır;

1. "Çok Mutsuz"
2. "Mutsuz"
3. "Ne Mutlu Ne Mutsuz"

4. “Mutlu”
5. “Çok Mutlu”

Sınıf etiketleri yardımıyla belli bir veri kümesi üzerinden oluşturulan modeller kullanılarak yeni örneklerin hangi sınıfa ait olacağının tahmin edilmesi genel olarak sınıflandırma problemi olarak adlandırılmaktadır. Gerçek hayat verisi olan Oxford Mutluluk Veri kümesi de bir sınıflandırma problemidir.

Oxford Mutluluk Veri kümesine ait özellikleri içeren bilgiler Tablo 3.2’de sunulmuştur.

Tablo 3.2 Oxford Mutluluk Veri Kümesi Özellikleri

Toplam Nitelik Sayısı	29
Nitelik Karakteristiği	Tamsayı
Toplam Sınıf Sayısı	5
Toplam Örnek Sayısı	469

Tablo 3.3 Mutluluk Veri Kümesi Örneği

E-posta Adresi	Kullanıcı adı	Genelde çok gülen biriyimdir.	Hayat güzeldir.
		4	5
		3	1
		3	3
		3	2
		3	3
		4	3
		4	4
		5	5
		4	5
		3	3
		4	3
		5	3
		5	4
		4	5
		4	3
		5	5
		1	3
		4	4
		1	3

Mutluluk veri kümesinde bulunan 469 verinin ilk 20 tanesine ait örnek, Tablo 3.3'te gösterilmiştir.

3.2 Algoritmanın Çalıştırılması ve Etkinlik Ölçütleri

Sincap Arama Algoritması ile Yapay Sinir Ağı Eğitimi esaslı olan bu çalışmanın gerçeklenmesi MATLAB dördüncü nesil programlama dili kullanılarak tamamlanmıştır. Ayrıca deney tasarımı ve veri kümeleri için karşılaştırma adımlarını da içeren çalışma ile ilgili tüm işlemler MATLAB programlama dili kullanılarak gerçekleştirilmiştir.

MATLAB (matrix laboratory), çok paradigmalı sayısal hesaplama yazılımı MathWorks tarafından geliştirilmektedir. MATLAB kullanıcıya, matris işleme, fonksiyon ve veri çizme, algoritma uygulama, kullanıcı ara yüzü oluşturma, optimizasyon, nümerik analiz,

istatistik, lineer cebir alanlarında kullanım ve C, C++, Java ve Fortran gibi diğer dillerde yazılmış programlarla bütünleşik çalışma gibi birçok imkân tanımaktadır.

MATLAB programı aynı zamanda 2D ve 3D grafik çizimi için de kullanılmaktadır. MATLAB mühendislik, bilim ve ekonomi gibi çeşitli alanlarda kullanılmasıyla beraber, endüstriyel işletmelerde olduğu kadar yaygın olarak akademik ve araştırma kurumlarında da kullanılmaktadır.

Eğitim verileri, modelin eğitilerek öğrendiği veriler iken; test verileri, modelin eğitilmeyen veriler üzerinde ne kadar iyi sınıflandırma başarısı gösterdiğini anlayabilmek için kullanılan verilerdir. Bu çalışmada kullanılacak veri kümelerinin eğitim ve test kümelerine parçalanması işlemi "basit bölümleme" esas alınarak gerçekleştirilmiştir. Verilerin %70'lik kısmı eğitim verilerini oluştururken, kalan %30'luk kısmı test verisi olarak kullanılmıştır. Karşılaştırmalarda ele alınan tüm veri kümeleri için bölümleme işlemi aynı şekilde yapılmıştır. Bu sayede veri kümesinde bulunan örneklerin %70'i ile ağın veriyi deneyimlemesi ve öğrenmesi sağlanırken, verinin ne kadar öğrenildiği tüm veri kümesinde bulunan örneklerin %30'u ile test edilmektedir.

Veri kümelerinin basit bölümleme tekniği ile elde edilmiş eğitim ve test kümelerindeki toplam örnek sayıları Tablo 3.4'te gösterildiği gibidir.

Tablo 3.4 Veri Kümelerinin Eğitim ve Test Kümelerine Bölümlenmesi

Veri Kümesi	**Toplam Örnek Sayısı**	**Eğitim için Kullanılan Toplam Örnek Sayısı (%70)**	**Test için Kullanılan Toplam Örnek Sayısı (%30)**
Oxford Mutluluk	469	329	140
Wine	178	125	53
Ionosphere	351	246	105
WDBC	569	399	170
Zoo	101	71	30
Glass	214	150	64
Nursery	12960	9072	3888

Sincap Arama Algoritması kullanılarak ağırlıkları optimize edilmiş YSA modelinin sözde kodu Tablo 3.5'te verilmiştir. Tablo 3.5'te verilen ve uygulanan sözde koda göre, öncelikle YSA ve SSA'ya ait başlangıç parametreleri belirlenmektedir. Bu parametreler, her bir veri kümesi için, tam faktöriyel deney tasarımı yapılarak belirlenecektir. Başlangıç parametrelerinin belirlenmesi adımından sonra, ağ eğitimi gerçekleştirilmesi ile ilgili adımlar tamamlanmıştır. YSA'ya ait en iyi ağırlıkların elde edilmesi işleminde sincap arama algoritması devreye girmektedir. Sınıflandırma başarısı en iyi olan ağırlıklar elde edildikten sonra doğruluk ve MSE etkinlik ölçütlerinin hesaplanması adımları gerçekleştirilmiştir.

Doğruluk etkinlik ölçütünün, eğitim ve test kümeleri için hesaplanması işlemi için Eşitlik 3.1 kullanılırken, MSE değerinin hesaplanması adımında ise Eşitlik 3.2 kullanılmıştır. Algoritmaya ait küresel en iyi ağırlıkların bulunması işlemi ve gerekli hesaplamaların yapılması adımları tamamlandıktan sonra ise bu değerlerin ekrana yazdırılması ile sözde kodda belirtilen tüm adımlar tamamlanmaktadır.

Tablo 3.5 SSA ile Eğitilmiş YSA Modelinin Sözde Kodu

1. Başlangıç Parametrelerinin Belirlenmesi
    - Optimizasyon Parametreleri
        - *Eğitim Verileri*
        - *Test Verileri*
        - *İterasyon Sayısı*
        - *Toplam Sincap Sayısı*
        - *Süzülme Sabitinin* (G_c)
        - *Süzülme Mesafesi* (d_g)
        - *Yırtıcı Hayvan Olasılığı* (P_{dp})
        - *Alt Sınır ve Üst Sınır Değerleri* (LB, UB)
        - *Meşe Palamudu Ağacı Sayısı* (FS_{at})
        - *Ceviz Ağacı Sayısı* (FS_{ht})
        - *Normal Ağaç Sayısı* (FS_{nt})
    - Yapay Sinir Ağları Parametreleri
        - *Gizli Katman Sayısı*
        - *Nöron Sayısı*
        - *Eğitim Fonksiyonu*
2. Yapay Sinir Ağlarının Eğitimi ve Başlangıç Değerlerinin Elde Edilmesi

    $Ağ = eğit(eğitim\ girişleri, eğitim\ çıkışları)$

    $Başlangıç\ Ağırlıkları = Ağırlıkları\ Al(Ağ)$

    $Boyut = boyut(Başlangıç\ Ağırlıkları)$
3. Başlangıç Popülasyonunun Oluşturulması

    $konum_{i,j} = LB + (UB - LB) * rastgele\ sayı$

    $konum_{i,j} = konum'_{i,j}$

$j = Boyut$

4. Başlangıç Popülasyonunun Uygunluk Değerlerinin Hesaplanması

$Ağ = Ağırlıkları\ Ayarla(Ağ, konum(:, i))$

$Çıkış = Benzetim\ (Ağ, Eğitim\ Girişleri)$

$Uygunluk_i = Ortalama\ Hata\ Karesi(Çıkış - Eğitim\ Çıkışları)$

5. Başlangıç Popülasyonu Uygunluk Değerlerine Göre Sırala

$[Uygunluk\ Sırası] = Sırala(Uygunluk)$

$konum = konum(:, Uygunluk\ Sırası)$

$FS_{ht} = konum(:,1)$

$FS_{at} = konum(:, FS_{ht}sayısı + 1 : FS_{ht}sayısı + FS_{at}sayısı)$

$FS_{nt} = konum(:, FS_{ht}sayısı + FS_{at}sayısı + 1 : end)$

6. **while** iter<iter$_{max}$

$En\ İyi(:, iter) = konum(:,1)$

Sincapların Hareket Ettirilerek Yeni Konumların Belirlenmesi

If (rastgele sayı1>=P_{dp})

$FS_{at_{i,j}} = FS_{at_{i,j}} + d_g * Rastgele\ Sayı * G_c * \left(FS_{ht} - FS_{at_{i,j}}\right)$

Else

$FS_{at_{i,j}} = LB + (UB - LB) * Rastgele\ Sayı$

End

If (rastgele sayı2>=P_{dp})

$FS_{nt_{i,j}} = FS_{nt_{i,j}} + d_g * Rastgele\ Sayı * G_c * \left(FS_{at_{i,j}} - FS_{nt_{i,j}}\right)$

Else

$FS_{nt_{i,j}} = LB + (UB - LB) * Rastgele\ Sayı$

End

If (rastgele sayı3>=P_{dp})

$FS_{nt_{i,j}} = FS_{nt_{i,j}} + d_g * Rastgele\ Sayı * G_c * \left(FS_{ht} - FS_{nt_{i,j}}\right)$

Else

$FS_{nt_{i,j}} = LB + (UB - LB) * Rastgele\ Sayı$

End

$Yeni\ Konum = [FS_{ht}, FS_{at}, FS_{nt}]$

Mevsimsel Sabiti Hesapla (S_c)

$$S_c^t = \sqrt{\sum_{k=1}^{d} \left(FS_{at,k}^t - FS_{ht,k}\right)^2}$$

Mevsimsel Sabitin Minimum Değerini Hesapla (S_{min})

$$S_{min} = \frac{10.E^{-6}}{(365)^{t/(t_m/2,5)}}$$

If ($S_c < S_{min}$)

Uçan Sincapların Rastgele Konumlandırılması

$FS_{nt}^{new} = LB + Lévy(n) * (UB - LB)$

End

Minimum Mevsimsel Sabit Değerinin Güncellenmesi (S_{min})

$$S_{min} = \frac{10.E^{-6}}{(365)^{t/(t_m/2,5)}}$$

End

Ceviz ağacındaki uçan sincap optimum çözümdür.

End

7. Yeni Konumların Uygunluk Değerlerinin Hesaplanması
   $Ağ = Ağırlıkları\ Ayarla(Ağ, Yeni\ Konum(:, i))$
   $Çıkış = Benzetim\ (Ağ, Eğitim\ Girişleri)$
   $Uygunluk_i = Ortalama\ Hata\ Karesi(Çıkış - Eğitim\ Çıkışları)$
8. Yeni Konumların Uygunluk Değerlerine Göre Sıralanması
   $[Uygunluk\ Sırası] = Sırala(Uygunluk)$
   $Yeni\ Konum = Yeni\ Konum(:, Uygunluk\ Sırası)$
   $FS_{ht} = Yeni\ Konum(:,1)$
   $FS_{at} = Yeni\ Konum(:, FS_{ht}sayısı + 1 : FS_{ht}sayısı + FS_{at}sayısı)$
   $FS_{nt} = Yeni\ Konum(:, FS_{ht}sayısı + FS_{at}sayısı + 1 : end)$
9. Durdurma kriteri sağlanıyorsa 11. Adıma git, sağlanmıyorsa iter=iter+1 ve 6. adıma geri dön
10. En İyiler için Uygunluk Değerini Kontrol Et ve Sırala, Küresel En İyiyi Belirle
    $Ağ = Ağırlıkları\ Ayarla(Ağ, En\ İyi(:, i))$
    $Çıkış = Benzetim\ (Ağ, Eğitim\ Girişleri)$
    $Uygunluk_i = Ortalama\ Hata\ Karesi(Çıkış - Eğitim\ Çıkışları)$
    $[Uygunluk\ Sırası] = Sırala(Uygunluk)$
    $En\ İyi = En\ İyi(:, Uygunluk\ Sırası)$
    $Küresel\ En\ İyi = En\ İyi(:,1)$
11. Küresel En İyiyi Yapay Sinir Ağları Ağırlığı Olarak Ayarla ve Test Verisi İçin Test Et
    $Ağ = Ağırlıkları\ Ayarla(Ağ, Küresel\ En\ İyi)$
    $Çıkış = Benzetim\ (Ağ, Test\ Girişleri)$
    $Test\ Doğruluğu = (say(Test\ Verisi\ Çıkışları == Çıkış))/100$
    $Test\ Hatası = Ortalama\ Hata\ Karesi(Test\ Verisi\ Çıkışları - Çıkış)$

SSA kullanılarak eğitilen yapay sinir ağlarının başarısı eğitim ve test verileri üzerinde doğruluk (accuracy) değerleri ve ortalama kareli hata değeri (MSE) göz önünde bulundurularak değerlendirilmiştir. Doğruluk değeri, tahminleme işleminde kaç tane değerin gerçek değerle aynı olduğunu % değeri olarak hesaplamaktadır.

Doğruluk etkinlik değeri aşağıda yer alan Eşitlik 3.1'de belirtildiği gibi formüle edilmektedir;

$$Doğruluk = \frac{Doğru\ Tahmin\ Sayısı}{Toplam\ Örnek\ Sayısı} * 100 \quad (3.1)$$

Bir öğrenme yönteminin veri kümeleri üzerindeki performansını değerlendirmek için etkinlik ölçütleri kullanılmaktadır. MSE değeri kullanılarak hesaplanan ağırlık değerlerinin gerçek değerlere ne kadar yaklaştığı gözlemlenmektedir. Bu nedenle etkinlik ölçütleri, öğrenme performansını etkilemekte olan önemli kriterlerden bir tanesidir. Bu çalışmada etkinlik ölçütü olarak Hata Kareleri Ortalaması (MSE) kullanılmıştır.

