



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



GELİŞTİRİLMİŞ KAR ERİME
OPTİMİZASYONU ALGORİTMASI
KULLANILARAK YAPAY SİNİR AĞININ
EĞİTİMİ

Feyza Nur ÖZDEMİR

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ekim-2024
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Feyza Nur Özdemir tarafından hazırlanan “Geliştirilmiş Kar Erime Optimizasyonu Algoritması Kullanılarak Yapay Sinir Ağının Eğitimi” adlı tez çalışması 11/10/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Tahir SAĞ

Danışman

Dr. Öğr. Üyesi Ahmet ÖZKİŞ

Üye

Dr. Öğr. Üyesi Murat KARAKOYUN

İmza

.....

.....

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../20.. gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Havvanur UÇBEYİAY
FBE Müdürü

Not: Bu tez kapsamında aşağıda bilgileri verilen makale çalışması yayınlanmıştır.

*“Kar Erime Optimizasyonu Algoritması ile Çok Katmanlı Yapay Sinir Ağının Eğitimi.”
Çukurova Üniversitesi Mühendislik Fakültesi Dergisi 39.2 (2024): 391-406.*

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Feyza Nur ÖZDEMİR

Tarih: 11.10.2024

ÖZET

YÜKSEK LİSANS TEZİ

GELİŞTİRİLMİŞ KAR ERİME OPTİMİZASYONU ALGORİTMASI KULLANILARAK YAPAY SİNİR AĞININ EĞİTİMİ

Feyza Nur ÖZDEMİR

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ahmet ÖZKİŞ

2024, 70 Sayfa

Jüri

Dr. Öğr. Üyesi Ahmet ÖZKİŞ

Doç. Dr. Tahir SAĞ

Dr. Öğr. Üyesi Murat KARAKOYUN

Bu çalışmada, yakın zamanda geliştirilen bir metasezgisel algoritma olan Kar Erime Optimizasyonu (SAO) algoritmasının Levy uçuşu yönteminden faydalanılarak performansı iyileştirilmiş ve LevySAO algoritması önerilmiştir. Önerilen algoritma yapay sinir ağları ile hibrit modellenerek YSA'nın eğitiminde kullanılmıştır. Önerilen algoritmanın Levy uçuşundan faydalanma olasılığını belirleyen LevyProb parametresi 0,1, 0,2, ..., 0,9, 1,0 değerleri için test edilmiş ve en uygun sonucun LevyProb =0,9 olduğu görülmüştür. Önerilen YSA-LevySAO modeli, literatürde iyi bilinen 15 farklı veri seti – aggregation, aniso, balance, ecoli, glass, iris, iris2D, liver, mouse, pathbased, seeds, smiley, varied, vertebral3, wine – üzerinde çalıştırılmış ve dört farklı metrik – sensitivity, specificity, precision ve F1-score – için performans değerleri hesaplanmıştır. Elde edilen sonuçlar literatürde iyi bilinen 9 farklı algoritma – AAA, CS, DEA, GWO, HBA, MPA, RSA, SAO, SCA – ile oluşturulan hibrit YSA modelleri ile karşılaştırılarak ortalama başarı sıralamaları (OBS) hesaplanmıştır. Sonuçlar incelendiğinde, önerilen YSA-LevySAO modelinin dört metriğin tamamında karşılaştırılan modellerden daha üstün bir performansa sahip olarak ilk sırada yer aldığı gözlemlenmiştir. Modellerin performansları galibiyet/mağlubiyet/beraberlik açısından bire bir karşılaştırıldığında önerilen modelin diğer modellerin tamamından daha fazla sayıda galibiyet elde ettiği görülmüştür. YSA-LevySAO modelinin elde ettiği sonuçların istatistiksel olarak anlamlı olup olmadığını belirlemek için Wilcoxon işaretli sıralar testi uygulanmıştır. İstatistik sonuçları, önerilen modelin veri setlerinin birçoğunda %95 güven aralığında anlamlı sonuçlar ürettiğini göstermektedir. Ayrıca modellerin eğitim aşamasındaki hata yakınsama performansları incelendiğinde önerilen modelin birçok veri setinde karşılaştırılan modellerden daha hızlı bir yakınsama performansına sahip olduğu görülmüştür.

Anahtar Kelimeler: Çok Katmanlı Algılayıcı (ÇKA), Kar Erime Optimizasyonu (SAO), Levy Uçuşu, Metasezgisel Algoritma, Sınıflandırma, Yapay Sinir Ağları (YSA)

ABSTRACT

MS THESIS

TRAINING THE ARTIFICIAL NEURAL NETWORK USING THE IMPROVED SNOW ABLATION OPTIMIZER ALGORITHM

Feyza Nur ÖZDEMİR

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF NECMETTİN ERBAKAN UNIVERSITY THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Asst. Prof. Dr. Ahmet ÖZKİŞ

2024, 70 Pages

Jury

Asst. Prof. Dr. Ahmet ÖZKİŞ

Assoc. Prof. Dr. Tahir SAĞ

Asst. Prof. Dr. Murat KARAKOYUN

In this study, the performance of the recently developed metaheuristic algorithm, Snow Ablation Optimizer (SAO), was improved by utilizing the Levy flight method and the LevySAO algorithm was proposed. The proposed algorithm was used in the training of the ANN by hybrid modeling with artificial neural networks. The LevyProb parameter, which determines the possibility of benefiting from the Levy flight, was tested for values of 0,1, 0,2, ..., 0,9, 1,0 and the most appropriate result was found to be LevyProb = 0,9. The proposed ANN-LevySAO model was run on 15 different data sets – aggregation, aniso, balance, ecoli, glass, iris, iris2D, liver, mouse, pathbased, seeds, smiley, varied, vertebral3, wine – well known in the literature, and performance values were calculated for four different metrics – sensitivity, specificity, precision and F1-score –. Elde edilen sonuçlar literatürde iyi bilinen 9 farklı algoritma – AAA, CS, DEA, GWO, HBA, MPA, RSA, SAO, SCA – ile oluşturulan hibrit YSA modelleri ile karşılaştırılarak ortalama başarı sıralamaları (OBS) hesaplanmıştır. The results obtained were compared with hybrid ANN models created with 9 different algorithms well-known in the literature – AAA, CS, DEA, GWO, HBA, MPA, RSA, SAO, SCA – and average success rankings (ASR) were calculated. When the results were examined, it was observed that the proposed ANN-LevySAO model ranked first with a superior performance than the compared models in all four metrics. When the performances of the models were compared one-to-one in terms of win/lost/draw, it was seen that the proposed model achieved more wins than all other models. Wilcoxon’s signed-rank test was applied to determine whether the results obtained from the ANN-LevySAO model were statistically significant. Statistical results show that the proposed model produces significant results within a 95% confidence interval in most of the data sets. In addition, when the error convergence performances of the models during the training phase were examined, it was seen that the proposed model had a faster convergence performance than the models compared in many data sets.

Keywords: Artificial Neural Network (ANN), Classification, Levy Flight, Metaheuristic Algorithm (MA), Multi Layer Perceptron (MLP), Snow Ablation Optimizer (SAO)

ÖNSÖZ

Bu çalışmanın hazırlanması sırasında ve lisansüstü eğitim öğretim hayatım boyunca desteklerini hiçbir zaman esirgemeyen değerli danışman hocam Dr.Öğr.Üyesi Ahmet ÖZKIŞ'a ve her daim yanımda olan, beni cesaretlendiren aileme sonsuz teşekkürlerimi sunarım.

Feyza Nur ÖZDEMİR
KONYA-2024



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	ix
ŞEKİLLER LİSTESİ	xi
ÇİZELGELER LİSTESİ	xii
1. GİRİŞ.....	1
1.1. Optimizasyon	3
1.1.1. Klasik optimizasyon yöntemleri	4
1.1.2. Metasezgisel optimizasyon yöntemleri.....	5
1.2. Metasezgisel Optimizasyon Algoritmaları.....	6
1.2.1. Guguk kuşu arama (CS) algoritması.....	6
1.2.2. Diferansiyel evrim algoritması (DEA)	6
1.2.3. Gri kurt optimizasyonu (GWO) algoritması.....	7
1.2.4. Bal porsuğu algoritması (HBA).....	7
1.2.5. Yapay alg algoritması (AAA).....	7
1.2.6. Deniz yırtıcıları algoritması (MPA).....	8
1.2.7. Sürüngen arama algoritması (RSA).....	9
1.2.8. Sinüs kosinüs algoritması (SCA).....	9
2. KAYNAK ARAŞTIRMASI	10
2.1. YSA'nın Tarihçesi ve YSA-Metasezgisel Algoritmalarla Hibrit Kullanımı Kaynak Araştırması	10
2.2. SAO Algoritması Kaynak Araştırması.....	14
3. MATERYAL VE YÖNTEM.....	16
3.1. Yapay Sinir Ağları	16
3.1.1. Toplam fonksiyonları.....	20
3.1.2. Aktivasyon fonksiyonları.....	20
3.2. SAO.....	24
3.2.1. SAO algoritmasının avantaj ve dezavantajları.....	29
3.3. Deneysel Çalışma.....	29
3.4. Veri Setleri	30
3.5. Performans Metrikleri	31

3.5.1.	Karmaşıklık matrisi.....	31
3.5.2.	Ortalama karesel hata.....	32
4.	ÖNERİLEN YÖNTEM	34
4.1.	Levy Uçuşu	34
4.2.	LevySAO.....	35
4.3.	YSA-LevySAO	38
4.3.1.	YSA-LevySAO modelinde parametre ayarlaması.....	38
5.	ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	45
5.1.	YSA-LevySAO Modeli ve Diğer Modellerin Performans Karşılaştırması	45
5.2.	Yakınsama Grafikleri	51
5.3.	YSA-LevySAO Modeli ve Diğer Modellerin Bire Bir Performans Karşılaştırması (Wilcoxon Testi).....	53
6.	SONUÇLAR VE ÖNERİLER.....	60
6.1.	Sonuçlar.....	60
6.2.	Öneriler	60
7.	KAYNAKLAR.....	61

SİMGELER VE KISALTMALAR

Kısaltmalar

AAA: Yapay Alg Algoritması

ABC: Yapay Arı Kolonisi

ALO: Karınca Aslanı Optimizasyonu

ANN: Yapay Sinir Ağları

CS: Guguk Kuşu Arama

ÇKA: Çok Katmanlı Algılayıcı

DEA: Diferansiyel Evrim Algoritması

FA: Ateşböceği Algoritması

GA: Genetik Algoritma

GOA: Çekirge Optimizasyon Algoritması

GWO: Gri Kurt Optimizasyonu

HBA: Bal Porsuğu Algoritması

MPA: Deniz Yırtıcıları Algoritması

MSE: Ortalama Karesel Hata

PSO: Parçacık Sürü Optimizasyonu

ReLU: Doğrultulmuş Doğrusal Birim

RSA: Sürüngen Arama Algoritması

SAO: Kar Erime Optimizasyonu

SCA: Sinüs Kosinüs Algoritması

TKA: Tek Katmanlı Algılayıcı

TLBO: Öğretim Öğrenme Tabanlı Optimizasyon

UCI: University of California Irvine

WOA: Balina Optimizasyon Algoritması

YSA: Yapay Sinir Ađı

YSA-LevySAO: Yapay Sinir Ađı-Levy Kar Erime Optimizasyonu



ŞEKİLLER LİSTESİ

Şekil 3.1. Biyolojik sinir ağı hücresi (Bulut, 2018).....	17
Şekil 3.2. Bir yapay sinir hücresinin temel bileşenleri.....	18
Şekil 3.3. Çok katmanlı algılayıcı mimarisi	18
Şekil 3.4. Basamak fonksiyonu ve türevi	21
Şekil 3.5. Doğrusal fonksiyon ve türevi	21
Şekil 3.6. Sigmoid fonksiyonu ve türevi	22
Şekil 3.7. Hiperbolik tanjant fonksiyonu ve türevi.....	23
Şekil 3.8. ReLU fonksiyonu ve türevi	23
Şekil 3.9. Karın doğadaki halleri	24
Şekil 3.10. Brownian hareketi (Mörters ve Peres, 2010)	25
Şekil 3.11. İkili popülasyon oluşturma mekanizması.....	28
Şekil 3.12. SAO algoritmasının sözde kodu (Deng ve Liu, 2023)	28
Şekil 3.13. ÇKA mimarisinde ağırlıkların vektör modellenmesi	30
Şekil 3.14. Karmaşıklık matrisi	31
Şekil 4.1. Levy uçuşu (X.-S. Yang ve ark., 2013).....	34
Şekil 4.2. LevySAO algoritmasının sözde kodu	36
Şekil 4.3. LevySAO algoritmasının akış diyagramı.....	37
Şekil 5.1. Algoritmaların oluşturduğu modellerin yakınsama grafikleri.....	51

ÇİZELGELER LİSTESİ

Çizelge 3.1. Biyolojik ve yapay sinir hücresi elemanları.....	17
Çizelge 3.2. Toplam fonksiyonları.....	20
Çizelge 3.3. Çalıştırma kriterleri.....	29
Çizelge 3.4. Veri setleri ve özellikleri.....	31
Çizelge 4.1. YSA-LevySAO modeli varyantlarının sensitivity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	41
Çizelge 4.2. YSA-LevySAO modeli varyantlarının specificity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	42
Çizelge 4.3. YSA-LevySAO modeli varyantlarının precision metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	43
Çizelge 4.4. YSA-LevySAO modeli varyantlarının F1-score metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	44
Çizelge 5.1. YSA-LevySAO modeli ve diğer modellerin sensitivity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	47
Çizelge 5.2. YSA-LevySAO modeli ve diğer modellerin specificity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	48
Çizelge 5.3. YSA-LevySAO modeli ve diğer modellerin precision metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	49
Çizelge 5.4. YSA-LevySAO modeli ve diğer modellerin F1-score metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri	50
Çizelge 5.5. YSA-LevySAO modeli ve diğer modellerin sensitivity metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları.....	56
Çizelge 5.6. YSA-LevySAO modeli ve diğer modellerin specificity metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları.....	57
Çizelge 5.7. YSA-LevySAO modeli ve diğer modellerin precision metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları.....	58
Çizelge 5.8. YSA-LevySAO modeli ve diğer modellerin F1-score metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları.....	59

1. GİRİŞ

İnsanlık var olduğundan beri çok çeşitli problemlerle karşı karşıya kalmıştır. Yakın geçmişimize kadar bu problemlerin çözümü için sadece kendi akıl ve zekâsını kullanabilen insanoğlu, dijitalleşme ile adeta çağ atlayarak çoğu problemin çözümünde bilgisayarları kullanmaya başlamıştır. Teknolojinin ilerlemesiyle beraber depolanabilen veri miktarının da çok hızlı bir şekilde artması, Turing (2009) başta olmak üzere bilim insanlarına şu soruyu sordurmuştur: “İnsanın yapabildiği bazı görevler makinelere öğretilir mi?” Bu soru, günümüzdeki yapay zekâ kavramına giden yolu açmıştır. Yapay zekâ, insan ve hayvan gibi biyolojik canlılardaki karar verme, tahmin etme ve çözüm bulma kabiliyetlerinin makinelere öğretilmesidir (Öztemel, 2003). Bu amacı gerçekleştirmek için bilim insanları çeşitli çalışmalar yapmıştır. Rosenblatt (1958) tarafından geliştirilen basit algılayıcı modeli yapay zekâ alanının öncü çalışmalarından biri olmuştur. Rumelhart ve ark. (1986) ise perceptron modelinin doğrusal olmayan problemleri çözmesini sağlayarak yapay sinir ağları (YSA) algoritmasının geliştirilmesine katkı sunmuştur.

YSA, insan beyninin yapabildiği eylemleri yapmayı öğrenebilen ve benzer çıktıları üretebilen bir algoritmadır. Öğrenebilme, öğrendiği bilgilerden yeni kurallar çıkarabilme, keşfedebilme gibi biyolojik sinir ağlarının sahip olduğu özelliklere sahiptir (Öztemel, 2003). YSA algoritması günümüzde görüntü işleme (Ciregan ve ark., 2012; Krizhevsky ve ark., 2012; Liu ve ark., 2023), zaman serisi tahmini (Bas ve ark., 2024; Dalal ve ark., 2023; Egrioglu ve ark., 2022), regresyon analizi (Ergun ve Tayfun, 2020; Jayasimha ve ark., 2022; Selim ve ark., 2023; Tam ve ark., 2022), sınıflandırma (Ghiassi ve Burnley, 2010; Xu ve ark., 2000; Yaman ve ark., 2024) ve kümeleme (Elangasinghe ve ark., 2014; Erilli ve ark., 2011; Nithya ve ark., 2020) gibi pek çok gerçek dünya probleminin çözümünde başarıyla uygulanmaktadır. YSA nöron denilen yapılardan oluşur. YSA'nın matematiksel modellemesinde, her bir nöron, ağırlık olarak adlandırılan sayısal bir değerle çarpılarak toplama fonksiyonuna gider. Sonrasında aktivasyon fonksiyonuna giderek sonuca ulaşılır. YSA'nın yaptığı bu harekete ileri yayılım (forward propagation) adı verilir. Ağırlık değerlerinin yeni değerlerle güncellenmesi hareketine ise geri yayılım (backward propagation) denmektedir. YSA'da geri yayılım sırasında ağırlıkların güncellenmesi için genellikle türev tabanlı teknikler kullanılmaktadır. Ancak bu teknikler bazen yerel minimum noktalara takılabilmektedir. Bu sorunun üstesinden

gelmek için geri yayılım fazı yerine arařtırmacılar metasezgisel algoritmalarından faydalanabilmektedir.

Metasezgisel algoritmalar, doęadaki fiziksel olaylardan ve canlıların besin arama ve hayatta kalma davranıřlarından ilham alınarak bilim insanları tarafından geliřtirilen problem çözüme yöntemleridir. Metasezgisel algoritmalar esnek yapıları sayesinde sayısal optimizasyon (Deng ve Liu, 2023; Karaboga ve Akay, 2009; Uymaz ve ark., 2015), kapasitesiz tesis yerleřimi (Cinar ve Kiran, 2018; Ahmet Özkıř ve Karakoyun, 2023), rüzgâr türbini yerleřimi (M. Aslan ve ark., 2023; Çelik ve ark., 2018), knapsack (Kong ve ark., 2008) ve yapay sinir aęları (YSA)'nın eęitilmesi (Irmak ve ark., 2023; Karaboga ve Akay, 2007; Mirjalili, 2015) gibi birçok gerçek dünya probleminin çözümünde başarı ile uygulanabilmektedir. Günümüze kadar çok sayıda metasezgisel algoritma geliřtirilmiř olsa da No Free Lunch (NFL) (Wolpert ve Macready, 1997) teoremindeki, hiçbir algoritma tüm problem türlerinde en başarılı sonuca ulaşmayı garanti edemez düşüncesinden hareketle halen yeni algoritmalar önerilmektedir. Karın erimesi esnasında meydana gelen fiziksel olaylardan ilham alınarak Deng ve Liu (Deng ve Liu, 2023) tarafından 2023 yılında önerilen Kar Erime Optimizasyonu (Snow Ablaton Optimizer, SAO) algoritması da bu alandaki güncel algoritmalarından biridir.

Bu tez çalışmasında öncelikle, SAO algoritmasının yerel arama kabiliyetini iyileřtirmek için Levy uçuřu (Houssein ve ark., 2020; X.-S. Yang ve ark., 2013) adı verilen bir arama stratejisinden faydalanılarak LevySAO yöntemi geliřtirilmiřtir. Geliřtirilen bu yöntem, YSA ile hibrit kullanılarak YSA-LevySAO hibrit modeli önerilmiřtir. Bu model sayesinde, YSA'nın geri yayılım fazında karşılařabildięi yerel minimumlara takılma sorununun üstesinden gelinmesi amaçlanmıřtır. Önerilen YSA-LevySAO hibrit modeli, literatürde iyi bilinen 15 farklı sınıflandırma veri setinin – aggregation, aniso, balance, ecoli, glass, iris, iris2D, liver, mouse, pathbased, seeds, smiley, varied, vertebral3 ve wine – eęitiminde kullanılmıřtır. Modelin başarısı sensitivity, specificity, precision ve F1-score metrikleri üzerinden hesaplanmıřtır. Önerilen modelin sonuçları, literatürde iyi bilinen 9 farklı metasezgisel algoritma – AAA, CS, DEA, GWO, HBA, MPA, RSA, SAO ve SCA – ile geliřtirilen hibrit YSA modelleri ile karşılařtırılmıřtır. Modellerin, her bir veri seti için başarı sıraları kaydedilmiř ve tüm veri setlerindeki ortalama başarı sırası (OBS) hesaplanmıřtır. Sonuçlar incelendięinde, YSA-LevySAO hibrit modelinin dört metrikte de en iyi OBS deęerini elde ettięi gözlenmiřtir. Önerilen YSA-LevySAO hibrit modeli ile dięer modeller galibiyet/maęlubiyet/beraberlik aęısından birebir karşılařtırıldıęında, önerilen modelin en

iyi performansı elde ettiği ve çoğu veri setinde Wilcoxon işaretli sıralar testine göre %95 güven aralığında istatistiksel olarak anlamlı sonuçlar ürettiği gözlenmiştir.

Tez, temel olarak 6 bölümden oluşmaktadır. Birinci bölümde tez çalışmasının konusuna giriş yapılmıştır. Optimizasyonun tanımı yapılarak klasik ve metasezgisel optimizasyon yöntemlerinden bahsedilmiştir. Tezde performans karşılaştırması için kullanılan metasezgisel algoritmalar tanıtılmıştır.

İkinci bölümde YSA'nın tarihçesi ve çalışma prensibi anlatılmıştır. YSA'nın metasezgisel algoritmalar ile hibrit kullanıldığı çalışmalara yer verilmiştir. SAO algoritmasının literatürde kullanıldığı mevcut çalışmalar incelenmiştir.

Üçüncü bölümde tez çalışmasında kullanılan materyal ve yöntem anlatılmaktadır. Bu bölümde ilk olarak orijinal YSA algoritmasının temel çalışma adımları açıklanmıştır. Ardından SAO algoritması adım adım anlatılarak algoritmanın avantaj ve dezavantajlarından bahsedilmiştir. Daha sonra ise çalışmada kullanılan veri setleri, performans metrikleri ve çalıştırma kriterleri hakkında bilgi verilmiştir.

Dördüncü bölümde tezde önerilen YSA-LevySAO hibrit modelinden bahsedilmiştir. Öncelikle orijinal SAO algoritmasına Levy uçuşu eklenerek önerilen LevySAO algoritması anlatılmıştır. Daha sonra LevySAO algoritması ve YSA'nın hibrit kullanımı ile geliştirilen YSA-LevySAO modeli tanıtılmış ve parametre ayarlaması yapılmıştır.

Beşinci bölümde tezin araştırma sonuçlarına yer verilmiştir. Geliştirilen YSA-LevySAO modeli, literatürdeki iyi bilinen 9 farklı algoritma ile oluşturulan hibrit modellerle sensitivity, specificity, precision ve F1-score metriklerinden elde edilen ortalama, standart sapma ve başarı sırası değerleri üzerinden kıyaslanmıştır. Daha sonra YSA-LevySAO modeli ve diğer modellerin bire bir performans karşılaştırma ve Wilcoxon işaretli sıralar testine göre istatistiksel anlamlılık sonuçları verilmiştir. Son olarak, modellerin eğitim aşamasındaki MSE değerlerine göre veri setlerindeki hata yakınsama grafikleri verilmiştir.

Altıncı bölümde, tez çalışmasında elde edilen sonuçlar genel olarak değerlendirilmiş ve sonraki çalışmalarla ilgili olarak önerilerde bulunulmuştur.

1.1. Optimizasyon

Optimizasyon, bir problemin ya da sürecin olabilecek en iyi duruma getirilmesi anlamına gelmektedir (El-Kenawy ve ark., 2024). Örneğin bir şirketin kârını maksimize etmek için kârı etkileyen parametreler optimize edilebilir. Problemden ya da süreçte

istenene göre optimize etme işlemi ve optimum kavramı değişebilmektedir. Trafikte yer alan bir yayanın karşıdan karşıya geçerken adımlarını hızlandırarak yoldan geçme süresini minimize etmesi minimizasyon, bir merceğin açısının değiştirilerek Güneş ışığının daha dar bir yerde toplanarak o noktadaki ısının maksimize edilmesi maksimizasyon işlemi olarak adlandırılabilir (A Özkış, 2017). Rüzgâr türbinleri, güneş panelleri gibi enerji sistemleri için maksimum verimliliğin sağlanması maksimizasyon ve havaalanlarındaki uçak iniş-kalkış zamanlamalarının optimize edilerek trafik gecikmelerin ve yakıt tüketiminin minimize edilmesi işlemi minimizasyon örneği olarak verilebilir. Bu tür örneklere genel olarak optimizasyon denmektedir.

Optimizasyon yöntemleri, klasik ve metasezgisel yöntemler olarak temelde ikiye ayrılmaktadır. Aşağıda klasik ve metasezgisel yöntemlerin özellikleri, birbirlerine göre avantaj ve dezavantajları anlatılmıştır.

1.1.1. Klasik optimizasyon yöntemleri

Klasik optimizasyon yöntemleri, genellikle daha kolay ve küçük boyutlu problemler için optimum sonucu bulmak amacıyla uygulanan optimizasyon yöntemleridir. Klasik optimizasyon yöntemleri deterministik yapılarından dolayı tekrar tekrar uygulansa dahi her seferinde aynı çözümü vermektedir. Bu tür yöntemler, tek değişkenli fonksiyonlarda türev alınarak kullanırken çok değişkenli fonksiyonlarda Gradyan ve Hessian matrisleri yardımıyla çalışmaktadır (Yılmaz, 2022). Problem türüne göre hesaplama hızı değişiklik gösterebilmektedir. Bazı gerçek dünya problemlerinde klasik optimizasyon yöntemleri makul zamanda çözüm verememektedir. Makul zamanda çözülmesi zor ya da mümkün olmayan problemlere NP-hard problemler denmektedir.

Klasik optimizasyon yöntemleri genellikle probleme özgü bir çözüm yöntemi sunar. Bu yüzden bu yöntemlerin kullanılabilirdiği problem türü oldukça azdır. Örneğin bir yöntem bir problem türünde çok iyi bir performans sergilerken diğer problem türlerinde iyi bir performans sergileyemeyebilir. Problemin boyutu arttıkça hesaplama maliyeti, bellek tüketimi ve geçen zaman da artmaktadır (Akay, 2009). Ayrıca türev tabanlı teknikler yapısına uygun olmayan problemlerde yerel optimum noktalara takılabilmektedir.

Dolayısıyla klasik optimizasyon yöntemlerinin avantajları şöyle sıralanabilir:

Yapısına uygun olan problemler için;

- Bulunan çözüm optimum ve tam çözümdür. Problemlerin analitik çözümünü elde etmek için kullanılırlar.

- Karmaşık ve çok boyutlu olmayan basit problem türlerinde kolay uygulanabilir olup hızlı yakınsayarak çözümü elde edebilirler.

Dezavantajları ise aşağıdaki gibidir (Akay, 2009):

Yapısına uygun olmayan problemler için;

- Genellikle çok boyutlu karmaşık problemlerin çözümünü elde edememektedirler. Dolayısıyla gerçek dünya problemlerinde uygulanması zordur.
- Bir problemde çok iyi sonuç veren bir yöntem başka bir problem türünde iyi sonuç vermeyebilir.
- Türev tabanlı teknikler genellikle yerel optimum noktalara takılı kalabilmektedir.
- Hafıza tüketimini artırarak problemi çözmeden sonlanmasına sebep olabilmektedirler.

Gerçek dünya problemlerine uygulanmasının zorluğu ve yerel optimum noktalara takılma gibi dezavantajlardan dolayı klasik optimizasyon yöntemleri yerine metasezgisel optimizasyon yöntemleri kullanılmaktadır.

