



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**DESTEK VEKTÖR MAKİNELERİ
PARAMETRELERİNİN META SEZGİSEL
OPTİMİZASYON ALGORİTMALARI İLE
BELİRLENMESİ**

Burak TEZCAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

**Ağustos - 2018
KONYA
Her Hakkı Saklıdır**

TEZ KABUL VE ONAYI

Burak TEZCAN tarafından hazırlanan “Destek Vektör Makineleri Parametrelerinin Meta Sezgisel Optimizasyon Algoritmaları İle Belirlenmesi” adlı tez çalışması 15/08/2018 tarihinde aşağıdaki jüri tarafından oy birliği /oy çokluğu ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Mesut GÜNDÜZ

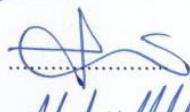
Danışman

Prof. Dr. Şakir TAŞDEMİR

Üye

Dr. Öğr. Üyesi Abdullah Erdal TÜMER

İmza


Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ
FBE Müdürü

TEZ BİLDİRİMİ

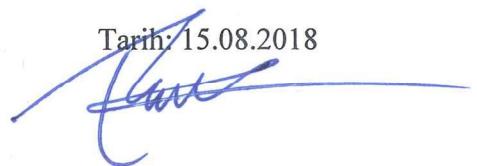
Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

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.

Burak TEZCAN

Tarih: 15.08.2018



ÖZET

YÜKSEK LİSANS TEZİ

DESTEK VECTÖR MAKİNELERİ PARAMETRELERİNİN META SEZGİSEL OPTİMİZASYON ALGORİTMALARI İLE BELİRLENMESİ

Burak TEZCAN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Şakir TAŞDEMİR

2018, 81 Sayfa

Jüri

Prof. Dr. Şakir TAŞDEMİR
Doç. Dr. Mesut GÜNDÜZ
Dr. Öğr. Üyesi Abdullah Erdal TÜMER

Destek Vektör Makineleri (DVM) sınıflandırma amacıyla 90'lı yıllarda ortaya çıkan popülerliği gittikçe artan bir algoritmadır. DVM'lerin birçok farklı alanda başarılı olduğu gözlemlenmiştir ve bu başarı sınıflandırma alanında en çok kullanılan yöntemlerden olmasını sağlamıştır. Son yıllarda, DVM eğitim aşamasının ağır hesaplama karmaşıklığı sebebiyle DVM parametre optimizasyonu araştırmaları hız kazanmıştır. DVM parametre seçimi çok önemlidir. Hatalı seçimler yöntemin sınıflandırma performansını etkileyebilecegi gibi hesaplama karmaşıklığını da artırabilir. Bu yüzden parametre ayarlanması için uygun optimizasyon algoritmaları aranmaktadır. Bu tez çalışmasında yukarıdaki problemler meta-sezgisel tabanlı optimizasyon modeli olarak ele alındı. Kullanılan yöntemler Karga Arama Algoritması (KAA) ve Salp Sürüsü Algoritması (SSA) olarak bilinen yeni yöntemlerdir. Bu iki algoritma meta-sezgisel yöntemlerin özelliklerini başarılı bir şekilde taşımaktadır.

Bu tez çalışmasında KAA ve SSA algoritmaları kullanılarak DVM optimizasyon modelleri oluşturulmuştur. Bu modeller DVM sınıflandırma doğruluğu değerlerini uygunluk değerleri olarak ele alarak DVM parametrelerini KAA ve SSA üzerinden geliştirmektedir. İşlem sonunda elde edilen en iyi parametre değerleri 10 katlamalı çapraz doğrulama kullanılan DVM uygulaması ile değerlendirildi.

Çalışmadaki deneyler UCI (*machine learning repository of University of California at Irvine*) veri tabanında alınan farklı standart veri setleri üzerinde denendi. Elde edilen sonuçlar doğruluk, duyarlılık, seçicilik ve AUC kriterleri kullanılarak karşılaştırılmıştır. Ayrıca sonuçların ROC eğrileri ve kutu grafiklerine yer verilmiştir. Tüm deneyler lineer, polinom ($d=1,2,3$) ve RBF çekirdek fonksiyonları kullanılarak gerçekleştirilmiştir. Çalışmada kullanılan göğüs kanseri, diyabet, parkinson, sonar, şarap, cam ve ünlü harf veri setleri ile elde edilen en iyi doğruluk sonuçları yüzdesel olarak sırasıyla şöyledir: 97.22, 78.25, 95.38, 88.50, 97.78, 67.77 ve 99.62. KAA-DVM'nin SSA-DVM'ye göre daha başarılı olduğu görülmüştür. DVM çekirdek fonksiyonlarından RBF veri setlerinin çoğunluğunda en iyi sonuçları vermiştir. Literatür karşılaşmaları elde edilen modellerin diğer çalışmalar ile rekabet edebilecek seviyede olduklarını göstermiştir. Sonuçlar göstermektedir ki önerilen modeller uygun DVM parametrelerinin belirlenmesi yeteneğine sahiptir.

Anahtar Kelimeler: Destek Vektör Makineleri, Karga Arama Algoritması, Salp Sürüsü Algoritması, Meta-Sezgisel Optimizasyon, Parametre Ayarlaması

ABSTRACT

MS THESIS

DETERMINATION OF SUPPORT VECTOR MACHINE PARAMETERS BY META HEURISTIC OPTIMIZATION ALGORITHMS

Burak TEZCAN

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Prof. Dr. Şakir TAŞDEMİR

2018, 81 Pages

Jury

**Prof. Dr. Şakir TAŞDEMİR
Assoc. Prof. Dr. Mesut GÜNDÜZ
Asst. Prof. Dr. Abdullah Erdal TÜMER**

Support Vector Machines (SVMs) are first appeared in 1990s and its popularity has been grown over the years. SVMs are shown very successful results in a wide range of areas and become a widely used method for classification purposes. In recent years, due to high computational burden of the SVM training phase, SVM parameter optimization is on the rise. SVM parameter tuning is important because wrong selection of parameters can decrease its classification performance and add computational burden. For this reason, new processes are required for the parameter optimization. We consider these problems as a meta-heuristic optimization approach. The meta-heuristic approaches used in this study are Crow Search Algorithm (CSA) and Salp Swarm Algorithm (SSA). These two algorithms carry the advantages of meta-heuristics.