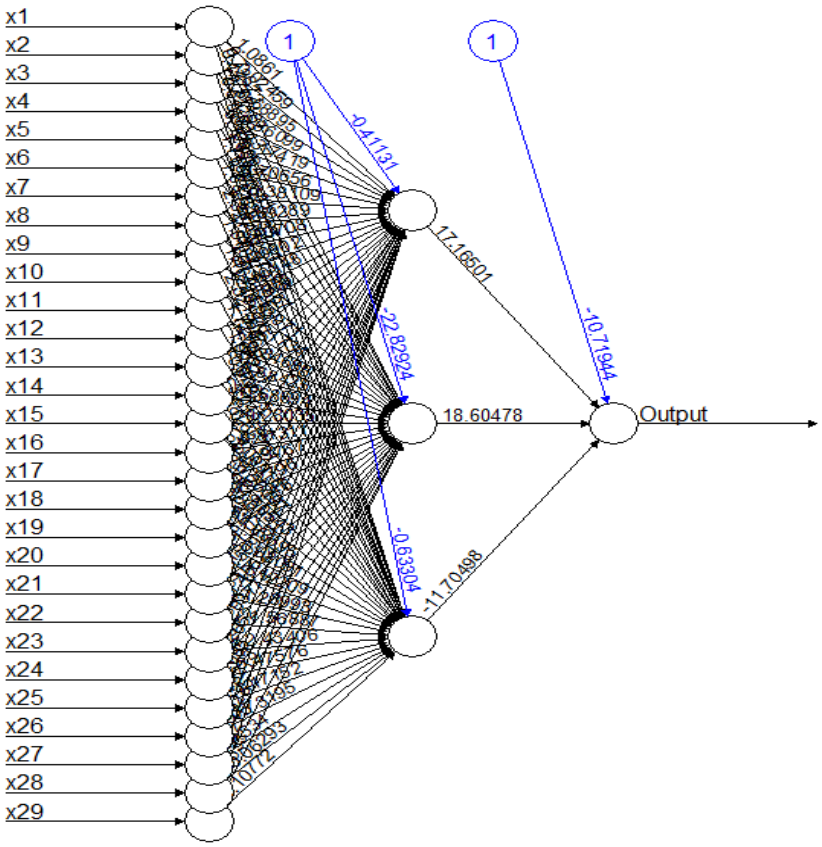
MSE değeri Eşitlik 3.2'de belirtildiği gibi formüle edilmektedir;

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{3.2}$$

Eşitlik 3.2'de belirtilen değerlerin ifadesi aşağıda verildiği gibidir.

$f_i = i.$örnek için tahmin değeri
$y_i = i.$örneğin gerçek çıktı değeri
$n =$ Tahminleme yapılan toplam örnek sayısı

Tahmin edilen değerler gerçek değerlere ne kadar yakınsa MSE değeri o kadar küçük olmaktadır. Tahminler gerçek değerlerden uzaklaştığında ise MSE değeri büyümektedir. Eğitilmiş yapay sinir ağlarının performansını; yırtıcı hayvan olasılığı, süzülme katsayısı, toplam sincap sayısı gibi SSA parametreleri ve YSA yapısında bulunan gizli katman sayısı, gizli katmanlarda bulunan nöron sayısı, iterasyon sayısı, aktivasyon fonksiyonu gibi birçok değişken etkilemektedir. Gerçek hayat verilerinden meydana gelen mutluluk verilerine ait başlangıçta belirlenen YSA yapısı Şekil 3.1'de gösterilmiştir.



Şekil 3.1 Mutluluk Verilerine Ait YSA Yapısı

Bu çalışmada SSA kullanılarak yapay sinir ağları eğitilmiş ve en iyi sınıflandırma doğruluğunu veren ağırlıklarının belirlenmesi işlemi gerçekleştirilmiştir. Çalışma sonucunda elde edilen ağırlık değerleri YSA'nın çalışma performansını doğrudan etkilemektedir.

Sincap Arama Algoritmasının güncelleme mekanizmasında kullanılacak olan sincapların ağaçlar üzerindeki başlangıç konumlarının oluşturulabilmesi için ve her iterasyonda daha başarılı performans sergileyebilmesi için bir uygunluk fonksiyonuna ihtiyaç duyulmaktadır. Bu çalışmada MSE değeri uygunluk fonksiyonu olarak kullanılmıştır.

MSE değerini minimize eden ağırlıkların bulunması ve eğitim, test verileri için doğruluk oranlarının maksimize edilmesi bu çalışmanın etkinlik ölçütleridir. En iyi sınıflandırma doğruluğunu veren parametrelerin belirlenmesi için çalışmada "Deney Tasarımı" gerçekleştirilmiştir.

3.3 Deney Tasarımı

Çalışmada doğru sonuçlara ulaşabilmek için doğru bir deney tasarımı yapmak, parametreleri doğru belirlemek oldukça önem teşkil etmektedir. Deney tasarımındaki amaç, deney hatalarını minimum düzeye indirebilmektir.

Mühendislik alanında oldukça yaygın kullanıma sahip olan faktöriyel tasarımlar Fisher ve Yates tarafından 1920'li yıllarda önerilmiş tasarımlardır [121]. Faktöriyel tasarım iki veya daha fazla faktörün etkilerini araştırmak için kullanılmaktadır [122]. Tam faktöriyel deney tasarımı en az iki parametreden oluşup, bu parametrelerin en az iki düzeyinin bulunması durumunda, tüm düzeylerin kombinasyonlarının incelenmesidir.

Bir deney tasarımında kullanılan parametrelerin hiçbiri birbirinden bağımsız değildir. Tüm parametreleri düzeylerini en iyi şekilde seçmek, modelin performansını oldukça yükseltmektedir. Bu yüzden bu çalışmada deney tasarımı gerçekleştirilmiştir.

Bu bölümde, her bir veri kümesi için SSA kullanarak eğitilmiş YSA modelinin parametrelerinin en iyi düzeylerinin belirlenmesi ve bu parametreler sayesinde modelin performansını arttırmak amaçlanmıştır.

Bu çalışmada her bir parametre için düzeylerin kombinasyonlarının incelenmesi ve bu parametrelerin düzeyleri arasından en iyi sınıflandırma başarısının elde edilmesi MATLAB yazılım dili kullanılarak gerçekleştirilmiştir.

Başlangıçta tam faktöriyel deney tasarımı toplamda 8 parametre için 3 düzeyli olarak gerçekleştirilmiştir.

Bu parametrelerden bir tanesi "Aktivasyon Fonksiyonu" olarak belirlenmiştir. Aktivasyon fonksiyonuna ait düzeyler aşağı verildiği gibidir;

- Doğrusal
- Sigmoid
- Tanjant hiperbolik

Ancak bu parametreler deney tasarımında gözlemlendiğinde doğrusal ve tanjant hiperbolik aktivasyon fonksiyonlarının veri kümeleri üzerinde başarılı performans göstermediği gözlemlenmiştir.

Bu nedenle deney tasarım parametreleri içerisinden aktivasyon fonksiyonu çıkarılmıştır. Çalıştırılan modelde yapay sinir ağına ait aktivasyon fonksiyonu "Sigmoid" olarak belirlenmiştir. Aktivasyon fonksiyonunun sabit bir düzey olarak belirlenmesi ile deney tasarımı için kullanılacak parametre sayısı 7 olarak güncellenmiştir.

SSA ile yapay sinir ağlarının eğitiminde yapılan denemeler göz önünde bulundurularak, deney tasarımı parametre düzeyleri belirlenmiştir. Sincap Arama Algoritması parametre düzeyleri eşit aralıklarda arttırılmış ve azaltılmıştır. G_c (Süzülme Katsayısı) ve P_{dp} (Yırtıcı Hayvan Olasılığı) parametrelerinde öncü çalışmadaki değerleri de birer düzey olarak ele alınmıştır [1].

Bu çalışmada kullanılan veri kümeleri farklı özelliklerden oluşmaktadır. Bu nedenle her veri kümesinin ayrı ayrı ele alınması ile farklı parametreler elde edilebilir. Ancak bu şekilde her bir veri kümesi için parametrelerin doğru düzeylerinin kullanılması söz konusudur. Tam faktöriyel deney tasarımında MSE değerleri temel alınarak her bir veri kümesi için faktörlerin ana etkileri dikkate alınmıştır.

Deney Tasarımı parametrelerine ait düzeyler Tablo 3.6'da gösterilmiştir.

Tablo 3.6 Deney Tasarımı Parametreleri

YSA Parametreleri	Gizli Katman Sayısı	1	2	3
	Gizli Katman Nöron Sayısı	10	15	20
	İterasyon Sayısı	500	1000	1500
SSA Parametreleri	Toplam Sincap Sayısı	50	75	100
	Meşe Palamudu Ağacı Sayısı	5	10	15
	G_c (Süzülme Katsayısı)	1,7	1,9	2,1
	P_{dp} (Yırtıcı Hayvan Olasılığı)	0,1	0,2	0,3

3.3.1 UCI Referans Sınıflandırma Veri Kümeleri Deney Tasarımı

Önerilen SSA kullanılarak eğitilen YSA modelinde Tablo 3.6'da belirlenen deney tasarımı parametre düzeyleri her bir veri kümesi için uygulanmış ve bu sonuçlar arasından en başarılı sınıflandırma performansları gözlemlenerek Tablo 3.7'de sunulmuştur.

Tablo 3.7'de her bir veri kümesi için belirlenmiş en iyi parametre düzeylerinin bilgisi bulunmaktadır. Veri kümeleri için seçilen parametre düzeyleri ile model kurulduğunda en başarılı sınıflandırma performansının elde edildiği gözlemlenmiştir. En iyi deney tasarımı parametre düzeyleri, her veri kümesi için değişkenlik gösterebilmektedir. Bu durum veri kümelerinin özelliklerinin farklı oluşundan kaynaklanmaktadır.

Süzülme katsayısı (G_c), yırtıcı hayvan olasılığı (P_{dp}) ve iterasyon sayısı parametreleri ise her bir veri kümesi için aynı değer olarak deney tasarımı sonucunda belirlenmiştir. Sincap Arama Algoritması öncü çalışmasında belirtilen Süzülme katsayısı (G_c) ve Yırtıcı Hayvan Olasılığı (P_{dp}) parametreleri en iyi parametreler olarak gözlemlenmiştir [1].

Bu değerler aşağıda belirtildiği gibidir.

Süzülme katsayısı (G_c) = 1.9
Yırtıcı Hayvan Olasılığı (P_{dp}) = 0.1

Gizli Katman sayısı 6 veri kümesi için de 2 katman ve 1000 iterasyon sayısı en iyi parametreler olarak belirlenmiştir. Sincap Arama Algoritması parametrelerinden olan;

toplam sincap sayısı ve meşe palamudu ağacı sayısı veri kümeleri için oldukça değişken deney tasarımı sonuçları ortaya çıkarmaktadır.

Wine ve Glass veri kümelerinin özellikleri birbirinden farklı olmasına rağmen tüm en iyi deney tasarımı parametreleri aynı düzeydir. Bu iki veri kümesi dışında diğer veri kümelerinin parametre düzeyleri kombinasyonları farklı elde edilmiştir.

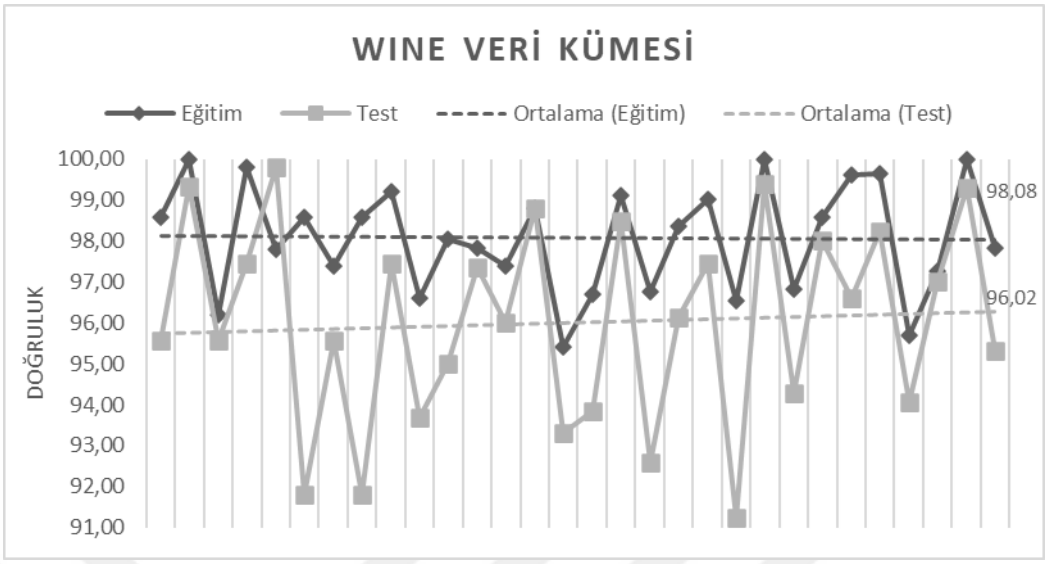
Tablo 3.7'de belirtilen en iyi deney tasarımı parametreleri kullanılarak her bir veri kümesi için önerilen SSA ile yapay sinir ağlarının eğitimi algoritması 30'ar kez çalıştırılmış ve sonuçlar elde edilmiştir. Veri kümeleri için yapılan bu gözlemlerin her biri birbirinden bağımsızdır.

Bu çalışmada kullanılan 6 veri kümesinde SSA ile eğitilmiş YSA modelinin 30'ar kez çalıştırılması sonucunda elde edilen ortalama MSE, Eğitim Doğruluğu, Test Doğruluğu, değerleri ve standart sapmalar gözlemlenmiştir.

Tablo 3.7 Her Bir Veri Kümesi için En İyi Deney Tasarım Parametreleri

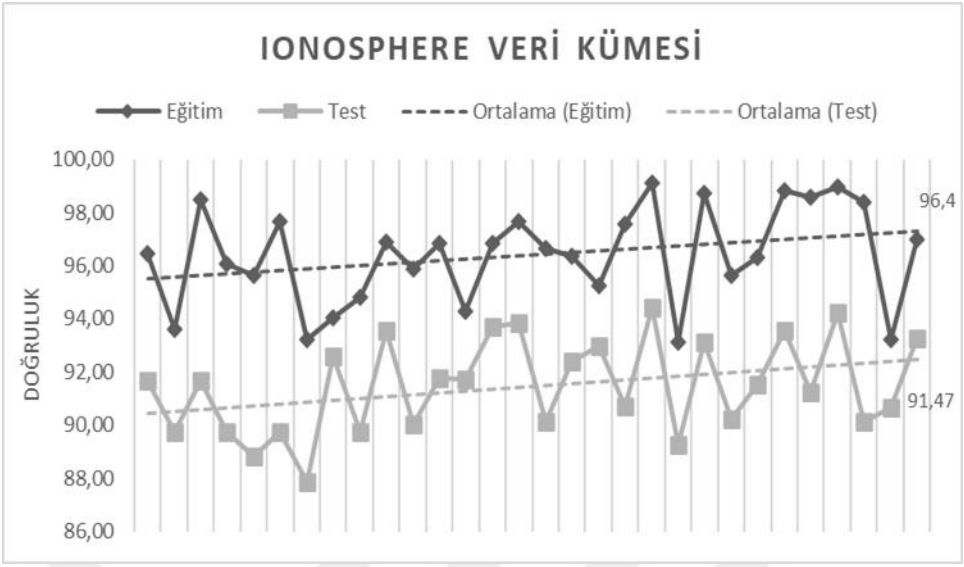
Veri Kümesi / Parametreler	**Wine**	**Ionosphere**	**WDBC**	**Zoo**	**Glass**	**Nursery**
İterasyon Sayısı	1000	1000	1000	1000	1000	1000
Gizli Katman Sayısı	2	2	2	2	2	2
Nöron Sayısı	20	15	15	15	20	15
Toplam Sincap Sayısı	75	100	75	50	75	50
Meşe Palamudu Ağacı Sayısı	10	10	5	15	10	10
G_c (Süzülme Katsayısı)	1.9	1.9	1.9	1.9	1.9	1.9
P_{dp} (Yırtıcı Hayvan Olasılığı)	0.1	0.1	0.1	0.1	0.1	0.1

Wine Veri Kümesi: Veri kümesi için 30 kez çalıştırma sonucunda eğitim doğruluğu %98.08 ±1.33 olarak elde edilmiştir. Test verilerinin sınıflandırma doğruluğu ise %96.02 ± 2.46'dır. Veri kümesine ait MSE değeri 0.013 ± 0.009 olarak elde edilmiştir. Sincap arama algoritması ile eğitilmiş yapay sinir ağının Wine veri kümesine ait doğruluk değerleri ve ortalama değerler Şekil 3.2'de gösterilmiştir.