1.1.2. Metasezgisel optimizasyon yöntemleri

Metasezgisel optimizasyon yöntemleri; büyük boyutlu ve karmaşık problemleri çözmek için geliştirilmiş, doğadan esinlenerek oluşturulan optimizasyon yöntemleridir. Arama uzayını verimli şekilde kullanarak sezgisel yöntemleri üst seviyede birleştiren yöntemlerdir (Aytuğ, 2013). Metasezgisel algoritmalar evrimsel yaklaşımlar ve sürü zekâsı yaklaşımları olarak iki gruba ayrılabilir.

Metasezgisel optimizasyon yöntemleri sezgiselliklerinden dolayı deterministik özellik taşımazlar. Aynı problem üzerinde tekrar tekrar çalıştırıldıklarında farklı sonuçlar verirler. Klasik optimizasyon yöntemlerinin yetersiz kaldığı problemlerde, kabul edilebilir bir zamanda makul çözümler üretebilen yaklaşımlardır. Klasik optimizasyon yöntemleri yapılarına uygun olmayan problemlerde optimal çözüme ulaşamamakta ya da ulaşması uzun zaman alabilmektedir. Bu yüzden gerçek dünyadaki karmaşık problemlerde klasik yaklaşımlar başarılı olamayabilmektedir. Metasezgisel optimizasyon yöntemleri yerel optimum noktalara takılabilirler ancak yerel optimum noktalardan kurtulmak için çeşitli stratejilere sahiptirler. Klasik yöntemlere göre de genellikle hızlıdır (Akay, 2009). Bu gibi nedenlerden dolayı metasezgisel optimizasyon

yöntemleri gerçek dünyadaki karmaşık problemlerin çözümünde daha çok tercih edilmektedir.

1.2. Metasezgisel Optimizasyon Algoritmaları

1.2.1. Guguk kuşu arama (CS) algoritması

CS algoritması, X.-S. Yang ve Deb (2009) tarafından geliştirilmiş bir algoritmadır. Bu algoritma, guguk kuşlarının yavru parazitliği gibi ilginç üreme davranışlarına ve Levy uçuşu mekanizmasına dayanır ve guguk kuşlarının uçuş özelliklerini birleştirir (Pu ve ark., 2023). X.-S. Yang ve Deb (2009) bu algoritmayı, aşağıdaki üç kuralla açıklamışlardır:

- Dişi guguk kuşu öncelikle rastgele bir yuva seçerek yumurtasını bırakır. Her seferde bir adet yumurta bırakmaktadır.
- En kaliteli yumurtalara sahip en iyi yuvalar bir sonraki nesillere aktarılır.
- Ev sahibi kuş genellikle guguk kuşu tarafından bırakılan yumurtayı tespit ederek konakçı yumurtayı yuvasından atar ya da kendi yuvasını terk ederek yeni bir yuva kurar.

1.2.2. Diferansiyel evrim algoritması (DEA)

DEA, Storn ve Price (1997) tarafından önerilmiş bir algoritmadır. GA gibi evrimsel yöntemlere alternatif olarak önerilmiştir. DEA; basit, etkili ve performansı güçlü bir algoritmadır. Ancak mutasyon, çaprazlama ve seçim operatörlerinden ve parametre kontrolünden etkilenmektedir. DEA'nın temel adımları aşağıda verilmiştir (Ersin Kaya ve ark., 2021):

- Rastgele oluşturulan kromozomlar ile başlangıç popülasyonu oluşturulur.
- Ardından mutasyon işlemine geçilir. Daha sonra çaprazlama işlemi uygulanır. Mevcut birey, rastgele seçilen iki bireyin farkı alınarak $[0,2]$ aralığından seçilen ölçeklendirme faktörü ile çarpılır. Rastgele seçilen üçüncü birey ile toplanarak uygunluk fonksiyonuna bakılır.
- Mevcut birey ve yeni bireyin uygunluk fonksiyonu karşılaştırılarak daha iyi olan birey belirlenen seçim operatörü yardımıyla seçilerek bir sonraki popülasyona aktarılır.
- Algoritma durdurulana kadar işlem devam eder.

1.2.3. Gri kurt optimizasyonu (GWO) algoritması

GWO algoritması, Mirjalili ve ark. (2014) tarafından, gri kurtların avlanma ve liderlik özelliklerinden ilham alınarak oluşturulmuş metasezgisel bir algoritmadır. Alfa, beta, delta ve omega olarak isimlendirilen dört tür gri kurt ile liderlik özellikleri simüle edilmiştir. Alfa en iyi, beta ikinci en iyi, delta üçüncü en iyi çözümü ifade ederken, omega popülasyonun geriye kalan çözümlerini ifade eder. Gri kurt sürülerinde liderlik özelliğini bir erkek ve bir dişi gri kurt üstlenir ve alfa olarak adlandırılırlar. Beta gri kurtlar, alfalara karar verme gibi konularda yardımcı olan kurtlardır. Omega gri kurtlar en düşük seviye kurtlardır. Diğer kurtlara boyun eğmek zorundadırlar. Üç gruba da girmeyen kurtlara delta kurt denir. Yaşlılar ve avcılar bu gruba aittir. Gri kurtlar av hareket etmeyi durduğunda avlarına saldırırlar. Algoritmanın çalışma prensibinde elde edilen ilk üç en iyi çözüm kaydedilir ve omegaların konumları da diğer en iyi üç çözüme göre güncellenir. Yani alfa beta ve delta avın konumunu tahmin ederken diğer kurtlar avın etrafında rastgele konumlarını günceller (Mirjalili ve ark., 2014).

1.2.4. Bal porsuğu algoritması (HBA)

HBA, Hashim ve ark. (2022) tarafından önerilen, bal porsuğunun beslenme davranışını taklit eden bir metasezgisel algoritmadır. Bal porsuğu balı sevmesine rağmen bal bulmada iyi değildir. Bu yüzden yiyecek kaynağına ulaşmak için bal rehberi kuşunu takip eder veya koklar ya da kazar. Algoritma, kazma aşaması ve bal aşaması olarak iki faza ayrılır. Kazma aşamasında bal porsuğu, koku duyusunu kullanarak avlanmaktadır. Bu aşamada kardioid şeklinde bir hareket gerçekleştirir. Bal aşamasında bal rehberi kuşundan yardım alır. Sonunda balı bulup kovayı parçalayan bal porsuğu sayesinde her iki taraf da iş birliği ile kazanmış olur. Algoritmanın yiyecek bulma yeteneği, koku yoğunluğu, arama yönünü değiştiren bayrak gibi parametreleri bulunur. Bunlardan arama yönünü değiştiren bayrak parametresi -1 ve 1 değerlerinden birini alarak kazma ve bal aşamasına karar verir (C. Aslan ve ark., 2023). Sonlandırma kriteri sağlandığında ise algoritma durur.

1.2.5. Yapay alg algoritması (AAA)

AAA algoritması, Uymaz ve ark. (2015) tarafından geliştirilen, fotosentetik bir tür olan mikroalglerin davranışlarından esinlenerek oluşturulmuş bir algoritmadır. Algoritmanın temeli evrimsel süreç ve adaptasyon süreci ile mikroalglerin davranışlarına dayanmaktadır (Uymaz ve ark., 2015). Mikroalgler, ayrı çekirdek zarları ve klorofilleri

bulunan tek hücreli alglerdir (Atila ve ark., 2018). Algoritma; helisel hareket, evrimsel süreç ve adaptasyon süreci olmak üzere 3 aşamadan oluşmaktadır. AAA algoritması alg kolonilerinden oluşur. Her bir alg kolonisi bir çözümün elemanları olarak adlandırılan alg hücrelerini içermektedir. Koloni içindeki tüm hücreler uygun çevre şartlarına doğru birlikte ilerleyen bir bütündür. Öncelikle, ilk aşama olan helisel hareketinde her bir alg kolonisi, enerjisi bitene kadar bir ışık kaynağı belirler. Işık kaynağı, alg kolonisinin amaç fonksiyonu değerine göre, turnuva metoduyla belirlediği başka bir alg kolonisidir. Daha sonra turnuva metoduyla seçilen alg kolonisinin rastgele üç farklı alg hücresi seçilerek konum güncellemesi yapılır. Sonrasında alg kolonisinin açlık değeri güncellenerek evrimsel sürece geçilir. En büyük ve en küçük alg kolonisi seçilir. Adaptasyon sürecinde ise en aç alg kolonisi seçilerek bu alg kolonisi adaptasyona uğratılır. Bu süreçte koloni, hayatta kalabilmek için aynı ortamda yer alan en büyük koloniye benzemeye çalışır. En iyi alg kolonisi saklanarak durdurma kriterine uygunluğuna bakılır. Uyana kadar süreç devam ederek optimum çözüme ulaşılır (Türkoğlu, 2019).

1.2.6. Deniz yırtıcıları algoritması (MPA)

MPA algoritması, Faramarzi ve ark. (2020) tarafından geliştirilen bir metasezgisel algoritmadır. Algoritma, deniz yırtıcılarının beslenme stratejisini Levy ve Brownian hareketlerini kullanarak ele alır. Av matrisi, başlangıç popülasyonunun konumlarını tutmaktadır. Aynı zamanda en iyi uygunluk değerlerini tutan Elite matrisini oluşturmaktadır. Algoritma üç optimizasyon aşamasına ayrılır (Faramarzi ve ark., 2020). MPA algoritması, arama stratejisini maksimum iterasyon sayısının her üçte birinde değiştirmektedir. İlk iterasyon grubunda keşfe, orta iterasyon grubunda dengeli aramaya ve son grupta da sömürüye önem vermektedir. İlk aşamada av, avcıdan daha hızlı hareket etmektedir. Bu aşama, algoritmanın keşif sürecinde olduğu iterasyonlarda gerçekleşir. İkinci aşamada av ve avcı aynı hıza sahiptirler. Bu aşamada popülasyonun yarısı keşif için kullanılırken diğer yarısı sömürü için kullanılır ve keşfin sömürüye dönüştüğü aşamadır. Üçüncü aşamada ise avcı avdan daha hızlı hareket etmektedir. Bu aşama sömürü sürecine odaklanılan aşamadır (Ahmet Özkış, 2024). Elite matris güncellemesi yapılır ve hafıza tasarrufu sağlanır. Tolerans değeri sağlanana kadar algoritma bu işleme devam eder.

1.2.7. Sürüngen arama algoritması (RSA)

RSA, Abualigah ve ark. (2022) tarafından timsahların avlanma davranışlarından ilham alınarak geliştirilmiş metasezgisel bir algoritmadır. Bu algoritmanın iki ana aşaması vardır. Bunlar, avın kuşatılması ve avlanması olmaktadır. Bu iki aşamanın da ikişer arama stratejisi vardır. Keşif aşamasına ait iki strateji, yüksek yürüyüş ve göbek yürüyüşü arama yöntemleridir. Sömürü aşamasının stratejileri ise av koordinasyonu ve av iş birliğidir. Sömürü arama stratejisinde yerel optimum noktalara takılmaktan kaçınmaya çalışılmaktadır. Rastgele aday çözümlerle algoritma başlatıldıktan sonra keşif ve sömürü sürecinde ikişer strateji uygulanır. Bu stratejiler belirli tolerans değerlerinde uygulanarak tolerans değeri istenene ulaştığında algoritma durdurulmaktadır (Abualigah ve ark., 2022).

1.2.8. Sinüs kosinüs algoritması (SCA)

SCA, Mirjalili (2016) tarafından, matematikte aynı isme sahip olan sinüs ve kosinüs fonksiyonlarından ilham alınarak önerilmiş metasezgisel bir algoritmadır. Keşif aşaması için sinüs, sömürü aşaması için kosinüs fonksiyonu kullanılmıştır. Rastgele çözümlerle algoritma başlatılır. En iyi çözümler kaydedilerek bu çözümlere göre diğer çözümler de güncellenir. İterasyon arttıkça sinüs kosinüs fonksiyonlarının aralıkları, arama alanının kullanımını vurgulamak amacıyla güncellenir. Tolerans değeri maksimuma ulaşmaya kadar optimizasyon süreci devam etmektedir. Sinüs ve kosinüs fonksiyonları 1 ve -1 arasında değer döndürdüğünde arama alanının doğru bölgesinden yararlanılmış olup bu değerlerden farklı değer döndürdüğünde arama alanında farklı bölgeler keşfedilmektedir (Mirjalili, 2016).

2. KAYNAK ARAŞTIRMASI

2.1. YSA'nın Tarihçesi ve YSA-Metasezgisel Algoritmalarla Hibrit Kullanımı

Kaynak Araştırması

YSA terimi, 1940'lı yıllarda ortaya atılmış ve sonraki yıllarda büyük ilgi gören bir konu haline gelmiştir (Öztemel, 2003). McCulloch ve Pitts (1943) yaptıkları bir çalışmada nörofizyolojik olarak sinir sisteminden, buna bağlı olarak da sinir ağlarından bahsetmişlerdir. Böylece sinir ağlarının ilk kez matematiksel modeli oluşturulmuştur. Yayınladıkları çalışma ile büyük etki uyandırmışlardır. Hebb Donald (1949) ise "The Organization of Behavior" isimli kitabında, yapay hücrelerden oluşan bir YSA'nın ağırlık değerlerini eğiten bir öğrenme kuralı olan Hebb Öğrenme Kuralı'nı önermiştir. Rosenblatt (1958) basit algılayıcı modeli (perceptron) yani tek katmanlı algılayıcıyı geliştirmiştir. Widrow ve Hoff (1960), ADALINE ve MADALINE modelini geliştirmişlerdir. Minsky ve Papert (1969), yazdıkları "Perceptron" (Algılayıcılar) kitabında tek katmanlı algılayıcıların XOR problemi gibi doğrusal olmayan problemlere çözüm üretemediğinin göstermişlerdir. Bu olay yapay zekâ ve YSA'nın duraklama döneminin başlamasına neden olmuştur. Bu dönemde YSA hakkında yapılan çalışmalar yok denecek kadar azalmış ve bu alanda çalışma yapmak boş bir iş olarak görülmüştür. Rumelhart ve ark. (1986) tarafından çok katmanlı algılayıcı geliştirilmiştir. Bu gelişme ile doğrusal olmayan problemlerin de çözülebileceği ortaya çıkmış, YSA ve yapay zekâ ile ilgili çalışmalar kaldığı yerden hızlanarak devam etmiştir.

Literatürde, YSA ve metasezgisel algoritmaların hibrit kullanıldığı çok sayıda çalışma olduğu görülmektedir. Karaboga ve Akay (2007); ABC, DEA ve PSO algoritmalarını YSA eğitiminde kullanarak performanslarını karşılaştırmışlardır. YSA eğitiminde XOR problemi, 3-Bit Parity problemi ve 4-bit Encoder-Decoder problemlerini kullanmışlardır. Qiao ve ark. (2021), bir sınıflandırıcı geliştirerek sınıflandırıcının parametrelerini optimize etmek için WOA'nın performansını iyileştirmiş ve YSA'yı eğitmek için kullanmışlardır. Değerlendirme için yirmi üç kıyaslama test fonksiyonundan yardım almışlardır. Şaban Gülcü (2022), yerel optimum noktalara takılmasını önlemek için YSA'yı Yusufçuk Algoritması ile eğiterek DA-MLP modelini geliştirmiştir. Turkoglu ve Kaya (2020), AAA ile YSA'yı eğitmiş ve UCI veri setindeki 10 farklı veri seti üzerinde performansını analiz etmişlerdir. Maleki ve ark. (2019) kurutucularda nem tahmini yapmak için YSA ve GA'dan yararlanmışlardır. Aljarah ve ark. (2018), YSA'nın bağlantı ağırlıklarının optimizasyonunda WOA'yı kullanmışlardır. Özden ve İşeri (2023), YSA eğitimi aşamasında türev tabanlı tekniklerin yerel optimuma takılma sorununu

önlemek için COOT Optimizasyon Algoritmasını kullanarak hibrit bir model geliştirmişlerdir. Bairathi ve Gopalani (2021), Salp Sürüsü Algoritmasını geliştirerek ileri beslemeli sinir ağlarının eğitiminde kullanmışlardır. Ebubekir Kaya (2022); ABC, CS algoritması, PSO, JAYA, SCA ve WOA gibi 16 farklı metasezgisel algoritma ile YSA'yı eğiterek kapsamlı bir performans kıyaslaması yapmıştır. Song ve ark. (2022), elastik GWO algoritması ile geliştirdikleri MLP modelini (EGWO-MLP) öğrencilerin başarı durumunu tahmin etmek için kullanmışlardır. J. Li ve ark. (2024), müzikte duygu tanıma için sinir ağları ile DEA'yı birlikte kullanmışlardır. Thankachan ve ark. (2024), CS tabanlı bir YSA modeli kullanarak çelik yapıların sağlığını izlemişlerdir. Narayanan ve ark. (2023); kablosuz sensör ağlarının savunmasızlığını gidermek, saldırı tespiti ve saldırıların önlenmesi amacıyla YSA-PSO hibrit modelini geliştirmişlerdir. Ansari ve ark. (2020), şirketlerin iflas etme ihtimalini tahmin etmek için Manyetik Optimizasyon ve PSO algoritmaları ile YSA'yı eğitmiş ve başarılı sonuçlara ulaşmışlardır.

YSA ve metasezgisel algoritmalar, inşaat mühendisliği problemlerinin çözümünde de yaygın olarak kullanılmaktadır. Kazemi ve ark. (2023); atık dökümhane kumu içeren betonun basınç dayanımını tahmin etmek için Karınca Kolonisi Optimizasyonu Algoritması ve PSO'dan yararlanarak geliştirdikleri ACO-YSA ve PSO-YSA modellerini kullanmışlardır. Wu ve ark. (2023), boru hatlarındaki hasarları tespit etmek amacıyla geliştirilmiş WOA tabanlı bir YSA modeli (IWOA) tasarlamışlardır. IWOA-BPNN modelini geri yayımlı sinir ağının ağırlık ve eşik değerlerini optimize etmek için kullanmışlardır. Tran-Ngoc ve ark. (2019), köprü ve kirişlerdeki hasarları tespit etmek için eğitim parametreleri CS algoritması tarafından belirlenen bir YSA modeli kullanmışlardır. Bui ve ark. (2018), yüksek performanslı betonun basınç ve çekme dayanıklılığını tahmin etmek için modifiye edilmiş FA tabanlı bir YSA modelinden yararlanmışlardır. Jitchaijaroen ve ark. (2024), kum üzerine inşa edilmiş olan şerit temellerinin taşıma kapasitesini değerlendiren bir çalışma yapmışlardır. Emperyalist Rekabetçi Algoritma (ICA), ALO, TLBO ve PSO'nun geliştirilmiş bir versiyonunu kullanarak gizli nöronların optimum sayısını seçmeyi amaçlayan çeşitli modeller tasarlayarak karşılaştırmışlardır. YSA-TLBO modelinin diğer modellerden taşıma kapasitesi faktörünün tahmininde üstün başarılı olduğunu ortaya çıkarmıştır.

Chen ve ark. (2021) katı oksit yakıt hücrelerinin optimum operasyonel değişkenlerini elde etmek için YSA ve GWO tabanlı bir model geliştirerek bu modelin üretim maliyetini düşürdüğünü ortaya koymuşlardır.

Enerji tüketimi konusunda da benzer çalışmalar yapılmıştır. Bui ve ark. (2020), binalardaki enerji tüketim maliyetlerini tahmin etmek için Elektromanyetizma Tabanlı FA (EFA) tabanlı bir YSA modeli geliştirmişlerdir. Talaat ve ark. (2020), enerji üretimi ve elektrik satın alma planlamasına yardımcı olması için GOA ile ileri beslemeli bir YSA modeli geliştirmişlerdir. Li ve ark. (2023) ise, YSA ve Gri Kurt Optimizasyonu (GWO) algoritmasını geliştirerek hibrit şekilde kullanmışlar ve mikro şebeke isimli çevre koruma sisteminin daha iyi çalışması için bir yöntem önermişlerdir. Bu sayede mikro şebeke kullanıcılarının aralarında güç paylaşımı sağlayan sis odaklı proje ile verimli bir çevre oluşturmayı amaçlamışlardır. Sonuç olarak simülasyon sonunda evlerin ithal ettikleri güç ile enerji maliyetlerinin yarı yarıya azaldığını ifade etmişlerdir.

YSA ile metasezgisel algoritmalar sağlık alanında da kullanılmaktadır. Thamaraimanalan ve Ramalingam (2024), vücuda takılan sensörlerin sıklıkla arızalanması ve elde edilen verilerin doğruluğunun anlaşılabilmesi nedeniyle anormal verileri bulmak için YSA-GOA modeli geliştirmişlerdir. Si ve ark. (2022), tıbbi veri sınıflandırma problemlerinde kullanmak amacıyla YSA eğitiminde Denge Optimize Edici algoritmasından faydalanmışlardır. Zamani ve Nadimi-Shahraki (2024), kronik hastalıkların teşhisinde kullanılmak üzere Evrimsel Karga Arama Algoritması ile optimize edilen bir YSA modeli geliştirmişlerdir. Kullandıkları hibrit model ile karşılaştırdıkları rakip modellere oranla üstün bir başarı elde etmişlerdir. Gong ve ark. (2020), beyin tümörü bölgesini doğru tespit etmek için kaos tabanlı iyileştirilmiş bir WOA kullanan YSA modeli geliştirmişlerdir.

Bendine ve ark. (2023) Lichtenberg Algoritması (LA) kullanmışlardır. Yıldırım fırtınalarından ilham alınarak oluşturulmuş bu algoritma ile YSA'yı hibrit şekilde modellemişlerdir. Başarılı sonuçlar elde etmişlerdir. S. Yang ve ark. (2023) YSA ve Genetik Algoritma (GA)'yı maya fermantasyonu için kullanmışlardır. Çeşitli istatistiksel testlerle modellerinin başarılı olduğunu göstermişlerdir. C. Wang ve ark. (2024), katı oksit yakıt hücresinde iç sıcaklık farkını azaltmak amacıyla YSA ve GA içeren bir model önermişlerdir. Öncelikli olarak YSA eğitilerek bir yedek model oluşturulmuştur. Daha sonra uyguladıkları tüp içi reformer kullanımının hücre içindeki sıcaklık farkını en aza indirmesi için genetik algoritma optimizasyonları gerçekleştirilmiştir. Sonuç olarak hücre içi sıcaklık farkı oldukça aşağıya çekilerek başarı elde edilmiştir.

Bernard ve ark. (2022) ise YSA ve metasezgisel algoritmaları eğitim alanında kullanmışlardır. Öğrenme stillerinin eğitim hayatında öğrencilerin güçlü ve zayıf yönlerini anlamasının önemine dikkat çekerek bir hibrit model oluşturmuşlardır. YSA ile

Karınca Kolonisi Optimizasyonu (ACO) Algoritması kullanmışlardır. Öğrenme stillerinin belirlenmesinde kesinliği artırmak amacını gütmüşlerdir. Çalışmalarının sonuçları, diğer yaklaşımlardan daha iyi değerler vermiştir. Şaban Gülcü (2022) ise çalışmalarında hayvanların göç davranışlarından ilham alarak oluşturulan orijinal Hayvan Göçü Optimizasyon (AMO) algoritmasını geliştirerek YSA ile hibrit kullanmıştır. Dokuz farklı algoritma ve orijinal AMO algoritmasını kıyaslayarak sonuçların daha başarılı olduğunu saptamıştır. S. Yang ve ark. (2023) çalışmalarında, Kutup Ayısı Optimizasyonu (PBO) ile YSA'yı optimize etmek istemişlerdir. Yenilenebilir enerji kaynaklarına bir yenisini daha ekleyebilmek için ötektik tuzları kullanmışlardır. Ötektik tuzların termofiziksel özelliklerini kullanan yeni bir tahmin modeli önermişlerdir. Doğruluk oranlarında başarılı sonuçlar elde etmişlerdir.

Karakoyun (2024), AAA'nın konum güncelleme tekniğine yeni bir seçim mekanizması ekleyerek algoritmayı geliştirmiş ve geliştirdiği algoritmaya MsAAA adını vermiştir. Bu algoritma ile YSA'yı eğiterek 8 farklı metasezgisel algoritmanın performansı ile karşılaştırmıştır. 21 farklı veri setinden elde ettiği sonuçlar, geliştirilen algoritmanın diğer algoritmalarından daha iyi sonuçlar verdiğini göstermiştir. Guo ve ark. (2024), YSA'yı inşaat mühendisliği alanında kullanmışlardır. Betonun basınç dayanımı birçok parametreye bağlı olduğundan betonun davranışını tahmin etmek zor bir konudur. Bu konu için, Afrika Akbabası Optimizasyon Algoritması'nı Elman Sinir Ağları ile hibrit kullanarak bir model tasarlamışlardır. Sonuç olarak çalışma, istenen özellikleri tahmin etmede modelin yüksek güce sahip olduğunu göstermiştir.

Ahmet Özkış (2024), MPA'nın yerel arama stratejisinin ve global arama yeteneğinin olumsuz etkisini iyileştirmek amacıyla MultiPopMPA isimli algoritmayı önermiştir. YSA ile hibrit bir model oluşturarak 21 farklı sınıflandırma veri seti üzerinde YSA eğitimi için kullanmıştır. Elde edilen sonuçlar, 8 farklı algoritma ile karşılaştırılmıştır. Önerilen algoritmanın diğer algoritmalarından daha başarılı olduğu görülmüştür. Maheswari ve Nagarajan (2024), ses ve metin tabanlı iletişim sağlamak için bir Chatbot tasarlamışlardır. Koloni Avı Algoritması ve İkili İmparator Penguen Optimizasyonu algoritmalarını hibritleyerek parametreleri optimize etmiş ve evrişimli sinir ağlarından yardım almışlardır. Sonuç olarak verilen sorgu için doğru verilerin yüksek tahmini sağlanarak başarı elde edilmiştir.

Monika ve Bhat (2022), Dünya Kupası Optimizasyonu algoritması ile dalgacık evrişimli sinir ağını hibrit kullanılarak bir model oluşturmuşlardır. Önerdikleri modeli, sosyal medyada suç tahmini için kullanmışlardır. Twitter'dan aldıkları verileri

temizleyerek özellik çıkarma işlemi uygulamışlar ve hibrit modelleri ile suç tespitini gerçekleştirmişlerdir. Bas ve ark. (2020), çarpımsal nöron modeli YSA için metasezgisel bir algoritma olan Yapay Yarasa Algoritması ve klasik algoritmalarından olan Geri Yayılım Algoritması ile hibrit bir model oluşturmuşlardır. Önerdikleri modeli, üç farklı gerçek dünya zaman serisi verisine uygulamışlardır. Çeşitli algoritmalar ile karşılaştırmışlardır. Sonuçlar önerilen yöntemin diğer yöntemlere göre daha iyi olduğunu göstermiştir. Rojas ve ark. (2024), ALO algoritmasını, çaprazlama ve mutasyon operatörleri gibi genetik operatörler kullanarak geliştirmiş ve GALO algoritmasını önermişlerdir. Önerdikleri bu algoritmayı YSA'nın eğitilmesi için kullanmışlardır. Algoritmanın değerlendirilmesi için 20 tıbbi veri seti kullanılmış ve 31 farklı algoritma ile test edilmiştir. Kullanılan tıbbi veri setlerinin çoğunda sınıflandırma kalitesinin iyileştiği görülmüştür. Gölcük ve ark. (2023), YSA'yı eğitmek için iyileştirilmiş bir Aritmetik Optimizasyon Algoritması önermişlerdir. Ağırlık ve bias değerleri optimize edilmiştir. Algoritmanın performansı, 12 dinamik sınıflandırma problemi üzerinde test edilmiştir. Çeşitli metasezgisel algoritmalarla karşılaştırılmıştır. Sonuçlar iyileştirilmiş AOA'nın karşılaştırılan algoritmalarından daha iyi performans sergilediğini göstermiştir.