In this thesis, two models using CSA and SSA are created for SVM parameter optimization. These models use SVM classification accuracy as fitness values and improves parameters using CSA and SSA. After the parameter optimization process, SVM evaluation is done using the best parameter values over 10-fold cross validation.

Different standard datasets from UCI (*machine learning repository of University of California at Irvine*) database were used in the study. The results are compared over accuracy, sensitivity, specificity and AUC. Also, ROC curves and boxplots of the evaluation phase were given in the study. All experiments are done using three different SVM kernels: linear, polynomial ($d=1,2,3$) and RBF. The results for the datasets breast cancer, diabetes, Parkinson, sonar, wine, glass and vowel are 97.22, 78.25, 95.38, 88.50, 97.78, 67.77 and 99.62 respectively. CSA model gave better results than SSA model. The RBF kernel has shown the best results for most of the datasets. Literature comparisons show that our models can compete with other studies. The experiments show that our models are successful to find suitable SVM parameters.

Keywords: Support Vector Machines, Crow Search Algorithm, Salp Swarm Algorithm, Meta-heuristic optimization, Parameter tuning

ÖNSÖZ

Yüksek lisans tezi olan bu çalışmada, “Destek Vektör Makineleri Parametrelerinin Meta Sezgisel Optimizasyon Algoritmaları İle Belirlenmesi” araştırılmıştır. Tez çalışması sürecinde değerli katkılarını ve desteklerini esirgemeyen değerli Danışman Hocam Prof. Dr. Şakir TAŞDEMİR’e, Dr. Öğretim Üyesi Tahir SAĞ’a, değerli anabilim dalı başkanı ve öğretim üyelerine, her an benimle olan ve destekleyen aileme ve tüm sevdiklerime sonsuz teşekkürlerimi sunarım.

Burak TEZCAN
KONYA-2018



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ	1
1.1. Çalışmanın Önemi	2
1.2. Çalışmanın Amacı ve Yöntemi	4
1.3. Kaynak Araştırması	5
1.2.1. Kaynak araştırması değerlendirmesi	9
1.3. Tezin Organizasyonu	10
2. OPTİMİZASYON.....	12
2.1. Sezgisel ve Meta-sezgisel Kavramları	13
2.2. Karga Arama Algoritması (Crow Search Algorithm)	14
2.2.1. KAA matematiksel modeli	15
2.2.2. KAA optimizasyon modeli	17
2.3. Salp Sürüsü Algoritması (Salp Swarm Algorithm)	19
2.3.1. SSA matematiksel modeli	20
2.3.2. SSA simülasyonları	21
2.2.3. SSA optimizasyon modeli	24
2.4. Bölüm Değerlendirmesi	24
3. DESTEK VEKTÖR MAKİNELERİ (SUPPORT VECTOR MACHINES).....	26
3.1. Maximal Margin Ayırıcı	26
3.2. Maximal Margin Sınıflandırıcı	28
3.3. Esnek Margin Sınıflandırıcı	29
3.4. Çekirdek Fonksiyonlarının Etkisi	31
3.5. Bölüm Değerlendirmesi	33
4. DVM PARAMETRELERİNİN OPTİMİZASYONU	35
4.1. Veri Setleri	35
4.2. Deney Ortamı ve Başlangıç Değerleri	37
4.3. Performans Değerlendirme Kriterleri	38
4.4. Deneyler	40

4.4.1. Göğüs kanseri veri seti deneyleri	40
4.4.2. Diyabet veri seti deneyleri	45
4.4.3. Parkison veri seti deneyleri	49
4.4.4. Sonar veri seti deneyleri	53
4.5. Çok Sınıflı Veri Setleri ve Literatür Karşılaştırması	57
4.6. Bölüm Değerlendirmesi	58
5. SONUÇLAR VE ÖNERİLER	60
KAYNAKLAR	63
ÖZGEÇMİŞ	68



ŞEKİL LİSTESİ

Şekil 2.1 KAA'nın Durum 1 için akış şeması (a) $uu < 1$ için (b) $uu > 1$ için. Karga i kesik çizgili alanın her yerine hareket edebilir.....	2
Şekil 2.2 (a) Tekil Salp canlısı (b) Salp Sürüsü (Salp zinciri).....	3
Şekil 2.3 (sol resim) Tekil Salp Canlısı (NEFSC, 2013) - (sağ resim) Salp Zinciri.....	3
Şekil 2.4 Salp zincirinin sabit besin kaynağı etrafında 2-boyutlu uzayda hareketi	4
Şekil 2.5 100 iterasyon sonunda, 2-boyutlu uzayda sabit ve mobil besin kaynakları etrafındaki arama geçmişi	4
Şekil 2.6 Salp zincirinin sabit besin kaynağı etrafında 3-boyutlu uzayda hareketi	5
Şekil 2.7 100 iterasyon sonunda, 3-boyutlu uzayda sabit ve mobil besin kaynakları etrafındaki arama geçmişi	5
Şekil 3.1 Hiperdüzlem $1+2X_1+3X_2=0$ görülmektedir. Mavi bölge $1+2X_1+3X_2>0$ 'ı ifade der. Kırmızı bölge ise $1+2X_1+3X_2<0$ 'ı ifade eder.....	7
Şekil 3.2 Üç farklı ayırcı hiperdüzlem durumu	8
Şekil 3.3 İki sınıf bulunmaktadır; mavi ve mor. Maximal margin hiperdüzlem koyu gösterilmiştir. Margin çizgili ısimların herhangibirinden hiperdüzleme kadar olan alandır. Çizgili kısımların üzerinde bulunan mavi noktalar destek vektörleridir .	9
Şekil 3.4 Bir gözlemin değişmesinin hiperdüzleme etkisi.....	10
Şekil 3.5 C parametresinin hiperdüzlem üzerine etkisi	11
Şekil 3.6 (sol) polinom çekirdek fonksiyonu ($d=3$) uygulanmış örnekler. (sağ) RBF çekirdek fonksiyonu uygulanmış örnekler.....	14
Şekil 4.1 KAA-DVM ve SSA-DVM modelleri akış şeması	16
Şekil 4.2 Kutu grafiği yapısı	20
Şekil 4.3 Bir kötü huylu göğüs aspirasyon iğnesi görüntüsü.....	21
Şekil 4.4 KAA-DVM'nin göğüs kanseri veri seti üzerindeki kutu grafiği	22
Şekil 4.5 SSA-DVM'nin göğüs kanseri veri seti üzerindeki kutu grafiği	24