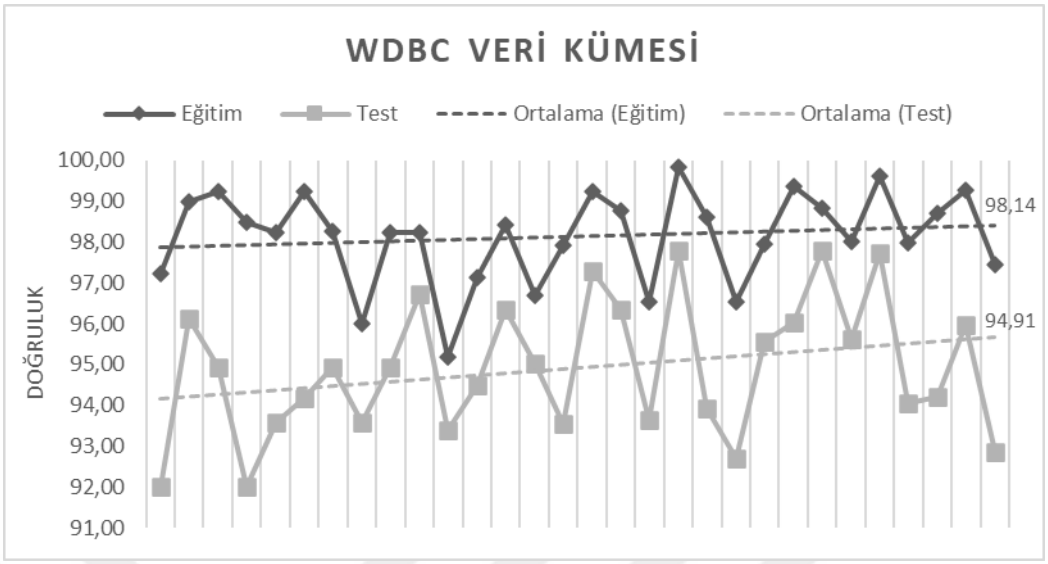
Şekil 3.2 Wine Veri Kümesi için Eğitim ve Test Doğrulukları

Ionosphere Veri Kümesi: Veri kümesi için eğitim doğruluğu %96.4 ±1.84 iken, test doğruluğu %91.47 ± 1.77 olarak gözlemlenmiştir. Veri kümesi için hesaplanan MSE değeri 0.036 ± 0.014 olarak elde edilmiştir. Algoritmanın 30 kez çalıştırılması sonucu elde edilen eğitim doğruluklarına ait standart sapma değeri 1.84 iken, test doğruluklarına ait standart sapma değeri 1.77 şeklindedir. Bu veri kümesine ait doğruluk değerleri Wine veri kümesinde olduğu gibi her çalıştırma sonucunda küçük bir aralıkta değişmektedir. Bu fark test doğruluğu için daha azdır. Bu veri kümesi 34 girdi niteliğinden ve 2 sınıftan oluşmaktadır. Veri kümesine ait örnek sayısının da 351 olduğu göz önünde bulundurulduğunda yöntemin kolay öğrenebildiği ve verileri tahminleyebildiği görülmektedir. Bu sayede Ionosphere veri kümesine ait doğruluk değerleri, diğer veri kümeleri ile karşılaştırıldığında, yöntemin bu veri kümesi üzerinde iyi sınıflandırma performansı elde ettiği söylenebilir. Ionosphere veri kümesine ait doğruluk değerleri ve bu doğruluk değerleri baz alınarak elde edilmiş ortalama eğitim ve test doğruluk değerleri Şekil 3.3'te verilmektedir.



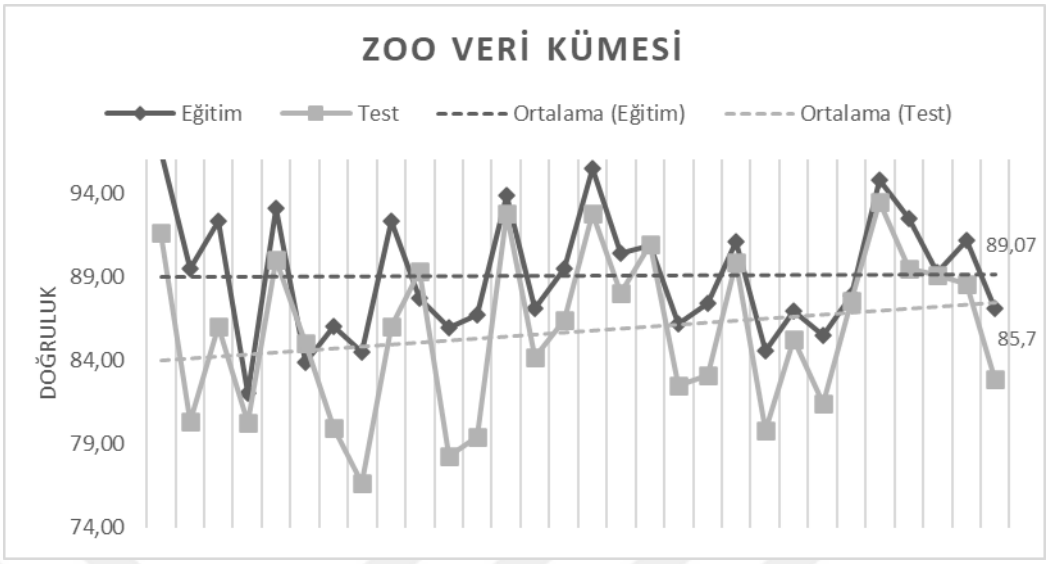
Şekil 3.3 Ionosphere Veri Kümesi için Eğitim ve Test Doğrulukları

WDBC Veri Kümesi: Veri kümesi için Eğitim doğruluğu %98.14 ± 1.13 olarak elde edilmiştir. Test doğruluğu ise 30 kez çalıştırma sonucunda %94.91 ± 1.66'dır. Bu çalışmada kullanılan WDBC veri kümesi için MSE değeri ise 0.036 ± 0.012 olarak gözlemlenmiştir. Veriler eğitim ve test verilerine parçalandığında toplam 569 örneği bulunan veri kümesinin 170 adet örneği test işlemi için ayrılmıştır. Test verilerinin de ortalama %94.91'i doğru sınıflandırılmıştır. Bu durumda SSA ile eğitilmiş YSA modeli kullanarak WDBC veri kümesine ait 170 örneğin 161 tanesi doğru sınıflandırılmıştır. Yöntemde kullanılan veri kümeleri içerisinde en başarılı sınıflandırma performansı Wine veri kümesinden sonra bu veri kümesi için elde edilmiştir. WDBC veri kümesine ait doğruluk değerleri Şekil 3.4'te gösterilmiştir.



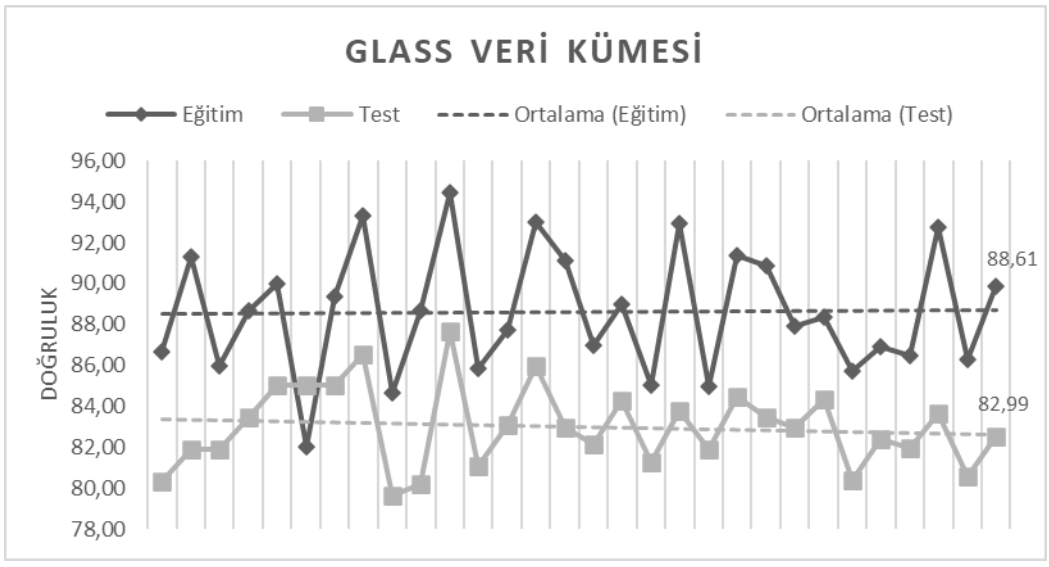
Şekil 3.4 WDBC Veri Kümesi için Eğitim ve Test Doğrulukları

Zoo Veri Kümesi: Veri kümesi için Eğitim doğruluğu 30 kez çalıştırma sonucunda %89.07 ± 3.7 olarak elde edilmiştir. Test doğruluğu %85.7 ± 4.78 olarak gözlemlenmiştir. Bu çalışmada kullanılan Zoo veri kümesi için test doğruluğunun aralığı diğer veri kümelerine oranla daha geniştir. Eğitim doğruluğuna ait standart sapma değerleri incelendiğinde de eğitim doğruluğu minimum ve maksimum doğruluk değerlerinin aralığı diğer veri kümelerine oranla daha geniş olduğu gözlemlenmiştir. Zoo veri kümesi diğer veri kümelerine göre eğitim ve test doğruluk değerlerinin en düşük olduğu veri kümelerinden bir tanesidir. Veri kümesine ait 17 girdi niteliği bulunurken toplam sınıf sayısı ise 7'dir. Kullanılan diğer veri kümeleri ile karşılaştırıldığında Zoo veri kümesine ait sınıf sayısı en fazla olan iki veri kümesinden bir tanesidir. Çıktı sayısının diğer veri kümelerinden daha fazla olmasının yanı sıra, veri kümesinde bulunan örnek sayısı da diğer veri kümelerine göre daha azdır. Veri kümesinin eğitim ve test şeklinde bölünmesi durumu göz önünde bulundurulduğunda ağa eğitim için sunulan toplam örnek sayısı 71'dir. Ağın test işlemi için toplam 30 örnek kullanılmıştır. Bu durum ağın sınıflandırma başarısını etkileyebilmektedir. Bu veri kümesine ait minimum, maksimum ve ortalama değerler dahil olmak üzere, yöntemin 30 kez çalıştırılması sonucunda elde edilen doğruluklar Şekil 3.5'te verilmiştir.



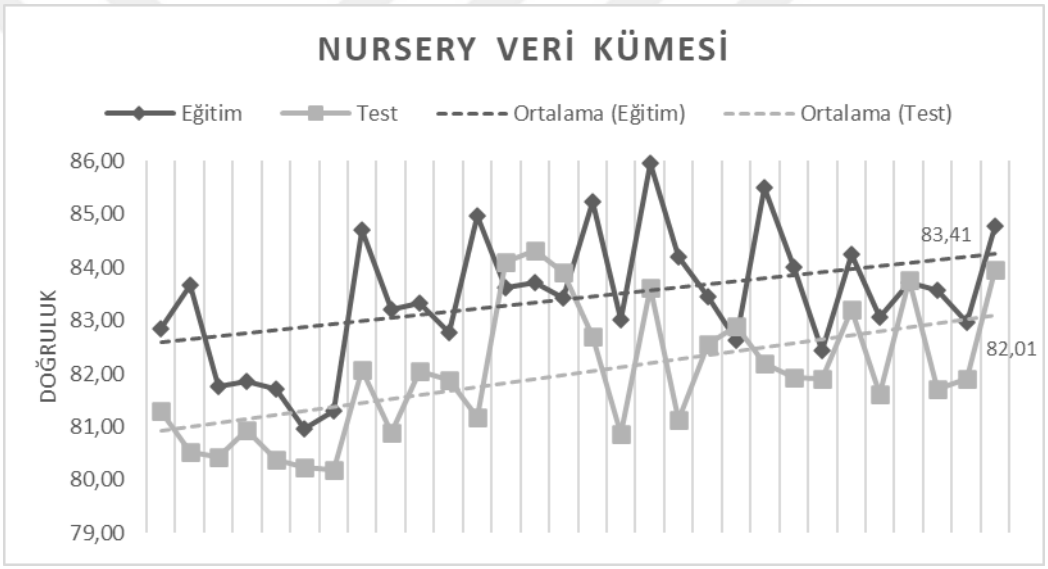
Şekil 3.5 Zoo Veri Kümesi için Eğitim ve Test Doğrulukları

Glass Veri Kümesi: Veri kümesi için Eğitim doğruluğu 30 kez çalıştırma sonucunda %88.61 ± 3.05 olarak gözlemlenirken; test doğruluğu %82.99 ± 3.96 olarak elde edilmiştir. Glass Veri kümesine ait MSE değeri ise 0.246 ± 0.076'dir. Toplam 214 örnekten oluşan bu veri kümesinin 150 örneği ile ağ eğitimi yapılırken verinin %30'una karşılık gelen 64 örnek ile ağın eğitimi test edilmiştir. Veri kümesine ait doğruluklar Şekil 3.6'da gösterilmiştir.



Şekil 3.6 Glass Veri Kümesi için Eğitim ve Test Doğrulukları

Nursery Veri Kümesi: Veri kümesi için Eğitim doğruluğu %83.41± 1.22 olarak gözlemlenirken; test doğruluğu %82.01 ± 1.25 olarak elde edilmiştir. Nursery veri kümesi için eğitim ve test sınıflandırma başarılarının hatasını ifade eden MSE değeri ise 0.235 ± 0.02 olarak gözlemlenmiştir. Bu veri kümesi çalışmada kullanılan diğer veri kümelerine oranla en fazla örnek sayısına sahip veri kümesidir. Toplam 12960 örnekten oluşan veri kümesinde toplam 8 girdi niteliği bulunurken, 5 farklı çıktı sınıfı bulunmaktadır. Veri kümesinde toplam verinin 9072 tanesi eğitim verisi olarak ağa sunulurken, 3888 örnek test verisi olarak kullanılmıştır. Veri kümesine ait doğruluklar grafik olarak Şekil 3.7'de sunulmuştur.