2.2. SAO Algoritması Kaynak Araştırması

Jia ve ark. (2023), orijinal SAO algoritmasında sıcaklık artışı ile su moleküllerinin su buharına dönüşeceğini ve keşif ile sömürü arası dengeyi bozabileceğini düşünerek ısı transferini ve yoğunlaşma stratejisini kullanarak orijinal SAO algoritmasında bir iyileştirme yapmışlardır. Çeşitli benchmark ve mühendislik problemleri üzerinde test ederek geliştirdikleri algoritmanın başarısını göstermişlerdir. Algoritma, tek modlu fonksiyon problemlerinin hepsinde rakip algoritmaların tamamından daha iyi sonuçlar verirken basit çok modlu problemlerin ve test fonksiyonlarının ise çoğunda daha iyi sonuçlar elde etmiştir. Ismaeel ve ark. (2024), jeneratör ünitelerinin optimum yerleşimi için SAO algoritmasından faydalanmışlardır. 6 durumun ele alındığı çalışmada 700-1000 MW'lık 2 yükteki 6, 1000-2000 MW'lık 2 yükteki 10 ve 2000-3000 MW'lık 2 yükteki 20 jeneratör kullanarak optimum güç uyumsuzluğu değerlerini SAO algoritması kullanarak minimuma çekmek amaçlanmıştır. Çalışma, SAO'nun diğer algoritmalara göre yakıt maliyetini azalttığını ortaya koymuştur.

Xiao ve ark. (2024), orijinal SAO'nun olumsuz faktörlerini azaltmak için algoritmayı geliştirerek MSAO isimli yeni bir varyasyonunu önermişlerdir. Dört geliştirme stratejisine sahip algoritma için ilk olarak iyi nokta kümesi başlatma stratejisi

oluşturulmuştur. Kaliteli bir popülasyon ile arama bölgesine olabildiğince hızlı girmek amaçlanmıştır. İkinci olarak aç gözlü seçim metodu ile bir sonraki iterasyon için daha iyi bireyler seçilmiştir. Üçüncü olarak sömürü aşamasında DEA'dan yararlanılarak daha yüksek yakınsama doğruluğu elde edilmesi amaçlanmıştır. Son olarak ise Dinamik Lens Karşıtlığı Tabanlı Öğrenme stratejisi geliştirilerek yerel optimuma takılma riski azaltılmak istenmiştir. Bununla beraber MSAO algoritmasının bazı iyileştirilmiş algoritmalar ile karşılaştırıldığında daha iyi sonuçlar verdiği görülmüştür.

Abd Elaziz ve ark. (2024), hedef tespitinde kullanılan ve bir uzaktan algılama tekniği olan hava fotoğrafçılığında elde ettikleri verilerin segmentasyon sürecinde SAO algoritmasını geliştirerek geliştirilmiş versiyonundan faydalanmışlardır. Hava görüntülerini segmentlemek için kullanılacak eşik seviyelerinin optimumunu keşfetme aşamasında, Kapsamlı öğrenme ve Çift çekici mekanizmasının güçlü yönleri alınarak SAO algoritma modifiye edilmiştir. Böylece araştırmacılar SAO'nun keşif ve sömürü aşamalarını geliştirmişlerdir. Geliştirdikleri yeni varyasyona DCSAO adını vererek CEC2022 kıyaslama fonksiyonunda ve farklı metasezgisel algoritmalar ile karşılaştırmışlardır. Elde ettikleri sonuçlar DCSAO'nun diğer algoritmalarından daha üstün olduğunu göstermiştir. Pandya ve ark. (2024), karmaşık güç sistemlerinde kullanılmak üzere SAO algoritmasını geliştirmişlerdir. Geliştirdikleri bu varyasyona MOSAO adını vermişlerdir. Algoritmayı Esnek Alternatif Akım İletim Sistemleri (FACTS) cihazlarının yerleşim ve boyutlandırılması için kullanmışlardır.

Ding ve ark. (2024), Otonom yankı durumu ağı (AESN)'nin hiper parametrelerini optimize etmek için SAO algoritmasını da dahil ettikleri AESN-SAO isimli bir tahmin yöntemi önermişlerdir. Rössler sistemi, Colpitts osilatörü, Lorenz-63 sistemi ve iklim Lorenz-63 sistemi isimli dört kaotik sistem üzerinde ve dört metasezgisel algoritma ile karşılaştırma yapılarak test edilmiştir. Bazı durumlarda kaynak kullanımını daha aza indirdiği için hesaplamalı kaynak kullanımını azaltmak amacıyla kullanımının uygun olabileceği sonucuna ulaşılmıştır. Maddaiah ve Narayanan (2023), YSA'yı sabitlemek için SAO algoritmasını geliştirerek yardım almışlardır. Algoritmaya ISAO adını vererek YSA'nın ağırlıklarını ve bias değerlerini başlatmak için önceden eğitilmiş (pre-trained) ağırlık ve bias değerlerini bulmak amacıyla kullanmışlardır. Geliştirdikleri algoritmayı, MNIST veri seti üzerinde test ederek üç farklı algoritma ile karşılaştırmışlardır. ISAO algoritması çapraz entropi, eğitim-test doğruluğu ve karşılaştırdıkları diğer algoritmaların başarıları açısından daha iyi sonuçlar vermiştir. Lu ve ark. (2023) DE algoritmasının güçlü küresel arama yetenekleri sergilemesinden esinlenerek SAO ve DE

algoritmalarının güçlü yönlerini birleştiren DESAO isimli yeni bir algoritma önermişlerdir. IEEE CEC2017 benchmark test fonksiyonlarından alınan deneysel sonuçlar önerdikleri algoritmanın hızlı yakınsama oranında üstün olduğunu ortaya koymuştur.

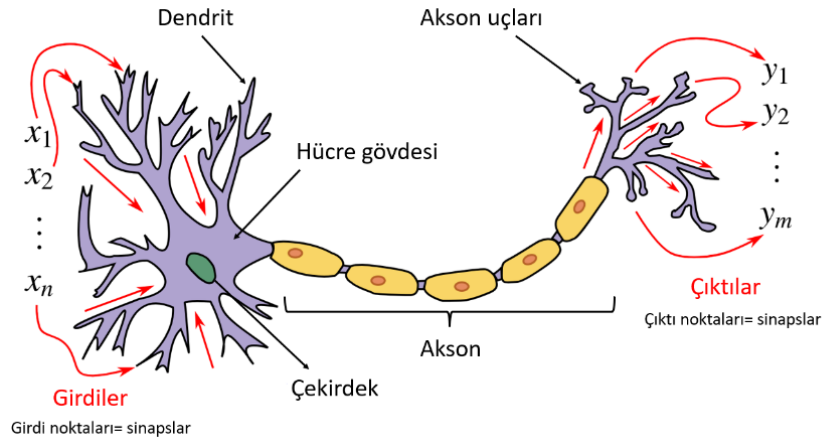
Anten dizilerini tasarlamak ve geliştirmek için birçok algoritmadan yararlanılmıştır. Yuan ve ark. (2024) seyrek dağıtılmış bir anten dizisinin optimizasyon sürecinde kaotik haritalama özelliğini SAO algoritmasına entegre ederek CSAO isimli algoritma önermişlerdir. CSAO algoritması ile karşılaştırılan diğer evrimsel algoritmalara bakıldığında CSAO algoritmasının daha iyi performans gösterdiği sonucuna ulaşılmıştır. Yaptıkları çalışma, anten dizilerini optimize etmede ve diğer elektromanyetizma ile ilgili problemlerde CSAO algoritmasının kullanılabilirliğini göstermiştir.

3. MATERYAL VE YÖNTEM

Bu bölümde, YSA ve YSA'ları oluşturan elemanlar, toplam ve aktivasyon fonksiyonlarından bahsedilerek SAO algoritması, deneysel çalışma, veri setleri ve performans metrikleri detaylı olarak anlatılmaktadır.

3.1. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), biyolojik sinir ağlarından ilham alınarak geliştirilen, sinir ağlarının yaptıklarını taklit edebilen bir sistemdir. Biyolojik sinir ağları, beş duyu organı vasıtasıyla dışarıdan gelen bilgileri, çeşitli anlama ve algılama yöntemleri kullanarak sonuçlar üretir. Milyarlarca sinir hücresi bir araya gelerek sinir sistemini oluşturmaktadır (Öztemel, 2003). YSA da benzer şekilde girdilere çeşitli işlemler uygulayarak sonuç üreten bir sistemdir. Şekil 3.1.'de bir biyolojik sinir ağı hücresi ve hücrenin içeriği gösterilmektedir.



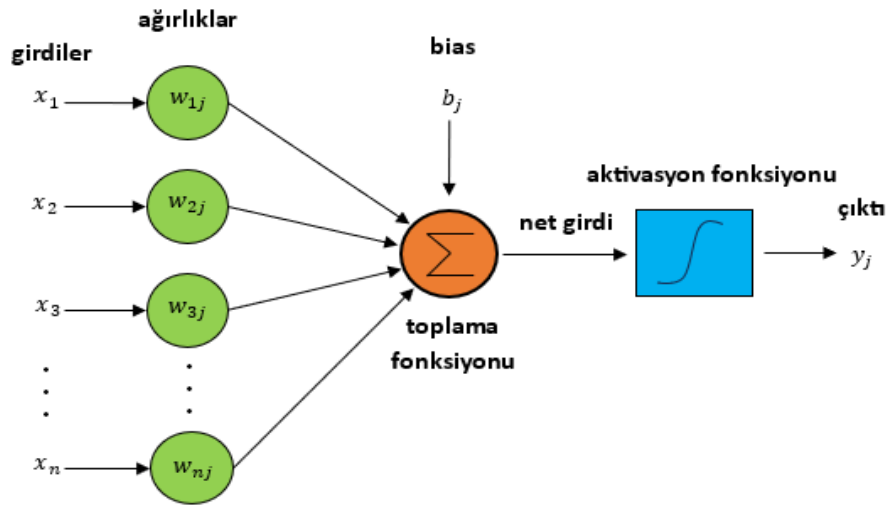
Şekil 3.1. Biyolojik sinir ağı hücresi (Bulut, 2018)

Çizelge 3.1.'de görüldüğü üzere biyolojik sinir hücresi üzerinde, dendritte taşınan bilgi, yapay sinir hücresinde girdi elemanına, biyolojik sinir hücresindeki sinaps bağlantıları, yapay sinir hücresindeki ağırlıklara, biyolojik sinir hücresindeki çekirdek, yapay sinir hücresinde toplam fonksiyonuna, biyolojik sinir hücresindeki hücre gövdesi, yapay sinir hücresindeki aktivasyon fonksiyonuna ve biyolojik sinir hücresindeki akson, yapay sinir hücresindeki çıktı elemanına karşılık gelmektedir.

Çizelge 3.1. Biyolojik ve yapay sinir hücresi elemanları

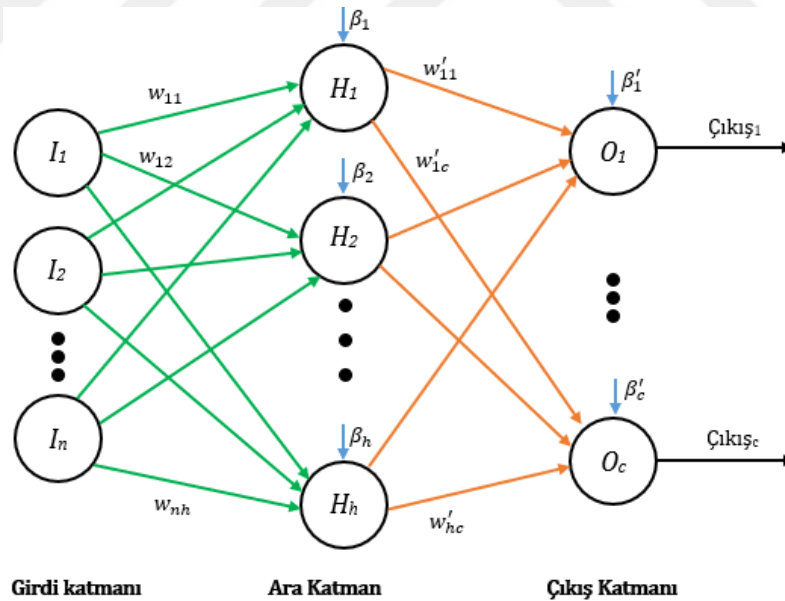
Biyolojik Sinir Hücresi Elemanı	Yapay Sinir Hücresi Elemanı
Çekirdek	Toplam fonksiyonu
Akson	Çıktı
Dendritte taşınan bilgi	Girdi
Sinaps bağlantıları	Ağırlık
Hücre gövdesi	Aktivasyon fonksiyonu

YSA'nın en küçük birimi yapay sinir hücresi olarak adlandırılır. Yapay zekâ sisteminin öğrenmesi istenilen problemin öznitelikleri, yapay sinir hücresine girdi olarak verilir. Bu girdiler ağırlık isimli başka sayısal değerlerle çarpılarak toplama fonksiyonuna gider, burada net girdi hesaplanır. Net girdi hesaplandıktan sonra aktivasyon fonksiyonuna gider, ardından sonuç elde edilir. Şekil 3.2.'de bir yapay sinir hücresinin; girdi, çıktı, toplama fonksiyonu, aktivasyon fonksiyonu, ağırlık ve bias değerleri görülmektedir (Öztemel, 2003).



Şekil 3.2. Bir yapay sinir hücresinin temel bileşenleri

YSA mimarisi, Tek Katmanlı Algılayıcı (TKA) ve Çok Katmanlı Algılayıcı (ÇKA) olarak ikiye ayrılmaktadır. Şekil 3.2.'deki görsel, en basit haliyle bir sinir hücresinin ve TKA mimarisinin yapısını göstermektedir.



Şekil 3.3. Çok katmanlı algılayıcı mimarisi

Şekil 3.3.'deki görsel, bir ÇKA mimarisine örnektir. ÇKA mimarisi; girdi katmanı, ara (gizli) katman ve çıktı katmanında oluşur. Ara katmandaki nöron sayısı ve ara katman sayısının kaç olması gerektiğine dair kesin bir bilgi yoktur. Bu çalışmada, ara

katmandaki nöron sayısı Mirjalili (2015) ile Turkoglu ve Kaya (2020) tarafından kullanılan Denklem (3.1)'deki yöntemle hesaplanmıştır.

$$araKatmandakiNöronSayısı = 2 * \text{öznitelikSayısı} + 1 \quad (3.1)$$

YSA insan gücüne gerek duymadan kendi kendine öğrenebilen mekanizmalardır (Anderson ve McNeill, 1992). Performansını kesin olarak neyin etkilediği hakkında tam olarak bilgi sahibi olunamasa da birçok çalışma sonucunda eğitim algoritması, veri seti ön işleme, parametrelerin doğru ayarlanması gibi çeşitli faktörlerin performansını etkileyebileceği sonucuna ulaşılmıştır (Ataseven, 2013). YSA hakkındaki ilk çalışmalar TKA mimarisi üzerine olmuştur. Ancak TKA'ların sadece doğrusal problemleri öğrenebildiğinin anlaşılması üzerine, doğrusal olmayan problemleri de öğrenebilen ÇKA mimarisi doğmuştur. Günümüzde ÇKA mimarisi ile çok sayıda çalışma (Gupta ve Deep, 2020) (P. Li ve ark., 2024) (Raziani ve ark., 2022) yapılmaktadır. Bu çalışmada da ÇKA mimarisi kullanılmıştır.

$$Net\ Girdi = Net(eski) + \sum_i (G_i A_i) \quad (3.2)$$

$$f(Net) = \frac{1}{1 + e^{-Net}} \quad (3.3)$$

Literatürde birçok toplam ve aktivasyon fonksiyonu yer almaktadır. Bu çalışmada toplam fonksiyonu olarak Denklem (3.2)'de verilen kümülatif toplam fonksiyonu, aktivasyon fonksiyonu olarak ise Denklem (3.3)'te verilen sigmoid fonksiyonu tercih edilmiştir. İleri yayılım hareketi olarak bilinen bu hareket sonrasında, veriler doğru sınıflandırılmamışlarsa ağırlıklar birtakım işlemler yardımıyla güncellenir. Ağırlıkların güncellenmesi hareketi ise geri yayılım olarak adlandırılır.

Katmanlardaki nöron sayısı aşağıda belirtildiği şekilde ayarlanmıştır:

- Giriş Katmanı: Veri setinin içerdiği öznitelik sayısına göre belirlenmektedir.
- Ara Katman: Denklem (3.1)'e göre hesaplanmaktadır.
- Çıkış Katmanı: Veri setindeki sınıf sayısına göre belirlenmektedir.

YSA'da geri yayılım sırasında ağırlıkları güncellemek için genelde türev tabanlı teknikler kullanılır. Problemlerin zorluk seviyesi arttıkça, türev tabanlı teknikler yerel

minimumu takılma sorunuyla karşılaşarak başarısız sonuçlar üretebilmektedir. Bunun yerine, ağırlık değerlerinin güncellenmesinde metasezgisel algoritmalarından faydalanarak yerel minimuma takılma ihtimali azaltılabilir. Bu sebeple, bu çalışmada YSA'nın ağırlık değerleri önerilen LevySAO algoritması ile güncellenmiştir.

3.1.1. Toplam fonksiyonları

Çizelge 3.2. Toplam fonksiyonları

Net giriş	Açıklama
Çarpım $\text{Net Girdi} = \prod_i G_i A_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak net girdi hesaplanır.
Maksimum $\text{Net Girdi} = \max(G_i A_i), i = 1, 2 \dots N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Minimum $\text{Net Girdi} = \min(G_i A_i), i = 1, 2 \dots N$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Çoğunluk $\text{Net Girdi} = \sum_i \text{sgn}(G_i A_i)$	N adet girdi içinden ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücresinin net girdisi olarak kabul edilir.
Kümülatif toplam $\text{Net Girdi} = \text{Net}(\text{eski}) + \sum_i (G_i A_i)$	Hücreye gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücresinin net girdisi bulunur.

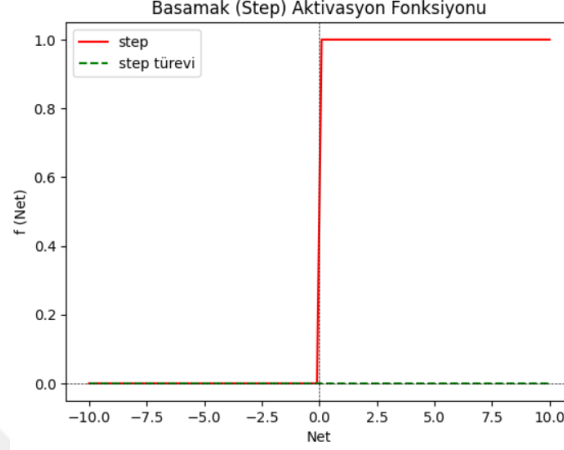
3.1.2. Aktivasyon fonksiyonları

Hücreye gelen net girdiyi çeşitli işlemlerle işleyerek hücrenin girdiye karşılık oluşturduğu çıktıyı belirleyen fonksiyonlara aktivasyon fonksiyonu denmektedir (Öztemel, 2003). Aktivasyon fonksiyonları gelen bilgiyi belirli değerler arasına çekerek normalize işlemi yapmaktadır. Bu fonksiyonlar, nöronların bilgi yayılımını ve aktivasyon seviyelerini doğrudan etkilemektedir. Bu nedenle hangi aktivasyon fonksiyonunun seçildiği problem için önem arz etmektedir (Y. Wang ve ark., 2024). Aşağıda, literatürde yaygın olarak kullanılan aktivasyon fonksiyonlarından beş tanesi tanımlanmıştır.

Basamak (Step) Fonksiyonu: Net girdinin belirlenen eşik değerinin altında ya da üstünde olmasına göre 0 veya 1 değerini alan fonksiyondur (Öztemel, 2003). Hücrenin çıktısı 0 veya 1 olabilir. Bu yüzden ikili sınıflayıcı olarak kullanılır. Basamak (step)

fonksiyonunun formülü Denklem (3.4)'teki gibidir. Şekil 3.4.'te basamak fonksiyonunun ve türevinin grafiği verilmektedir (Irmak, 2022).

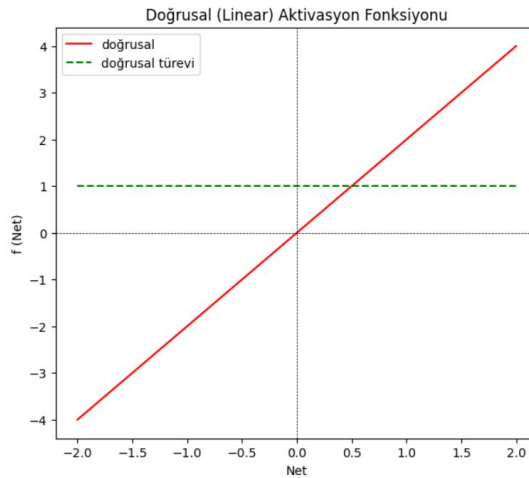
$$f(Net) = \begin{cases} 0, & Net < 0 \\ 1, & Net \geq 0 \end{cases} \{0,1\} \quad (3.4)$$



Şekil 3.4. Basamak fonksiyonu ve türevi

Doğrusal (Linear) Fonksiyon: Net girdinin, doğrudan çıktı olarak değer aldığı fonksiyondur. Fonksiyonun çıktısı $-\infty$ ile $+\infty$ arasında değer alabilmektedir. Doğrusal fonksiyonun türevi sabit değer aldığından dolayı geri yayılım sırasında sorun yaşanmaktadır. Bu yüzden ÇKA'larda tercih edilmez. Doğrusal fonksiyonunun formülü Denklem (3.5)'teki gibidir. Şekil 3.5.'te doğrusal fonksiyonunun ve türevinin grafiği verilmektedir.

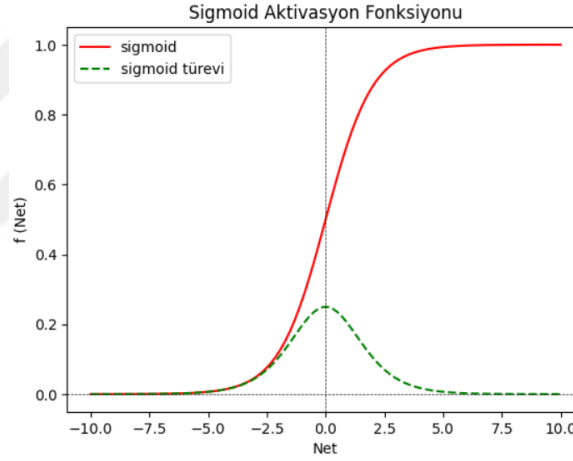
$$f(Net) = Net, (-\infty, +\infty) \quad (3.5)$$



Şekil 3.5. Doğrusal fonksiyon ve türevi

Sigmoid Fonksiyonu: Gelen Net girdiyi, 0 ile 1 arasındaki değerlere dönüştüren fonksiyondur (Koçak ve Şiray, 2021). Grafik bakımından S şekline benzemektedir. Sigmoid fonksiyonu türevi alınabildiğinden dolayı geri yayılım sırasında da kullanılabilen bir fonksiyondur. Bu yüzden çok tercih edilen aktivasyon fonksiyonlarından biridir. Ancak bir dezavantaj olarak türev grafiğinin uç noktalarına bakıldığında 0'a çok yakın değerler almasından dolayı öğrenme olayı minimum gerçekleşmektedir. Bu olaya gradyan kaybolması (vanishing gradient) denilmektedir (Irmak, 2022). Ancak en çok tercih edilmesinin sebeplerinden biri de ikili sınıflandırma problemleri dışındaki çoklu sınıflandırma problemleri için de kullanımına olanak sağlamasından dolayıdır. Bu çalışmada da bu sebepten ötürü sigmoid fonksiyonu tercih edilmiştir. Sigmoid fonksiyonunun formülü Denklem (3.6)'daki gibidir. Şekil 3.6.'da sigmoid fonksiyonunun ve türevinin grafiği verilmektedir.

$$f(Net) = \frac{1}{1 + e^{-Net}}, (0,1) \quad (3.6)$$

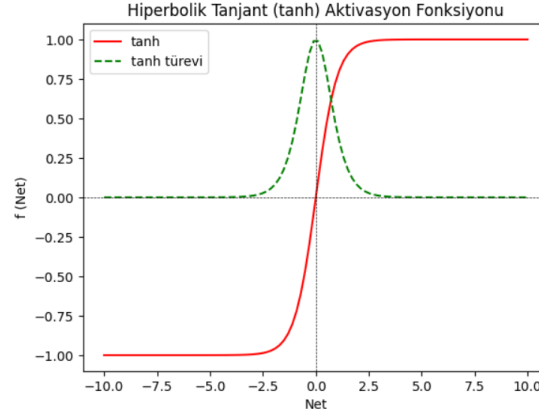


Şekil Error! No text of specified style in document.6. Sigmoid fonksiyonu ve türevi

Hiperbolik Tanjant Fonksiyonu: Gelen Net girdiyi, -1 ile 1 arasındaki değerlere dönüştüren fonksiyondur. Sigmoid fonksiyonundan farklı olarak türevi daha dik bir fonksiyondur ve negatif değerlere karşılık negatif değerler üretebilmektedir. Türevinin daha dik olması daha geniş bir aralıkta daha çok değer alabileceği ve öğrenmenin daha hızlı olacağı anlamına gelmektedir. Ancak hiperbolik tanjant fonksiyonunda da türev uçlarının 0'a yakınlığından kaynaklı gradyan kaybolması problemi yaşanmaktadır (Irmak, 2022).

Hiperbolik tanjant fonksiyonunun formülü Denklem (3.7)'deki gibidir. Şekil 3.7.'de hiperbolik tanjant fonksiyonunun ve türevinin grafiği verilmektedir.

$$f(Net) = \tanh(Net) = \frac{e^{Net} - e^{-Net}}{e^{Net} + e^{-Net}}, (-1,1) \quad (3.7)$$

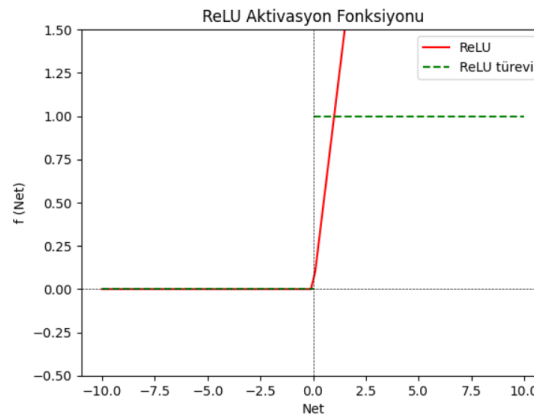


Şekil 3.7. Hiperbolik tanjant fonksiyonu ve türevi

ReLU Fonksiyonu: Negatif Net girdileri 0 değerine, 0 ve pozitif Net girdileri doğrudan çıktı değerine atayan fonksiyona ReLU (Rectified Linear Unit, Doğrultulmuş Doğrusal Birim) fonksiyonu denir. Hızı ve sadeliğinden dolayı özellikle derin öğrenme gibi konularda çok tercih edilmektedir. Negatif Net girdileri 0 değerine atadığından dolayı negatif girdileri kullanan nöronlara ölü nöron denmektedir. Bu yüzden negatif Net girdilere 0 değeri yerine başka değerler atayan LReLU gibi çeşitli yeni aktivasyon fonksiyonları türetilmiştir (Koçak, 2022). ReLU aktivasyon fonksiyonu $[0, \infty)$ arasında değer almaktadır ve türevlenebilmektedir.

ReLU fonksiyonunun formülü Denklem (3.8)'deki gibidir. Şekil 3.8.'de ReLU fonksiyonunun ve türevinin grafiği verilmektedir.