Şekil 4.6 KAA-DVM ve SSA-DVM'nin göğüs kanseri veri seti üzerindeki karşılaştırmalı kutu grafiği	24
Şekil 4.7 . KAA-DVM göğüs kanseri veri seti test verileri üzerindeki ROC eğrisi	25
Şekil 4.8 SSA-DVM göğüs kanseri veri seti test verileri üzerindeki ROC eğrisi	25
Şekil 4.9 KAA-DVM'nin diyabet veri seti üzerindeki kutu grafiği	27
Şekil 4.10 SSA-DVM'nin diyabet veri seti üzerindeki kutu grafiği	29
Şekil 4.11 KAA-DVM ve SSA-DVM'nin diyabet veri seti üzerindeki karşılaştırmalı kutu grafiği	29
Şekil 4.12 KAA-DVM diyabet veri seti üzerindeki ROC eğrisi	30
Şekil 4.13 SSA-DVM diyabet veri seti üzerindeki ROC eğrisi.....	30
Şekil 4.14 KAA-DVM parkinson veri seti üzerindeki kutu grafiği.....	33
Şekil 4.15 SSA-DVM parkinson veri seti üzerindeki kutu grafiği	33
Şekil 4.16 KAA-DVM ve SSA-DVM'nin parkinson veri seti üzerindeki karşılaştırmalı kutu grafiği	34
Şekil 4.17 KAA-DVM parkinson veri seti üzerindeki ROC eğrisi	35
Şekil 4.18 SSA-DVM parkinson veri seti üzerindeki ROC eğrisi.....	35
Şekil 4.19 KAA-DVM sonar veri seti üzerindeki kutu grafiği.....	38
Şekil 4.20 SSA-DVM sonar veri seti üzerindeki kutu grafiği	39
Şekil 4.21 KAA-DVM ve SSA-DVM'nin sonar veri seti üzerindeki karşılaştırmalı kutu grafiği	39
Şekil 4.22 KAA-DVM sonar veri seti üzerindeki ROC eğrisi	41
Şekil 4.23 SSA-DVM sonar veri seti üzerindeki ROC eğrisi.....	41
Şekil 4.24. KAA ve SSA model performanslarının grafiksel literatür karşılaştırması	55

ÇİZELGE LİSTESİ

Çizelge 2.1 Karga Arama Algoritması sözde kodu	4
Çizelge 2.2 SSA algoritmasının sözde kodu.....	7
Çizelge 4.1 Deneylerde kullanılan veri setleri	17
Çizelge 4.2 Algoritmaların başlangıç değerleri	19
Çizelge 4.3 Wisconsin göğüs kanseri veri setinin öznitelikleri	22
Çizelge 4.4 Wisconsin göğüs kanseri veri seti üzerinde detaylı sınıflandırma sonuçları	23
Çizelge 4.5 KAA ve SSA'nın göğüs kanseri veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması.....	27
Çizelge 4.6 Diyabet veri setinin öznitelikleri	27
Çizelge 4.7 Pima Kızılderilileri diyabet veri seti üzerinde detaylı sınıflandırma sonuçları	28
Çizelge 4.8 KAA ve SSA'nın diyabet veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması	32
Çizelge 4.9 Parkinson veri setinin 22 niteliği	32
Çizelge 4.10 Parkinson veri seti üzerinde detaylı sınıflandırma sonuçları	33
Çizelge 4.11 KAA ve SSA'nın parkinson veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması	38
Çizelge 4.12 Sonar veri seti üzerinde detaylı sınıflandırma sonuçları	39
Çizelge 4.13 KAA ve SSA'nın sonar veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması	43
Çizelge 4.14 KAA ve SSA model performanslarının literatür karşılaştırması.....	45
Çizelge 4.15 Literatür karşılaştırması yapılan çalışmalar.....	46

SİMGELER VE KISALTMALAR

Simgeler

B_j	: $j.$ boyuttaki besin kaynağı
$\beta_0, \beta_1, \beta_2$: Hiperdüzlem katsayıları
C	: Hiperdüzlem ayarlama parametresi
c_1	: Rastgele sayı
c_2	: Rastgele sayı
c_3	: Rastgele sayı
d	: Polinom çekirdek fonksiyonu derecesi
f	: Uygunluk Fonksiyonu
FO	: Farkındalık Olasılığı
itr_{max}	: Azami iterasyon sayısı
l	: Mevcut iterasyon
L	: Azami iterasyon sayısı
lb_j	: $j.$ boyut için alt sınır
$m^{i,itr}$: i kargasının itr iterasyonundaki hafızası
N	: Karga Sayısı
$SCCO2$: Süperkritik karbondioksit
r_i	: 0-1 aralığında rastgele sayı
t	: Zaman
ub_j	: $j.$ boyut için üst sınır
uu	: Uçuş Uzaklığı
$x^{i,itr}$: i kargasının itr iterasyonundaki pozisyonu
v_o	: Hız
γ	: Çekirdek fonksiyonu parametresi