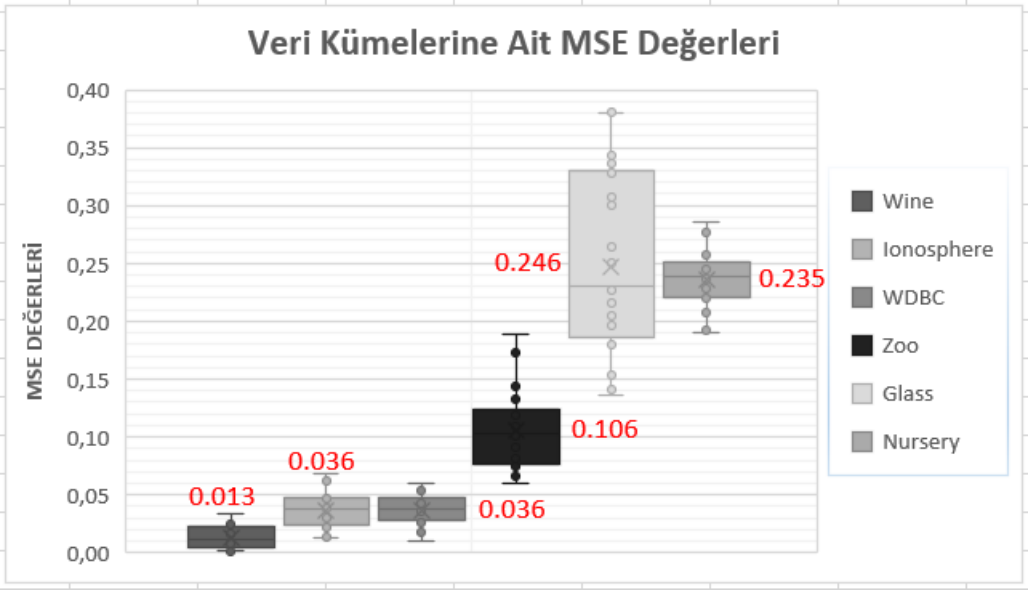


Şekil 3.7 Nursery Kümesi için Eğitim ve Test Doğrulukları

SSA ile eğitilmiş YSA modelinin UCI makine öğrenme deposundan alınan veri kümeleri üzerinde 30 kez çalıştırma sonucunda elde edilmiş MSE değerlerini içeren kutu grafiği Şekil 3.8'de verilmiştir.

Doğruluk değerleri gözlemlendiğinde en iyi sınıflandırma başarısı elde edilen Wine veri kümesi için MSE değerleri de en düşük değerler olarak elde edilmiştir. MSE veri kümelerinin yöntemde uygulanması sonucunda elde edilen hatayı ifade ettiği göz önünde bulundurulduğunda, doğruluk değerleri ile MSE değerlerinin ters orantılı bir şekilde değiştiği görülmektedir.

Veri kümelerine ait standart sapma değerleri incelendiğinde Wine, WDBC, Ionosphere ve Nursery veri kümeleri için MSE değerleri küçük bir aralıkta değişirken, Zoo ve Glass veri kümeleri için MSE değerleri diğer veri kümelerine göre daha geniş bir aralıkta değişmektedir. Bu iki veri kümesi, diğer veri kümeleri ile karşılaştırıldığında en fazla sınıf sayısına sahip iki veri kümesidir. MSE değerlerinin de bu veri kümelerinde, diğer veri kümelerinden daha yüksek standart sapmaya sahip olduğu gözlemlenmiştir. Nursery veri kümesine ait MSE değeri ortalaması diğer veri kümelerine göre daha yüksek olmasına rağmen bu veri kümesine ait standart sapma Zoo ve Glass veri kümelerinden daha düşüktür. Bu yüzden MSE değeri daha küçük bir aralıkta değişkenlik göstermektedir. MSE değerlerine ait standart sapmalar küçükten büyüğe sıralandığında; Wine, WDBC, Ionosphere, Nursery, Zoo ve Glass şeklindedir. Şekil 3.8'de MSE değerlerinin her bir veri kümesi için ortalama değerleri de ayrıca belirtilmiştir.



Şekil 3.8 Her Bir Veri Kümesine Ait MSE Değerleri

Tablo 3.8'de UCI makine öğrenme deposundan alınan tüm veri kümeleri için eğitim ve test verilerinin SSA ile eğitilmiş YSA modelinin deney tasarımı parametreleri kullanılarak elde edilen en iyi, en kötü ve ortalama sınıflandırma doğrulukları, MSE değerleri ve aynı zamanda bu değerlere ait standart sapmalar belirtilmiştir.

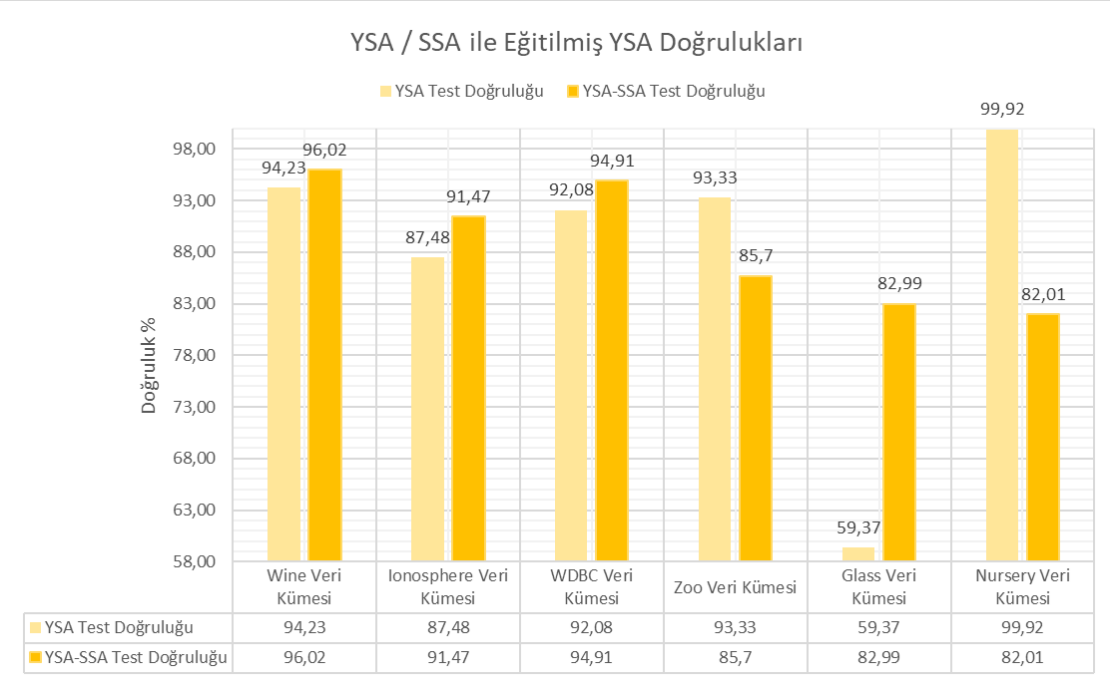
Tablo 3.8 Her Bir Veri Kümesinin Ortalama, Minimum, Maksimum MSE, Standart Sapma ve Doğruluk Değerleri

Veri Kümesi		**MSE Değeri**	**Eğitim Doğruluğu**	**Test Doğruluğu**
Wine Veri Kümesi	**Minimum**	0.001	95.42	91.24
	Ortalama	0.013	98.08	96.02
	Maksimum	0.034	100	99.8
	STD	0.009	1.33	2.46
Ionosphere Veri Kümesi	**Minimum**	0.012	93.13	87.86
	Ortalama	0.036	96.41	91.47
	Maksimum	0.068	99.10	94.42
	STD	0.014	1.84	1.77
WDBC Veri Kümesi	**Minimum**	0.011	95.17	92
	Ortalama	0.037	98.14	94.91
	Maksimum	0.060	99.83	97.8
	STD	0.012	1.13	1.66
Zoo Veri Kümesi	**Minimum**	0.060	82.01	76.67
	Ortalama	0.106	89.07	85.7
	Maksimum	0.188	96.55	93.49
	STD	0.032	3.7	4.78
Glass Veri Kümesi	**Minimum**	0.136	82	79.62
	Ortalama	0.247	88.61	82.99
	Maksimum	0.380	94.45	87.69
	STD	0.076	3.05	3.96
Nursery Veri Kümesi	**Minimum**	0.191	80.97	80.19
	Ortalama	0.235	83.41	82.01
	Maksimum	0.286	85.95	84.32
	STD	0.02	1.22	1.25

Tablo 3.8 göz önünde bulundurulduğunda SSA kullanılarak eğitilmiş YSA modelinin her bir veri kümesine uygun deney tasarım parametreleri ile 30'ar kez çalıştırıldığında elde edilen değerler arasında doğruluk değerleri en yüksek veri kümesi Wine veri kümesi olarak gözlemlenmektedir. Yöntemin veri kümelerine ait ortalama test sınıflandırma başarısı sırasıyla; Wine, WDBC, Ionosphere, Zoo, Glass ve Nursery şeklindedir.

Veri kümelerine ait doğruluk değerlerinin anlamlandırılabilmesi için, yalnızca YSA yöntemi kullanılarak aynı parametreler ile YSA kodlanmış ve veri kümelerine ait doğruluk değerleri gözlemlenmiştir. SSA ile eğitilmiş YSA modelinin sınıflandırma başarısı üzerindeki etkisini gözlemlemek amacıyla her bir veri kümesi için deney tasarımında elde edilen en iyi YSA parametre düzeyleri kullanılarak YSA modeli çalıştırılmıştır. YSA yönteminde eğitim fonksiyonu olarak Levenberg-Marquardt algoritması ve ileri beslemeli geri yayılım YSA modeli kullanılmıştır.

İki yöntemin çalıştırılması ve birbiri ile karşılaştırılmasına ait test doğruluk değerlerini içeren histogram grafiği Şekil 3.9'da verilmiştir.



Şekil 3.9 UCI Veri Ambarından Alınan 6 Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması Histogram Grafiği

YSA'ya ait sınıflandırma doğrulukları elde edilip SSA ile eğitilmiş YSA modeli ile karşılaştırılmıştır. Bu karşılaştırmanın test doğruluk değerleri Tablo 3.9'da verildiği gibidir. Değerler UCI makine öğrenme deposundan alınan 6 veri kümesi için her iki yöntemin 30 kez çalıştırılması sonucu elde edilen ortalama test doğruluğundan oluşmaktadır. Tablodaki koyu işaretli değerler veri kümeleri için hangi yöntemin sınıflandırma başarısının daha iyi olduğunu ifade etmektedir.

Tablo 3.9 Her Bir Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması

Veri Kümesi	**YSA Ortalama Test Doğruluğu**	**YSA+SSA Ortalama Test Doğruluğu**
Wine	%94.23	**%96.02**
Ionosphere	%87.48	**%91.47**
WDBC	%92.08	**%94.91**
Zoo	**%93.33**	%85.7
Glass	%59.37	**%82.99**
Nursery	**%99.92**	%82.01

Tablo 3.9 incelendiğinde SSA ile eğitilmiş YSA modelinin Wine, Ionosphere, WDBC ve Glass veri kümeleri için daha iyi sınıflandırma performansı elde ettiği görülmektedir. 6 veri kümesinden 4 tanesi üzerinde YSA başarısı SSA ile ağ eğitilerek yükseltilmiş ve bu yöntem kullanılarak ağın veri kümesi üzerindeki sınıflandırma başarısı arttırılmıştır. Ancak, Zoo ve Nursery veri kümeleri için ağı SSA ile eğitmek sınıflandırma başarısı üzerinde artışa sebep olmamaktadır. Bu iki veri kümesi için YSA daha yüksek başarı elde etmiştir.

UCI makine öğrenme deposundan alınan ve bu çalışmada kullanılan 6 veri kümesinin sınıflandırma başarısını daha doğru bir şekilde yorumlayabilmek için daha önceki çalışmaları göz önünde bulundurmak gerekmektedir. Yöntemin sınıflandırma performansının daha iyi yorumlanabilmesi, bu çalışmada kullanılan veri kümelerinin önceki çalışmalarının sınıflandırma performansının incelenmesi ile mümkündür.

Literatürde Yapay sinir ağı eğitimi çalışmaları oldukça yaygındır. SSA ile eğitilmiş YSA modelinin sınıflandırma başarısı, üç farklı metasezgisel algoritma kullanılarak eğitilmiş YSA modelinin sınıflandırma başarısı ile karşılaştırılmıştır. UCI makine öğrenme deposundan alınan 6 veri kümesi için metasezgisel algoritmalar kullanılarak eğitilen YSA modellerinin, SSA ile eğitilen YSA modeli ile karşılaştırılmasına ait doğruluk değerleri Tablo 3.10'da gösterilmiştir. Wine, Ionosphere ve WDBC veri kümelerine ait karşılaştırma doğruluk değerleri 2019 yılında Faris ve arkadaşları tarafından [123] yapılmış çalışmadan elde edilirken, Glass, Zoo veri kümelerine ait karşılaştırma doğruluk değerleri McLean'nin 2019 yılında yapmış olduğu yüksek lisans tez çalışmasından [124] , Shokouhifa ve Sabet'in 2010 yılında yaptığı çalışmadan [125] ve Hassanin ve arkadaşlarının 2016 yılındaki çalışmasından elde edilmiştir [126]. Tablo 3.10 genetik algoritma (GA), parçacık sürü optimizasyonu algoritması (PSO) ve yapay arı kolonisi algoritması (ABC) kullanılarak eğitilmiş YSA'nın sınıflandırma doğruluk değerlerinin sincap arama algoritması (SSA) ile eğitilmiş YSA'nın sınıflandırma doğruluk değerlerini içermektedir. Tabloda koyu işaretli gösterilen değerler en iyi sınıflandırma performansının elde edildiği YSA eğitim yöntemleridir.

Tablo 3.10 Metasezgisel Algoritmalar ile YSA Eğitimi Yapılan Yöntemlerin Sınıflandırma Performansları Karşılaştırma Tablosu

Veri Kümesi	**Doğruluk %**	**GA**	**PSO**	**ABC**	**SSA**
Wine	Ortalama ± STD	89.39 ± 4.54	75.96 ± 5.23	75.35 ± 7.00	**96.2** ± **2.46**
	Maksimum	[95.46]	[84.85]	[89.39]	**[99.8]**
Ionosphere	Ortalama ± STD	83.06 ± 2.77	77.42 ± 3.86	80.06 ± 2.93	**91.47 ± 1.77**
	Maksimum	[87.5]	[85.0]	[85.83]	**[94.42]**
WDBC	Ortalama ± STD	**95.96 ± 1.1**	93.06 ± 1.52	94.19 ± 1.26	94.91 ± 1.66
	Maksimum	**[98.45]**	[95.88]	[97.42]	[97.8]
Zoo	Ortalama ± STD	**93.93**	90.75 ± 0.06	91.62 ± 0.06	85.7 ± 4.78
	Maksimum	-	-	-	[93.49]
Glass	Ortalama ± STD	53.5 ± 0.28	64.98 ± 0.06	66.16 ± 0.06	**82.99** ± **3.96**
	Maksimum	-	-	-	**[87.69]**
	Maksimum				

Tabloda yer alan doğruluk değerleri veri kümelerinin test verileri için elde edilmiş doğruluk değerlerini ifade etmektedir. Nursery veri kümesine ait GA, PSO ve ABC metasezgisel algoritmaları kullanılarak YSA eğitimi çalışması sonuçları elde edilememiştir. Nursery veri kümesi için elde edilen sınıflandırma başarısı %82.01 ± 1.25'dir. Tablo 3.10 incelendiğinde geliştirilen yöntemin Wine, Ionosphere ve Glass veri kümeleri için karşılaştırılan metasezgisel algoritmalara göre daha yüksek sınıflandırma başarısı elde ettiği gözlemlenmektedir. Zoo ve WDBC veri kümesine ait SSA ile eğitilmiş YSA modelinin sınıflandırma performansı diğer metasezgisel algoritmalara oldukça yakındır. Önerilen yöntem, Glass veri kümesi için diğer metasezgisel algoritmalar kullanılarak yapılan YSA eğitimlerinin sınıflandırma performansına kıyasla oldukça yüksektir.