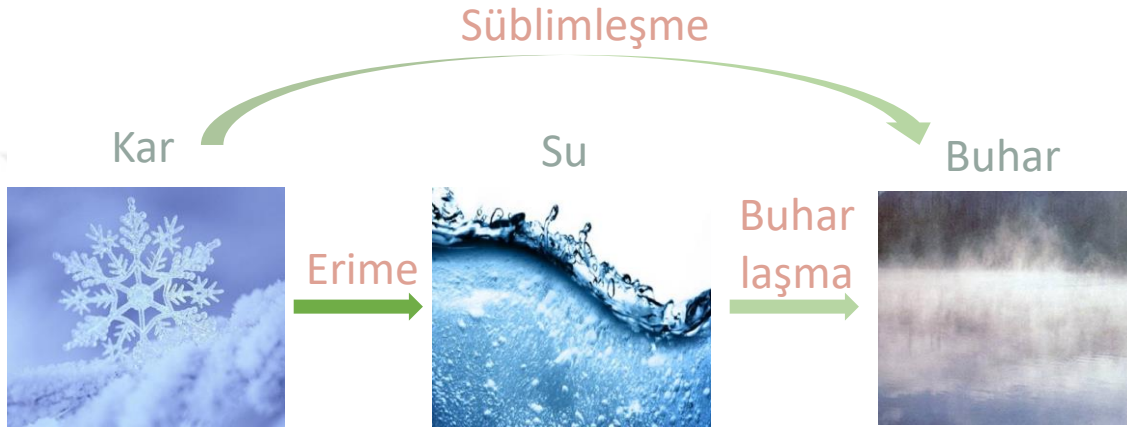
$$f(Net) = \begin{cases} 0, & Net < 0 \\ Net, & Net \geq 0 \end{cases}, [0, \infty) \quad (3.8)$$



Şekil 3.8. ReLU fonksiyonu ve türevi

3.2. SAO

Kar Erime Optimizasyonu (SAO) algoritması, 2023 yılında önerilmiş yeni bir metasezgisel algoritmadır (Deng ve Liu, 2023). SAO; karın suya, suyun da buhara dönüşmesi olayından ve karın doğrudan süblimleşme yoluyla buhara dönüşmesi olayından ilham alınarak geliştirilen doğa esinli bir algoritmadır. Süblimleşme, katı maddelerin ısı ile eriyip sıvı hale geçmeden gaz hale geçmesi anlamına gelmektedir. Şekil 3.9.'da, bahsedilen doğa olayları görsel olarak sunulmuştur (Deng ve Liu, 2023).



Şekil 3.9. Karın doğadaki halleri

Metasezgisel algoritmalarındaki en büyük eksikliklerden biri algoritmaların adımlarından biri olan keşif (exploration) ve sömürü (exploitation) arası oluşan dengesizlik halidir. SAO bu iki durumu dengelemek için oluşturulmuş bir algoritmadır (Deng ve Liu, 2023). SAO algoritmasının başlıca dört aşaması vardır. Bunlar; başlatma aşaması, keşif aşaması, sömürü aşaması ve ikili popülasyon oluşturma aşamasıdır.

Başlatma Aşaması: Popülasyon, Denklem (3.9) ile arama uzayı sınırları içerisinde rastgele başlatılır.

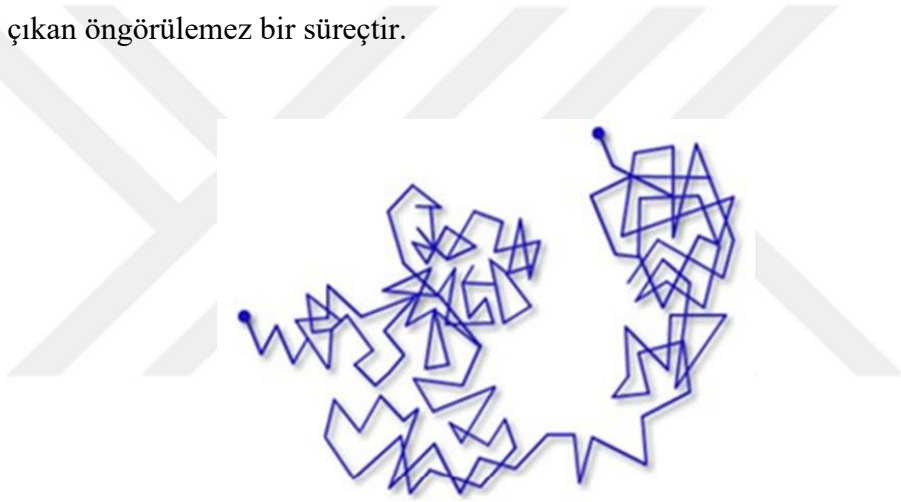
$$Z = L + \theta \times (U - L) = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,Dim-1} & Z_{1,Dim} \\ Z_{2,1} & Z_{2,2} & \cdots & Z_{2,Dim-1} & Z_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Z_{N-1,1} & Z_{N-1,2} & \cdots & Z_{N-1,Dim-1} & Z_{N-1,Dim} \\ Z_{N,1} & Z_{N,2} & \cdots & Z_{N,Dim-1} & Z_{N,Dim} \end{bmatrix}_{N \times Dim} \quad (3.9)$$

Burada Z , popülasyonu temsil eden bir matristir. N popülasyondaki ajan sayısını, Dim problemin boyutunu, L arama uzayının alt sınır değerini, U arama uzayının üst sınır değerini ve θ ise $[0, 1]$ aralığında rastgele seçilen bir sayıyı ifade eder.

Keşif Aşaması: Kar veya su buharlaştığında düzensiz bir şekilde dağılmaya başlar. Denklem (3.10)'da verilen Brownian hareketi bu düzensiz dağılımı taklit eder.

$$f_{BM}(x; 0,1) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{x^2}{2}\right) \quad (3.10)$$

Şekil 3.10.'da verilen Brownian hareketi, rastgele bir hareket türüdür. Özellikle sıvı ve gazlarda bulunan parçacıkların, çevre moleküllerle sürekli çarpışması sonucunda ortaya çıkan öngörülemez bir süreçtir.



Şekil Error! No text of specified style in document..10. Brownian hareketi (Mörters ve Peres, 2010)

Popülasyondaki arama ajanlarının konumları Brownian hareketinden faydalanılarak Denklem (3.11)'deki gibi güncellenir.

$$Z_i(t+1) = Elite(t) + BM_i(t) \otimes (\theta_1 \times (G(t) - Z_i(t)) + (1 - \theta_1) \times (\bar{Z}(t) - Z_i(t))) \quad (3.11)$$

Burada, $i = 1, 2, \dots, N$ konumu güncellenen ajanın indisini, t o anki iterasyonu, $BM_i(t)$ Brownian hareketini, \otimes girdilerin çarpımını, θ_1 $[0,1]$ aralığında rastgele oluşturulan bir sayıyı, $G(t)$ o ana kadar elde edilen en iyi çözümü, $\bar{Z}(t)$ popülasyonun sütun bazında ağırlıklarının ortalamasını içeren vektörü temsil etmektedir. Denklem

(3.12)'deki gibi hesaplanmaktadır. $\theta_1 \times (G(t) - Z_i(t))$ ve $(1 - \theta_1) \times (\bar{Z}(t) - Z_i(t))$ ifadeleri bireyler arasındaki etkileşimi ifade etmektedir. θ_1 , mevcut en iyi bireye doğru hareketi ve en iyi uygunluğa sahip $N/2$ bireyin merkez konumuna doğru hareketi kontrol eder. En iyi uygunluğa sahip üç birey ve en iyi uygunluğa sahip $N/2$ bireyin konumlarının ortalaması $Z_c(t)$, *EliteHavuzu* adı verilen bir matriste tutulur. $Elite(t)$, *EliteHavuzu* içinden rastgele seçilen bir çözümü ifade eder. Bu şekilde elit bireyler de konum belirlemede etkin rol oynarlar. Bu işlem, Denklem (3.13)'te gösterilmiştir.

$$\bar{Z}(t) = \frac{1}{N} \sum_{i=1}^N Z_i(t), \quad (3.12)$$

$$Elite(t) \in [G(t), Z_{second}(t), Z_{third}(t), Z_c(t)], \quad (3.13)$$

$Z_{second}(t)$, mevcut popülasyondaki ikinci en iyi bireyi, $Z_{third}(t)$ ise mevcut popülasyondaki üçüncü en iyi bireyi temsil etmektedir. $Z_c(t)$, Denklem (3.14)'deki gibi hesaplanmaktadır.

$$Z_c(t) = \frac{1}{N_1} \sum_{i=1}^{N_1} Z_i(t), \quad (3.14)$$

N_1 , en iyi uygunluğa sahip $N/2$ birey (uygunluk değerleri ilk %50'de yer alan bireylerin) sayısını temsil etmektedir. $Z_i(t)$, i . en iyi birey anlamına gelmektedir.

Sömürü Aşaması: Karın su haline geçme davranışını taklit eder. Bu aşamada arama uzayında yayılmak yerine, mevcuttaki en iyi çözümlerin yardımıyla yerel arama yapılması amaçlanır. Sömürü aşamasında popülasyon Denklem (3.15) ile güncellenir.

$$Z_i(t+1) = M \times G(t) + BM_i(t) \otimes (\theta_2 \times (G(t) - Z_i(t)) + (1 - \theta_2) \times (\bar{Z}(t) - Z_i(t))) \quad (3.15)$$

Burada, M kar erime oranını, θ_2 ise $[-1, 1]$ aralığındaki rastgele bir sayıyı temsil eder. $\theta_2 \times (G(t) - Z_i(t))$ ve $(1 - \theta_2) \times (\bar{Z}(t) - Z_i(t))$ terimleri yardımıyla umut vadeden bölgeleri kullanma ihtimalleri artmaktadır.

Kar erime oranı ise Denklem (3.16) ile ifade edilmektedir. T günlük ortalama sıcaklığı, T_1 , genelde 0 olarak seçilen temel sıcaklığı ifade eder.

$$M = DDF \times (T - T_1), \quad (3.16)$$

Fark ifadesi T olarak alınırsa Denklem (3.17) elde edilir.

$$M = DDF \times T, \quad (3.17)$$

DDF; 0,35 ile 0,6 arasında değişen derece-gün faktörünü temsil eder. Her yinelemede güncellenir ve Denklem (3.18)'deki gibi hesaplanır.

$$DDF = 0.35 + 0.25 \times \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1}, \quad (3.18)$$

$$M = \left(0.35 + 0.25 \times \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1}\right) \times T(t), T(t) = e^{\frac{-t}{t_{max}}} \quad (3.19)$$

Denklem (3.19)'da t_{max} sonlandırma kriterini, e Euler sayısını temsil etmektedir.

İkili Popülasyon Oluşturma Aşaması: Bu aşamada keşif ve sömürü arasında denge oluşturmak amaçlanmıştır. Su haline gelen karın bir kısmı arama esnasında buharlaşabilir ve merkezîyetçiliğini kaybedebilir. Bu dengenin sağlanması için ikili popülasyon yapısı oluşturulmuştur. Popülasyon arama sürecinin başında eşit sayıda arama ajanından oluşacak şekilde rastgele iki alt popülasyona ayrılır. Alt popülasyonlardan ilki ($index_a$) keşiften, ikincisi ($index_b$) sömürüden sorumludur. Her iterasyonda ikinci alt popülasyonun ajan sayısı birer azaltılırken, ilkinin ajan sayısı birer artırılır. Bu mekanizmanın sözde kodu Şekil 3.11.'de sunulmuştur.

İkili popülasyon oluşturma mekanizması

N: popülasyondaki ajan sayısı

Başlatma: $t = 0, t_{max}, N_a = N_b = N/2$

While ($t < t_{max}$)

If $N_a < N$

$N_a = N_a + 1$

$N_b = N_b - 1$

End if

$t = t + 1$

End while

Şekil 3.11. İkili popülasyon oluşturma mekanizması

Keşif ve sömürü aşamalarındaki konum güncelleme denklemlerinin bir arada gösterilmiş hali (tam konum güncelleme denklemi) Denklem (3.20)'de verilmiştir.

$$Z_i(t+1) = \begin{cases} Elite(t) + BM_i(t) \otimes (\theta_1 \times (G(t) - Z_i(t)) \\ \quad + (1 - \theta_1) \times (\bar{Z}(t) - Z_i(t))), i \in index_a \\ M \times G(t) + BM_i(t) \otimes (\theta_2 \times (G(t) - Z_i(t)) \\ \quad + (1 - \theta_2) \times (\bar{Z}(t) - Z_i(t))), i \in index_b \end{cases} \quad (3.20)$$

Tüm popülasyon bir konum matrisidir. Dolayısıyla Denklem (3.20)'deki $index_a$ ve $index_b$, tüm konum matrisinde alt popülasyonlardaki bireylerin satır numaralarını içeren bir dizi indeksi belirtir. SAO algoritmasının sözde kodu Şekil 3.12.'de sunulmuştur.

Başlangıç için sürüdeki birey sayısı, ağırlık, tolerans değeri belirlenir:

$$Z_i (i = 1, 2, \dots, N), t = 0, t_{max}, N_a = N_b = \frac{N}{2}$$

Uygunlukları hesaplanır.

Mevcut en iyi birey kaydedilir.

While ($t < t_{max}$)

Kar erime hızı hesaplanır.

Tüm popülasyon rastgele iki alt popülasyona ayrılır.

ForEach birey

Denklem (3.20)' ye göre her ajanın konumu güncellenir.

End ForEach

Uygunluk fonksiyonu değerlendirme yapılır.

En iyi ajan güncellenir.

End While

En iyi ajan alınır.

Şekil 3.12. SAO algoritmasının sözde kodu (Deng ve Liu, 2023)

3.2.1. SAO algoritmasının avantaj ve dezavantajları

SAO algoritmasının içinde barındırdığı bazı avantaj ve dezavantajlar bulunmaktadır. Algoritmanın avantajları aşağıda belirtilmiştir (Pandya ve ark., 2024):

- Yerel minimuma takılma ihtimalini gevşetilmiş yakınsama mekanizması sayesinde daha aza indirmektedir.
- Sürekli, kesikli, tek amaçlı veya çok amaçlı gibi çeşitli problem türleri için uyarlanabildiğinden dolayı kullanıcı dostu bir algoritmadır.
- Keşif ve sömürü aşamalarını kolaylıkla birleştirerek hızlı yakınsama yapabilir.
- Kolay anlaşılabilir ve kolayca uygulanabilir bir yapıya sahiptir.

Yeni geliştirilen bir algoritma olmasından dolayı bünyesinde bazı dezavantajları da barındırmaktadır. Bunlar da şöyle sıralanabilir:

- Yüksek boyutlu veya çok modlu (birden fazla yerel optimuma sahip) problemlerde salınım hareketli yakınsama eğilimleri gösterebilmektedir.
- Aldatıcı optimum noktalara takılı kalabilmektedir.

Bu yüzden SAO algoritmasının geliştirilmesine yönelik çeşitli çalışmalar yapılmaktadır.

3.3. Deneysel Çalışma

Önerilen YSA-LevySAO modeli, 15 farklı sınıflandırma veri seti üzerinde Çizelge 3.3.'de verilen kriterlere göre çalıştırılmıştır.

Çizelge 3.3. Çalıştırma kriterleri

Popülasyon boyutu	50
Alt sınır-üst sınır	[-10,10]
Maksimum uygunluk hesaplama	20000
İterasyon sayısı	400
Çalıştırma sayısı	30

Ağın girdi kısmındaki nöron sayıları veri setindeki öznitelik sayısı ile belirlenmiştir. Ara katmandaki nöron sayıları ($2 \times \text{öznitelik sayısı} + 1$) denkleminde yararlanılarak belirlenmiştir. Çıktı katmanındaki nöron sayısı ise veri setindeki sınıf sayısına göre belirlenmiştir. Ağırlık değerleri [-10, 10] arama uzayında oluşturulmuştur.

Bu çalışmada ÇKA mimarisi üzerindeki ağırlıklar vektör olarak modellenmiştir. ÇKA mimarisi üzerindeki ağırlıkların vektör olarak modellenmesi Şekil 3.13.'te gösterilmiştir.

w_{11}	w_{12}	...	w_{nm}	w'_{11}	w'_{1k}	...	w'_{mk}	β_1	β_2	...	β'_k
----------	----------	-----	----------	-----------	-----------	-----	-----------	-----------	-----------	-----	------------

Şekil 3.13. ÇKA mimarisinde ağırlıkların vektör modellenmesi

Çalışma içerisinde 50 popülasyon bulunmakta olup 20000 maksimum uygunluk hesaplanmış (maxFES) ve 400 iterasyon için 30 kez çalıştırma yapılmıştır. Elde edilen sonuçlar sensitivity, specificity, precision ve F1-score isimli literatürde sıklıkla kullanılan 4 karşılaştırma metriği için ortalama, standart sapma ve OBS değerleri verilmiştir.

Önerilen model, 9 farklı hibrit YSA modeli ile ortalama başarı sırası, bire-bir başarı ve hata yakınsama performansı açısından adil karşılaştırma kriterlerine uygun olarak karşılaştırılmıştır. Ayrıca Wilcoxon işaretli sıralar testi ile önerilen modelin elde ettiği sonuçların istatistiksel açıdan anlamlı olup olmadığı da değerlendirilmiştir.

Çalışma Matlab yazılım dilinde kodlanmıştır. Kullanılan veri setleri ve karşılaştırma metrikleri ilgili konu başlıklarında daha detaylı anlatılmıştır.

3.4. Veri Setleri

Çalışmada UCI Machine Learning Repository (Markelle Kelly) ve literatürde (Qaddoura ve ark., 2021) yer alan aggregation, aniso, balance, ecoli, glass, iris, iris2D, liver, mouse, pathbased, seeds, smiley, varied, vertebral3 ve wine isimli 15 farklı sınıflandırma veri seti kullanılmıştır. Veri setleri %75 eğitim, %25 test şeklinde ayrılmıştır. Örnek sayısının 150'den az olması durumunda tüm veri seti hem eğitim hem de test aşamalarında kullanılmıştır (Mirjalili, 2015). Veri setleri hakkında detaylı bilgi Çizelge 3.4.'te sunulmuştur.

Çizelge 3.4. Veri setleri ve özellikleri

Veri Seti No	Veri Seti Adı	Örnek Sayısı	Eğitim Verisi	Test Verisi	Öznitelik Sayısı	Sınıf Sayısı
1	aggregation	788	594	194	2	7
2	aniso	1500	1125	375	2	3
3	balance	625	469	156	4	3
4	ecoli	336	252	84	7	7
5	glass	214	162	52	9	6
6	iris	150	150	0	4	3
7	iris2D	150	150	0	2	3
8	liver	345	259	86	6	3
9	mouse	490	368	122	2	3
10	pathbased	400	226	74	2	3
11	seeds	400	226	74	2	3
12	smiley	500	375	125	2	4
13	varied	1500	1125	375	2	3
14	vertebral3	310	233	77	6	3
15	wine	178	133	45	13	3

3.5. Performans Metrikleri

3.5.1. Karmaşıklık matrisi

Çalışmada, literatürde sıklıkla kullanılan performans ölçüm yöntemlerinden sensitivity, specificity, precision ve F1-score isimli metrikler kullanılmıştır. Bu metrikler, karmaşıklık matrisi adı verilen bir matris yardımıyla elde edilir. Karmaşıklık matrisi, modelin doğruluğunu, veri setlerinin sınıflandırma başarısını ölçmeye yarayan ve literatürde sıklıkla kullanılan bir performans ölçüm yöntemidir. Karmaşıklık matrisinde tahmin edilen sınıf etiketi ve gerçek sınıf etiketi üzerinden Şekil 3.14.'te verilen durumlara göre çeşitli hesaplamalar yapılır.

		Tahmin	
		Negatif	Pozitif
Gerçek Durum	Negatif	Doğru Negatif (True Negative, TN)	Yanlış Pozitif (False Pozitive, FP)
	Pozitif	Yanlış Negatif (False Negative, FN)	Doğru Pozitif (True Pozitive, TP)

Şekil Error! No text of specified style in document.14. Karmaşıklık matrisi

Tahmini sınıf pozitif ve gerçek sınıf pozitif ise doğru pozitif (True Positive, TP), tahmini sınıf negatif ve gerçek sınıf negatif ise doğru negatif (True Negative, TN),

tahmini sınıf pozitif ve gerçek sınıf negatif ise yanlış pozitif (False Positive FP), tahmini sınıf negatif ve gerçek sınıf pozitif ise yanlış negatif (False Negative, FN) olarak isimlendirilir. Veri setindeki her bir örnek için bu dört durumdan bir tanesi oluşur. Bu durumların oluşma sayılarına göre sensitivity, specificity, precision ve F1-score metrikleri Denklem (3.21)-Denklem (3.24)'deki gibi hesaplanır.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (3.21)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (3.22)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3.23)$$

$$\text{F1-score} = 2 * \frac{\text{Sensitivity} * \text{Precision}}{\text{Sensitivity} + \text{Precision}} \quad (3.24)$$

- Sensitivity, doğru pozitif değerlerin toplam gerçek pozitif değerlerine bölünmesi ile elde edilir. Pozitif sınıfa ait değerlerden doğru tahminlerin oranının ne olduğu hakkında bilgi verir.
- Specificity, gerçek negatif değerlerin toplam gerçek negatif değerlere bölünmesi ile elde edilir. Modelin gerçek negatif değerlerin ne kadarını doğru bir şekilde tahmin ettiğini gösterir.
- Precision, doğru pozitif değerlerin toplam tahmini pozitif değerlere bölünmesiyle elde edilir. Bu metrik, doğru tahminlerin ne oranda olduğu hakkında bilgi verir.
- F1-score, sensitivity ve precision metriğinin harmonik ortalamasıdır. Eşit dağılmayan veri setleri için hatalı modeller oluşturmanın önüne geçmesi bakımından önemli bir metriktir.

Metrikler [0,1] arasında değer alırlar ve 1'e yaklaştıkça metriklerin başarısı artmaktadır.

3.5.2. Ortalama karesel hata

YSA mimarisi eğitim aşamasında ileri yayılım yaptıktan sonra veri setindeki gerçek sınıf etiketleri ve tahmin edilen sınıf etiketlerine bakar. Buna göre her bir örnek için bir hata değeri hesaplar. Bu çalışmada, hata hesabı için Denklem (3.25)'te verilen ortalama karesel hata (MSE) yöntemi kullanılmıştır.

$$MSE = \frac{1}{r} \sum_{i=1}^r \sum_{j=1}^k (y_j - \hat{y}_j)^2 \quad (3.25)$$

Burada r eğitim setindeki örnek sayısını, k sınıf etiketi sayısını, y_j j. çıktı için beklenen değeri ve \hat{y}_j ise j. çıktı için YSA tarafından tahmin edilen değeri ifade etmektedir.



4. ÖNERİLEN YÖNTEM

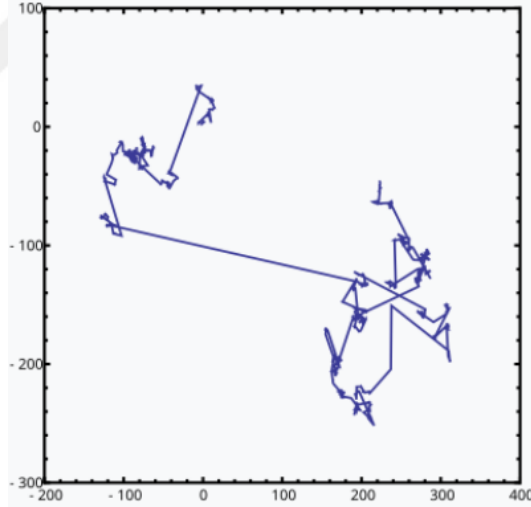
Bu tez çalışmasında;

- SAO algoritmasına Levy uçuşu tekniği uygulanarak LevySAO algoritması geliştirilmiştir.
- YSA'nın geriye yayılım fazında yerel minimuma takılma sorununu çözmek için ağırlık değerlerini, LevySAO algoritması ile güncelleyen YSA-LevySAO hibrit modeli önerilmiştir.

Bölüm 4.1.'de Levy uçuşu, Bölüm 4.2.'de LevySAO algoritması anlatılmıştır. Bölüm 4.3.'te ise önerilen YSA-LevySAO hibrit modeli hakkında detaylara yer verilmiştir.

4.1. Levy Uçuşu

Levy uçuşu, 1937 yılında Fransız matematikçi Paul Levy (Houssein ve ark., 2020; X.-S. Yang ve ark., 2013) tarafından geliştirilen Şekil 4.1.'de görüldüğü gibi bir tür rastgele yürüyüştür.



Şekil 4.1. Levy uçuşu (X.-S. Yang ve ark., 2013)

Matematik ve modellemede pek çok rastgele yürüyüş hareketi vardır. Bu hareketlerden farklı olarak Levy uçuşu, adım uzunluğunu bir güç yasasına göre belirlemektedir. Levy uçuşu, anormal yayılma olarak adlandırılan yayılmaları modellemek için kullanılır (Brown ve ark., 2007). Levy uçuşunun adım uzunluğu Denklem (4.1)'deki Levy dağılımına göre belirlenir.

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left(-\frac{\gamma}{2(s-\mu)}\right) \frac{1}{(s-\mu)^{\frac{3}{2}}}, & 0 < \mu < \infty \\ 0, & \text{diğer durumlarda} \end{cases} \quad (4.1)$$

Arılar, kuşlar, karıncalar, köpek balıkları, ton balıkları gibi pek çok hayvan türü yiyecek arama stratejilerinde Levy uçuşunu kullanmaktadır. Ayrıca Levy uçuşu yöntemi, metasezgisel algoritmaların yerel minimuma takılma problemine önemli ölçüde çözüm sunabilmektedir. Bu yüzden Levy uçuşu birçok metasezgisel algoritmanın performansını artırmak için de kullanılmıştır (Bhullar ve ark., 2022) (Jamal ve ark., 2023) (J. Gao ve ark., 2021) (Saji ve Barkatou, 2021) (Mohseni ve ark., 2021).

Denklem (4.1)'deki μ ve s sırasıyla iletim parametreleridir. Levy uçuşunun adım uzunluğu Denklem (4.2)'deki gibi hesaplanır.

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (4.2)$$

Burada u ve v değişkenleri normal bir dağılım izlemektedir ve β sabit bir parametredir (Y. Gao ve ark., 2023).

4.2. LevySAO

Orijinal SAO algoritmasının yerel minimuma takılma sorunu vardır. Bu sorunun üstesinden gelmek için Levy uçuşundan faydalanan yeni bir konum güncelleme mekanizması geliştirilmiştir. Bu mekanizmanın matematiksel modeli Denklem (4.3)'te verilmiştir.

$$Z_i(t+1) = Elite(t) + 0.01 * LF(1, Dim, 1) \quad (4.3)$$

Burada, LF Levy uçuşunu, Dim problemin boyut sayısını temsil eder.

Önerilen LevySAO algoritmasında, her iki konum güncelleme mekanizmasından da faydalanılmaktadır. Hangi mekanizmanın kullanılacağı $[0, 1]$ aralığında değer alan LevyProb adı verilen bir parametreye göre belirlenir. Her bir parçacık için $[0, 1]$ aralığında rastgele bir sayı üretilir. Üretilen sayı LevyProb parametresinden büyükse rastgele iki alt popülasyona bölünen bireylerden ilk grubun konumu orijinal SAO algoritmasının konum güncelleme mekanizmasına göre, değilse Levy uçuşundan faydalanan Denklem (4.3)'te gösterilen konum güncelleme mekanizması kullanılarak

konumlar güncellenir. LevyProb parametresi, LevySAO'nun Levy uçuşu oranını belirten önemli bir parametredir. Levy uçuşu tekniğinden hangi oranla yararlanılacağı bu parametre yardımıyla belirlenir.

Önerilen algoritmanın sözde kodu Şekil 4.2.'de, akış diyagramı ise Şekil 4.3.'te sunulmuştur.

Başlangıç için sürüdeki birey sayısı, ağırlık, tolerans değeri belirlenir:

$$Z_i (i = 1, 2, \dots, N), t = 0, t_{max}, N_a = N_b = \frac{N}{2}, levyProb = 0.9$$

Fitness (uygunluk) fonksiyonu değerlendirilmesi yapılır.

Mevcut en iyi birey kaydedilir.

While ($t < t_{max}$)

Kar erime hızı Denklem (3.19)'a göre hesaplanır.

Tüm popülasyon N_a ve N_b isimli rastgele iki alt popülasyona bölünür.

Foreach (birey)

[0,1] arasında rastgele sayı seçilir.