Kısaltmalar

ACO	: Karınca Kolonisi Algoritması
ACROA	: Yapay Kimyasal Reaksiyon Algoritması
AFSA	: Yapay Balık Sürüsü Algoritması
ALO	: Karıncaaslanı Optimizasyonu
AUC	: Area Under the ROC Curve
BA	: Yarasa Algoritması
CALO	: Kaotik Karıncaaslanı Optimizasyon Algoritması
CMA-ES	: Covariance Matrix Adaptation Evolution Strategy
CPSO	: Kaotik Parçacık Sürüsü Optimizasyonu
CSA	: Crow Search Algorithm – Guguk-kuşu Algoritması
ÇD	: Çapraz Doğrulama
DA	: Yusufçuk Algoritması
DVM	: Destek Vektör Makineleri
EHA	: Fil Sürüsü Algoritması
EMS	: Esnek Margin Sınıflandırıcı
FOA	: Ateşböceği Algoritması
fmGA	: Fast Messy Genetic Algorithm
FPA	: Çiçek Tozlaşması Algoritması

GA	:	Genetik Algoritma
GS	:	Izgara Arama
GSA	:	Kandilböceği Sürü Algoritması
GWO	:	Gri kurt Optimizasyon Algoritması
HGA	:	Hibrid Genetik Algoritma
HS	:	Armoni Arama Algoritması
IQR	:	Interquertile Range
KAА	:	Karga Arama Algoritması
LMD	:	Lokal Mean Decomposition
MAFSA	:	Modifiye Edilmiş Yapay Balık Süresi Algoritması
MMH	:	MAximal Margin Hiperdüzlem
MNIST	:	Modified National Institute of Standards and Technology
MÖ	:	Makine Öğrenmesi
MVO	:	Çoklu Evren Optimizasyon Algoritması
NEFSC	:	Northeast Fisheries Science Center
NFL	:	No Free Lunch
NGHS	:	Global-En iyi Armoni Arama Algoritması
NPSVM	:	Paralel Olmayan Destek Vektör Makineleri
PSO	:	Parçacık Süresi Optimizasyonu
RBF	:	Radial Basis Function
ROC	:	Receiver Operating Characteristics
RMSE	:	Root Mean Square Error
SSA	:	Salp Süresi Algoritması - Salp Swarm Algorithm
SSO	:	Sosyal Örümcek Algoritması
SVM	:	Support Vector Machines
TWSVM	:	Twin-SVM
UCI	:	University of California, Irvine Machine Learning Repository
WOA	:	Balina Optimizasyon Algoritması
YRM	:	Yapısal Risk Minimizasyonu

1. GİRİŞ

Son yıllarda öğrenme yöntemleri kullanarak veriler içerisindeki desenleri ortaya çıkarmak araştırmacıların ilgi odağı olmuştur. Bu durum Makine Öğrenmesi (MÖ) yöntemlerinin literatürde yoğun olarak kullanılmasını tetiklemiştir. Destek Vektör Makineleri (DVM) geçmiş yıllarda çok farklı bilim dallarında yoğun olarak kullanılmıştır. Bu alanlar biyometri (Vatsa ve ark., 2005), kemoinformatik (Jean-Pierre ve ark., 2007), sağlık (Xian, 2010), görüntü işleme (Gumus ve ark., 2010), sinyal işleme (Subasi ve Ismail Gursoy, 2010) meteoroloji (Chen ve ark., 2011b), ekonomi (Pai ve ark., 2014) vb. alanlarıdır. İstatistiksel öğrenme teorisine dayanan DVM, Cortes ve Vapnik (1995) tarafından geliştirilmiştir.

DVM, *maximal margin (azami aralık)* varsayıımına dayanmaktadır. Bu varsayıım verilerin çeşitli çekirdek fonksiyonları ile yüksek boyutlu özellik uzaylarına dönüştürüldüğü bir ortamda en iyi hiperdüzlem ile ayırlmasını ifade eder. DVM en iyi desen tanıma yöntemlerinden biri kabul edilse de hesaplama karmaşıklığı yüksektir. Çekirdek fonksiyonları DVM'lerin etkin çalışmasında önemli rol oynarlar. Bu çekirdek fonksiyonları çeşitli parametreler içermektedir. Parametreler DVM performansını doğrudan etkiler. Bu parametrelerin doğru belirlenmesi için optimizasyon algoritmalarına ihtiyaç duyulur. Parametrelerin doğru belirlenmesi sayesinde eğitim aşamasında sınıflandırma hatalarına daha toleranslı modeller oluşturulabilir.

Optimizasyon, verilen bir problem için en uygun çözümün bulunması işlemidir. Belirli bazı problemler için tek bir çözüm bulunabilse de diğer problemlerin birden fazla çözümü bulunabilir. Optimizasyon problemlerinin çözümünde sezgisel yöntemler sıkılıkla kullanılmaktadır. Bu nedenle sezgisel optimizasyon uygulamalarında “en iyi” çözüm, bulunabilecek en iyi ve en kesin çözümü değil, çözümler arasında diğerlerinden yeterince üstün olanı ifade eder (Nocedal ve Wright, 2006).

Optimizasyon problemlerinin çözümlerine yönelik ilk yaklaşımlar, çoğunlukla salt matematiksel işlemlere dayanıyordu. Bu çözümlerde karar değişkenleri türevin sıfır olduğu noktalar bulunarak elde edilir. Fakat çok değişkenli ve kısıtlı, lineer olmayan problemlerde bu yöntemlerin uygulanması her zaman mümkün değildir. Ayrıca birden fazla lokal optimum olabileceğiinden çözümün bu noktalarda takılması söz konusu olabilir (Miettinen, 1999; Stadler, 2013). Matematiksel olarak yapılan optimizasyon çözümlerinin sakıncalarını (türev işlemleri, karmaşıklık, lokal optimuma takılma gibi) ortadan kaldırmak amacıyla meta-sezgisel algoritmalar geliştirilmeye başlandı (Mirjalili

ve ark., 2014). Bu yöntemler rastgele operatörler kullanan basit kavamlar olarak geliştirildiler. Bu rastgelelik problem çözümünün lokal optimuma takılmasının önüne geçti. Meta-sezgisel algoritmalar optimizasyon problemlerini bir kara kutu olarak ele alır. Bu nedenle arama uzayının türevini almaya gerek kalmaz. Dolayısıyla meta-sezgisel algoritmalar uygulanması basit yöntemler olmaları, karmaşık matematiksel işlemler gerektirmemeleri ve lokal optimuma nadiren takılmaları gibi özellikleri sayesinde geniş kullanım alanına ulaştılar.