Tablo 3.9'da SSA kullanılarak eğitimi yapılan YSA modelinin ortalama test doğrulukları göz önünde bulundurulduğunda 6 veri kümesi içerisinden 4 tanesi için YSA'nın sınıflandırma performansını iyileştirdiği gözlemlenmiştir. Tablo 3.10'da bulunan diğer üç metasezgisel algoritmanın veri kümeleri için sınıflandırma doğrulukları incelendiğinde ise yine 6 veri kümesinden 4 tanesi için sınıflandırma başarısının iyileştiğini söylemek mümkündür.

Sincap Arama Algoritması kullanılarak eğitilen YSA'nın, UCI makine öğrenme deposundan alınan ve bu çalışmada kullanılan 6 veri kümesinde 30 kez çalıştırılması sonucu elde edilen değerler ile diğer yöntemler arasında anlamlı bir farklılık olup olmadığını test etmek için istatistiksel analiz gerçekleştirilmiştir. İstatistiksel analizler gerçekleştirilirken test doğruluk değerleri esas alınmıştır.

Merkezi limit teoremindeki örneklem örnek sayısı 30 veya üzerinde olduğunda normal dağılım söz konusudur varsayımı dikkate alınarak, gruplar bağımlı olduğu, önerilen yöntem ve yalnızca YSA yöntemi 30'ar kere çalıştırıldığı için Eşleştirilmiş-t testi istatistiksel analiz için kullanılmıştır. UCI makine öğrenme deposundan alınan 6 veri kümesi için metasezgisel algoritmalar kullanılarak eğitilen YSA modellerinin, SSA ile eğitilen YSA modeli ile karşılaştırılmasına ait sonuçların istatistiksel analizinde, tek örneklem-t testi kullanılmıştır. T-testi, iki örneklem grubunun ortalamaları arasında istatistiksel olarak anlamlı bir fark olup olmadığını ortaya koymaktadır.

Bu tez çalışmasında istatistiksel analiz işlemlerinin tamamı IBM SPSS yazılım platformu aracılığıyla gerçekleştirilmiştir.

Hipotez olarak, SSA kullanılarak eğitilen YSA'nın test doğruluk değerlerinin, daha önceki çalışmalarda GA, ABC ve PSO algoritmaları kullanılarak eğitimi yapılan YSA çalışmalarında elde edilen doğruluk değerinden ve aynı parametrelere sahip YSA yönteminden α=0.05 anlamlılık düzeyinde farklı olup olmadığı test edilmiştir.

T testi için kurulan hipotezler aşağıda belirtildiği gibidir;

H0 Hipotezi: SSA kullanılarak eğitilen YSA ve literatürdeki diğer yöntemler arasında istatistiksel olarak anlamlı bir fark yoktur.

H1 Hipotezi: SSA kullanılarak eğitilen YSA ve literatürdeki diğer yöntemler arasında istatistiksel olarak anlamlı bir fark vardır.

Tablo 3.11 GA, ABC ve PSO algoritmaları kullanılarak eğitilen YSA modellerinin, SSA ile eğitilen YSA modeli ile karşılaştırılmasına ait tek örneklem-t testi p değerlerini göstermektedir. Tabloda koyu işaretli değerler karşılaştırılan algoritmaların UCI makine öğrenme deposundan alınan 5 veri kümesi için istatistiksel olarak anlamlı bir farklılık olduğunu ifade etmektedir (<0.05).

Tablo 3.11 Tek Örneklem-t testi p Değerleri

		YSA-SSA YSA-GA	**YSA-SSA YSA-ABC**	**YSA-SSA YSA-PSO**
Veri Kümeleri				
p-değeri	**Wine**	**0.000**	**0.000**	**0.000**
	Ionosphere	**0.000**	**0.000**	**0.000**
	WDBC	**0.002**	**0.025**	**0.000**
	Zoo	**0.000**	**0.000**	**0.000**
	Glass	**0.000**	**0.000**	**0.000**

Tablo incelendiğinde 5 veri kümesi için %95 güven aralığında p değeri<0.05 olarak elde edilmiştir. Nursery veri kümesine ait test doğruluk değerleri ile ilgili sonuçlara literatürde ulaşılamamıştır. Bu yüzden yapılan t testine Nursery veri kümesi dahil edilmemiştir. Bu durum sonucunda H0 hipotezi (SSA kullanılarak eğitilen YSA ve literatürdeki diğer yöntemler arasında istatistiksel olarak anlamlı bir fark yoktur.) reddedilmiş, H1 hipotezi (SSA kullanılarak eğitilen YSA ve literatürdeki diğer yöntemler arasında istatistiksel olarak anlamlı bir fark vardır.) kabul edilmiştir. UCI makine öğrenme deposundan alınan tüm veri kümeleri için önerilen yöntem ile literatürdeki yöntemler arasında anlamlı bir farklılık vardır. Tablo 3.10'da verilen GA, ABC, PSO ve SSA ile eğitilen yapay sinir ağlarının ortalama doğruluk değerleri de dikkate alındığında bu anlamlı farklılıklar şu şekilde özetlenebilir; Wine, Ionosphere ve Glass veri kümelerinde YSA-SSA algoritması ile elde edilen ortalama test doğrulukları, YSA-GA, YSA-ABC ve YSA-PSO algoritmaları ile elde edilen sonuçlardan daha iyidir ve ortalama test doğrulukları arasında YSA-SSA yönünde istatistiksel olarak anlamlı bir farklılık vardır. WDBC veri kümesinde YSA-SSA algoritması ile elde edilen ortalama test doğrulukları YSA-ABC ve YSA-PSO algoritmalarından daha yüksektir ve bu algoritmalar arasında elde edilen ortalama test doğrulukları bakımından YSA-SSA yönünde istatistiksel olarak anlamlı farklılık vardır. Bu veri kümesinde YSA-GA algoritması YSA-SSA algoritmasından daha iyi sonuç vermiştir, dolayısıyla YSA-GA ve YSA-SSA algoritmaları ile elde edilen ortalama doğruluklar arasında anlamlı farklılık vardır ve bu fark YSA-GA algoritması yönündedir. WDBC veri kümesinde YSA-SSA algoritması YSA-GA algoritmasından daha iyi sonuç elde edememiştir. Zoo veri kümesinde ise karşılaştırmada dikkate alınan diğer tüm algoritmalar YSA-SSA algoritmasından daha iyi sonuçlar vermiştir ve ortalama uygunluk değerleri arasında anlamlı farklılıklar vardır.

Tablo 3.12 yalnızca YSA yönteminin 30 kez çalıştırılması sonucu elde edilen test doğruluk değerlerinin SSA ile eğitilen YSA modeli ile karşılaştırılmasına ait Eşleştirilmiş-t testi analiz sonuçlarını göstermektedir. Tabloda koyu işaretli değerler karşılaştırılan algoritmaların UCI makine öğrenme deposundan alınan 6 veri kümesi için istatistiksel olarak anlamlı bir farklılık olduğunu ifade etmektedir (<0.05). Bu durum sonucunda H0 hipotezi (SSA kullanılarak eğitilen YSA ve YSA arasında istatistiksel olarak anlamlı bir fark yoktur.) reddedilmiş, H1 hipotezi (SSA kullanılarak eğitilen YSA ve YSA arasında istatistiksel olarak anlamlı bir fark vardır.) kabul edilmiştir.

Tablo 3.12 Eşleştirilmiş-t testi p Değerleri

		YSA-SSA/YSA
Veri Kümeleri		
p-değeri	**Wine**	**0.000**
	Ionosphere	**0.000**
	WDBC	**0.000**
	Zoo	**0.000**
	Glass	**0.000**
	Nursery	**0.000**

Tablo incelendiğinde 6 veri kümesi için p değeri<0.05 olarak elde edilmiştir. UCI makine öğrenme deposundan alınan tüm veri kümeleri için Sincap arama algoritması kullanılarak eğitilen YSA ile yalnızca YSA çalıştırılması arasında anlamlı bir farklılık vardır. Bu farklılık Wine, Ionosphere, WDBC ve Glass veri kümelerinde YSA-SSA algoritması yönündedir yani YSA-SSA algoritması YSA'dan dahi iyi ortalama uygunluk değerine sahiptir. Zoo ve Nursery veri kümelerinde ise YSA algoritması, YSA-SSA algoritmasından daha iyi uygunluklar vermiştir, istatistiksel olarak anlamlı farklılık YSA algoritması yönündedir.

3.3.2 Gerçek Hayat Verisi Deney Tasarımı

UCI makine öğrenme deposundan alınan veri kümelerine ek olarak çalışmada bir gerçek hayat verisi kullanılmıştır. Oxford Mutluluk veri kümesi deney tasarımı adımları, UCI makine öğrenme deposundan alınan 6 veri kümesinde olduğu gibi MATLAB yazılım dili kullanılarak gerçekleştirilmiştir.

Toplamda 7 parametreden oluşan deney tasarımı 3 düzeyli olarak incelenmiş ve daha önce kullanılan parametre düzeyleri ele alınmıştır. Oxford Mutluluk veri kümesi için tam faktöriyel deney tasarımında kullanılan parametreler içerisinde en iyi sınıflandırma performansı sergileyen parametre düzeyleri Tablo 3.13'de gösterilmiştir.

Tablo 3.13 Oxford Mutluluk Veri Kümesi En İyi Deney Tasarım Parametreleri

Oxford Mutluluk Veri Kümesi			
YSA Parametreleri		**SSA Parametreleri**	
İterasyon Sayısı	1000	Toplam Sincap Sayısı	50
Gizli Katman Sayısı	3	Meşe Palamudu Ağacı Sayısı	15
Nöron Sayısı	15	G_c (Süzülme Katsayısı)	1.9
		P_{dp} (Yırtıcı Hayvan Olasılığı)	0.1

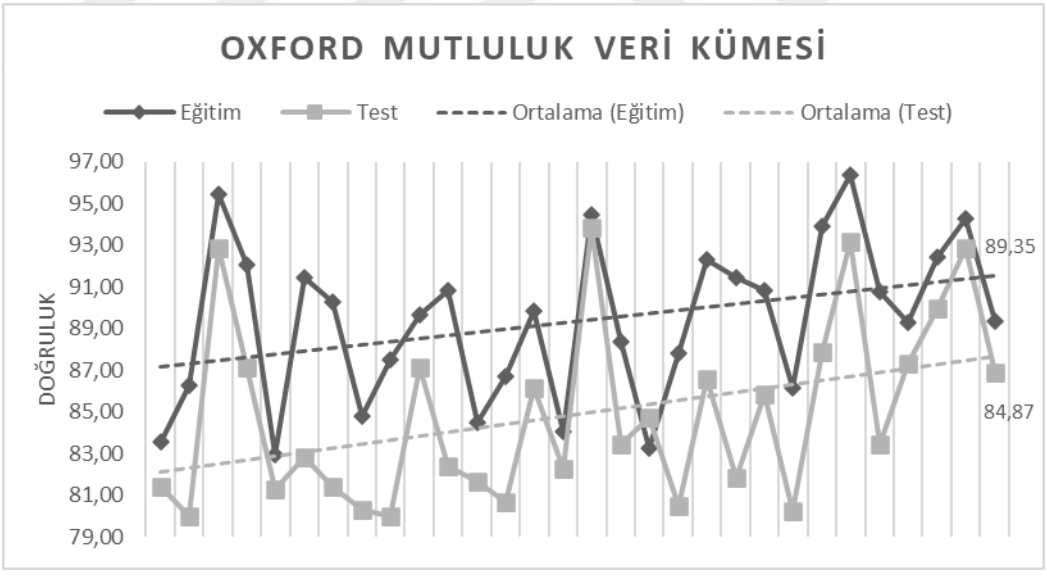
Süzülme katsayısı (G_c) ve yırtıcı hayvan olasılığı (P_{dp}) parametreleri UCI makine öğrenme deposundan alınan veri kümelerinde de olduğu gibi öncü çalışmada belirtilen değerde olacak şekilde elde edilmiştir [1]. YSA'ya ait parametrelerden bir tanesi olan iterasyon sayısı parametresinde diğer veri kümeleri ile aynı düzey elde edilmiştir. Diğer veri kümelerinde gizli katman sayısı parametresine ait en iyi düzey 2 gizli katman sayısı olarak elde edilirken, Oxford Mutluluk Veri Kümesi için gizli katman sayısı 3 olarak gözlemlenmiştir. Aktivasyon fonksiyonu olarak yine sigmoid aktivasyon fonksiyonu kullanılmıştır.

Tablo 3.13'de belirtilen en iyi deney tasarımı parametre düzeyleri kullanılarak Oxford Mutluluk veri kümesi için SSA ile eğitilmiş YSA modeli 30'ar kez çalıştırılmış ve sonuçlar gözlemlenmiştir. Bu veri kümesi için yapılan gözlemlerin her biri birbirinden bağımsızdır. Oxford Mutluluk veri kümesi için SSA ile eğitilmiş YSA modelinin 30'ar kez çalıştırılması sonucunda elde edilen MSE, Eğitim Doğrulukları, Test Doğrulukları gözlemlenmiştir. Ortalama, minimum, maksimum MSE, eğitim doğrulukları, test doğrulukları ve bu değerlerin standart sapmaları Tablo 3.14'te verilmiştir.

Tablo 3.14 Oxford Mutluluk Veri Kümesi MSE ve Doğruluk Değerleri

Gerçek Hayat Veri Kümesi		**MSE Değeri**	**Eğitim Doğruluğu**	**Test Doğruluğu**
Oxford Mutluluk Veri Kümesi	**Minimum**	0.023	82.95	80
	Ortalama	0.107	89.35	84.87
	Maksimum	0,180	96.34	93.86
	STD	0.038	3.787	4.32

Oxford Mutluluk Veri Kümesi için Eğitim doğruluğu 30 kez çalıştırma sonucunda %89.36 ± 3.787 olarak gözlemlenirken, test doğruluğu %84.87 ± 4.32 olarak elde edilmiştir. Bu çalışmada kullanılan gerçek hayat verisi olan Oxford Mutluluk veri kümesi için MSE değeri ise 0.107 ± 0.038'tür. Veri kümesine ait eğitim ve test doğruluk değerleri Şekil 3.10'da gösterildiği gibidir.