Bu sayı *levyProb* parametresinden büyükse rastgele iki alt popülasyona bölünen bireylerden ilk grubun konumu orijinal SAO algoritmasının konum güncelleme mekanizması olan Denklem (3.11)'e göre güncellenir, değil ise Levy uçuşundan faydalanan Denklem (4.3)'e göre güncellenir.

İkinci grubun konumu Denklem (3.15)'e göre güncellenir.

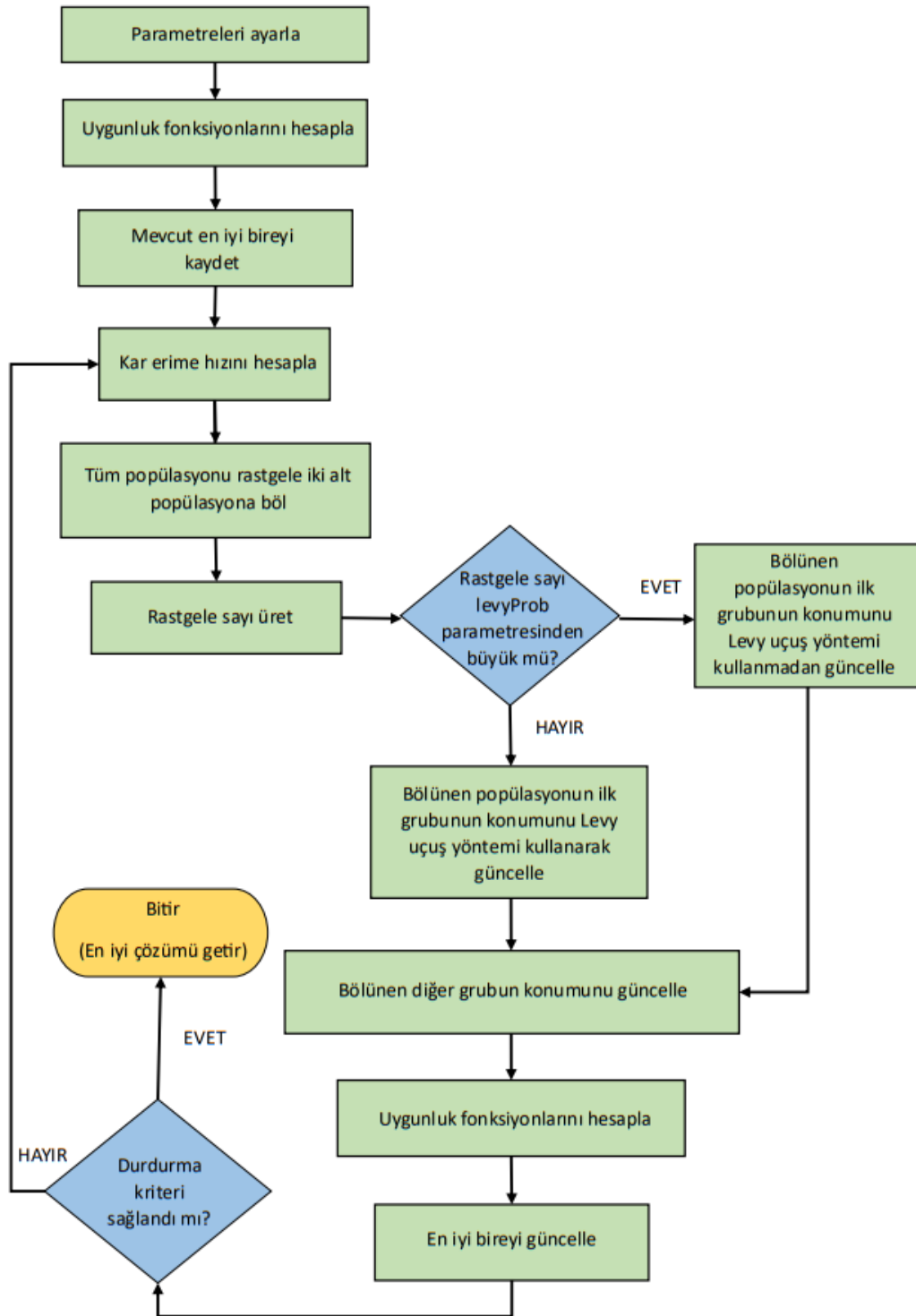
End Foreach

Fitness (uygunluk) fonksiyonu değerlendirilmesi yapılır.

En iyi birey güncellenir.

End While

Şekil 4.2. LevySAO algoritmasının sözde kodu



Şekil 4.3. LevySAO algoritmasının akış diyagramı

4.3. YSA-LevySAO

Önerilen YSA-LevySAO modelinde YSA'nın hatanın geri yayılım fazı yerine geliştirilen LevySAO algoritması kullanılmıştır. Önerilen model ile ilgili parametre ayarlaması ve performans sonuçları bölümün devamında verilmiştir.

4.3.1. YSA-LevySAO modelinde parametre ayarlaması

Önerilen YSA-LevySAO hibrit modelinde Levy uçuşundan yararlanma olasılığını ifade eden LevyProb parametresinin en uygun değerini belirleyebilmek için parametre ayarlama çalışmaları yapılmıştır. Aşağıda parametre ayarlaması adım adım açıklanarak ilgili çizelgelerde belirtilmiştir.

Bu aşamada, önerilen algoritmadaki "LevyProb" parametresinin en uygun değerini belirlemek için parametre ayarlama çalışması yapılmıştır. LevyProb değeri 0,1, 0,2, ..., 0,9, 1,0 olarak ayarlanmış ve toplamda 10 farklı LevySAO varyantı oluşturulmuştur. Her bir varyant ile YSA hibrit kullanılarak 10 farklı hibrit model geliştirilmiştir. Tüm modeller Çizelge 3.3.'te verilen kriterlere göre çalıştırılmış ve elde edilen metrik sonuçları Çizelge 4.1.-Çizelge 4.4.'te sunulmuştur. Çizelgelerdeki ort sütunu 30 çalıştırma sonucundaki ortalama metrik değerlerini, std sütunu standart sapmayı, sıra sütunu ise ort değerine göre modelin başarı sırasını ifade eder. Çizelgelerin en alt satırında ise her bir modelin tüm veri setlerindeki ortalama başarı sırası (OBS) verilmiştir.

Çizelge 4.1'de sensitivity metriği için elde edilen sonuçlar incelendiğinde, YSA-LevySAO09 varyantının liver ve wine veri setlerinde birinci olduğu ve OBS açısından da 4,133 değeri ile ilk sırayı aldığı gözlenmektedir. OBS açısından YSA-LevySAO01 varyantının 4,6 değeri ile 2., YSA-LevySAO02 varyantının 4,933 değeri ile 3., YSA-LevySAO07 varyantının 5,2 değeri ile 4., YSA-LevySAO04 varyantının 5,267 değeri ile 5., YSA-LevySAO01 varyantının 5,4 ile 6., YSA-LevySAO08 varyantının 5,867 değeri ile 7., YSA-LevySAO05 ve YSA-LevySAO06 varyantının 6,2 değeri ile 8., orijinal SAO algoritmasının olduğu YSA-SAO'nun 6,533 değeri ile 9. ve YSA-LevySAO03 varyantının 6,667 değeri ile 10. sırayı elde ettiği görülmüştür. Sonuç olarak sensitivity metriği için orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,533 OBS değeri ile 9. sırayı alırken, YSA-LevySAO09 varyantı 4,133 OBS değeri ile ilk sırada yer almıştır.

Çizelge 4.2'de specificity metriği için elde edilen sonuçlar incelendiğinde, YSA-LevySAO09 varyantının liver, vertebral3 ve wine veri setlerinde birinci olduğu ve OBS açısından da 3,8 değeri ile ilk sırayı aldığı gözlenmektedir. OBS açısından YSA-

LevySAO07 varyantının 4,4 değeri ile 2., YSA-LevySAO01 varyantı ve YSA-LevySAO04 varyantı 5,067 değeri ile 3., YSA-LevySAO02 varyantının 5,133 değeri ile 4., YSA-LevySAO08 varyantının 5,4 değeri ile 5., YSA-LevySAO1 varyantının 5,933 ile 6., YSA-LevySAO06 varyantının 6 değeri ile 7., YSA-LevySAO05 varyantının 6,067 değeri ile 8., orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,33 değeri ile 9. ve YSA-LevySAO03 varyantının 6,733 değeri ile 10. olduğu görülmüştür. Sonuç olarak specificity metriği için orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,33 OBS değeri ile 9. sırayı alırken, YSA-LevySAO09 varyantı 3,8 OBS değeri ile ilk sırada yer almıştır.

Çizelge 4.3.'te precision metriği için elde edilen sonuçlar incelendiğinde, YSA-LevySAO09 varyantının wine veri setinde birinci olduğu ve OBS açısından da 4,933 değeri ile ilk sırayı aldığı gözlenmektedir. OBS açısından YSA-LevySAO1 varyantının 5 değeri ile 2., YSA-LevySAO04 varyantı 5,067 değeri ile 3., YSA-LevySAO08 varyantının 5,133 değeri ile 4., YSA-LevySAO02 varyantı ve YSA-LevySAO07 varyantı 5,267 değeri ile 5., YSA-LevySAO01 varyantının 5,8 ile 6., YSA-LevySAO06 varyantının 5,867 değeri ile 7., YSA-LevySAO05 varyantının 6,2 değeri ile 8., orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,533 değeri ile 9. ve YSA-LevySAO03 varyantının 7 değeri ile 10. olduğu görülmüştür. Sonuç olarak precision metriği için orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,533 OBS değeri ile 9. sırayı alırken, YSA-LevySAO09 varyantı 4,933 OBS değeri ile ilk sırada yer almıştır.

Çizelge 4.4'te F1-score metriği için elde edilen sonuçlar incelendiğinde, YSA-LevySAO09 varyantının liver ve wine veri setinde birinci olduğu ve OBS açısından da 4,6 değeri ile ilk sırayı aldığı gözlenmektedir. OBS açısından YSA-LevySAO07 varyantının 4,733 değeri ile 2., YSA-LevySAO01 varyantı 4,933 değeri ile 3., YSA-LevySAO099 varyantının 5,133 değeri ile 4., YSA-LevySAO02 varyantı 5,267 değeri ile 5., YSA-LevySAO04 varyantının 5,333 ile 6., YSA-LevySAO08 varyantının 5,467 değeri ile 7., YSA-LevySAO06 varyantının 6,2 değeri ile 8., YSA-LevySAO05 varyantının 6,333 değeri ile 9., orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,867 değeri ile 10. ve YSA-LevySAO03 varyantının 7,2 değeri ile 11. olduğu görülmüştür. Sonuç olarak F1-score metriği için orijinal SAO algoritmasını kullanan YSA-SAO modeli 6,867 OBS değeri ile 10. sırayı alırken, YSA-LevySAO09 varyantı 4,6 OBS değeri ile ilk sırada yer almıştır.

Çizelge 4.1.-Çizelge 4.4.'te sonuçlar birlikte değerlendirildiğinde, YSA-LevySAO09 modeli varyantının 4 metriğin tamamında en iyi OBS değerlerini elde ederek

diğer varyantlardan ve orijinal SAO algoritmasını kullanan YSA-SAO modelinden daha üstün bir performansa sahip olduğu gözlenmiştir. Bu sebeple, LevyProb parametresi 0,9 olarak ayarlanmış olan LevySAO09 varyantı çalışmada önerilen algoritma olarak belirlenmiştir. Çalışmanın geri kalan kısmında LevySAO09 algoritması LevySAO olarak ifade edilecektir.



Çizelge 4.1. YSA-LevySAO modeli varyantlarının sensitivity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-SAO			YSA-LevySAO01			YSA-LevySAO02			YSA-LevySAO03			YSA-LevySAO04			YSA-LevySAO05		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,355	0,122	11	0,521	0,145	7	0,53	0,132	5	0,473	0,155	10	0,549	0,134	3	0,49	0,149	9
aniso	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
balance	0,684	0,084	5	0,684	0,077	4	0,69	0,085	2	0,663	0,053	11	0,666	0,074	10	0,691	0,066	1
ecoli	0,339	0,067	11	0,425	0,084	10	0,442	0,087	9	0,45	0,105	8	0,468	0,081	7	0,493	0,097	4
glass	0,231	0,074	11	0,354	0,08	10	0,399	0,065	8	0,412	0,082	4	0,411	0,09	5	0,41	0,055	6
iris	0,989	0,006	3	0,99	0,005	1	0,98	0,059	4	0,968	0,082	11	0,98	0,059	5	0,979	0,059	7
iris2D	0,97	0,009	1	0,967	0,008	3	0,967	0,008	2	0,946	0,076	10	0,964	0,007	4	0,942	0,075	11
liver	0,396	0,033	11	0,41	0,026	4	0,41	0,025	3	0,412	0,034	2	0,408	0,032	8	0,406	0,027	9
mouse	0,989	0,061	8	0,989	0,061	7	1	0,001	3	0,989	0,061	6	1	0,001	2	0,999	0,003	4
pathbased	0,6	0,076	2	0,602	0,06	1	0,585	0,092	6	0,598	0,032	3	0,58	0,049	8	0,591	0,051	4
seeds	0,728	0,07	10	0,762	0,054	4	0,772	0,076	3	0,748	0,059	9	0,752	0,085	7	0,749	0,078	8
smiley	0,43	0,085	1	0,389	0,094	4	0,352	0,097	10	0,411	0,075	2	0,39	0,088	3	0,379	0,107	7
varied	0,982	0,003	1	0,982	0,002	3	0,982	0,002	4	0,971	0,058	9	0,982	0,002	2	0,982	0,003	2
vertebral3	0,7	0,065	11	0,741	0,066	1	0,72	0,062	6	0,718	0,062	8	0,715	0,075	9	0,713	0,064	10
wine	0,683	0,077	11	0,833	0,144	9	0,834	0,143	8	0,838	0,134	6	0,86	0,134	5	0,805	0,147	10
OBS	6,533 (9)			4,6 (2)			4,933 (3)			6,667 (10)			5,267 (5)			6,2 (8)		
Veri Setleri	YSA-LevySAO06			YSA-LevySAO07			YSA-LevySAO08			YSA-LevySAO09			YSA-LevySAO1					
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra			
aggregation	0,555	0,153	2	0,498	0,145	8	0,564	0,141	1	0,535	0,148	4	0,528	0,111	6			
aniso	1	0	1	1	0	1	1	0	1	0,989	0,061	2	1	0	1			
balance	0,678	0,079	8	0,683	0,057	6	0,679	0,077	7	0,669	0,057	9	0,687	0,076	3			
ecoli	0,482	0,085	5	0,482	0,077	6	0,513	0,081	2	0,512	0,079	3	0,524	0,072	1			
glass	0,392	0,091	9	0,417	0,098	3	0,407	0,091	7	0,423	0,063	2	0,431	0,072	1			
iris	0,979	0,059	8	0,979	0,06	6	0,978	0,059	9	0,989	0,005	2	0,977	0,059	10			
iris2D	0,953	0,054	9	0,962	0,016	6	0,962	0,006	5	0,959	0,008	8	0,961	0,004	7			
liver	0,405	0,029	10	0,408	0,03	7	0,409	0,028	6	0,413	0,027	1	0,41	0,023	5			
mouse	0,966	0,102	10	1	0	1	0,989	0,061	6	0,999	0,003	5	0,988	0,061	9			
pathbased	0,589	0,047	5	0,579	0,101	9	0,569	0,04	11	0,582	0,083	7	0,576	0,041	10			
seeds	0,753	0,086	6	0,782	0,039	1	0,717	0,103	11	0,774	0,073	2	0,756	0,082	5			
smiley	0,373	0,112	8	0,383	0,09	6	0,364	0,096	9	0,348	0,093	11	0,385	0,091	5			
varied	0,981	0,002	5	0,98	0,003	8	0,981	0,002	6	0,982	0,002	3	0,981	0,003	7			
vertebral3	0,725	0,071	5	0,731	0,053	3	0,729	0,065	4	0,733	0,061	2	0,719	0,093	7			
wine	0,881	0,137	2	0,838	0,168	7	0,874	0,14	3	0,908	0,099	1	0,867	0,123	4			
OBS	6,2 (8)			5,2 (4)			5,867 (7)			4,133 (1)			5,4 (6)					

Çizelge 4.2. YSA-LevySAO modeli varyantlarının specificity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-SAO			YSA-LevySAO01			YSA-LevySAO02			YSA-LevySAO03			YSA-LevySAO04			YSA-LevySAO05		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,911	0,026	11	0,937	0,026	9	0,944	0,025	6	0,936	0,031	10	0,952	0,023	1	0,937	0,031	8
aniso	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
balance	0,919	0,017	10	0,921	0,019	9	0,922	0,019	7	0,917	0,015	11	0,921	0,021	8	0,93	0,019	2
ecoli	0,945	0,013	11	0,955	0,01	10	0,955	0,013	9	0,96	0,011	8	0,963	0,01	6	0,961	0,012	7
glass	0,849	0,015	11	0,885	0,024	10	0,898	0,019	7	0,9	0,022	6	0,898	0,023	8	0,901	0,019	3
iris	0,995	0,003	2	0,995	0,003	1	0,99	0,03	3	0,984	0,041	9	0,99	0,03	4	0,989	0,03	6
iris2D	0,985	0,004	1	0,983	0,004	3	0,983	0,004	2	0,973	0,038	10	0,982	0,004	4	0,971	0,038	11
liver	0,729	0,033	11	0,743	0,026	4	0,743	0,025	3	0,746	0,034	2	0,741	0,032	8	0,739	0,027	9
mouse	0,994	0,03	7	0,994	0,03	8	1	0,001	3	0,994	0,03	6	1	0,001	2	1	0,002	4
pathbased	0,796	0,04	1	0,795	0,032	2	0,787	0,048	6	0,792	0,015	3	0,784	0,024	8	0,79	0,026	4
seeds	0,864	0,035	10	0,881	0,027	4	0,886	0,038	3	0,874	0,029	9	0,876	0,043	7	0,875	0,039	8
smiley	0,845	0,044	1	0,824	0,053	3	0,807	0,056	9	0,834	0,044	2	0,823	0,05	4	0,817	0,056	6
varied	0,991	0,001	1	0,991	0,001	2	0,991	0,001	4	0,985	0,029	9	0,991	0,001	2	0,991	0,001	1
vertebral3	0,901	0,015	6	0,905	0,022	2	0,901	0,018	5	0,9	0,019	8	0,899	0,028	9	0,897	0,023	11
wine	0,855	0,031	11	0,922	0,063	8	0,922	0,065	9	0,923	0,06	7	0,935	0,058	4	0,908	0,066	10
OBS	6,33 (9)			5,067 (3)			5,133 (4)			6,733 (10)			5,067 (3)			6,067 (8)		
Veri Setleri	YSA-LevySAO06			YSA-LevySAO07			YSA-LevySAO08			YSA-LevySAO09			YSA-LevySAO1					
	ort	std	sıra	ort	Std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra			
aggregation	0,949	0,023	2	0,94	0,026	7	0,947	0,024	4	0,949	0,023	3	0,946	0,02	5			
aniso	1	0	1	1	0	1	1	0	1	0,994	0,03	2	1	0	1			
balance	0,924	0,021	6	0,928	0,016	3	0,93	0,019	1	0,925	0,013	5	0,926	0,019	4			
ecoli	0,963	0,007	5	0,964	0,007	3	0,966	0,008	2	0,963	0,012	4	0,967	0,007	1			
glass	0,896	0,027	9	0,902	0,024	1	0,901	0,024	4	0,901	0,02	2	0,9	0,018	5			
iris	0,989	0,03	5	0,989	0,03	5	0,989	0,03	7	0,995	0,002	2	0,989	0,029	8			
iris2D	0,976	0,027	9	0,981	0,008	6	0,981	0,003	5	0,979	0,004	8	0,981	0,002	7			
liver	0,738	0,029	10	0,742	0,03	7	0,742	0,028	6	0,746	0,027	1	0,743	0,023	5			
mouse	0,983	0,051	10	1	0	1	0,994	0,03	6	0,999	0,002	5	0,994	0,03	9			
pathbased	0,788	0,024	5	0,783	0,051	9	0,778	0,019	11	0,784	0,042	7	0,781	0,02	10			
seeds	0,876	0,043	6	0,891	0,019	1	0,858	0,051	11	0,887	0,036	2	0,878	0,041	5			
smiley	0,812	0,055	8	0,821	0,051	5	0,806	0,054	10	0,804	0,053	11	0,814	0,051	7			
varied	0,991	0,001	5	0,99	0,001	8	0,991	0,001	6	0,991	0,001	3	0,99	0,001	7			
vertebral3	0,9	0,023	7	0,903	0,018	3	0,902	0,019	4	0,905	0,019	1	0,899	0,048	10			
wine	0,944	0,061	2	0,924	0,078	6	0,941	0,062	3	0,955	0,045	1	0,934	0,059	5			
OBS	6 (7)			4,4 (2)			5,4 (5)			3,8 (1)			5,933 (6)					

Çizelge 4.3. YSA-LevySAO modeli varyantlarının precision metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-SAO			YSA-LevySAO01			YSA-LevySAO02			YSA-LevySAO03			YSA-LevySAO04			YSA-LevySAO05		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,302	0,111	11	0,463	0,155	7	0,48	0,138	4	0,431	0,173	9	0,501	0,149	3	0,43	0,146	10
aniso	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
balance	0,67	0,1	8	0,663	0,091	9	0,676	0,101	6	0,656	0,084	11	0,657	0,093	10	0,703	0,105	1
ecoli	0,286	0,082	11	0,386	0,096	10	0,395	0,12	9	0,427	0,119	8	0,464	0,107	7	0,47	0,123	6
glass	0,145	0,098	11	0,302	0,11	10	0,354	0,088	9	0,363	0,091	8	0,384	0,126	2	0,373	0,08	5
iris	0,989	0,006	2	0,99	0,005	1	0,975	0,09	4	0,957	0,124	11	0,973	0,099	6	0,971	0,1	9
iris2D	0,97	0,009	1	0,967	0,008	2	0,967	0,008	3	0,933	0,126	10	0,964	0,007	4	0,931	0,118	11
liver	0,432	0,038	10	0,435	0,028	9	0,441	0,027	3	0,439	0,046	5	0,436	0,034	8	0,437	0,03	7
mouse	0,985	0,076	8	0,985	0,076	7	0,999	0,003	3	0,986	0,076	6	1	0,002	2	0,999	0,004	4
pathbased	0,647	0,116	1	0,567	0,109	6	0,589	0,169	5	0,605	0,136	2	0,596	0,126	4	0,599	0,128	3
seeds	0,745	0,134	10	0,801	0,076	4	0,802	0,11	2	0,785	0,092	6	0,778	0,127	8	0,778	0,127	7
smiley	0,345	0,065	1	0,312	0,081	6	0,285	0,089	10	0,336	0,067	2	0,317	0,077	3	0,306	0,094	7
varied	0,982	0,003	1	0,982	0,002	2	0,982	0,002	6	0,965	0,089	11	0,982	0,002	4	0,982	0,003	5
vertebral3	0,656	0,124	11	0,718	0,123	4	0,711	0,095	6	0,708	0,099	8	0,708	0,14	9	0,711	0,086	7
wine	0,591	0,136	11	0,785	0,202	9	0,795	0,205	8	0,8	0,198	7	0,833	0,189	5	0,757	0,209	10
OBS	6,533 (9)			5,8 (6)			5,267 (5)			7 (10)			5,067 (3)			6,2 (8)		
Veri Setleri	YSA-LevySAO06			YSA-LevySAO07			YSA-LevySAO08			YSA-LevySAO09			YSA-LevySAO1					
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra			
aggregation	0,52	0,173	2	0,46	0,154	8	0,526	0,169	1	0,476	0,167	5	0,469	0,129	6			
aniso	1	0	1	1	0	1	1	0	1	0,983	0,091	2	1	0	1			
balance	0,681	0,094	2	0,671	0,073	7	0,68	0,113	3	0,68	0,109	5	0,68	0,096	4			
ecoli	0,482	0,101	5	0,486	0,079	4	0,513	0,098	2	0,49	0,115	3	0,534	0,064	1			
glass	0,366	0,099	7	0,375	0,105	4	0,386	0,128	1	0,367	0,075	6	0,382	0,093	3			
iris	0,972	0,094	7	0,973	0,091	5	0,972	0,097	8	0,989	0,005	3	0,97	0,096	10			
iris2D	0,946	0,089	9	0,962	0,015	6	0,962	0,006	5	0,959	0,007	8	0,961	0,004	7			
liver	0,441	0,039	2	0,429	0,05	11	0,438	0,028	6	0,44	0,032	4	0,443	0,021	1			
mouse	0,957	0,128	10	1	0	1	0,986	0,076	6	0,999	0,004	5	0,984	0,076	9			
pathbased	0,566	0,127	7	0,558	0,173	9	0,528	0,109	11	0,552	0,166	10	0,56	0,111	8			
seeds	0,776	0,129	9	0,823	0,026	1	0,727	0,175	11	0,801	0,113	3	0,791	0,114	5			
smiley	0,299	0,092	8	0,313	0,081	5	0,292	0,085	9	0,284	0,088	11	0,313	0,083	4			
varied	0,982	0,002	7	0,98	0,003	10	0,982	0,002	8	0,982	0,002	3	0,981	0,003	9			
vertebral3	0,706	0,126	10	0,726	0,094	1	0,724	0,098	2	0,718	0,108	5	0,723	0,136	3			
wine	0,863	0,178	2	0,813	0,225	6	0,847	0,188	3	0,91	0,115	1	0,837	0,179	4			
OBS	5,867 (7)			5,267 (5)			5,133 (4)			4,933 (1)			5 (2)					

Çizelge 4.4. YSA-LevySAO modeli varyantlarının F1-score metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-SAO			YSA-LevySAO01			YSA-LevySAO02			YSA-LevySAO03			YSA-LevySAO04			YSA-LevySAO05		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,294	0,12	11	0,461	0,153	7	0,475	0,137	6	0,422	0,162	10	0,498	0,147	3	0,429	0,154	9
aniso	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1
balance	0,663	0,083	7	0,654	0,068	8	0,666	0,08	6	0,648	0,064	11	0,651	0,078	10	0,68	0,074	1
ecoli	0,302	0,072	11	0,396	0,085	10	0,407	0,105	9	0,431	0,108	8	0,457	0,091	7	0,47	0,109	6
glass	0,158	0,077	11	0,297	0,093	10	0,359	0,072	8	0,365	0,088	7	0,371	0,1	5	0,37	0,065	6
iris	0,989	0,006	3	0,99	0,005	1	0,977	0,08	4	0,961	0,11	11	0,976	0,083	5	0,974	0,083	8
iris2D	0,97	0,009	1	0,967	0,008	3	0,967	0,008	2	0,938	0,107	10	0,964	0,007	4	0,935	0,104	11
liver	0,378	0,049	11	0,404	0,034	2	0,402	0,034	3	0,401	0,05	4	0,398	0,045	8	0,396	0,039	9
mouse	0,987	0,07	8	0,987	0,07	7	0,999	0,002	3	0,987	0,07	6	1	0,002	2	0,999	0,004	4
pathbased	0,525	0,104	1	0,525	0,082	2	0,503	0,108	4	0,5	0,033	6	0,491	0,038	9	0,503	0,066	5
seeds	0,715	0,105	10	0,763	0,066	4	0,77	0,096	3	0,747	0,08	7	0,747	0,113	6	0,742	0,105	9
smiley	0,378	0,069	1	0,343	0,084	4	0,314	0,092	10	0,367	0,069	2	0,348	0,08	3	0,336	0,098	7
varied	0,981	0,003	5	0,982	0,002	3	0,981	0,002	6	0,967	0,079	11	0,982	0,002	2	0,982	0,003	1
vertebral3	0,657	0,087	11	0,709	0,098	3	0,697	0,077	6	0,697	0,077	7	0,689	0,101	10	0,693	0,077	8
wine	0,614	0,097	11	0,799	0,179	9	0,801	0,181	8	0,805	0,172	7	0,835	0,168	5	0,763	0,184	10
OBS	6,867 (10)			4,933 (3)			5,267 (5)			7,2 (11)			5,333 (6)			6,333 (9)		
Veri Setleri	YSA-LevySAO06			YSA-LevySAO07			YSA-LevySAO08			YSA-LevySAO09			YSA-LevySAO1					
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra			
aggregation	0,51	0,164	1	0,453	0,151	8	0,509	0,153	2	0,485	0,155	4	0,475	0,119	5			
aniso	1	0	1	1	0	1	1	0	1	0,985	0,081	2	1	0	1			
balance	0,666	0,08	5	0,666	0,06	4	0,668	0,085	3	0,653	0,056	9	0,67	0,081	2			
ecoli	0,476	0,091	4	0,475	0,075	5	0,504	0,087	2	0,488	0,097	3	0,518	0,064	1			
glass	0,355	0,099	9	0,38	0,099	2	0,372	0,104	4	0,376	0,072	3	0,386	0,075	1			
iris	0,975	0,081	7	0,975	0,081	6	0,974	0,082	9	0,989	0,005	2	0,973	0,082	10			
iris2D	0,949	0,076	9	0,962	0,016	6	0,962	0,006	5	0,959	0,009	8	0,961	0,004	7			
liver	0,392	0,042	10	0,399	0,045	7	0,4	0,041	6	0,406	0,033	1	0,401	0,032	5			
mouse	0,961	0,116	10	1	0	1	0,987	0,07	6	0,999	0,003	5	0,986	0,07	9			
pathbased	0,507	0,061	3	0,495	0,116	7	0,479	0,036	11	0,489	0,089	10	0,493	0,037	8			
seeds	0,744	0,115	8	0,787	0,038	1	0,699	0,14	11	0,771	0,094	2	0,753	0,101	5			
smiley	0,329	0,097	8	0,342	0,084	6	0,322	0,088	9	0,311	0,09	11	0,343	0,085	5			
varied	0,981	0,002	7	0,98	0,003	10	0,981	0,002	8	0,982	0,002	4	0,981	0,003	9			
vertebral3	0,691	0,093	9	0,71	0,07	1	0,709	0,08	2	0,707	0,081	4	0,705	0,116	5			
wine	0,859	0,167	2	0,808	0,208	6	0,852	0,171	3	0,898	0,116	1	0,839	0,159	4			
OBS	6,2 (8)			4,733 (2)			5,467 (7)			4,6 (1)			5,133 (4)					

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

YSA mimarisi öğrenme gerçekleştirirken, ileri yayılım yaptıktan sonra veri setlerindeki gerçek sınıf etiketleri ve tahmin edilen sınıf etiketlerine bakar. Buna göre her bir örnekleme için bir hata hesaplar. Daha sonrasında bu hatanın en aza indirilmesi için ağırlıkları güncelleme işlemi olan geri yayılım kısmına geçilir. Bu çalışmada geri yayılım algoritması olarak türev tabanlı teknikler yerine metasezgisel bir algoritma olan LevySAO algoritması kullanılarak yerel çözüme takılma ihtimali en aza indirilmiştir.