Doğadan esinlenen algoritmaların büyük çoğunluğunu biyolojiden esinlenen algoritmalar oluşturmaktadır (Fister jr ve ark., 2013). Bazı karmaşık problemlerin çözümünde biyolojiden esinlenen algoritmaların sürü zekâsı temelli yöntemler kullanılmaktadır. Sürü zekâsı bazı canlıların grup olarak kolektif davranışlarına dayanmaktadır. Örneğin karıncalar sürüsünün devamını merkezi bir kontrol olmadan kolektif bir şekilde sağlarlar. Daha açık bir ifadeyle kimse karıncalara nereye gitmeleri gerektiğini ya da besin kaynağının nerede olduğunu söylemez, fakat karıncalar iş birliği içerisinde çok uzaktaki besinleri bile bulabilirler. Karınca Kolonisi Optimizasyonu (ACO) (Dorigo ve ark., 2006) karıncaların bu kolektif sürü zekalarına dayanmaktadır. Biyolojiden esinlen algoritmaların bir kısmı ise sürü zekâsına dayanmaz. Bu algoritmalar genellikle canlıların etkileyici üreme, beslenme sistemlerinden ve zeki davranışlarından esinlenmişlerdir. Örneğin Çiçek Tozlaşma Algoritması (FPA) biyolojiden esinlenen bir algoritmadır ancak sürü zekâsı temelli değildir (Yang ve ark., 2014).

Bu çalışmada meta-sezgisel yöntemlerin büyük ve popüler kısmını oluşturan doğadan esinlenen algoritmalar kullanılmıştır. Bunlar arasında bir sürü zekâsı temelli algoritma olan Salp Sürüsü Algoritması (SSA) (Mirjalili ve ark., 2017) ve kargaların zeki davranışlarına dayanan Karga Arama Algoritması (KAA) (Askarzadeh, 2016) kullanılmıştır.

1.1. Çalışmanın Önemi

Destek vektör makineleri için çekirdek fonksiyonu optimizasyonu devam eden bir araştırma konusudur. Bu optimizasyon işlemi için çok çeşitli algoritmalar kullanılmakta ve bu alanda sürekli yeni algoritmalar önerilmektedir. Wolpert ve Macready (1997) araştırmalarında her optimizasyon problemini çözebilecek tek bir optimizasyon

metodolojisi olmadığını ifade etmektedirler (NFL – No Free Lunch Teorisi). Bu amaçla optimizasyon alanında yeni algoritmalar yazılmaktadır.

Glover (1986)'a göre tüm modern doğadan esinlenen metodlar meta-sezgisel olarak tanımlanmaktadır. Mevcut trend zor problemlerin çözümünde meta-sezgisel optimizasyonları kullanılır. Bu alanda meta-sezgisel algoritmalar başarılı sonuçlar vermişlerdir (Blum ve Roli, 2003; Yang, 2010c). Bu nedenle son 20 yılda meta-sezgisel algoritmalar literatürde hızla yükselmiştir (Yang ve Wiley, 2010). En çok bilinen meta-sezgisel algoritmalar şunlardır: doğal seçilime dayanan Genetik Algoritma (GA) (Holland, 1992), kuş ve balık sürülerinin davranışlarına dayanan Parçacık Sürüsü Optimizasyonu (PSO) (Kennedy ve Eberhart, 1995), müzik uydurma yöntemine dayanan Armoni Arama Algoritması (HS) (Zong Woo ve ark., 2001), guguk kuşlarının kuluçka parazitleri ile ilgili Guguk Kuşu Algoritması (CSA) (Yang ve Deb, 2009), yarasaların eko-lokasyon mekanizmasına dayanan Yarasa Algoritması (BA) (Yang, 2010a) ve tropik ateşböceklerinin oluşturduğu ışık desenlerine dayanan Ateşböceği Algoritması (FA) (Yang, 2010b) vb.

(Tharwat ve ark., 2017) BA yöntemi ile DVM parametrelerini optimize edecek bir model önermişlerdir. BA en yeni geliştirilen optimizasyon tekniklerindendir. BA algoritması yarasaların ekolokasyon davranışları ile uzaklık bulma ve engeller ile arasında ayırmayı yapabilmesi özelliklerini kullanır. Yarasalar yüksek bir ses çıkarırlar ve sesin gecikmesinden faydalananarak çevrelerindeki nesnelerin yerlerini algılarlar. Yansıyan ve seken seslerin arasındaki ilişki yarasaya nesnenin tipi hakkında bilgi verir. Yazalar geliştirdikleri model ile DVM için en uygun parametreleri belirlemeye çalışmışlardır. Uygulamalar UCI veri bankasından 9 adet veri seti üzerinde denenmiştir. Sonuçta görülmüşür ki BA algoritması önerilen model DVM optimal parametrelerinin belirlenmesinde başarılı olmuştur ve lokal optimuma takılma sorunu yaşamamıştır. BA-DVM algoritmasının popüler optimizasyon algoritmaları olan Parçacık Sürüsü Optimizasyonu ve Genetik Algoritmadan daha düşük sınıflandırma hatası verdiği belirlenmiştir. Dolayısıyla doğadan esinlenen meta-sezgisel algoritmaların başarılı sonuçlar elde edilmesi yönünde bekleniler her zaman yüksektir. Sürekli olarak BA ve benzeri yeni algoritmalar literatüre eklenmektedir. Ancak yeni çıkan bu algoritmalar DVM alanında henüz uygulamaya tabi tutulmamış olabiliyorlar. Bu yeni algoritmaların çeşitli alanlardaki problemler üzerinde DVM ile uygulanması, başarısının gözlemlenmesi ve DVM literatürüne kazandırılması önemlidir.