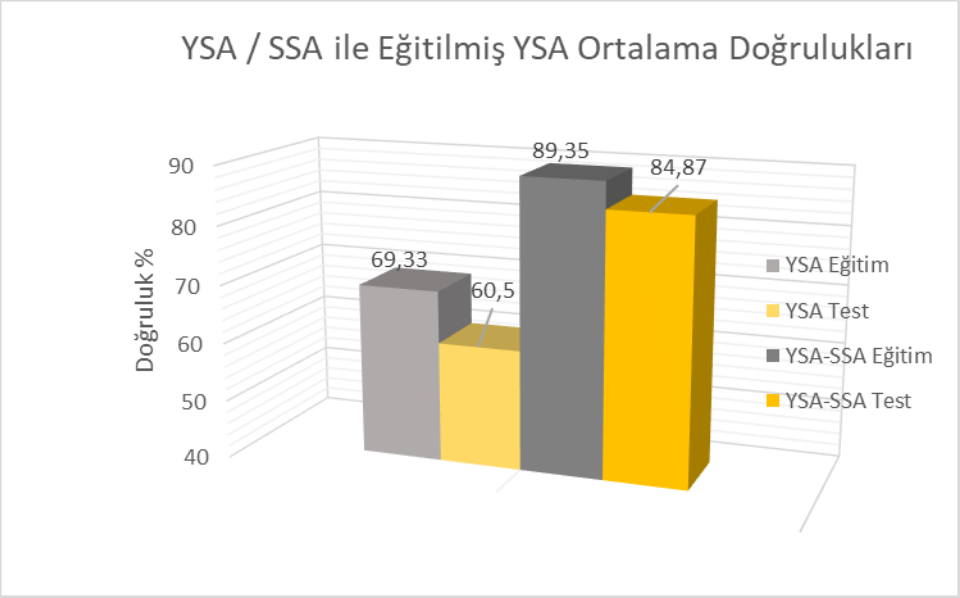


Şekil 3.10 Oxford Mutluluk Veri Kümesi için Eğitim ve Test Doğrulukları

UCI makine öğrenme deposundan alınan veri kümeleri için daha önceki çalışmalar göz önünde bulundurularak yöntemin sınıflandırma başarısı hakkında yorum yapabilmek mümkündür. Ancak, Oxford mutluluk veri kümesi gerçek hayat verisi olduğu için diğer yöntemlerle karşılaştırılması gerekmektedir.

SSA ile eğitilmiş YSA modelinin Oxford Mutluluk veri kümesi üzerinde elde edilen doğruluklarını karşılaştırmak için öncelikle YSA kullanılmıştır. YSA modeli oluşturulurken Oxford Mutluluk verisi için deney tasarımında elde edilen en iyi parametre düzeyleri ele alınmıştır. Veri kümesinin eğitim ve test kümelerine parçalanması işlemi %70 eğitim %30 test verisi olacak şekilde basit bölümleme ile gerçekleştirilmiştir. YSA yönteminde eğitim fonksiyonu olarak Levenberg-Marquardt algoritması ve ileri beslemeli geri yayılım YSA modeli kullanılmıştır.

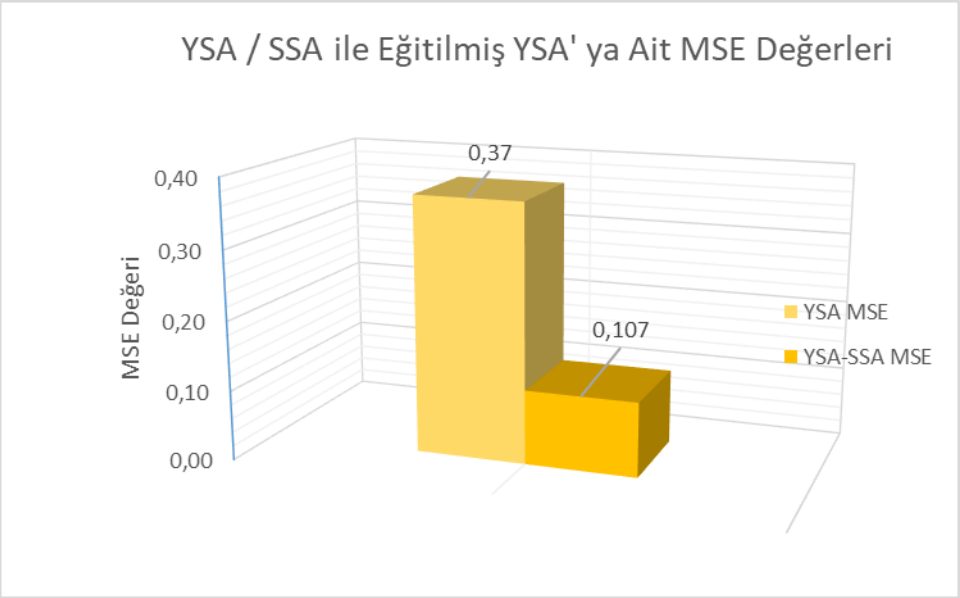
YSA modeli Oxford Mutluluk veri kümesi için 30 kez çalıştırılmış ve sonuçlar gözlemlenmiştir. Bu sonuçlara göre veri kümesine ait YSA modelinde elde edilen ortalama eğitim doğruluğu %69.63 iken, ortalama test doğruluğu %60.5'tir. Bu değerler gözlemlendiğinde YSA modelinden elde edilen doğruluk değerlerinin tamamı SSA ile eğitilmiş YSA modeli doğruluğundan daha düşüktür. Yani YSA ağırlıklarının SSA kullanılarak optimize edilmesi yöntemiyle birlikte Oxford Mutluluk veri kümesi için sınıflandırma başarısı yükselmiştir. Sınıflandırma başarısının yükselmesi ile veri kümesine ait MSE değeri de düşmektedir. İki yöntemin çalıştırılması ve birbiri ile karşılaştırılmasına ait histogram grafikleri Şekil 3.11 ve Şekil 3.12'de verilmiştir. İki yöntemin ortalama eğitim ve ortalama test doğruluk değerleri Şekil 3.11'de gösterilmiştir.



Şekil 3.11 Oxford Mutluluk Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin Doğruluk Karşılaştırması Histogram Grafiği

Şekil 3.11'de verilen histogram grafiği göz önünde bulundurulduğunda Oxford Mutluluk veri kümesi için YSA modelinin SSA-YSA modeline göre daha yüksek MSE elde etmesi beklenmektedir. Şekil 3.12'de verilen MSE değerleri karşılaştırıldığında sonuçların da bu yönde olduğu görülmektedir.

YSA başarısı SSA ile ağ eğitilerek yükseltilmiş ve bu yöntem kullanılarak ağın veri kümesi üzerindeki sınıflandırma başarısı arttırılmıştır. SSA kullanılarak eğitilen ağın ortalama MSE değerinin daha düşük olması, YSA modelinden daha düşük hata elde edildiğini göstermektedir. Bu grafiğe göre, SSA kullanılarak YSA'nın eğitilmesi sonucunda MSE değerlerinde yüksek bir düşüş elde edilmiştir. Her iki yöntemin 30 kez çalıştırılması sonucu elde edilen MSE değerlerini içeren histogram grafiği Şekil 3.12'de verilmiştir.



Şekil 3.12 Oxford Mutluluk Veri Kümesi için YSA ve SSA ile Eğitilmiş YSA Yöntemlerinin MSE Karşılaştırması Histogram Grafiği

Bir gerçek hayat verisi olan Oxford Mutluluk veri kümesinin her iki yöntem için sonuçları incelenmiş olup, SSA ile eğitilmiş YSA modeli, aynı parametrelere sahip YSA modelinden her çalıştırmada daha yüksek sınıflandırma başarısı ve daha düşük MSE değeri elde etmiştir. Bu bilgilerin tamamına dayanarak ağı SSA ile eğitmek gerçek hayat verisi üzerindeki sınıflandırma başarısını iyileştirecek şekilde etki etmiştir denilebilir.

Yöntemin Oxford Mutluluk veri kümesine ait sınıflandırma başarısını daha iyi yorumlayabilmek için WEKA programı içerisinde bulunan 14 sınıflandırıcı yöntemi bu veri kümesi için çalıştırılmış, sonuçların karşılaştırılması yapılmış ve yöntemlerin sınıflandırma başarıları, doğruluk değerleri gözlemlenerek yorumlanmıştır.

Bu sınıflandırma yöntemleri [127];

BayesNET (Bayes Ağları); Bayes Ağı veri modelleme ve durum geçişi ifade etmek için kullanılan güçlü yöntemlerden birisidir. Bu ağların özelliği istatistiksel ağlar olmaları ve düğümler arası geçiş yapan noktaların arama algoritmalarını ve kalite ölçülerini kullanarak seçilmesidir. Bayes ağları yönlü dönüşsüz ağlardır. Her düğüm ayrı bir değişkeni ifade eder. Sınıflandırma problemleri için kullanıma uygundur. Bayes Teoremine dayanır.

Naive Bayes (Naif Bayes Sınıflandırıcısı); İstatistiksel sınıflandırma yöntemlerinden bir tanesidir. Özelliklerin çıktı üzerindeki olasılıklarının incelenerek oluşturulan bir ağ yapısı ile sistemdeki verilerin sınıflandırılmasını sağlar. Bayes Teoremine dayanır.

Logistic (Lojistik Regresyon); Belirli değişkenleri kullanarak bir değişkenin değerini tahminleyecek bir model üretmeyi hedefler. Analiz sonucunda değerlerin gerçekleşme olasılıkları hesaplanır. Bu olasılıklar baz alınarak sınıflandırma yapılır. Doğrusal olmayan verilerle çalışırken elverişli bir sınıflandırma yöntemidir.

IBK (KNN- K En Yakın Komşu Algoritması); KNN algoritması en basit ve en çok kullanılan sınıflandırma algoritmalarından biridir. Sınıflandırılmak istenen yeni bireyin daha önceki bireylerden k tanesine yakınlığına bakılmasıdır. Diğer Denetimli Öğrenme algoritmalarının aksine, eğitim aşamasına sahip değildir.

JRip Sınıflandırıcısı; Algoritma, tamamı pozitif örneklerden oluşan bir kural seti kurar ve gürültülü büyük veri kümeleri üzerinde etkili performans sergiler. Hata Azaltma için Tekrarlayan Artırımlı Budama (RIPPER) kuralını uygular

Bagging (Bootstrap Aggregating); Eğitim veri kümesinin ayrı örneklerini oluşturan ve her örnek için bir sınıflandırıcı oluşturan bir topluluk yöntemidir. Bu çoklu sınıflandırıcıların sonuçları daha sonra birleştirilir. Eğitim verisine bağlı olarak sınıflandırma ve regresyon yapabilir ve varyans azaltılır.

MCC (Multi Class Classifier, Çok Sınıflı Sınıflandırıcı); Çok sınıflı veri kümelerini işlemek için kullanılan bir sınıflandırıcıdır. Doğruluk değerini yükseltmek için hata düzeltme işlemi uygulamaktadır. Verilere ait örnek ağırlıklar oluşturarak hatayı gözlemleyip ağırlıkları güncelleme işlemi ile yeniden örneklendirme temeline dayanır.

Stacking; Weka içerisinde bulunan bu sınıflandırıcı çeşitli sınıflandırma yöntemlerini birleştirerek sınıflandırma veya regresyon işlemi gerçekleştirmektedir.

Decision Table (Karar Tablosu); Karar tabloları girdi değerlerinin farklı kombinasyonları ile oluşan farklı çıktı değerlerini inceleyerek eylemleri ilişkilendirir. Neden ve sonuçları matris şeklinde listelemektedir. Matrisin her satırı özgün bir kombinasyonu ifade eder. Hipotezler oldukça basit ifade edilir ve anlaşılırdır.

J48 (C4.5 Ağacı); ID3 ağacının geliştirilmiş halidir. Ağaç üzerinde entropi hesabı yapılıp ağacın karar noktaları belirlenip dallanmalar sonucu oluşturulmaktadır. ID3 ağacından en büyük farkları normalleştirme kullanması ve budama işlemi gerçekleştirmesidir.

Kstar; Örnek tabanlı bir sınıflandırıcıdır. Bir test örneğinin sınıfı benzerlik gösteren eğitim örneklerinin sınıfına dahil edilir. Entropi tabanlı olan bu sınıflandırıcı mesafe fonksiyonları kullanmaktadır.

LWL (Locally Weighted Learning, Yerel Ağırlıklı Öğrenme); Veriye ait örnek ağırlıkları atayabilmek için örnek tabanlı bir algoritma kullanmaktadır. Sınıflandırma veya doğrusal regresyon işlemleri gerçekleştirebilmektedir.

ICO (Iterative Classifier Optimizer, Yinelemeli Sınıflandırıcı); Yineleme sayısını optimize ederek test verileri üzerindeki doğruluk değerini yükseltmeyi hedeflemektedir.

Random Forest (Rastgele Orman); En sık kullanılan sınıflandırıcılardan bir tanesidir. Hem regresyon hem sınıflandırma problemlerine uygulanabilir, kullanımı kolay ve anlaşılırdır. Rastgele ağaçlardan oluşan bir orman inşa eder. Sınıflandırma işlemi esnasında birçok karar ağacı oluşturur ve sınıflandırma doğruluk değerini yükseltmeyi hedeflemektedir. Bireysel oluşan karar ağaçlarının bir araya gelmesiyle oluşur. Her karar ağacı bağlı olduğu veri kümesinden rastgele seçilen birer alt kümeyi ifade etmektedir.

Tablo 3.15 Sınıflandırıcı Yöntemlerin Sınıflandırma Performansları Karşılaştırma Tablosu

Yöntem	**Doğruluk**	**Yöntem**	**Doğruluk**	**Yöntem**	**Doğruluk**
BayesNet	%80.85	Bagging	%81.56	Kstar	%70.92
Naive Bayes	%82.98	MCC	%73.05	LWL	%75.18
Logistic	%78.72	Stacking	%56.03	ICO	%78.72
IBK	%75.18	Decision Table	%63.12	Random Forest	%82.27
JRip	%74.47	J48	%70.20	**YSA-SSA**	**%84.87**

Önerilen yöntem 14 farklı sınıflandırıcı ile karşılaştırılmış ve sonuçlar gözlemlenmiştir. Tablo 3.15'te yer alan doğruluk değerleri test verileri için elde edilmiş doğruluk değerlerini ifade etmektedir. Tabloda koyu işaretli olarak verilmiş doğruluk değeri en iyi sınıflandırma performansının elde edildiği yöntemdir.

WEKA veri madenciliği yazılımı içerisinde bulunan diğer sınıflandırıcı yöntemlerle karşılaştırıldığında, gerçek hayat verisi için Sincap Arama Algoritması ile eğitimi yapılan YSA modelinin diğer yöntemlere kıyasla daha yüksek sınıflandırma performansı elde ettiği gözlemlenmektedir.

Sincap Arama Algoritması kullanılarak eğitilen YSA'nın gerçek hayat verisi olan Oxford Mutluluk veri kümesinde 30 kez çalıştırılması sonucu elde edilen değerler ile diğer yöntemler arasında anlamlı bir farklılık olup olmadığını test etmek için istatistiksel analiz gerçekleştirilmiştir. İstatistiksel analizler gerçekleştirilirken UCI makine öğrenme deposundan alınan veri kümelerinde olduğu gibi test doğruluk değerleri esas alınmıştır.

SSA ile eğitimi yapılan YSA ve yalnızca YSA yöntemi 30'ar kere çalıştırıldığı için Eşleştirilmiş-t testi istatistiksel analiz için kullanılmıştır. WEKA veri madenciliği yazılımda bulunan algoritmalar ile karşılaştırılmasına ait sonuçların analizi için ise, tek örneklem-t testi kullanılmıştır.

Gerçek hayat veri kümesi için H0 ve H1 hipotezleri UCI makine öğrenme deposundan alınan veri kümelerinde kurulan hipotezler ile aynı şekildedir.

Buna göre önerilen yöntemin yalnızca YSA yöntemi ile karşılaştırılmasına ait Eşleştirilmiş-t testi için kurulan hipotezler aşağıda belirtildiği gibidir;

H0 Hipotezi: SSA kullanılarak eğitilen YSA ve yalnızca YSA yöntemi ile eğitilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark yoktur.