YSA, Matlab ortamında kodlanmış ve aggregation, aniso, balance, ecoli, glass, iris, iris2D, liver, mouse, pathbased, seeds, smiley, varied, vertebral3, wine isimli 15 farklı veri seti kullanılarak sonuçlar elde edilmiştir. Toplam fonksiyonu olarak kümülatif toplam fonksiyonu, aktivasyon fonksiyonu olarak sigmoid fonksiyonu tercih edilirken ağırlıklar vektörel olarak temsil edilmiştir.

Her bir veri seti aynı şartlar altında 30 kez çalıştırılmıştır. Her çalıştırma için iterasyon sayısı 400 ve popülasyon sayısı 50 olmak üzere 20000 kez maksimum uygunluk hesaplaması (maximum Fitness Evaluations, maxFEs) yapılmıştır. YSA'nın ağırlık değerleri, [-10, 10] arama uzayında değerler almıştır. Veri setleri %75 eğitim, %25 test verisi olarak ayrılmıştır. Ancak veri setindeki toplam örnek sayısı 150 ve altında ise veri setinin tamamı hem eğitim hem de test aşamalarında kullanılmıştır.

Algoritmaların başarısını ölçen ve literatürde sıklıkla kullanılan 4 metrik vardır. Bunlar: sensitivity, specificity, precision ve F1-score' dur. Hibrit YSA modellerinin içinde yer alan algoritmaların bu metriklere göre elde ettikleri başarı sıralamaları (rankları), 1'e yakınlığına göre numaralandırılmıştır. Yani maksimum sonucu veren modeldeki algoritma, ilgili veri seti için en başarılı algoritmadır. Daha sonra başarı sırası sütununun (rank sütunu) ortalaması alınıp minimum ortalamaya sahip modeldeki algoritma, seçilen veri setleri için en başarılı algoritma seçilmiştir. Aşağıda 10 algoritmanın YSA ile hibrit oluşturulan modellerinin 15 farklı veri seti ile 30 kez çalıştırılması sonucu bu metriklere göre elde ettikleri ortalama, standart sapma ve başarı sıralamaları verilmiştir.

5.1. YSA-LevySAO Modeli ve Diğer Modellerin Performans Karşılaştırması

Çizelge 5.1.'de sensitivity metrik sonuçları incelendiğinde, YSA-LevySAO modelinin 15 veri setinin 7'sinde – aggregation, ecoli, glass, iris, liver, mouse ve wine – ilk sırayı aldığı ve OBS açısından da 2,533 değeri ile en başarılı sonucu elde ettiği gözlenmiştir. OBS açısından YSA-AAA modeli 2,6 değeri ile 2. olurken orijinal SAO

algoritmasının oluşturduğu model olan YSA-SAO modeli ise 2,867 değeri ile 3. sırada yer almıştır. Diğer algoritmalar ise bu metrik için dikkat çekici bir başarı gösterememiştir.

Çizelge 5.2.'de specificity metrik sonuçları incelendiğinde, YSA-LevySAO modeli 15 veri setinin 9'unda – aggregation, balance, ecoli, glass, iris, liver, mouse, vertebral3 ve wine – ilk sırayı aldığı ve OBS açısından da 2,333 değeri ile en başarılı sonucu elde ettiği gözlenmiştir. OBS açısından YSA-AAA modeli 2,533 değeri ile 2. olurken orijinal SAO algoritmasının oluşturduğu model YSA-SAO modeli ise 3,267 değeri ile 3. sırada yer almıştır. Diğer algoritmaların yer aldığı modeller ise bu metrik için dikkat çekici bir başarı gösterememiştir.

Çizelge 5.3.'te precision metrik sonuçları incelendiğinde, YSA-LevySAO modelinin 15 veri setinin 7'sinde – aggregation, balance, ecoli, glass, liver, mouse ve wine – ilk sırayı aldığı ve OBS açısından da 2,6 değeri ile en başarılı sonucu elde ettiği gözlenmiştir. OBS açısından YSA-AAA modeli 2,8 değeri ile 2. olurken YSA-SAO modeli ise 3,133 değeri ile 3. sırada yer almıştır. Diğer algoritmaların yer aldığı modeller ise bu metrik için dikkat çekici bir başarı gösterememiştir.

Çizelge 5.4.'te F1-score metrik sonuçları incelendiğinde, YSA-LevySAO modeli 15 veri setinin 8'inde – aggregation, ecoli, glass, iris, liver, mouse, varied ve wine – ilk sırayı aldığı ve OBS açısından da 2,4 değeri ile en başarılı sonucu elde ettiği gözlenmiştir. OBS açısından YSA-AAA modeli 2,6 değeri ile 2. olurken YSA-SAO modeli ise 3,133 değeri ile 3. sırada yer almıştır. Diğer algoritmaların yer aldığı modeller ise bu metrik için dikkat çekici bir başarı gösterememiştir.

Çizelge 5.1.-Çizelge 5.4.'te sonuçlar birlikte değerlendirildiğinde, YSA-LevySAO modeli birçok veri setinde ilk sırayı elde ettiği görülmektedir. Ayrıca 4 metriğin tamamında en iyi OBS değerini elde ederek ortalama başarı açısından diğer modellerden daha üstün bir performansa sahip olduğu gözlenmiştir. Ek olarak LevySAO ve CS algoritmaları, Levy uçuşundan faydalanan iki farklı metasezgisel algoritmadır. Ancak bu iki algoritmanın yer aldığı modellerin başarı sıralamaları Çizelge 5.1.-Çizelge 5.4.'ten de görüldüğü üzere birbirinden çok farklıdır. Bu fark, LevySAO algoritmasının Levy uçuşundan faydalanan diğer metasezgisel algoritmalarından farklı olduğunu açıklayarak literatüre önemli bir katkı yapıldığını gösterir.

Çizelge 5.1. YSA-LevySAO modeli ve diğer modellerin sensitivity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-LevySAO			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-AAA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,535	0,148	1	0,294	0,050	8	0,355	0,105	4	0,340	0,121	5	0,329	0,117	6	0,388	0,076	2	0,310	0,068	7	0,220	0,058	10	0,355	0,122	3	0,257	0,031	9
aniso	0,989	0,061	4	1,000	0,001	2	1,000	0,000	1	0,999	0,002	3	0,989	0,061	5	1,000	0,000	1	1,000	0,000	1	0,912	0,094	7	1,000	0,000	1	0,981	0,016	6
balance	0,669	0,057	4	0,604	0,037	8	0,682	0,080	2	0,605	0,017	7	0,626	0,040	6	0,658	0,097	5	0,674	0,092	3	0,542	0,083	10	0,684	0,084	1	0,565	0,073	9
ecoli	0,512	0,079	1	0,261	0,038	8	0,289	0,054	7	0,341	0,069	3	0,299	0,071	6	0,385	0,058	2	0,314	0,068	5	0,146	0,036	10	0,339	0,067	4	0,221	0,061	9
glass	0,423	0,063	1	0,243	0,067	5	0,185	0,042	10	0,269	0,080	4	0,227	0,086	7	0,338	0,044	2	0,303	0,056	3	0,204	0,051	8	0,231	0,074	6	0,196	0,055	9
iris	0,989	0,005	1	0,966	0,014	7	0,982	0,011	5	0,953	0,078	8	0,970	0,083	6	0,984	0,006	4	0,988	0,007	3	0,769	0,124	10	0,989	0,006	2	0,881	0,084	9
iris2D	0,959	0,008	6	0,946	0,013	7	0,965	0,007	3	0,960	0,003	5	0,871	0,140	8	0,965	0,006	2	0,960	0,002	4	0,693	0,067	10	0,970	0,009	1	0,858	0,096	9
liver	0,413	0,027	1	0,377	0,031	7	0,389	0,037	5	0,377	0,034	8	0,387	0,029	6	0,399	0,030	2	0,391	0,024	4	0,355	0,037	10	0,396	0,033	3	0,359	0,032	9
mouse	0,999	0,003	1	0,974	0,021	7	0,998	0,004	3	0,977	0,061	6	0,880	0,182	8	0,996	0,005	4	0,998	0,003	2	0,546	0,137	10	0,989	0,061	5	0,768	0,122	9
pathbased	0,582	0,083	3	0,554	0,065	7	0,560	0,050	5	0,540	0,053	8	0,566	0,082	4	0,583	0,078	2	0,559	0,064	6	0,384	0,160	10	0,600	0,076	1	0,458	0,155	9
seeds	0,774	0,073	3	0,752	0,074	4	0,712	0,105	8	0,724	0,084	7	0,750	0,083	5	0,776	0,037	2	0,784	0,054	1	0,577	0,082	10	0,728	0,070	6	0,614	0,076	9
smiley	0,348	0,093	7	0,306	0,087	8	0,405	0,080	4	0,354	0,099	6	0,371	0,095	5	0,417	0,068	3	0,422	0,061	2	0,250	0,000	10	0,430	0,085	1	0,263	0,048	9
varied	0,982	0,002	2	0,970	0,010	8	0,981	0,002	4	0,979	0,003	7	0,981	0,002	3	0,980	0,004	5	0,980	0,003	6	0,893	0,052	10	0,982	0,003	1	0,949	0,023	9
vertebral3	0,733	0,061	2	0,638	0,047	8	0,649	0,045	7	0,695	0,051	4	0,660	0,056	5	0,748	0,053	1	0,658	0,046	6	0,619	0,054	10	0,700	0,065	3	0,631	0,026	9
wine	0,908	0,099	1	0,663	0,057	7	0,635	0,049	9	0,701	0,119	4	0,703	0,111	3	0,766	0,110	2	0,681	0,080	6	0,594	0,145	10	0,683	0,077	5	0,638	0,059	8
OBS	2,533 (1)			6,733 (8)			5,133 (5)			5,667 (7)			5,533 (6)			2,6 (2)			3,933 (4)			9,667 (10)			2,867 (3)			8,733 (9)		

Çizelge 5.2. YSA-LevySAO modeli ve diğer modellerin specificity metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

Veri Setleri	YSA-LevySAO			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-AAA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,949	0,023	1	0,903	0,009	8	0,912	0,021	3	0,910	0,027	5	0,909	0,028	6	0,915	0,016	2	0,907	0,018	7	0,884	0,021	10	0,911	0,026	4	0,895	0,012	9
aniso	0,994	0,030	5	1,000	0,001	2	1,000	0,000	1	1,000	0,001	3	0,994	0,030	4	1,000	0,000	1	1,000	0,000	1	0,956	0,047	7	1,000	0,000	1	0,991	0,008	6
balance	0,925	0,013	1	0,895	0,027	8	0,925	0,017	2	0,900	0,014	7	0,908	0,013	5	0,904	0,029	6	0,917	0,025	4	0,846	0,071	10	0,919	0,017	3	0,865	0,063	9
ecoli	0,963	0,012	1	0,932	0,010	8	0,939	0,013	6	0,949	0,011	3	0,935	0,017	7	0,954	0,007	2	0,941	0,012	5	0,890	0,018	10	0,945	0,013	4	0,914	0,020	9
glass	0,901	0,020	1	0,854	0,016	5	0,839	0,011	10	0,863	0,019	4	0,852	0,026	6	0,871	0,009	2	0,866	0,013	3	0,843	0,013	8	0,849	0,015	7	0,841	0,013	9
iris	0,995	0,002	1	0,983	0,007	6	0,991	0,006	4	0,977	0,039	7	0,985	0,042	5	0,992	0,003	3	0,994	0,004	2	0,885	0,062	9	0,995	0,003	1	0,940	0,042	8
iris2D	0,979	0,004	6	0,973	0,007	7	0,982	0,003	3	0,980	0,002	5	0,936	0,070	8	0,983	0,003	2	0,980	0,001	4	0,846	0,034	10	0,985	0,004	1	0,929	0,048	9
liver	0,746	0,027	1	0,711	0,031	7	0,723	0,036	5	0,710	0,034	8	0,720	0,029	6	0,733	0,030	2	0,725	0,024	4	0,689	0,037	10	0,729	0,033	3	0,694	0,031	9
mouse	0,999	0,002	1	0,985	0,012	7	0,999	0,002	3	0,988	0,028	6	0,942	0,088	8	0,998	0,003	4	0,999	0,002	2	0,774	0,071	10	0,994	0,030	5	0,887	0,056	9
pathbased	0,784	0,042	3	0,773	0,032	7	0,775	0,025	6	0,765	0,026	8	0,777	0,042	4	0,787	0,040	2	0,775	0,032	5	0,686	0,083	10	0,796	0,040	1	0,723	0,079	9
seeds	0,887	0,036	3	0,876	0,037	4	0,856	0,052	8	0,862	0,042	7	0,875	0,042	5	0,888	0,019	2	0,892	0,027	1	0,789	0,041	10	0,864	0,035	6	0,807	0,038	9
smiley	0,804	0,053	7	0,783	0,052	8	0,840	0,047	4	0,810	0,058	6	0,817	0,054	5	0,847	0,040	1	0,847	0,037	2	0,750	0,000	10	0,845	0,044	3	0,757	0,028	9
varied	0,991	0,001	2	0,985	0,005	8	0,990	0,001	4	0,989	0,002	7	0,990	0,001	3	0,990	0,002	5	0,990	0,001	6	0,947	0,026	10	0,991	0,001	1	0,974	0,012	9
vertebral3	0,905	0,019	1	0,878	0,017	8	0,882	0,013	7	0,902	0,009	3	0,885	0,022	6	0,904	0,015	2	0,889	0,015	5	0,865	0,040	10	0,901	0,015	4	0,876	0,011	9
wine	0,955	0,045	1	0,846	0,025	7	0,831	0,026	9	0,863	0,052	4	0,864	0,049	3	0,891	0,047	2	0,855	0,034	6	0,800	0,077	10	0,855	0,031	5	0,833	0,031	8
OBS	2,333 (1)			6,667 (8)			5 (5)			5,533 (7)			5,40 (6)			2,533 (2)			3,80 (4)			9,6 (10)			3,267 (3)			8,667 (9)		

Çizelge 5.3. YSA-LevySAO modeli ve diğer modellerin precision metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

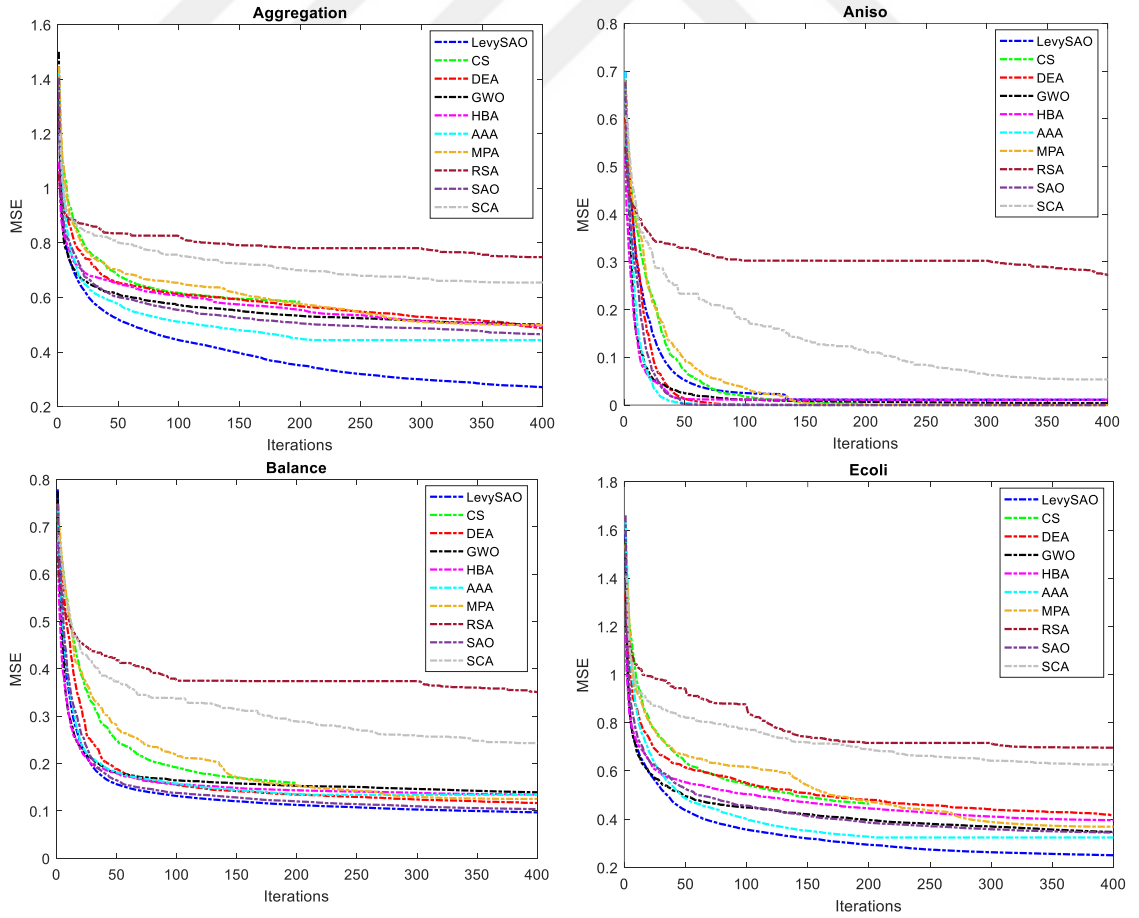
Veri Setleri	YSA-LevySAO			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-AAA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,476	0,167	1	0,233	0,048	8	0,291	0,092	5	0,297	0,110	4	0,268	0,125	6	0,353	0,073	2	0,255	0,056	7	0,155	0,060	10	0,302	0,111	3	0,198	0,029	9
aniso	0,983	0,091	4	1,000	0,001	2	1,000	0,000	1	0,999	0,002	3	0,983	0,091	4	1,000	0,000	1	1,000	0,000	1	0,926	0,096	6	1,000	0,000	1	0,982	0,014	5
balance	0,680	0,109	1	0,599	0,091	7	0,671	0,084	2	0,579	0,009	8	0,613	0,082	6	0,646	0,098	5	0,648	0,090	4	0,544	0,033	10	0,670	0,100	3	0,559	0,031	9
ecoli	0,490	0,115	1	0,233	0,054	8	0,239	0,062	6	0,294	0,091	3	0,234	0,086	7	0,373	0,077	2	0,268	0,089	5	0,091	0,047	10	0,286	0,082	4	0,158	0,050	9
glass	0,367	0,075	1	0,156	0,072	5	0,098	0,066	10	0,196	0,083	4	0,138	0,092	7	0,278	0,069	2	0,232	0,102	3	0,138	0,100	8	0,145	0,098	6	0,114	0,076	9
iris	0,989	0,005	2	0,967	0,014	6	0,982	0,011	5	0,941	0,128	8	0,958	0,129	7	0,985	0,006	4	0,988	0,007	3	0,728	0,201	10	0,989	0,006	1	0,885	0,117	9
iris2D	0,959	0,007	6	0,946	0,013	7	0,965	0,007	3	0,960	0,004	5	0,818	0,224	9	0,965	0,006	2	0,961	0,002	4	0,641	0,180	10	0,970	0,009	1	0,858	0,137	8
liver	0,440	0,032	1	0,408	0,066	7	0,420	0,088	5	0,379	0,099	8	0,411	0,066	6	0,430	0,034	4	0,436	0,034	2	0,287	0,118	10	0,432	0,038	3	0,347	0,104	9
mouse	0,999	0,004	1	0,974	0,026	6	0,998	0,004	2	0,972	0,083	7	0,851	0,231	8	0,995	0,007	4	0,997	0,005	3	0,523	0,187	10	0,985	0,076	5	0,768	0,191	9
pathbased	0,552	0,166	6	0,624	0,095	2	0,567	0,097	5	0,528	0,132	7	0,514	0,142	8	0,597	0,118	3	0,579	0,116	4	0,334	0,203	10	0,647	0,116	1	0,474	0,228	9
seeds	0,801	0,113	3	0,795	0,071	4	0,740	0,133	8	0,749	0,136	6	0,754	0,152	5	0,817	0,032	2	0,827	0,046	1	0,511	0,145	10	0,745	0,134	7	0,549	0,167	9
smiley	0,284	0,088	7	0,242	0,079	8	0,331	0,071	4	0,285	0,089	6	0,302	0,087	5	0,343	0,061	3	0,348	0,054	1	0,192	0,008	10	0,345	0,065	2	0,204	0,045	9
varied	0,982	0,002	2	0,971	0,009	8	0,981	0,002	4	0,979	0,003	7	0,981	0,002	3	0,981	0,004	5	0,980	0,003	6	0,915	0,033	10	0,982	0,003	1	0,952	0,021	9
vertebral3	0,718	0,108	2	0,559	0,091	8	0,577	0,116	6	0,650	0,126	4	0,574	0,121	7	0,748	0,068	1	0,585	0,115	5	0,523	0,060	10	0,656	0,124	3	0,546	0,083	9
wine	0,910	0,115	1	0,567	0,115	7	0,532	0,083	8	0,607	0,172	5	0,620	0,164	3	0,736	0,171	2	0,607	0,146	4	0,513	0,213	10	0,591	0,136	6	0,528	0,084	9
OBS	2,6 (1)			6,2 (8)			4,933 (5)			5,667 (6)			6,067 (7)			2,8 (2)			3,533 (4)			9,6 (10)			3,133 (3)			8,667 (9)		

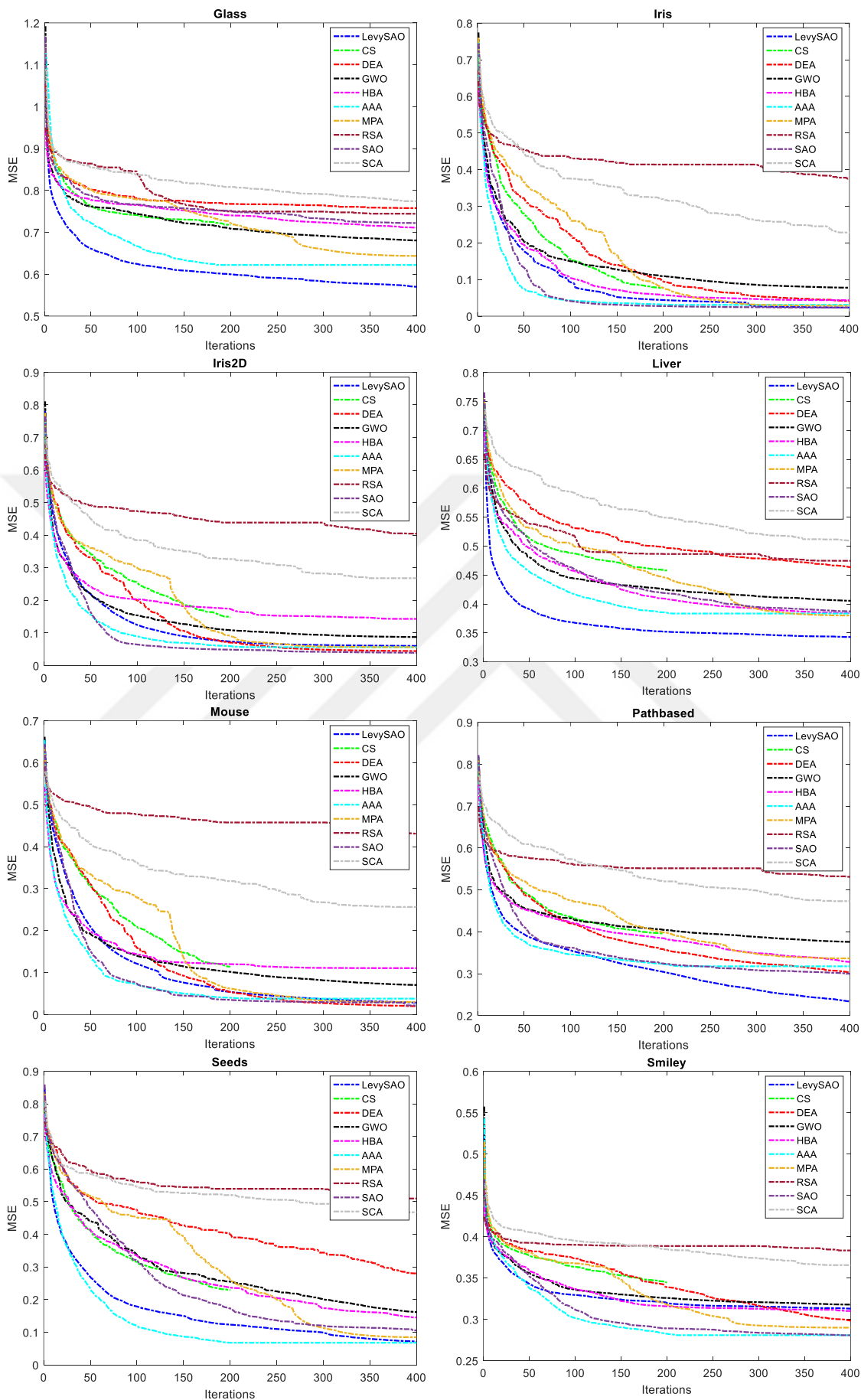
Çizelge 5.4. YSA-LevySAO modeli ve diğer modellerin F1-score metriği için elde ettiği ortalama, standart sapma ve başarı sırası değerleri