1.2. Çalışmanın Amacı ve Yöntemi

Lineer olarak ayrılabilen sınıflandırma problemlerinde, DVM, sınıfları ayıran en ideal hiperdüzlemi bulmayı amaçlar. Bu ayırcı hiperdüzlem ile sınıfların en yakın noktalarına olan aralıklar (margin) mümkün olduğunda geniş olmalıdır. Uzaklıkların geniş olması sınıflandırmanın daha başarılı olmasını sağlar. Lineer olarak ayrılabilen veriler üzerinde DVM'ler yüksek doğruluk oranları verir. Ancak destek vektör makineleri lineer olarak ayrılamayan veriler üzerinde etkili sonuçlar vermez. Bu problemi düzeltmek için verileri çok boyutlu uzaya haritalayıp lineer olarak ayrılabilecek hale getirebilecek çekirdek fonksiyonları kullanılır. Ayrıca bazı noktalar hiperdüzlemi ihlal edebilirler. Ne kadar noktanın hiperdüzlemi ihlal edebileceğini belirleyen parametreler devreye girer. Bu nedenle, etkili sonuçlar alabilmek için DVM parametrelerinin uygun olarak belirlenmesi önemli ve gereklidir.

DVM algoritması çekirdek fonksiyonu tabanlı bir yaklaşımdır. Çekirdek fonksiyonları DVM'leri güçlü kıلان en önemli özellikdir. Çekirdek fonksiyonları sayesinde DVM'ler lineer olmayan örnek uzaylarında sınıflandırma yapabilirler. DVM'ler bu fonksiyonları kullanmadan da lineer olmayan uzaylarda sınıflandırma yapabilse de hem doğruluk performansı hem de hesaplama karmaşıklığı bakımından çok zayıf kalmaktadır.

Tez çalışmasında DVM'lerin belli parametrelerinin belirlenmesi için literatürde daha önce kullanılmayan iki algoritmanın kullanılması ve performanslarının değerlendirilmesi hedeflenmiştir. Bu amaçla kullanılan iki algoritmada yakın zamanda literatüre girmiştir. Karga Arama Algoritması (KAA) 2016 yılının, Salp Sürüsü Algoritması (SSA) ise 2017 yılının çalışmasıdır. İki çalışmada ilgili makalelerinde literatürde onde gelen optimizasyon algoritmaları ile karşılaştırmalarına yer vermişler ve başarılı sonuçlar elde ettiklerini göstermişlerdir.

Çalışmada lineer, polinom ve RBF çekirdek fonksiyonları kullanılmıştır ve performansları gözlemlenmiştir. RBF çekirdek fonksiyonu hakkında bilgi olmayan veri setleri için ilk tercih edilen fonksiyondur. Polinom çekirdek fonksiyonu özellikle doğal dil işleme ve görüntü işleme alanlarında popüler olmuştur.

Bu çalışmada iki optimizasyon algoritmasının DVM parametrelerinin belirlenmesi için modeller oluşturuldu. İki model birbirleriyle ve literatürdeki benzer bir çalışmada yer alan başarılı algoritmaların sonuçları ile karşılaştırıldı. Deneylerde kullanılan veri setleri literatürde sıkça yer alan standart UCI veri setlerinden seçilmiştir.

1.3. Kaynak Araştırması

DVM optimal parametrelerinin belirlenmesinde Izgara Arama (GS) yöntemi yoğun olarak kullanılmıştır. Bu yöntemde parametreler çözüm uzayında sabit adım aralıklarıyla yer alırlar. Her parametre kombinasyonunun değeri çeşitli performans kriterleri ile belirlenir (Vatsa ve ark., 1997). Ancak bu parametre çözümü az sayıda parametre ayarı için uygundur fakat zaman alıcı ve karmaşık yapıdadır (Chapelle ve ark., 2002).

Ali ve Smith-Miles (2003) DVM'lerin iyi genelleme yapabilmesinde polinom çekirdek fonksiyonunun optimal derecesinin belirlenmesinin kritik bir rolü olduğunu ortaya koymuşlardır. Uygun çekirdek fonksiyonu parametrelerini seçmek veri madenciliğinde önemli bir faktördür. Bu çalışmada otomatik bir parametre seçme yaklaşımı önerilmiş ve DVM üzerinde test edilmiştir. Yeni yaklaşım farklı boyutlarda tek sınıflı ve çok sınıflı sınıflandırma problemleri üzerinde denenmiştir.

(Friedrichs ve Igel, 2005) DVM parametrelerinin deneme yanlışla bulunması yönünde araştırma yapmışlardır. Sonlu sayıda değerin denenmesi ve minimum hata oranına sahip değerlerin saklanması prosedürü uygulanır. Bu yöntem arama uzayında kapsamlı bir tarama gerektirir .

Bir başka araştırmada birini-dışarıda-bırak (Leave One Out) yönteminin eğim (gradient) versiyonu DVM parametre optimizasyonu için kullanılmıştır (Friedrichs ve Igel, 2005). DVM parametrelerini belirlemek amacıyla evrimsel bir yaklaşım önerilmiştir (CMA-ES). Seçilen veri setleri üzerinde CMA-ES'nin Izgara Arama'dan daha iyi sonuçlar verdiği görülmüştür.

Wu ve ark. (2009) günlük maksimum elektrik yükünün belirlenmesi problemi üzerinde DVM ve parametre optimizasyonu için hibrid genetik algoritma (Li ve Aggarwal, 2000) kullanılmışlardır. HGA-SVR adı verilen model optimal çekirdek fonksiyonunun belirlenmesinde ve uygun çekirdek fonksiyonu parametrelerin kullanılmasında önceki çalışmalara kıyasla daha başarılı olmuştur. Elektrik yükünün belirlenmesi ile ilgili çalışmalar arasında optimal çekirdek fonksiyonunu başarıyla belirleyebilmiş ve en düşük hata oranına ulaşmıştır.

Karinca sürüsü optimizasyonu (ACO) da destek vektör makineleri parametre optimizasyonu için kullanılmıştır (Zhang ve ark., 2010). Bu çalışmada karinca sürüsü optimizasyonu ile oluşturulan model detaylı bir şekilde açıklanmıştır. Yapılan çalışma

beş farklı veri seti (Göğüs Kanseri, Diyabet, Kalp, Tiroit, Titanik) üzerinde uygulanmıştır ve var olan bazı teknikler ile karşılaştırılmıştır. Deneysel sonuçlar göstermektedir ki ACO-SVM modeli DVM parametrelerinin belirlenmesinde uygun sonuçlar vermektedir.