H1 Hipotezi: SSA kullanılarak eğitilen YSA ve yalnızca YSA yöntemi ile eğitilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark vardır.

Tablo 3.16 Gerçek hayat veri kümesi için YSA modelinin, SSA ile eğitilen YSA modeli ile karşılaştırılmasına ait Eşleştirilmiş-t testi analiz sonucunu göstermektedir (<0.05).

Tablo 3.16 Oxford Mutluluk Veri Kümesi için Eşleştirilmiş-t testi p Değerleri

	SSA-YSA/YSA
p-değeri	0.000

Oxford Mutluluk veri kümesi için p değeri<0.05 olarak elde edilmiştir. H0 hipotezi reddedilerek H1 hipotezi kabul edilmiştir yani gerçek hayat veri kümesi için Sincap arama algoritması kullanılarak eğitilen YSA ile yalnızca YSA çalıştırılması arasında %95 güven aralığında anlamlı bir farklılık vardır. Bu farklılık Şekil 3.11'den de görülebileceği gibi YSA-SSA yönündedir, YSA-SSA algoritması daha iyi sonuçlar vermiştir.

SSA ile eğitilen YSA'nın ile WEKA paket programı içerisinde bulunan algoritmalar ile karşılaştırılmasına ait Tek örneklem-t testi için kurulan hipotezler aşağıda belirtildiği gibidir;

H0 Hipotezi: SSA kullanılarak eğitilen YSA ve belirtilen WEKA algoritması ile elde edilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark yoktur.

H1 Hipotezi: SSA kullanılarak eğitilen YSA ve belirtilen WEKA algoritması algoritması ile elde edilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark vardır.

Tablo 3.17 SSA ile eğitilen YSA modeli ile WEKA paket programı içerisinde bulunan algoritmaların test doğruluk değerleri sonuçlarının karşılaştırılmasına ait tek örneklem-t testi analiz sonuçlarını (p değerleri) göstermektedir.

Tablo 3.17 Oxford Mutluluk Veri Kümesi için Tek Örneklem-t testi p Değerleri

Yöntem	YSA-SSA/ Bayesnet	YSA-SSA/ Bagging	YSA-SSA/ Kstar
p-değeri	**0.000**	**0.000**	**0.000**
Yöntem	YSA-SSA/ NaiveBayes	YSA-SSA/ MCC	YSA-SSA/ LWL
p-değeri	**0.023**	**0.000**	**0.000**
Yöntem	YSA-SSA/ Logistic	YSA-SSA/ Stacking	YSA-SSA/ ICO
p-değeri	**0.000**	**0.000**	**0.000**
Yöntem	YSA-SSA/ IBK	YSA-SSA/ DT	YSA-SSA/RF
p-değeri	**0.000**	**0.000**	**0.003**
Yöntem	YSA-SSA/ JRIP	YSA-SSA/ J48	
p-değeri	**0.000**	**0.000**	

SSA ile eğitilen YSA modeli ile WEKA paket programı içerisinde bulunan algoritmaların t testi sonuçları göz önünde bulundurulduğunda tüm p değerleri 0.05'ten küçük elde edilmiştir. Bu durum sonucunda H0 hipotezi (SSA kullanılarak eğitilen YSA ve belirtilen WEKA algoritması ile elde edilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark yoktur.) reddedilmiş, H1 hipotezi (SSA kullanılarak eğitilen YSA ve belirtilen tüm WEKA algoritmaları algoritması ile elde edilen ortalama doğruluk değerleri arasında istatistiksel olarak anlamlı bir fark vardır.) kabul edilmiştir. Elde edilen istatistiksel analiz sonuçları ve Tablo 3.15'te verilen ortalama doğruluklar değerlendirildiğinde, YSA-SSA algoritmasının karşılaştırmada dikkate alınan ve WEKA paket programı içerisinde bulunan diğer tüm algoritmalardan istatistiksel olarak anlamlı olarak daha yüksek doğruluklar verdiği gözlemlenmektedir.

4. BÖLÜM

SONUÇ VE DEĞERLENDİRME

4.1 Sonuç ve Değerlendirme

Yapay Sinir Ağları çalışma prensibine göre ağırlıkların güncellenmesi mekanizması sınıflandırma başarısını doğrudan etkilemektedir. Bu yüzden YSA ağırlık güncelleme işlemi iyi bir öğrenme algoritması ile gerçekleştirilmelidir. Bu tez çalışmasında Sincap Arama Algoritması kullanılarak YSA ağırlıklarının güncellenmesi işlemleri gerçekleştirilmiştir. SSA kullanarak, en iyi ağırlıkların bulunması sayesinde YSA'nın sınıflandırma başarısı arttırılmıştır.

Önerilen bu yöntem UCI makine öğrenme deposundan elde edilen 6 farklı veri kümesi ve bir gerçek hayat verisi olan Oxford Mutluluk Veri Kümesi için test edilmiştir. SSA'ya ait parametreler ve YSA parametreleri değişimi söz konusu olduğunda veri kümeleri için nasıl etkiler oluşturduğu deney tasarımı ile gözlemlenmiş ve en iyi parametre düzeyleri belirlenerek her veri kümesi için sınıflandırma başarıları elde edilmiştir.

SSA kullanılarak en iyi ağırlıkların elde edildiği YSA modelinde UCI makine öğrenme deposu veri kümeleri üzerinde test edilip literatürde bu veri kümeleri için kullanılan sınıflandırıcıların performansı ile karşılaştırılmıştır. Gerçek hayat verisi için ise belirlenen deney tasarımı parametre düzeyleri ile yalnızca YSA çalıştırılmış ve ağı SSA ile eğitmenin YSA'nın sınıflandırma performansını iyileştirdiği sonucu elde edilmiştir. Ayrıca gerçek hayat verisi olan Oxford Mutluluk Veri Kümesi WEKA veri madenciliği yazılımında bulunan diğer sınıflandırıcı yöntemler ile karşılaştırılmıştır. Performanslar karşılaştırılırken hata fonksiyonu olarak MSE değerleri ve ağın sınıflandırma başarısını gözlemleyebilmek için doğruluk değerleri kullanılmıştır.

Bu tez çalışmasında Sincap Arama Algoritması kullanılarak Yapay Sinir Ağı eğitimi yapılarak YSA modelinin sınıflandırma performansı arttırılmıştır. Sincap arama algoritması YSA eğitiminde etkili sonuçlar elde etmektedir.

5. KAYNAKÇA

[1] Jain, M., Singh, V., & Rani, A. (2019). **A novel nature-inspired algorithm for optimization: Squirrel search algorithm.** Swarm and evolutionary computation, 44, 148-175.

[2] Şimşek ÖF. **Happiness Revisited: ontological well-being as a theory-based construct of subjective well-being.** J Happiness Stud 2009; 10:505-522.

[3] V. V. N. A. B. Đ. Yev, C. Köse, and S. Bayrak, "**An Artificial Neural Network Approach for Sign Language**," pp. 1–4.

[4] Andrew, R. J. (1991). **Neural and behavioural plasticity: The use of the domestic chick as a model.**

[5] Gilman, R., Holt, R. J., Kinney, E. R., Kowalczyk, R. S., Mishnev, S. I., Napolitano, J., ... & Temnykh, A. B. (1990). **Measurement of tensor analyzing power in electron-deuteron elastic scattering.** Physical review letters, 65(14), 1733.

[6] P. E. Kolattukudy *et al.*, "**Evidence for possible involvement of an elastolytic serine protease in aspergillosis,**" *Infect. Immun.*, vol. 61, no. 6, pp. 2357–2368, 1993.

[7] Charniak, E. (1985). I**ntroduction to artificial intelligence.** Pearson Education India.

[8] Ergezer, H., Dikmen, M., & Özdemir, E. (2003). **Yapay sinir ağları ve tanıma sistemleri.** PiVOLKA, 2(6), 14-17.

[9] Mehra, P., & Wah, B. W. (1992, June). **Adaptive load-balancing strategies for distributed systems.** In Proc. 2nd Int'l Conf. on Systems Integration (pp. 666-675).

[10] Aladag, C. H., Basaran, M. A., Egrioglu, E., Yolcu, U., & Uslu, V. R. (2009). **Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations.** Expert Systems with Applications, 36(3), 4228-4231.

[11] Kohonen, T. (1987). **Adaptive, associative, and self-organizing functions in neural computing.** Applied optics, 26(23), 4910-4918.

[12] Glover, F., & Laguna, M. (1998). **Tabu search. In Handbook of combinatorial optimization** (pp. 2093-2229). Springer, Boston, MA.

[13] Dorigo, M., & Di Caro, G. (1999, July). **Ant colony optimization: a new meta-heuristic.** In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470-1477). IEEE.

[14] Socha, K., & Dorigo, M. (2008). **Ant colony optimization for continuous domains.** European journal of operational research, 185(3), 1155-1173.

[15] Kennedy, J., & Eberhart, R. (1995, November). **Particle swarm optimization.** In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.

[16] Storn, R., & Price, K. (1997). **Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.** Journal of global optimization, 11(4), 341-359.

[17] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). **A new heuristic optimization algorithm: harmony search. simulation,** 76(2), 60-68.

[18] Karaboga, D. (2005). A**n idea based on honey bee swarm for numerical optimization** (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

[19] Yang, X. S. (2008). **Firefly algorithm. Nature-inspired metaheuristic**

algorithms, 20, 79-90.

[20] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "**Filter modeling using gravitational search algorithm,**" *Eng. Appl. Artif. Intell.*, 2011.

[21] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.

[22] Gandomi, A. H., & Alavi, A. H. (2012). **Krill herd: a new bio-inspired optimization algorithm.** Communications in nonlinear science and numerical simulation, 17(12), 4831-4845.

[23] Duman, E., Uysal, M., & Alkaya, A. F. (2012). **Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem.** Information Sciences, 217, 65-77.

[24] Pan, W. T. (2012). **A new fruit fly optimization algorithm: taking the financial distress model as an example.** Knowledge-Based Systems, 26, 69-74.

[25] James, J. Q., & Li, V. O. (2015). **A social spider algorithm for global optimization.** Applied Soft Computing, 30, 614-627.

[26] Yang, X. S. (2010). **Nature-inspired metaheuristic algorithms.** Luniver press.

[27] Alatas, B. (2010). **Chaotic bee colony algorithms for global numerical optimization.** Expert systems with applications, 37(8), 5682-5687.

[28] Yang, X. S. (2010). **A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization** (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.

[29] Goel, S., Sharma, A., & Bedi, P. (2011, December). **Cuckoo Search Clustering Algorithm: A novel strategy of biomimicry.** In 2011 World Congress on Information and Communication Technologies (pp. 916-921). IEEE.

[30] Wu, T. Q., Yao, M., & Yang, J. H. (2016). **Dolphin swarm algorithm.** Frontiers of Information Technology & Electronic Engineering, 17(8), 717-729.

[31] Chen, C. C., Tsai, Y. C., Liu, I. I., Lai, C. C., Yeh, Y. T., Kuo, S. Y., & Chou, Y. H. (2015, October). **A novel metaheuristic: Jaguar algorithm with learning behavior.** In 2015 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1595-1600).

[32] Li, M. D., Zhao, H., Weng, X. W., & Han, T. (2016). **A novel nature-inspired algorithm for optimization: Virus colony search.** Advances in Engineering Software, 92, 65-88.

[33] Mirjalili, S., & Lewis, A. (2016). **The whale optimization algorithm.** Advances in engineering software, 95, 51-67.

[34] Yazdani, M., & Jolai, F. (2016). **Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm.** Journal of computational design and engineering, 3(1), 24-36.

[35] Saremi, S., Mirjalili, S., & Lewis, A. (2017). **Grasshopper optimisation algorithm: theory and application.** Advances in Engineering Software, 105, 30-47.

[36] Askarzadeh, A. (2016). **A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm.** Computers & Structures, 169, 1-12.

[37] Jain, M., Maurya, S., Rani, A., & Singh, V. (2018). **Owl search algorithm: a novel nature-inspired heuristic paradigm for global optimization.** Journal of Intelligent & Fuzzy Systems, 34(3), 1573-1582.

[38] Samie Yousefi, F., Karimian, N., & Ghodousian, A. (2019). **Xerus Optimization Algorithm (XOA): a novel nature-inspired metaheuristic algorithm for solving global optimization problems.** Journal of Algorithms and Computation, 51(2), 111-126.

[39] Zaeimi, M., & Ghoddosian, A. (2020). C**olor harmony algorithm: an art-inspired metaheuristic for mathematical function optimization.** Soft Computing, 1-40.

[40] Battiti R., Tecchiolli G., 1993. **Training Neural Nets with the Reactive Tabu Search**, IEEE Transactions on Neural Network, 6(5): 1185-2000.

[41] Sexton R.S., Dorsey R.E., Johnson J.D., 1999. **Beyond backpropagation: Using simulated annealing for training neural networks**, Journal of Organizational and End User Computing, 11(3).

[42] Mendes R., Cortez P., Rocha M., Neves J., 2002. **Particle swarms for feedforward neural network training**, Proceedings of the 2002 International Joint Conference on Neural Networks, 12-17 May, pp. 1895-1899.

[43] Ilonen J.,Kamarainen J.K., Lampinen J., 2003. **Differential evolution training algorithm for feed-forward neural networks**, pp. 93–105, Neural Processing Letters 17(1), Kluwer Academic Publishers, Netherlands.

[44] Blum C., Socha K., 2005. **Training feed-forward neural networks with ant colony optimization: An application to pattern classification**, pp 1-6, IEEE Fifth International Conference on Hybrid Intelligent Systems, November 6-9, Rio de Janeiro, Brazil.

[45] Zhang R.J., Zhang J., Lok T.M., Lyu M.R., 2007. **A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training**, Applied Mathematics and Computation 185, 1026–1037.

[46] Özbakir L., Baykasoglu A., Kulluk S., Yapıcı H., 2009. **TACO-miner: An ant colony based algorithm for rule extraction from trained neural networks**, Expert Systems with Applications, 36: 12295–12305.

[47] Zamani M., Sadeghian A., 2010. **A variation of partical swarm optimization for training of artificial neural network**, pp. 131-144, 4th International Conference Nn Information Technology, June 3-5, Al-Zaytoonah University, Jordan, 348 pp.

[48] Kulluk S., Özbakır L., Baykasoğlu A., 2011. **Self-adaptive global best harmony search algorithm for training neural networks**, Procedia Computer Science, 3: 282–286.

[49] Zanchettin C., Ludermir T.B., Almeida L.M., 2011. **Hybrid training method for**

MLP: optimization of architecture and training, pp. 1097-1109, IEEE Transactions on Systems, Man, and Cybernetics, DOI: 10.1109/TSMCB.2011.2107035.

[50] Vasquez R. A., 2011. **Training spiking neural models using cuckoo search algorithm**, pp. 679-686, IEEE Congress of Evolutionary Computation, June 5-8, New Orleans, LA, USA.

[51] Mirjalili S., Hashim S.Z.M., Sardroudi R.M., 2012. **Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm**, Applied Mathematics and Computation, 218: 11125–11137.

[52] Kulluk S., 2013. **A novel hybrid algorithm combining hunting search with harmony search algorithm for training neural networks**, Journal of the Operational Research Society, 64(5): 748–761.