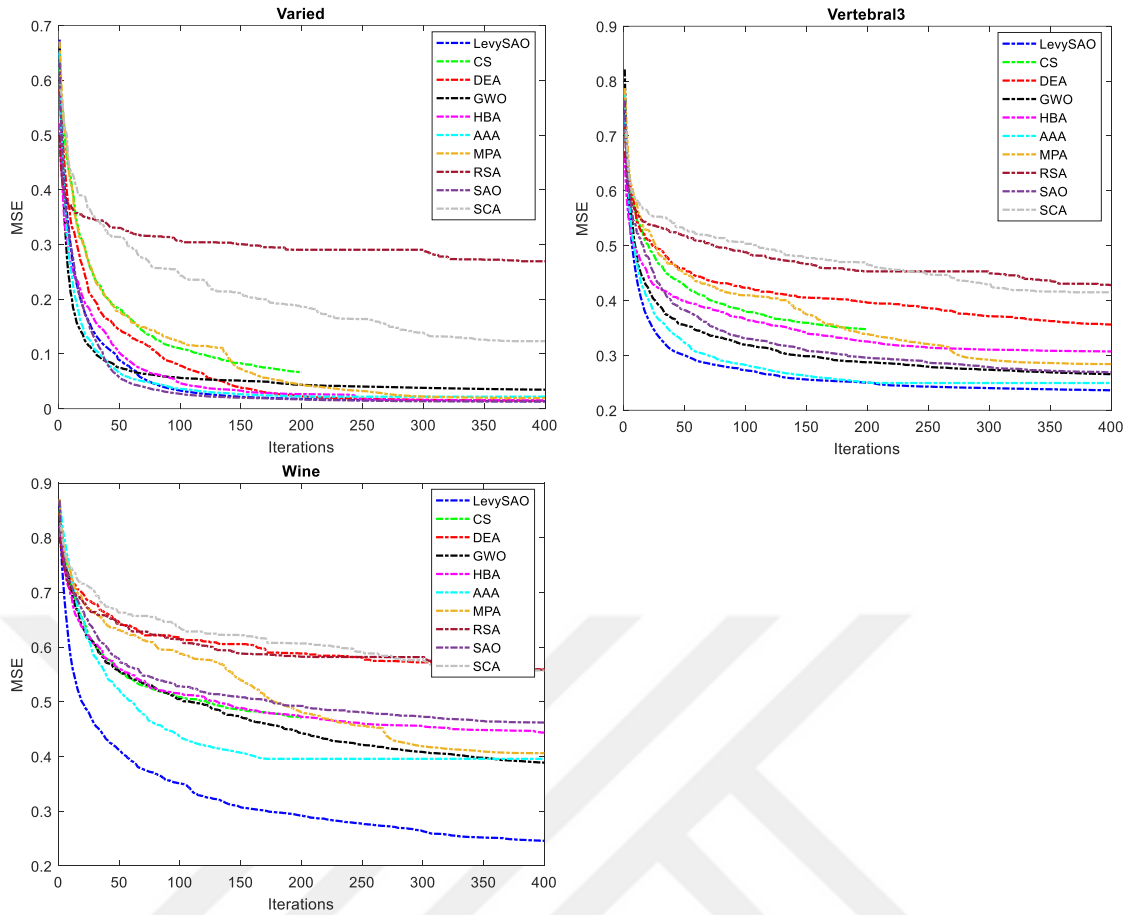
Veri Setleri	YSA-LevySAO			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-AAA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra	ort	std	sıra
aggregation	0,485	0,155	1	0,238	0,042	8	0,294	0,096	4	0,289	0,116	5	0,275	0,126	6	0,330	0,073	2	0,252	0,055	7	0,154	0,058	10	0,294	0,120	3	0,201	0,029	9
aniso	0,985	0,081	4	1,000	0,001	2	1,000	0,000	1	0,999	0,002	3	0,985	0,081	4	1,000	0,000	1	1,000	0,000	1	0,902	0,117	6	1,000	0,000	1	0,981	0,016	5
balance	0,653	0,056	3	0,583	0,042	7	0,669	0,078	1	0,583	0,017	8	0,604	0,041	6	0,632	0,086	5	0,652	0,090	4	0,509	0,100	10	0,663	0,083	2	0,537	0,081	9
ecoli	0,488	0,097	1	0,238	0,045	8	0,256	0,056	6	0,310	0,076	3	0,253	0,076	7	0,367	0,058	2	0,279	0,074	5	0,104	0,039	10	0,302	0,072	4	0,173	0,050	9
glass	0,376	0,072	1	0,175	0,064	5	0,113	0,046	10	0,212	0,079	4	0,160	0,092	6	0,277	0,043	2	0,239	0,068	3	0,142	0,069	8	0,158	0,077	7	0,124	0,057	9
iris	0,989	0,005	1	0,966	0,014	6	0,982	0,011	5	0,945	0,109	8	0,962	0,113	7	0,984	0,006	4	0,988	0,007	3	0,719	0,169	10	0,989	0,006	2	0,869	0,109	9
iris2D	0,959	0,009	6	0,946	0,013	7	0,965	0,007	3	0,960	0,003	5	0,837	0,193	9	0,965	0,006	2	0,960	0,002	4	0,607	0,098	10	0,970	0,009	1	0,843	0,126	8
liver	0,406	0,033	1	0,351	0,055	7	0,362	0,065	6	0,342	0,064	8	0,365	0,051	5	0,386	0,045	2	0,374	0,035	4	0,291	0,073	10	0,378	0,049	3	0,318	0,063	9
mouse	0,999	0,003	1	0,972	0,023	7	0,998	0,004	2	0,974	0,073	6	0,861	0,210	8	0,996	0,006	4	0,998	0,004	3	0,501	0,152	10	0,987	0,070	5	0,741	0,150	9
pathbased	0,489	0,089	3	0,467	0,077	7	0,480	0,061	5	0,457	0,046	8	0,487	0,096	4	0,515	0,102	2	0,476	0,081	6	0,305	0,136	10	0,525	0,104	1	0,373	0,133	9
seeds	0,771	0,094	3	0,751	0,075	4	0,698	0,126	8	0,714	0,111	7	0,733	0,120	5	0,780	0,037	2	0,787	0,053	1	0,499	0,101	10	0,715	0,105	6	0,541	0,110	9
smiley	0,311	0,090	7	0,268	0,080	8	0,361	0,073	4	0,313	0,092	6	0,331	0,088	5	0,374	0,063	3	0,378	0,055	2	0,217	0,008	10	0,378	0,069	1	0,229	0,046	9
varied	0,982	0,002	1	0,969	0,010	8	0,981	0,002	4	0,979	0,003	7	0,981	0,002	3	0,980	0,004	5	0,980	0,003	6	0,887	0,058	10	0,981	0,003	2	0,948	0,024	9
vertebral3	0,707	0,081	2	0,583	0,063	8	0,593	0,066	7	0,649	0,077	4	0,598	0,081	6	0,730	0,056	1	0,605	0,071	5	0,546	0,060	10	0,657	0,087	3	0,571	0,040	9
wine	0,898	0,116	1	0,587	0,072	7	0,555	0,060	9	0,634	0,148	4	0,643	0,140	3	0,728	0,140	2	0,616	0,100	5	0,514	0,176	10	0,614	0,097	6	0,561	0,070	8
OBS	2,4 (1)			6,6 (8)			5 (5)			5,733 (7)			5,6 (6)			2,6 (2)			3,933 (4)			9,6 (10)			3,133 (3)			8,6 (9)		

5.2. Yakınsama Grafikleri

Modellerin eğitim aşamasındaki MSE değerine göre hata yakınsama grafikleri Şekil 5.1.'de verilmiştir. Grafikler incelendiğinde YSA-LevySAO modelinin çoğu veri setinde diğer modellere göre daha hızlı yakınsama performansına sahip olduğu görülmektedir. Bu da hataları diğer modellerden daha hızlı minimize ettiği anlamına gelmektedir. YSA-LevySAO, aggregation, ecoli, glass, liver, pathbased, seeds vertebral3 ve wine veri kümelerinde orijinal SAO'nun bulunduğu YSA-SAO modelinden daha üstün yakınsama performansı göstermiştir. Orijinal SAO algoritmasının bulunduğu YSA-SAO modeli yalnızca iris2D ve wine veri kümelerinde YSA-LevySAO'dan daha iyi bir yakınsama performansına sahipken; iki modelin aniso, balance, iris, mouse ve çeşitli veri kümeleri üzerinde rekabetçi performansa sahip olduğu gözlemlenmiştir. Orijinal SAO algoritmasının bulunduğu YSA-SAO modelinin hata yakınsama hızı genel olarak yeterli gelmemiştir. Bu da Levy uçuşu yönteminin algoritmanın başarısına katkı sağladığını gösterir.







Şekil 5.1. Algoritmaların oluşturduğu modellerin yakınsama grafikleri

5.3. YSA-LevySAO Modeli ve Diğer Modellerin Bire Bir Performans

Karşılaştırması (Wilcoxon Testi)

Önerilen LevySAO algoritması YSA ile hibrit modellenerek YSA-LevySAO modeli geliştirilmiş ve diğer algoritmalar ile oluşturulan hibrit YSA modelleri ile sensitivity, specificity, precision ve F1-score metrikleri için elde ettiği değerlerin bire bir karşılaştırma sonuçları Çizelge 5.5.- Çizelge 5.8.'de sunulmuştur. Karşılaştırma sonuçları "1 - 1" sütunu altında 'w/l/d' harfleriyle belirtilmiştir. Bu harflerin anlamları şu şekildedir:

- win (w): YSA-LevySAO modeli, karşılaştırılan modelden daha başarılı sonuç elde etmiştir. Dolayısıyla galibiyet elde edilmiştir.
- lost (l): Karşılaştırılan model, YSA-LevySAO modelinden daha başarılı sonuç elde etmiştir. Dolayısıyla mağlubiyet elde edilmiştir.
- draw (d): YSA-LevySAO modeli ve karşılaştırılan model, aynı sonucu üretirek berabere kalmıştır.

İlgili sütunun en alt satırında 15 veri seti için toplam w/l/d sayıları verilmiştir.

Çizelge 5.5.'te sensitivity bire bir karşılaştırma sonuçları incelendiğinde YSA-LevySAO modelinin YSA-AAA ve YSA-SAO'ya 9, YSA-MPA'ya 10, YSA-DEA'ya 11, YSA-GWO'ya 12, YSA-CS ve YSA-HBA'ya 14, YSA-RSA ve YSA-SCA'ya 15 veri setinde üstünlük sağladığı görülmüştür. Toplamda sensitivity metriği için yapılan 135 karşılaştırmanın 109'unda önerilen YSA-LevySAO modeli kazanan taraf olmuştur.

Çizelge 5.6.'da specificity bire bir karşılaştırma sonuçları incelendiğinde YSA-LevySAO modelinin YSA-SAO'ya 9, YSA-AAA'ya 10, YSA-MPA'ya 11, YSA-DEA ve YSA-GWO'ya 12, YSA-HBA'ya 13, YSA-CS'ye 14, YSA-RSA ve YSA-SCA'ya 15 veri setinde üstünlük sağladığı görülmüştür. Toplamda specificity metriği için yapılan 135 karşılaştırmanın 111'inde önerilen YSA-LevySAO modeli kazanan taraf olmuştur.

Çizelge 5.7.'de precision bire bir karşılaştırma sonuçları incelendiğinde YSA-LevySAO modeli YSA-AAA ve YSA-SAO'ya 9, YSA-MPA'ya 10, YSA-DEA'ya 11, YSA-GWO'ya 12, YSA-CS ve YSA-HBA'ya 13, YSA-RSA ve YSA-SCA'ya 15 veri setinde üstünlük sağladığı görülmüştür. Toplamda precision metriği için yapılan 135 karşılaştırmanın 107'sinde önerilen YSA-LevySAO modeli kazanan taraf olmuştur.

Çizelge 5.8.'de F1-score bire bir karşılaştırma sonuçları incelendiğinde YSA-LevySAO modeli YSA-AAA'ya 9, YSA-SAO'ya 10, YSA-MPA ve YSA-DEA'ya 11, YSA-GWO'ya 12, YSA-HBA'ya 13, YSA-CS'ye 14, YSA-RSA ve YSA-SCA'ya 15 veri setinde üstünlük sağladığı görülmüştür. Toplamda F1-score metriği için yapılan 135 karşılaştırmanın 110'unda önerilen YSA-LevySAO modeli kazanan taraf olmuştur.

Çizelge 5.5.-Çizelge 5.8.'deki bire bir karşılaştırma sonuçları birlikte değerlendirildiğinde önerilen LevySAO algoritmasının yer aldığı YSA-LevySAO modelinin diğer modellerden açık bir şekilde daha başarılı olduğu görülmektedir.

Ancak önerilen LevySAO algoritmasının yer aldığı YSA-LevySAO modelinin elde ettiği başarılı sonuçların tesadüfen oluşup oluşmadığının da analiz edilmesi gerekir. Bu amaçla, önerilen model ve diğer modellerin metrik değerleri arasında %95 güven aralığında Wilcoxon işaretli sıralar testi (Wilcoxon ve ark., 1970) uygulanmış ve elde edilen sonuçlar Çizelge 5.5.-Çizelge 5.8.'deki p-değeri sütunlarında sunulmuştur. Tablolardaki p değerinin 0,05'ten küçük olması, YSA-LevySAO modelinin elde ettiği sonucun istatistiksel olarak anlamlı olduğunu ve tesadüfen oluşmadığı gösterir. Bu durumu göstermek için tablolardaki işaret sütununa “+” işareti konulmuştur. Aksi durumda, sonuçların tesadüfen oluşma ihtimali vardır ve bu durumu göstermek için işaret

sütununa “-” işareti konulmuştur. İlgili sütunun en alt satırında 15 veri seti için toplam +/- sayıları verilmiştir.

- Çizelge 5.5.’de sensitivity metriği için elde edilen sonuçlara bakıldığında YSA-AAA’da 8; YSA-HBA, YSA-MPA ve YSA-SAO’da 9; YSA-DEA’da 11; YSA-CS ve YSA-GWO’da 12, YSA-RSA ve YSA-SCA’da 15 veri setinde istatistiksel açıdan anlamlı sonuçlar elde edildiği görülmektedir.

- Çizelge 5.6.’da specificity metriği için elde edilen sonuçlara bakıldığında YSA-SAO’da 8; YSA-AAA, YSA-HBA ve YSA-MPA’da 9; YSA-DEA ve YSA-GWO’da 11; YSA-CS’de 12, YSA-RSA ve YSA-SCA’da 15 veri setinde istatistiksel açıdan anlamlı sonuçlar elde edildiği görülmektedir.

- Çizelge 5.7.’de precision metriği için elde edilen sonuçlara bakıldığında YSA-AAA, YSA-DEA, YSA-HBA, YSA-MPA ve YSA-SAO’da 8; YSA-CS’de 11; YSA-GWO’da 12; YSA-SCA’da 14 ve YSA-RSA’da 15 veri setinde istatistiksel açıdan anlamlı sonuçlar elde edildiği görülmektedir.

- Çizelge 5.8.’de F1-score metriği için elde edilen sonuçlara bakıldığında YSA-HBA’da 8; YSA-AAA, YSA-MPA ve YSA-SAO’da 9; YSA-DEA’da 10; YSA-CS’de 12; YSA-GWO’da 13; YSA-RSA ve YSA-SCA’da 15 veri setinde istatistiksel açıdan anlamlı sonuçlar elde edildiği görülmektedir.

Çizelge 5.5.-Çizelge 5.8.’deki Wilcoxon işaretli sıralar testi sonuçları birlikte değerlendirildiğinde önerilen YSA-LevySAO modelinin çoğu durumda istatistiksel açıdan anlamlı sonuçlar ürettiği görülmektedir.

Çizelge 5.5. YSA-LevySAO modeli ve diğer modellerin sensitivity metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları

Veri Setleri	YSA-AAA			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA																													
	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret																											
aggregation	w	4,637e-5	+	w	4,443e-8	+	w	1,335e-5	+	w	5,079e-6	+	w	3,423e-6	+	w	3,943e-7	+	w	1,967e-10	+	w	2,201e-5	+	w	9,044e-10	+																											
aniso	1	0,334	-	1	0,190	-	1	0,334	-	1	0,597	-	w	1	-	1	0,334	-	w	1,463e-10	+	1	0,334	-	w	4,935e-10	+																											
balance	w	0,078	-	w	6,609e-7	+	1	1	-	w	3,973e-9	+	w	2,195e-5	+	1	0,283	-	w	2,394e-9	+	1	0,819	-	w	5,250e-7	+																											
ecoli	w	1,357e-7	+	w	3,307e-11	+	w	6,615e-11	+	w	5,939e-9	+	w	2,860e-10	+	w	3,302e-10	+	w	1,439e-11	+	w	2,122e-9	+	w	3,674e-11	+																											
glass	w	2,949e-6	+	w	2,025e-10	+	w	2,807e-11	+	w	1,540e-8	+	w	1,748e-9	+	w	1,820e-8	+	w	3,495e-11	+	w	2,505e-10	+	w	5,370e-11	+																											
iris	w	0,002	+	w	8,732e-11	+	w	0,006	+	w	2,282e-9	+	w	0,203	-	w	0,628	-	w	1,766e-11	+	w	0,633	-	w	4,559e-11	+																											
iris2D	1	0,001	+	w	7,134e-5	+	1	0,004	+	1	0,454	-	w	0,095	-	1	0,760	-	w	5,695e-12	+	1	7,716e-7	+	w	3,904e-10	+																											
liver	w	0,098	-	w	9,768e-5	+	w	0,015	+	w	1,393e-4	+	w	0,002	+	w	0,003	+	w	5,649e-7	+	w	0,044	+	w	1,583e-7	+																											
mouse	w	0,009	+	w	1,371e-9	+	w	0,302	-	w	2,119e-5	+	w	4,541e-4	+	w	0,221	-	w	4,015e-12	+	w	0,720	-	w	4,053e-12	+																											
pathbased	1	0,342	-	w	0,017	+	w	0,017	+	w	3,2101e-4	+	w	0,022	+	w	0,011	+	w	6,041e-6	+	1	0,732	-	w	5,257e-5	+																											
seeds	1	0,864	-	w	0,248	-	w	0,022	+	w	0,008	+	w	0,4722	-	1	0,533	-	w	4,787e-10	+	w	0,021	+	w	9,197e-9	+																											
smiley	1	5,051e-4	+	w	0,111	-	1	0,002	+	1	0,420	-	1	0,245	-	1	7,232e-4	+	w	5,290e-6	+	1	8,175e-5	+	w	1,2482e-4	+																											
varied	w	0,170	-	w	3,918e-10	+	w	0,226	-	w	2,819e-4	+	w	0,256	-	w	0,005	+	w	6,902e-11	+	1	0,554	-	w	2,508e-11	+																											
vertebral3	1	0,549	-	w	2,333e-8	+	w	1,473e-7	+	w	0,0140	+	w	8,982e-6	+	w	1,670e-6	+	w	6,475e-10	+	w	0,038	+	w	6,800e-10	+																											
wine	w	7,916e-6	+	w	1,766e-9	+	w	6,036e-10	+	w	6,198e-7	+	w	9,071e-8	+	w	9,731e-9	+	w	1,494e-9	+	w	5,909e-9	+	w	6,088e-10	+																											
	9/6/0			8/7			14/1/0			12/3			11/4/0			11/4			12/3/0			12/3			14/1/0			9/6			10/5/0			9/6			15/0/0			15/0			9/6/0			9/6			15/0/0			15/0		

Çizelge 5.6. YSA-LevySAO modeli ve diğer modellerin specificity metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları

Veri Setleri	YSA-AAA			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret
aggregation	w	1,286e-6	+	w	3,352e-8	+	w	7,599e-7	+	w	1,861e-6	+	w	2,153e-6	+	w	1,473e-7	+	w	1,150e-9	+	w	1,108e-6	+	w	2,916e-9	+
aniso	1	0,334	-	1	0,190	-	1	0,334	-	1	0,597	-	1	1	-	1	0,334	-	w	1,463e-10	+	1	0,334	-	w	4,935e-10	+
balance	w	0,001	+	w	1,353e-6	+	w	0,976	-	w	2,690e-8	+	w	1,162e-5	+	w	0,237	-	w	1,544e-8	+	w	0,122	-	w	4,731e-5	+
ecoli	w	8,663e-5	+	w	9,756e-10	+	w	1,429e-8	+	w	9,514e-6	+	w	1,558e-8	+	w	1,254e-7	+	w	2,936e-11	+	w	7,043e-7	+	w	1,613e-10	+
glass	w	1,469e-7	+	w	5,527e-10	+	w	5,239e-11	+	w	9,813e-8	+	w	2,482e-8	+	w	1,833e-8	+	w	5,353e-11	+	w	1,813e-10	+	w	4,164e-11	+
iris	w	0,002	+	w	8,732e-11	+	w	0,006	+	w	2,282e-9	+	w	0,203	-	w	0,628	-	w	1,767e-11	+	d	0,633	-	w	4,556e-11	+
iris2D	1	0,001	+	w	7,807e-5	+	1	0,004	+	1	0,454	-	w	0,095	-	1	0,760	-	w	5,695e-12	+	1	7,716e-7	+	w	4,128e-10	+
liver	w	0,098	-	w	1,283e-4	+	w	0,018	+	w	1,393e-4	+	w	0,002	+	w	0,003	+	w	6,111e-7	+	w	0,044	+	w	2,183e-7	+
mouse	w	0,008	+	w	1,244e-9	+	w	0,302	-	w	1,965e-5	+	w	4,559e-4	+	w	0,221	-	w	4,026e-12	+	w	0,720	-	w	4,060e-12	+
pathbased	1	0,348	-	w	0,035	+	w	0,012	+	w	6,273e-4	+	w	0,010	+	w	0,023	+	w	7,417e-6	+	1	0,761	-	w	5,216e-5	+
seeds	1	0,899	-	w	0,215	-	w	0,023	+	w	0,008	+	w	0,471	-	1	0,552	-	w	4,695e-10	+	w	0,021	+	w	7,946e-9	+
smiley	1	2,8534e-4	+	w	0,158	-	1	0,001	+	1	0,376	-	1	0,264	-	1	8,066e-4	+	w	5,290e-6	+	1	3,088e-4	+	w	1,3425e-4	+
varied	w	0,170	-	w	3,918e-10	+	w	0,226	-	w	2,819e-4	+	w	0,256	-	w	0,005	+	w	6,902e-11	+	1	0,554	-	w	2,508e-11	+
vertebral3	w	0,668	-	w	3,507e-7	+	w	2,135e-6	+	w	0,399	-	w	1,204e-4	+	w	5,383	+	w	4,471e-8	+	w	0,237	-	w	4,259e-8	+
wine	w	6,965e-6	+	w	1,892e-9	+	w	6,176e-10	+	w	7,132e-7	+	w	8,006e-8	+	w	1,084e-8	+	w	1,650e-9	+	w	6,073e-9	+	w	7,533e-10	+
	10/5/0		9/6	14/1/0		12/3	12/3/0		11/4	12/3/0		11/4	13/2/0		9/6	11/4/0		9/6	15/0/0		15/0	9/5/1		8/7	15/0/0		15/0

Çizelge 5 7. YSA-LevySAO modeli ve diğer modellerin precision metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları

Veri Setleri	YSA-AAA			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret
aggregation	w	0,002	+	w	5,533e-8	+	w	2,682e-5	+	w	4,348e-5	+	w	6,062e-6	+	w	9,533e-7	+	w	4,860e-11	+	w	3,478e-5	+	w	2,366e-10	+
aniso	1	0,334	-	1	0,190	-	1	0,334	-	1	0,597	-	d	1	-	1	0,334	-	w	1,465e-10	+	1	0,334	-	w	4,992e-10	+
balance	w	0,055	-	w	2,305e-6	+	w	0,994	-	w	3,373e-8	+	w	1,856e-4	+	w	0,095	-	w	6,411e-10	+	w	0,535	-	w	1,623e-8	+
ecoli	w	3,592e-5	+	w	8,891e-10	+	w	1,411e-9	+	w	1,067e-7	+	w	2,670e-9	+	w	1,429e-8	+	w	2,936e-11	+	w	3,965e-8	+	w	8,988e-11	+
glass	w	4,347e-5	+	w	2,359e-10	+	w	9,519e-11	+	w	9,234e-9	+	w	6,901e-10	+	w	3,079e-6	+	w	2,497e-9	+	w	3,083e-9	+	w	1,156e-10	+
iris	w	0,011	+	w	2,417e-10	+	w	0,008	+	w	1,007e-8	+	w	0,249	-	w	0,780	-	w	1,939e-11	+	1	0,727	-	w	6,312e-11	+
iris2D	1	4,756e-5	+	w	0,004	+	1	0,004	+	1	0,024	+	w	0,095	-	1	0,011	+	w	6,209e-12	+	1	7,509e-7	+	w	2,425e-10	+
liver	w	0,267	-	w	0,017	+	w	0,554	-	w	0,010	+	w	0,012	+	w	0,679	-	w	4,564e-6	+	w	0,620	-	w	4,063e-5	+
mouse	w	0,011	+	w	3,602e-9	+	w	0,357	-	w	1,153e-5	+	w	3,990e-4	+	w	0,191	-	w	4,026e-12	+	w	0,760	-	w	4,063e-12	+
pathbased	1	0,255	-	1	0,064	-	1	0,689	-	w	0,274	-	w	0,253	-	1	0,446	-	w	3,042e-5	+	1	0,022	+	w	0,147	-
seeds	1	0,894	-	w	0,315	-	w	0,023	+	w	0,041	+	w	0,322	-	1	0,539	-	w	1,053e-8	+	w	0,029	+	w	1,601e-6	+
smiley	1	0,034	+	w	0,084	-	1	0,087	-	1	0,796	-	1	0,257	-	1	0,003	+	w	5,290e-6	+	1	0,003	+	w	1,078e-4	+
varied	w	0,643	-	w	1,056e-9	+	w	0,550	-	w	0,002	+	w	0,335	-	w	0,025	+	w	1,311e-10	+	1	0,253	-	w	3,096e-11	+
vertebral3	1	0,673	-	w	8,172e-7	+	w	1,328e-5	+	w	0,026	+	w	2,847e-5	+	w	2,188e-5	+	w	1,486e-8	+	w	0,062	-	w	1,457e-7	+
wine	w	5,436e-6	+	w	1,309e-9	+	w	2,889e-10	+	w	2,596e-7	+	w	4,119e-8	+	w	5,354e-9	+	w	1,809e-9	+	w	3,548e-9	+	w	3,217e-10	+
	9/6/0		8/7	13/2/0		11/4	11/4/0		8/7	12/3/0		12/3	13/1/1		8/7	10/5/0		8/7	15/0/0		15/0	9/6/0		8/7	15/0/0		14/1

Çizelge 5.8. YSA-LevySAO modeli ve diğer modellerin F1-score metriği için bire bir karşılaştırma ve Wilcoxon işaretli sıralar anlamlılık testi sonuçları

Veri Setleri	YSA-AAA			YSA-CS			YSA-DEA			YSA-GWO			YSA-HBA			YSA-MPA			YSA-RSA			YSA-SAO			YSA-SCA		
	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret	1 - 1	p-değeri	işaret
aggregation	w	8,663e-5	+	w	2,831e-8	+	w	4,745e-6	+	w	6,734e-6	+	w	2,581e-6	+	w	1,873e-7	+	w	7,222e-11	+	w	7,221e-6	+	w	2,865e-10	+
aniso	1	0,337	-	1	0,190	-	1	0,334	-	1	0,597	-	d	1	-	1	0,334	-	w	1,466e-10	+	1	0,334	-	w	5,021e-10	+
balance	w	0,046	+	w	7,556e-7	+	1	0,663	-	w	1,429e-8	+	w	2,833e-5	+	w	0,287	-	w	1,021e-9	+	1	0,767	-	w	9,421e-8	+
ecoli	w	1,108e-6	+	w	2,610e-10	+	w	5,573e-10	+	w	3,965e-8	+	w	1,174e-9	+	w	2,922e-9	+	w	2,398e-11	+	w	1,558e-8	+	w	4,972e-11	+
glass	w	1,026e-6	+	w	1,199e-10	+	w	3,899e-11	+	w	1,308e-8	+	w	1,452e-9	+	w	3,334e-8	+	w	1,355e-10	+	w	2,276e-10	+	w	4,311e-11	+
iris	w	3,243	+	w	4,994e-11	+	w	0,005	+	w	1,038e-9	+	w	0,176	-	w	0,497	-	w	1,966e-11	+	w	0,553	-	w	4,228e-11	+
iris2D	1	0,043	+	w	7,107e-6	+	1	0,004	+	1	5,349e-4	+	w	0,095	-	1	0,036	+	w	6,304e-12	+	1	8,079e-7	+	w	4,388e-10	+
liver	w	0,104	-	w	6,963e-5	+	w	0,008	+	w	4,038e-5	+	w	0,001	+	w	5,388e-4	+	w	5,655e-7	+	w	0,020	+	w	2,761e-7	+
mouse	w	0,008	+	w	1,247e-9	+	w	0,302	-	w	1,967e-5	+	w	4,559e-4	+	w	0,221	-	w	4,026e-12	+	w	0,720	-	w	4,069e-12	+
pathbased	1	0,918	-	w	0,015	+	w	0,225	-	w	0,004	+	w	0,292	-	w	0,077	-	w	3,188e-6	+	1	0,912	-	w	3,932e-5	+
seeds	1	0,767	-	w	0,099	-	w	0,016	+	w	0,006	+	w	0,308	-	1	0,641	-	w	5,828e-10	+	w	0,018	+	w	4,552e-9	+
smiley	1	7,044e-4	+	w	0,111	-	1	0,006	+	1	0,447	-	1	0,313	-	1	0,002	+	w	5,290e-6	+	1	6,102e-4	+	w	1,160e-4	+
varied	w	0,062	-	w	6,018e-10	+	w	0,132	-	w	1,709e-5	+	w	0,239	-	w	0,008	+	w	4,417e-11	+	w	0,927	-	w	3,424e-11	+
vertebral3	1	0,610	-	w	3,245e-7	+	w	8,126e-7	+	w	0,007	+	w	9,130e-6	+	w	6,915e-6	+	w	2,539e-9	+	w	0,032	+	w	1,676e-8	+
wine	w	4,581e-6	+	w	1,439e-9	+	w	4,242e-10	+	w	3,704e-7	+	w	5,778e-8	+	w	8,379e-9	+	w	1,653e-9	+	w	4,457e-9	+	w	5,184e-10	+
	9/6/0		9/6	14/1/0		12/3	11/4/0		10/5	12/3/0		13/2	13/1/1		8/7	11/4/0		9/6	15/0/0		15/0	10/5/0		9/6	15/0/0		15/0

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

Bu tez çalışmasında, iki farklı yerel minimuma takılma problemine çözüm bulunmuştur: İlk olarak, YSA mimarisinin hatanın geri yayılımında ağırlık güncelleme esnasında yerel minimuma takılma problemi için çözüm üretilmiştir. Bunun için SAO algoritmasından yardım alınmıştır. İkinci olarak SAO algoritmasının yerel minimuma takılma problemine çözüm üretilmiştir. Bunun için ise Levy uçuşu tekniğinden yardım alınarak LevySAO algoritması önerilmiştir.