Parçacık sürü optimizasyonu (PSO), DVM optimal parametrelerinin belirlenmesinde birçok defa kullanılmıştır. Subasi (2013) EMG sinyallerinde nöromusküler hastalıkların teşhisini için destek vektör makinelerini parçacık sürü optimizasyonu ile optimize etmiştir. Bir başka araştırmada yine parçacık sürü optimizasyonu, DVM parametrelerinin belirlenmesinde kullanılmıştır (Lin ve ark., 2008).

Pereira ve ark. (2013) DVM'lerin en uygun parametrelerini Armoni Arama algoritması (HS) ile optimize etmek için çalışmalar yürütmüşlerdir. DVM'lerin RBF (Radyal Tabanlı Fonksiyon) çekirdek fonksiyonunun parametrelerini belirlemeye çalışmışlardır. Bazı veri setleri için PSO ve GS'den daha başarılı sonuçlar elde etmişlerdir.

Ao ve ark. (2013), Yapay Kimyasal Reaksiyon Algoritması (ACROA) (Alatas, 2011) kullanarak DVM parametre optimizasyonu literatürüne girmiştir. Çalışmada ACROA-SVM modeli makaralı rulman hataları problemine uygulanmıştır. Makaralı rulman sinyalleri local mean decomposition (LMD) kullanarak ürün fonksiyonlarına dönüştürülmüştür. ACROA-SVM ve LMD'nin beraber kullanılması makaralı rulman probleminin doğruluk oranlarını artırmıştır ve maliyeti azaltmıştır.

Li ve Kong (2014) çalışmalarında Genetik Algoritma (GA) ile DVM parametre optimizasyonunu gerçekleştirmiştir. Heyelan tahmini uygulamaları heyelan araştırmaları alanında önemli bir yere sahiptir. Bu tahminin gerçekleştirilmesi için başarılı bir algoritma olan DVM kullanılmasına karar verilmiştir. Ancak DVM'nin başarısının doğru parametre seçiminde yattığını bile yazarlar GA tabanlı bir yaklaşım önermişlerdir. Üzerinde çalışacakları veri için Çin'in kuzeybatısında bir hidroelektrik mühendislik alanını seçmişlerdir. Heyelanın tek faktörlü ve çok faktörlü GA-SVM modelleri oluşturulmuştur. Sonuçta görülmüştür ki çok faktörlü GA-SVM modeli en düşük hata oranını vermiştir (RMSE – 0.00009).

Chou ve ark. (2014) çalışmalarında Hızlı Dağıtık Genetik Algoritma (fmGA) tabanlı DVM modeli oluşturulmuştur. fmGA-SVM modeli kamu-özel ortaklık projelerinin başlangıç aşamasında uyuşmazlık eğiliminin erken tahmininde kullanılmıştır. Modelin değerlendirilmesinde duyarlılık, seçicilik, doğruluk, kesinlik ve AUC değerleri kullanılmıştır. Deneysel karşılaştırmalar göstermektedir ki fmGA-SVM modeli diğer temel modeller (CART, CHAID, QUEST, C5.0) ve geçmiş çalışmalarla göre daha iyi

sonuçlar vermektedir. Sistem, uyuşmazlıkları engellemek için proaktif uyarılar vermektedir ve gerekli karar destek bilgisini sunmaktadır.

Lin ve ark. (2015) çalışmalarıyla Yapay Balık Sürüsü Algoritması'ni (AFSA) (Hu ve ark., 2010) DVM parametre optimizasyonunda kullanmışlardır. AFSA meta-sezgisel bir algoritmadır. Balık sürülerinin hareketleri ve bu hareketlerin altında yatan zeki mantık üzerine geliştirilmiştir. AFSA algoritmasında üç tip vardır: takip, sürü ve av. AFSA bu üç tipi tekrar tekrar her çözüm üzerinde uygulayarak optimal çözümü elde eder. Bu çalışma AFSA yönteminin DVM sınıflandırıcıları özellik seçimi ve parametre optimizasyonu için modifiye edilmiş versiyonudur (MAFSA). Deneysel sonuçlar MAFSA'nın az özellik seçimi yapılarak kullanılan UCI veri setlerinde orijinal AFSA'ya göre daha iyi sınıflandırma yaptığı ortaya koymuştur.

Dong ve Jian (2015), Kaotik Parçacık Sürüsü Optimizasyonu (CPSO) (Hefny ve Azab, 2010) ile yaptıkları çalışmada DVM parametrelerini hızlı ve etkili bir şekilde optimize etmeye çalışmışlardır. Kaotik harenketin başlangıç duyarlılığı ve frekansı nedeniyle PSO algoritması da uygulanmıştır. Deneyler göstermiştir ki geliştirilen CPSO optimum değerlere daha kolay ulaşabiliyor ve iterasyon sayısını düşürüyor.

Tuba ve ark. (2016) tarafından ateş böceklerinin sosyal ve ışık yayma özelliklerinden esinlenerek ortaya koyduğu Ateş Böceği Algoritmasını (FOA) (Yang, 2009) DVM parametre ayarlanmasında kullanılmışlardır. 13 veri seti üzerinde yapılan çalışmalarda, deneylerin birçoğunda sınıflandırma hata oranının düşürüldüğü görülmüştür. Özellik seçiminin öneminden ve üzerinde çalışmaya devam edileceğinden bahsedilmiştir.

Bamakan ve ark. (2016) tarafından yapılan çalışmada PSO DVM parametrelerinin belirlenmesinde kullanılmıştır. Paralel Olmayan DVM (NPSVM) ki Twin-SVM'nin bir uzantısıdır ve Twin-SVM'ye göre daha esnek ve üstündür. Twin-SVM'deki mevcut sorunların üstesinden gelerek büyük boyutlu veri setleri üzerinde uygulanabilir hale gelmiştir. Ancak başarılı bir NPSVM'nin en önemli zorluğu, onun farklı parametreleridir. Bu parametrelerin öğrenme aşamasında iyi ayarlanması gerekmektedir. Bu çalışmada NPSVM'nin parametre optimizasyonu için PSO kullanan hibrid bir yaklaşım önerilmiştir. Yapılan deneyler göstermektedir ki yeni model TWSVM ve NPSVM'ye göre daha iyi sınıflandırma başarısı göstermektedir. Üstelik bunu daha az bir hesaplama zamanı kullanarak yapmaktadır.