[53] Pereira L.A.M., Rodrigues D., Riberio P.B., Papa J.P., Weber S.A.T., 2014. **Social-spider optimization-based artificial neural networks training and ıts applications for parkinson's disease ıdentifıcation**, pp. 14-17, IEEE 27th International Symposium on Co.

[54] Sarangi P.P., Sahu A., Panda M., 2014. **Training a feed-forward neural network using artifcial bee colony with back-propagation algorithm**, pp. 511-519, Intelligent Computing, Networking and Informatics, Springer, New Delhi.

[55] Chen J.F., Do Q.H., Hsieh H.N., 2015. **Training artificial neural networks by a hybrid PSO-CS algorithm**, Algorithms, 8: 292-308.

[56] Jaddi N.S., Abdullah S., Abdul Malek M., 2017. **Master-leader-slave cuckoo search with parameter control for ANN optimization and its real-world application to water quality prediction**, Plos One, 12(1): 1-19.

[57] Nuaimi A., Abdullah R., 2017. **Neural network training using hybrid particlemove artificial bee colony algorithm for pattern classification**, Journal of Information and Communication Technology, 16(2): 314-334.

[58] Dash R., **Performance analysis of a higher order neural network withan**

improved shuffled frog leaping algorithm for currencyexchange rate prediction, Applied Soft Computing, 67: 215–231.

[59] Doğan, T., & Çötok, N. A. (2016). **Oxford mutluluk ölçeği kısa formunun Türkçe uyarlaması: Geçerlik ve güvenirlik çalışması**. Türk Psikolojik Danışma ve Rehberlik Dergisi, 4(36).

[60] Hills, P., & Argyle, M. (2002). **The Oxford Happiness Questionnaire: a compact scale for the measurement of psychological well-being**. Personality and individual differences, 33(7), 1073-1082.

[61] Argyle, M., Martin, M., & Crossland, J. (1989). **Happiness as a function of personality and social encounters. Recent advances in social psychology: An international perspective**, 189-203.

[62] Diener, E. D., Emmons, R. A., Larsen, R. J., & Griffin, S. (1985). **The satisfaction with life scale**. Journal of personality assessment, 49(1), 71-75.

[63] Veenhoven, R. (1989). **National wealth and individual happiness. In Understanding economic behaviour** (pp. 9-32). Springer, Dordrecht.

[64] Brunyé, T. T., Gagnon, S. A., Paczynski, M., Shenhav, A., Mahoney, C. R., & Taylor, H. A. (2013). **Happiness by association: Breadth of free association influences affective states**. Cognition, 127(1), 93-98.

[65] Abdel-Khalek, A. M. (2006). **Measuring happiness with a single-item scale. Social Behavior and Personality**: an international journal, 34(2), 139-150.

[66] Lyubomirsky, S., & Lepper, H. S. (1999). **A measure of subjective happiness: Preliminary reliability and construct validation**. Social indicators research, 46(2), 137-155.

[67] Ura, K., Alkire, S., Zangmo, T., & Wangdi, K. (2012). **A short guide to gross national happiness index**. The Centre for Bhutan Studies.

[68] Lu, L., & Shih, J. B. (1997). **Sources of happiness: A qualitative approach**. The Journal of Social Psychology, 137(2), 181-187.

[69] Furnham, A., & Cheng, H. (1999). **Personality as predictor of mental health and happiness in the East and West**. Personality and Individual Differences, 27(3), 395-403.

[70] Francis, L. J., & Katz, Y. J. (2000). **Internal consistency reliability and validity of the Hebrew translation of the Oxford Happiness Inventory**. Psychological Reports, 87(1), 193-196.

[71] Liaghatdar, M. J., Jafari, E., Abedi, M. R., & Samiee, F. (2008). **Reliability and validity of the Oxford Happiness Inventory among university students in Iran**. The Spanish journal of psychology, 11(1), 310-313.

[72] Lewis, C. A., Francis, L. J., & Ziebertz, H. G. (2002). **The internal consistency reliability and construct validity of the German translation of the Oxford Happiness Inventory**. North American Journal of Psychology.

[73] Jalloh, A. (2014). **Measuring happiness: Examining definitions and instruments. Illuminare: A Student Journal in Recreation**, Parks, and Tourism Studies, 12(1).

[74] Saraç, T. (2004). **Yapay Sinir Ağları, Seminer Projesi**. Gazi Üniversitesi Endüstri Mühendisliği Ana Bilim Dalı.

[75] Simon, H. (1999). **Neural networks: a comprehensive foundation**. Prentice hall.

[76] Koç, M. L., Balas, C. E., & Arslan, A. (2004). **Taş dolgu dalgakıranların yapay sinir ağları ile ön tasarımı**. Teknik Dergi, 15(74).

[77] McCullogh, W. S., & Pitts, W. (1943). **A logical calculus of the ideas immanent in nervous activity**. Bulletin of mathematical biophysics, 5(4), 115-133.

[78] Pirim, A. G. H. (2006). **Yapay zeka**. Journal of Yaşar University, 1(1), 81-93."

[79] Ağyar, Z. (2015). **Yapay sinir ağlarının kullanım alanları ve bir uygulama**. Mühendis ve Makine, 56(662), 22-23.

[80] Bayır, F. (2006). **Yapay sinir ağları ve tahmin modellemesi üzerine bir uygulama**. Yayınlanmamış Yüksek Lisans Tezi, İstanbul, İstanbul Üniversitesi

Sosyal Bilimler Enstitüsü.

[81] Gurney, K. (1997). **An introduction to neural networks**. CRC press.

[82] Çayıroğlu, İ. (2015). **İleri Algoritma Analizi-5 Yapay Sinir Ağları**. Karabük Üniversitesi Mühendislik Fakültesi.

[83] Şen, Z. (2004). **Yapay sinir ağları ilkeleri**. Su Vakfı.

[84] M. Assaad, 2018, **Neural networks and back-propagation explained in a simple way**, "https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e", (Erişim Tarihi: Mart 2020) .

[85] Elmas, Ç. (2003). **Yapay sinir ağları:(kuram, mimari, eğitim, uygulama)**. Seçkin Yayıncılık.

[86] Ataseven, B. (2013). **YAPAY SİNİR AĞLARI İLE ÖNGÖRÜ MODELLEMESİ**. Öneri Dergisi, 10(39), 101-115.

[87] Anderson, D., & McNeill, G. (1992). **Artificial neural networks technology**. Kaman Sciences Corporation, 258(6), 1-83.

[88] Öztemel, E. (2006). **Yapay Sinir Ağlan**, 2. Baskı, İstanbul, Papatya Yayıncılık.

[89] B. Ataseven, "**YapaSinir AğlariİIe ÖngörüModellemesi**," *Öneri Derg.*, vol. 10, no. 39, pp. 101-115–115, 2013.

[90] Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). **Artificial neural networks: A tutorial**. Computer, 29(3), 31-44.

[91] Rosenblatt, F. (1958). **The perceptron: A probabilistic model for information storage and organization in the brain**. Psychoanalytic Review, 65, 386-408.

[92] Freeman, J. A., & Skapura, D. M. (1991). N**eural networks: algorithms, applications, and programming techniques**. Addison Wesley Longman Publishing Co., Inc..

[93] Faghfouri, A. E., & Frish, M. B. (2011, May). **Robust discrimination of human**

footsteps using seismic signals. In Unattended Ground, Sea, and Air Sensor Technologies and Applications XIII (Vol. 8046, p. 80460D). International Society for Optics and Photoni.

[94] Talbi, E.G., 2009. **Metaheuristics From Design to Implementaton**, John Wiley&Sons Inc. Publication, Canada, 593 pp.

[95] Murty, K. G. (2003). **Optimization models for decision making**: Volume. University of Michigan, Ann Arbor.

[96] Alataş, B., Akın, E., & Özer, A. B. (2007). **Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları**. Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi.

[97] Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). **Henry gas solubility optimization: A novel physics-based algorithm**. Future Generation Computer Systems, 101, 646-667.

[98] Çelik, Y. (2013). **Optimizasyon problemlerinde bal arıları evlilik optımızasyonu algoritmasının (marriage in honey bee optimization-MBO) performansının geliştirilmesi** (Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).

[99] Blum, C., & Roli, A. (2003). **Metaheuristics in combinatorial optimization: Overview and conceptual comparison**. ACM computing surveys (CSUR), 35(3), 268-308.

[100] Canayaz, M., & Karcı, A. (2015). **Investigation of cricket behaviours as evolutionary computation for system design optimization problems**. Measurement, 68, 225-235.

[101] Brownlee, J. (2011). **Clever algorithms: nature-inspired programming recipes**. Jason Brownlee.

[102] Hu, H., Zhang, L., Bai, Y., Wang, P., & Tan, X. (2019). **A Hybrid Algorithm Based on Squirrel Search Algorithm and Invasive Weed Optimization for Optimization**. IEEE Access, 7, 105652-105668.

[103] Agrawal, S., Samantaray, L., Panda, R., & Dora, L. (2020). **A New Hybrid Adaptive Cuckoo Search-Squirrel Search Algorithm for Brain MR Image Analysis**. In Hybrid Machine Intelligence for Medical Image Analysis (pp. 85-117). Springer, Singapore.

[104] Luo, W., & Zheng, T. (2019). A**n Improved Squirrel Search Algorithm for Optimization**. Complexity, 2019, 1-31.

[105] Wang, Y., & Du, T. (2019). **An Improved Squirrel Search Algorithm for Global Function Optimization**. Algorithms, 12(4), 80.

[106] Agrawal, S., Panda, R., Kumari, S., Dora, L., & Abraham, A. **A New Hybrid Multifocus Image Fusion Model Using Single Optimum Gabor Filter A New Hybrid Multifocus Image Fusion Model Using Single Optimum Gabor Filter**.

[107] Chaudhry, R., Tapaswi, S., & Kumar, N. (2019). **A green multicast routing algorithm for smart sensor networks in disaster management**. IEEE Transactions on Green Communications and Networking, 3(1), 215-226.

[108] Swain, S. K., Prasad, S., & Krishna, M. V. **Fault Detection using Unsupervised Feature Learning on Big Data Based on Improved Weighted SoftMax Regression with Jaya Optimization**.

[109] Arbogast, B. S. (2007). **A brief history of the New World flying squirrels: phylogeny, biogeography, and conservation genetics**. Journal of Mammalogy, 88(4), 840-849.

[110] Vernes, K. (2001). **Gliding performance of the northern flying squirrel (Glaucomys sabrinus) in mature mixed forest of eastern Canada**. Journal of Mammalogy, 82(4), 1026-1033.

[111] Jackson, S. M. (2012). **Gliding mammals of the world**. Csiro Publishing.

[112] Thomas, R. B., & Weigl, P. D. (1998). **Dynamic foraging behavior in the southern flying squirrel (Glaucomys volans): test of a model**. The American midland naturalist, 140(2), 264-270.

[113] Bahlman, J. W., Swartz, S. M., Riskin, D. K., & Breuer, K. S. (2013). **Glide performance and aerodynamics of non-equilibrium glides in northern flying squirrels (Glaucomys sabrinus)**. Journal of The Royal Society Interface, 10(80), 20120794.

[114] Norberg, U. M. (1985). **Evolution of vertebrate flight: an aerodynamic model for the transition from gliding to active flight**. The American Naturalist, 126(3), 303-327.

[115] Stapp, P., Pekins, P. J., & Mautz, W. W. (1991). **Winter energy expenditure and the distribution of southern flying squirrels**. Canadian Journal of Zoology, 69(10), 2548-2555.

[116] Fister, I., Fister Jr, I., Yang, X. S., & Brest, J. (2013). **A comprehensive review of firefly algorithms**. Swarm and Evolutionary Computation, 13, 34-46.

[117] Wang, X., Wang, W., & Wang, Y. (2013, July). **An adaptive bat algorithm**. In International Conference on Intelligent Computing (pp. 216-223). Springer, Berlin, Heidelberg.

[118] Civicioglu, P., & Besdok, E. (2013). **A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms**. Artificial intelligence review, 39(4), 315-346.

[119] M. Jamil, X.-S. Yang, **A literature survey of benchmark functions for global optimisation problems**, Int. J. Math. Model. Numer. Optim. 4 (2) (2013) 150–194.

[120] Frank, A., & Asuncion, A. (2010). **UCI Machine Learning Repository** [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California. School of information and computer science, 213(11).

[121] Yates, F. (1935). **Complex experiments**. Supplement to the Journal of the Royal Statistical Society, 2(2), 181-247.

[122] Fisher, R. A. (1936). **Design of experiments**. Br Med J, 1(3923), 554-554.

[123] Faris, H., Mirjalili, S., & Aljarah, I. (2019). **Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme**. International Journal of Machine Learning and Cybernetics, 10(10), 2901-2920.

[124] McLean, R. (2019). **Swarm-based Algorithms for Neural Network Training**.

[125] Shokouhifar, M., & Sabet, S. (2010, December). **A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization**. In 3rd international conference on machine vision (ICMV 2010) (pp. 502-506).

[126] Hassanin, M. F., Shoeb, A. M., & Hassanien, A. E. (2016, December). **Grey wolf optimizer-based back-propagation neural network algorithm**. In 2016 12th International Computer Engineering Conference (ICENCO) (pp. 213-218). IEEE.

[127] https://www.cs.waikato.ac.nz/~ml/weka/", (Erişim tarihi: Temmuz 2020)

6. ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı:	Betül KARAKAŞ
Uyruğu:	Türkiye (T.C)
Doğum Tarihi ve Yeri:	13.07.1993 - Aydın
Medeni Durum:	Evli
E-posta:	betulakpinar.09@gmail.com
Yazışma Adresi:	Mevlâna Mah. Papatya Cad. Ediz Bey Apt. Kat:3 No:40/11 Talas-KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Endüstri Mühendisliği	2020
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2016
Lise	Özel Sultan Murat Eğitim Kurumları, Sivas	2011

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2017-2018	ALA Dekor	Bilgi İşlem Destek Personeli
2017-2018	AF/ AC Bilişim Kayseri	Yazılım ve Veri Tabanı Uzmanı

YABANCI DİL

İngilizce

YAYINLAR

1. Kahya Özyirmidokuz, E., Uyar, K., & Karakaş, B., (2019). AN APPLICATION OF MEASURING BRANDS'TURKISH HAPPINESS ON SOCIAL MEDIA . International Management Information Systems Conference IMISC 2019, İstanbul, Turkey

2. Kahya Özyirmidokuz, E., & Karakaş, B., (2018). Dijital Bağımlılık Mücadelede Sosyal Bilişimin Rolü. 5th International Management Informa Systems Conference (IMISC 2018) (pp.90-93). Ankara, Turkey

3. Kahya Özyirmidokuz, E., Uyar, K., Stoica, E. A. , & Karakaş, B., (2018). Exploring Turkish happiness themes on social media . International Conference Employment, Education and Entrepreneurship (EEE 2018) (pp.1-8). Belgrad, Serbia And Montenegro

4. Kahya Özyirmidokuz, E., Uyar, K., Stoica, E. A. , & Akpınar, B., (2018). Exploring Turkish Happiness Themes on Social Media. Raising the Level of Education and Employment for the Purposes of Creating New Business Opportunities (pp.246-255), Londrina: Silver and Smith Publishers.