Önerilen algoritmanın YSA ile hibrit olarak kullanılmasıyla YSA-LevySAO modeli geliştirilmiştir. Geliştirilen hibrit model 15 farklı veri kümesinde test edilmiş ve sensitivity, specificity, precision ve F1-score metrikleri açısından puanlanmıştır. YSA-LevySAO modelinin sonuçları, literatürde iyi bilinen 9 farklı algoritma ile geliştirilen hibrit YSA modelleriyle ortalama başarı sıralaması, bire bir performans karşılaştırması ve hata yakınsama performansları açısından karşılaştırılmıştır. Sonuçlar, LevySAO algoritmasının çoğu durumda orijinal SAO ve diğer karşılaştırma algoritmalarından daha üstün performans gösterdiğini göstermiştir. Eğitim aşamasındaki hata yakınsama performansları incelendiğinde, önerilen LevySAO algoritmasının yer aldığı YSA-LevySAO modelinin birçok veri kümesinde karşılaştırılan modellerden daha hızlı yakınsama performansına sahip olduğu görülmüştür. Ek olarak, önerilen YSA-LevySAO modelinin sonuçlarının %95 güven aralığıyla istatistiksel olarak anlamlı olup olmadığını belirlemek için Wilcoxon işaretli sıralar testi uygulanmıştır. Sonuçlar incelendiğinde, önerilen modelin çoğu durumda istatistiksel olarak anlamlı değerler ürettiği görülmüştür.

Özetle, önerilen LevySAO algoritmasının YSA ve orijinal SAO algoritmasının yerel minimuma takılma sorununu çözdüğü, performansının önemli ölçüde iyileştirildiği ve literatüre önemli bir katkı sağlandığı görülebilir.

6.2. Öneriler

Bu çalışmada önerilen LevySAO algoritması, sınıflandırma problemleri üzerinde çalıştırılmıştır. Gelecekteki çalışmalarda, LevySAO algoritması yerel minimuma takılma riski olan sürekli optimizasyon problemlerinde test edilebilir. Ayrıca algoritmada gerekli ayarlamalar yapılarak ikili veya kombinatoriyal problemlere de uygulanabilir. Büyük boyutlu veri setlerinde başarısını ölçmek amacıyla test edilebilir.

7. KAYNAKLAR

- Abd Elaziz, M., Al-qaness, M. A., Ibrahim, R. A., Ewees, A. A., & Shrahili, M. 2024. Multilevel thresholding Aerial image segmentation using comprehensive learning-based Snow ablation optimizer with double attractors. *Egyptian Informatics Journal*, 27, 100500.
- Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. 2022. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, 191, 116158.
- Akay, B. 2009. *Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi (Performance analysis of artificial bee colony algorithm on numerical optimization problems)*. Erciyes University, Turkey,
- Aljarah, I., Faris, H., & Mirjalili, S. 2018. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22, 1-15.
- Anderson, D., & McNeill, G. 1992. Artificial neural networks technology. *Kaman Sciences Corporation*, 258(6), 1-83.
- Ansari, A., Ahmad, I. S., Bakar, A. A., & Yaakub, M. R. 2020. A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction. *IEEE access*, 8, 176640-176650.
- Aslan, C., Seyyarer, E., & Uçkan, T. 2023. Honey Badger Optimizasyon Algoritması ile Üç Elemanlı Kafes Sisteminin Ağırlık ve Maliyet Minimizasyonu. *Çukurova Üniversitesi Mühendislik Fakültesi Dergisi*, 38(2), 441-449.
- Aslan, M., Gunduz, M., & Kiran, M. S. 2023. A Jaya-based approach to wind turbine placement problem. *Energy Sources, Part A: Recovery, Utilization, Environmental Effects*, 45(2), 3318-3337.
- Ataseven, B. 2013. Yapay sinir ağları ile öngörü modellemesi. *Öneri Dergisi*, 10(39), 101-115.
- Atila, Ü., Dorterler, M., & Şahin, İ. 2018. Yapay Alg Algoritmasının Tasarım Optimizasyon Problemlerindeki Performansı Üzerine Bir Çalışma: Basınç Yayı Örneği. *Bilişim Teknolojileri Dergisi*, 11(4), 349-355.
- Aytuğ, O. 2013. Metasezgisel Yöntemler ve Uygulama Alanları. *Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 17(2).
- Bairathi, D., & Gopalani, D. 2021. Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: multiple leader salp swarm algorithm. *Evolutionary Intelligence*, 14(3), 1233-1249.

- Bas, E., Egrioglu, E., & Cansu, T. 2024. Robust training of median dendritic artificial neural networks for time series forecasting. *Expert Systems with Applications*, 238, 122080.
- Bas, E., Egrioglu, E., & Yolcu, U. 2020. A hybrid algorithm based on artificial bat and backpropagation algorithms for multiplicative neuron model artificial neural networks. *Journal of Ambient Intelligence Humanized Computing*, 1-9.
- Bendine, K., Pereira, J. L. J., & Gomes, G. F. 2023. *Energy harvesting enhancement of nonuniform functionally graded piezoelectric beam using artificial neural networks and Lichtenberg algorithm*. Paper presented at the Structures.
- Bernard, J., Popescu, E., & Graf, S. 2022. Improving online education through automatic learning style identification using a multi-step architecture with ant colony system and artificial neural networks. *Applied Soft Computing*, 131, 109779.
- Bhullar, A. K., Kaur, R., & Sondhi, S. 2022. Optimization of fractional order controllers for AVR system using distance and levy-flight based crow search algorithm. *IETE Journal of Research*, 68(5), 3900-3917.
- Brown, C. T., Liebovitch, L. S., & Glendon, R. 2007. Lévy flights in Dobe Ju/'hoansi foraging patterns. *Human Ecology*, 35, 129-138.
- Bui, D.-K., Nguyen, T., Chou, J.-S., Nguyen-Xuan, H., & Ngo, T. D. 2018. A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. *Construction and Building Materials*, 180, 320-333.
- Bui, D.-K., Nguyen, T. N., Ngo, T. D., & Nguyen-Xuan, H. 2020. An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings. *Energy*, 190, 116370.
- Bulut, C. 2018. *Beyin bilgisayar arayüzü uygulamaları için EEG sinyal analizi*. Altınbaş Üniversitesi,
- Chen, X., Yi, Z., Zhou, Y., Guo, P., Farkoush, S. G., & Niroumandi, H. 2021. Artificial neural network modeling and optimization of the Solid Oxide Fuel Cell parameters using grey wolf optimizer. *Energy Reports*, 7, 3449-3459.
- Cinar, A. C., & Kiran, M. S. 2018. Similarity and logic gate-based tree-seed algorithms for binary optimization. *Computers Industrial Engineering*, 115, 631-646.
- Ciregan, D., Meier, U., & Schmidhuber, J. 2012, 16-21 June 2012. *Multi-column deep neural networks for image classification*. Paper presented at the 2012 IEEE conference on computer vision and pattern recognition, Providence, RI, USA.
- Çelik, İ., Yıldız, C., & Şekkelı, M. 2018. Rüzgâr Enerji Santrali kurulumunda rüzgâr türbinlerinin mikro yerleşimi için bir optimizasyon modeli. *Gazi University Journal of Science Part C: Design Technology*, 6(4), 898-908.

- Dalal, A.-A., AlRassas, A. M., Al-qaness, M. A., Cai, Z., Aseeri, A. O., Abd Elaziz, M., & Ewees, A. A. 2023. TLIA: Time-series forecasting model using long short-term memory integrated with artificial neural networks for volatile energy markets. *Applied Energy*, 343, 121230.
- Deng, L., & Liu, S. 2023. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Systems with Applications*, 225, 120069.
- Ding, L., Bai, Y.-L., Fan, M.-H., Song, W., & Ren, H.-h. 2024. Using a snow ablation optimizer in an autonomous echo state network for the model-free prediction of chaotic systems. *Nonlinear Dynamics*, 1-18.
- Egrioglu, E., Baş, E., & Chen, M.-Y. 2022. Recurrent dendritic neuron model artificial neural network for time series forecasting. *Information Sciences*, 607, 572-584.
- El-Kenawy, E.-S. M., Khodadadi, N., Mirjalili, S., Abdelhamid, A. A., Eid, M. M., & Ibrahim, A. 2024. Greylag goose optimization: nature-inspired optimization algorithm. *Expert Systems with Applications*, 238, 122147.
- Elangasinghe, M., Singhal, N., Dirks, K., Salmond, J., & Samarasinghe, S. 2014. Complex time series analysis of PM10 and PM2. 5 for a coastal site using artificial neural network modelling and k-means clustering. *Atmospheric Environment*, 94, 106-116.
- Ergun, U., & Tayfun, D. 2020. Jaya algoritması ile optimize edilmiş yapay sinir ağlarını kullanarak Türkiye’de elektrik enerjisi tüketiminin tahmini. *Gazi University Journal of Science Part C: Design Technology*, 8(3), 511-528.
- Erilli, N. A., Yolcu, U., Eğrioğlu, E., Aladağ, Ç. H., & Öner, Y. 2011. Determining the most proper number of cluster in fuzzy clustering by using artificial neural networks. *Expert Systems with Applications*, 38(3), 2248-2252.
- Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. 2020. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377.
- Gao, J., Gao, F., Ma, Z., Huang, N., & Yang, Y. 2021. Multi-objective optimization of smart community integrated energy considering the utility of decision makers based on the Lévy flight improved chicken swarm algorithm. *Sustainable Cities and Society*, 72, 103075.
- Gao, Y., Zhang, H., Duan, Y., & Zhang, H. 2023. A novel hybrid PSO based on levy flight and wavelet mutation for global optimization. *Plos one*, 18(1), e0279572.
- Ghiassi, M., & Burnley, C. 2010. Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems. *Expert Systems with Applications*, 37(4), 3118-3128.

- Gong, S., Gao, W., & Abza, F. 2020. Brain tumor diagnosis based on artificial neural network and a chaos whale optimization algorithm. *Computational Intelligence*, 36(1), 259-275.
- Gölcük, İ., Ozsoydan, F. B., & Durmaz, E. D. 2023. An improved arithmetic optimization algorithm for training feedforward neural networks under dynamic environments. *Knowledge-Based Systems*, 263, 110274.
- Guo, S., Kou, H., Bi, Y., & Mamlooki, M. 2024. Predicting the compressive strength of self-compacting concrete by developed African vulture optimization algorithm-Elman neural networks. *Scientific Reports*, 14(1), 20080.
- Gupta, S., & Deep, K. 2020. A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. *Applied Intelligence*, 50(4), 993-1026.
- Gülcü, Ş. 2022. An improved animal migration optimization algorithm to train the feed-forward artificial neural networks. *Arabian Journal for Science Engineering*, 47(8), 9557-9581.
- Gülcü, Ş. 2022. Training of the feed forward artificial neural networks using dragonfly algorithm. *Applied Soft Computing*, 124, 109023.
- Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. 2022. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics Computers in Simulation*, 192, 84-110.
- Hebb Donald, O. 1949. The organization of behavior. *J Clin Psychol. new york: Wiley*, 6(3), 335-307.
- Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. 2020. Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 94, 103731.
- Irmak, B. 2022. Yapay sinir ağlarının eğitimi için kelebek optimizasyonu algoritmasının iyileştirilmesi.
- Irmak, B., Karakoyun, M., & Gülcü, Ş. 2023. An improved butterfly optimization algorithm for training the feed-forward artificial neural networks. *Soft Computing*, 27(7), 3887-3905.
- Ismaeel, A. A., Houssein, E. H., Khafaga, D. S., Aldakheel, E. A., Abdelrazek, A. S., & Said, M. 2024. Performance of Snow Ablation Optimization for Solving Optimum Allocation of Generator Units. *IEEE access*.
- Jamal, R., Zhang, J., Men, B., Khan, N. H., Youssef, A.-R., Ebeed, M., & Kamel, S. 2023. Optimal scheduling of short-term hydrothermal with integration of renewable energy resources using Lévy spiral flight artificial hummingbird algorithm. *Energy Reports*, 10, 2756-2777.

- Jayasimha, S., Lingaraju, K., & Raju, H. 2022. Prediction of surface finish in extrusion honing process by regression analysis and artificial neural networks. *Applications in Engineering Science*, 10, 100105.
- Jia, H., You, F., Wu, D., Rao, H., Wu, H., & Abualigah, L. 2023. Improved snow ablation optimizer with heat transfer and condensation strategy for global optimization problem. *Journal of Computational Design Engineering Structures*, 10(6), 2177-2199.
- Jitchaijaroen, W., Kumar, D. R., Keawsawasvong, S., Wipulanusat, W., & Jamsawang, P. 2024. Hybrid artificial neural network models for bearing capacity evaluation of a strip footing on sand based on Bolton failure criterion. *Transportation Geotechnics*, 101347.
- Karaboga, D., & Akay, B. 2007. *Artificial bee colony (ABC) algorithm on training artificial neural networks*. Paper presented at the 2007 IEEE 15th Signal Processing and Communications Applications.
- Karaboga, D., & Akay, B. 2009. A comparative study of artificial bee colony algorithm. *Applied mathematics computational Intelligence*, 214(1), 108-132.
- Karakoyun, M. 2024. Artificial neural network training using a multi selection artificial algae algorithm. *Engineering Science Technology, an International Journal*, 53, 101684.
- Kaya, E. 2022. A comprehensive comparison of the performance of metaheuristic algorithms in neural network training for nonlinear system identification. *Mathematics*, 10(9), 1611.
- Kaya, E., Korkmaz, S., Sahman, M. A., & Cinar, A. C. 2021. DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets. *Expert Systems with Applications*, 169, 114482.
- Kazemi, R., Golafshani, E. M., & Behnood, A. 2023. Compressive strength prediction of sustainable concrete containing waste foundry sand using metaheuristic optimization-based hybrid artificial neural network. *Structural Concrete*.
- Koçak, Y. 2022. *Geleneksel regresyon analizi ile yapay sinir ağlarının farklı aktivasyon fonksiyonları kullanılarak karşılaştırılması*. Necmettin Erbakan University (Turkey),
- Koçak, Y., & Şiray, G. Ü. 2021. New activation functions for single layer feedforward neural network. *Expert Systems with Applications*, 164, 113977.
- Kong, M., Tian, P., & Kao, Y. 2008. A new ant colony optimization algorithm for the multidimensional knapsack problem. *Computers Operations Research*, 35(8), 2672-2683.

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
- Li, J., Soradi-Zeid, S., Yousefpour, A., & Pan, D. 2024. Improved differential evolution algorithm based convolutional neural network for emotional analysis of music data. *Applied Soft Computing*, 153, 111262.
- Li, P., Zhang, Y., Gu, J., & Duan, S. 2024. Prediction of compressive strength of concrete based on improved artificial bee colony-multilayer perceptron algorithm. *Scientific Reports*, 14(1), 6414.
- Li, Q., Cui, Z., Cai, Y., & Su, Y. 2023. Multi-objective operation of solar-based microgrids incorporating artificial neural network and grey wolf optimizer in digital twin. *Solar Energy*, 262, 111873.
- Liu, X., Zeng, S., Namaiti, A., & Xin, R. 2023. Comparison between three convolutional neural networks for local climate zone classification using Google Earth Images: A case study of the Fujian Delta in China. *Ecological Indicators*, 148, 110086.
- Lu, T., Li, H., Yang, Y., Yu, Z., Lei, Z., & Gao, S. 2023. *Differential Vectors Empower Snow Ablation Optimizer*. Paper presented at the 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC).
- Maddaiah, P. N., & Narayanan, P. P. 2023. *An Improved Snow Ablation Optimizer for Stabilizing the Artificial Neural Network*. Paper presented at the International Conference on Advances in Data-driven Computing and Intelligent Systems.
- Maheswari, B., & Nagarajan, R. 2024. EduChatbot: Implementing educational Chatbot for assisting the teaching-learning process by NLP-based hybrid heuristic adopted deep learning framework. *Kybernetes*.
- Maleki, B., Ghazvini, M., Ahmadi, M. H., Maddah, H., & Shamshirband, S. 2019. Moisture estimation in cabinet dryers with thin-layer relationships using a genetic algorithm and neural network. *Mathematics*, 7(11), 1042.
- Markelle Kelly, R. L., Kolby Nottingham. The UCI Machine Learning Repository. Retrieved from <https://archive.ics.uci.edu>
- McCulloch, W. S., & Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.
- Minsky, M., & Papert, S. 1969. An introduction to computational geometry. *Cambridge tiass.*, HIT, 479(480), 104.
- Mirjalili, S. 2015. How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43, 150-161.
- Mirjalili, S. 2016. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133.

- Mirjalili, S., Mirjalili, S. M., & Lewis, A. 2014. Grey wolf optimizer. *Advances in Engineering Software*, 69, 46-61.
- Mohseni, S., Brent, A. C., Burmester, D., & Browne, W. N. 2021. Lévy-flight moth-flame optimisation algorithm-based micro-grid equipment sizing: An integrated investment and operational planning approach. *Energy and AI*, 3, 100047.
- Monika, & Bhat, A. 2022. Automatic Twitter crime prediction using hybrid wavelet convolutional neural network with world cup optimization. *International Journal of Pattern Recognition Artificial Intelligence*, 36(05), 2259005.
- Mörters, P., & Peres, Y. 2010. *Brownian motion* (Vol. 30): Cambridge University Press.
- Narayanan, S. L., Kasiselvanathan, M., Gurumoorthy, K., & Kiruthika, V. 2023. Particle swarm optimization based artificial neural network (PSO-ANN) model for effective k-barrier count intrusion detection system in WSN. *Measurement: Sensors*, 29, 100875.
- Nithya, A., Appathurai, A., Venkatadri, N., Ramji, D., & Palagan, C. A. 2020. Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images. *Measurement*, 149, 106952.
- Özden, A., & İşeri, İ. 2023. COOT optimization algorithm on training artificial neural networks. *Knowledge and Information Systems*, 65(8), 3353-3383.
- Özkış, A. 2017. *Girdap arama ve yapay alg algoritmalarının çok amaçlı optimizasyon problemlerine uyarlanması*. Doktora Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü,
- Özkış, A. 2024. A multi-population-based marine predators algorithm to train artificial neural network. *Soft Computing*, 1-28.
- Özkış, A., & Karakoyun, M. 2023. Kapasitesiz tesis yerleşim problemleri için geliştirilmiş ikili güve alevi optimizasyon algoritması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 29(7), 737-751.
- Öztemel, E. 2003. *Yapay sinir ağları*. PapatyaYayincılık, İstanbul.
- Pandya, S. B., Kalita, K., Çep, R., Jangir, P., Chohan, J. S., & Abualigah, L. 2024. Multi-objective Snow Ablation Optimization Algorithm: An Elementary Vision for Security-Constrained Optimal Power Flow Problem Incorporating Wind Energy Source with FACTS Devices. *International Journal of Computational Intelligence Systems*, 17(1), 33.
- Pu, Y., Song, J., Wu, M., Xu, X., & Wu, W. 2023. Node location using cuckoo search algorithm with grouping and drift strategy for WSN. *Physical Communication*, 59, 102088.
- Qaddoura, R., Faris, H., Aljarah, I., & Castillo, P. A. 2021. Evocluster: an open-source nature-inspired optimization clustering framework. *SN Computer Science*, 2, 1-12.

- Qiao, W., Khishe, M., & Ravakhah, S. 2021. Underwater targets classification using local wavelet acoustic pattern and Multi-Layer Perceptron neural network optimized by modified Whale Optimization Algorithm. *Ocean Engineering*, 219, 108415.
- Raziani, S., Ahmadian, S., Jalali, S. M. J., & Chalechale, A. 2022. An efficient hybrid model based on modified whale optimization algorithm and multilayer perceptron neural network for medical classification problems. *Journal of Bionic Engineering*, 19(5), 1504-1521.
- Rojas, M. G., Olivera, A. C., & Vidal, P. J. 2024. A genetic operators-based Ant Lion Optimiser for training a medical multi-layer perceptron. *Applied Soft Computing*, 151, 111192.
- Rosenblatt, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
- Saji, Y., & Barkatou, M. 2021. A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem. *Expert Systems with Applications*, 172, 114639.
- Selim, A., Shuvo, S. N. A., Islam, M., Moniruzzaman, M., Shah, S., & Ohiduzzaman, M. 2023. Predictive models for dissolved oxygen in an urban lake by regression analysis and artificial neural network. *Total Environment Research Themes*, 7, 100066.
- Si, T., Bagchi, J., & Miranda, P. B. 2022. Artificial neural network training using metaheuristics for medical data classification: an experimental study. *Expert Systems with Applications*, 193, 116423.
- Song, Y., Meng, X., & Jiang, J. 2022. Multi-layer perception model with elastic grey wolf optimization to predict student achievement. *Plos one*, 17(12), e0276943.
- Storn, R., & Price, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341-359.
- Talaat, M., Farahat, M., Mansour, N., & Hatata, A. 2020. Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach. *Energy*, 196, 117087.
- Tam, V. W., Butera, A., Le, K. N., Da Silva, L. C., & Evangelista, A. C. 2022. A prediction model for compressive strength of CO₂ concrete using regression analysis and artificial neural networks. *Construction Building Materials*, 324, 126689.

- Thamaraimanalan, T., & Ramalingam, S. 2024. Hybrid Artificial Neural Network-based Grasshopper Optimization Algorithm for Anomaly Detection in Wireless Body Area Networks. *IETE Journal of Research*, 1-15.
- Thankachan, P., Fida, A., & Pillai, T. M. 2024. *Health monitoring of steel structures using Cuckoo Search algorithm-based ANN*. Paper presented at the Structures.
- Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., & Wahab, M. A. 2019. An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm. *Engineering Structures*, 199, 109637.
- Turing, A. M. 2009. *Computing machinery and intelligence*: Springer.
- Turkoglu, B., & Kaya, E. 2020. Training multi-layer perceptron with artificial algae algorithm. *Engineering Science Technology, an International Journal*, 23(6), 1342-1350.
- Türkoğlu, B. 2019. *Yapay sinir ağlarının yapay alg algoritması ile eğitimi*. Selçuk Üniversitesi,
- Uymaz, S. A., Tezel, G., & Yel, E. 2015. Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, 153-171.
- Wang, C., He, Q., Li, Z., Yu, J., Bello, I. T., Zheng, K., . . . Ni, M. 2024. A novel in-tube reformer for solid oxide fuel cell for performance improvement and efficient thermal management: A numerical study based on artificial neural network and genetic algorithm. *Applied Energy*, 357, 122030.
- Wang, Y., Yu, Y., Zhang, T., Song, K., Wang, Y., & Gao, S. 2024. Improved Dendritic Learning: Activation Function Analysis. *Information Sciences*, 121034.
- Widrow, B., & Hoff, M. 1960. *Adaptive switching circuits (pp. 96–104)*. Paper presented at the Institute of Radio Engineers (IRE) WESCON Convention record.
- Wilcoxon, F., Katti, S., & Wilcox, R. A. 1970. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1, 171-259.
- Wolpert, D. H., & Macready, W. G. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.
- Wu, L., Mei, J., & Zhao, S. 2023. Pipeline damage identification based on an optimized back-propagation neural network improved by whale optimization algorithm. *Applied Intelligence*, 53(10), 12937-12954.
- Xiao, Y., Cui, H., Hussien, A. G., & Hashim, F. A. 2024. MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications. *Advanced Engineering Informatics*, 61, 102464.

- Xu, B., Su, J., Dale, D., & Watson, M. 2000. Cotton color grading with a neural network *Textile Research Journal*, 70(5), 430-436.
- Yaman, S., Karakaya, B., & K k m, M. 2024. A neural network approach for classification of fault-slip data in geoscience. *Ain Shams Engineering Journal*, 15(1), 102325.
- Yang, S., Tian, X., Zhang, Q., Jiang, J., Dong, P., Tan, J., . . . Song, J. 2023. Microorganism inspired hydrogels: Optimization by response surface methodology and genetic algorithm based on artificial neural network. *European Polymer Journal*, 201, 112497.
- Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. 2013. *Swarm intelligence and bio-inspired computation: theory and applications*: Newnes.
- Yang, X.-S., & Deb, S. 2009. *Cuckoo search via L vy flights*. Paper presented at the 2009 World congress on nature & biologically inspired computing (NaBIC).
- Yılmaz, M. 2022. *Evrimsel hesaplama algoritmaları ile yapay sinir ađının bađlantı optimizasyonu*. Konya Teknik  niversitesi,
- Yuan, H., Liu, Y., Li, J., & Zhao, W. 2024. *Optimize Sparse Antenna Arrays with Chaotic Mapping Snow Ablation Optimizer*. Paper presented at the 2024 IEEE 14th International Conference on Electronics Information and Emergency Communication (ICEIEC).
- Zamani, H., & Nadimi-Shahraki, M. H. 2024. An evolutionary crow search algorithm equipped with interactive memory mechanism to optimize artificial neural network for disease diagnosis. *Biomedical Signal Processing Control*, 90, 105879.