Pereira ve ark. (2016) çalışmalarında bir evrimsel tabanlı algoritma olan Sosyal Örümcek Algoritmasını (SSO) (Yu ve Li, 2015) DVM parametrelerini belirlemede

kullanmışlar ve özellik çıkarma yapmışlardır. Model seçimi üç farklı şekilde gerçekleştirilmiştir: özellik çıkarımı, parametre ayarlaması ve özellik çıkarımı + parametre ayarlaması. Yoğun deneyler ve popüler evrimsel tabanlı algoritmalar (PSO, Global-En iyi Armoni Arama Algoritması (NGHS)) ile karşılaştırmalar SSO'nun DVM optimizasyonunda başarılı olduğunu göstermiştir. SSO-DVM çalışmada kullanılan 10 veri setinin 8'inde üstünlük sağlamıştır. Modeller arasında en iyisinin, "parametre ayarlaması + özellik çıkarımı" olduğu görülmüştür. Ayrıca SSO-SVM modeli güç dağıtım sistemlerindeki hırsızlık yakalama problemine de uygulanmıştır.

Ding ve ark. (2017) tarafından Kandilböceği Süre algoritması (GSA) (Krishnanand ve Ghose, 2005) üzerinde bir çalışma gerçekleştirilmiştir. Twin-DVM, çapraz bölgelere sahip veri setlerinde standart DVM'ye göre daha iyi sonuçlar vermektedir. Yapılan çalışma wawelet kernel fonksiyonunu önermektedir. Bu kernel hem kernel fonksiyonları havuzunu genişletiyor hem de twin-DVM'lerin performansını önemli ölçüde artırıyor. Ancak bu yeni model de diğer DVM'lerde olduğu gibi parametre seçimini iyi yapamıyor. Çalışmada DVM parametrelerini optimize etmek amacıyla GSA'yı kullanılmıştır. Deneysel sonuçlar göstermektedir ki önerilen yaklaşım verimlidir ve yüksek sınıflandırma başarısına sahiptir.

Sai ve Huajing (2017) çalışmalarında Balina Optimizasyon Algoritmasını (WOA) (Mirjalili ve Lewis, 2016) DVM regresyon modeli için önermişlerdir. Model durum gözleme işlemlerinde kullanılmıştır. Durum gözleme sistem güvenliği ve bakım işlemleri için çok önemlidir. WOA-SVM modeli bazı standart veri setleri üzerinde ve Tennessee Eastman Sürecinin durum takibinde kullanılmıştır. Her iki durumda da başarılı sonuçlar elde edilmiştir.

Bian ve ark. (2017) çalışmalarında Gri-Kurt Algoritmasını (GWO) (Mirjalili ve ark., 2014) DVM parametre optimizasyonunda kullanmışlardır. Süperkritik karbon dioksit (SCCO₂) içindeki çözünürlüğün tahmini, süperkritik uygulamaların geliştirilmesi için çok önemlidir. Aromatik bileşiklerin çözünürlüğünün belirlenebilmesi için birçok farklı model geliştirilmiştir. Bu çalışmada gri kurt optimizasyon algoritması tabanlı bir DVM önerilmektedir (GWO-SVM). Önerilen model SCCO₂'nun ısı, basınç ve yoğunluk değerlerini gri kurt algoritmasının giriş parametreleri olarak ve farklı çözünenlerin SCCO₂ içerisindeki çözünürlük değerlerini hedef parametreleri olarak belirlemiştir. 27 en çok kullanılan empirik model ile karşılaştırıldığında GWO-SVM en düşük ortalama mutlak göreli sapmaya sahiptir (3.20%).

Tharwat ve ark. (2017) tarafından gerçekleştirilen çalışmada Yarasa Algoritması (BA) DVM parametre optimizasyonu için kullanılmıştır. BA-SVM adı verilen model UCI veri tabanından alınan dokuz standart veri seti üzerinde uygulanmıştır. Model, GS, PSO ve GA ile karşılaştırılmıştır. Sonuçlar modelin parametreleri belirlemede yeterli olduğunu göstermiştir. Ayrıca model PSO ve GA'ya göre daha düşük hata oranları vermiştir. Çalışmada DVM çekirdek fonksiyonlarından RBF'in, Polinom çekirdek fonksiyonuna göre daha iyi sonuçlar verdiği görülmüştür.

Tuba ve ark. (2017) Fil Sürüsü Algoritmasını (EHA) (Wang ve ark., 2015) ile optimize edilmiş DVM modelini hastalık teşhisi için kullanmışlardır. Birbirinden ayırt etmesi çok zor olan altı erythemato-squamous hastalığı sınıflandırılmıştır. Literatürdeki başarılı algoritmalar ile karşılaştırmalar yapılmıştır ve tüm testlerde EHA-SVM modeli üstünlük sağlamıştır.

Tharwat ve Hassanien (2018) çalışmalarında Kaotik Karıncaaslanı Optimizasyonu (CALO) algoritmasını DVM parametre optimizasyonu için kullanmışlardır. Önerilen modelin değerlendirilmesi UCI veri tabanından alınan altı adet veri seti üzerinde yapılmıştır. Modelin başarısı GS, PSO, GA, Sosyal Duygusal Optimizasyon (SEOA) ve standart ALO ile karşılaştırılmıştır. Önerilen modelin optimal DVM parametrelerini belirlemede yeterli olduğu ve lokal optima sorununa takılmadığı belirlenmiştir. Deneylerde farklı kaotik haritalar kullanılmıştır. Logistic haritanın en iyi sonuçları verdiği görülmüştür.

Tharwat ve ark. (2018) Yusufçuk Algoritması (DA) (Mirjalili, 2016) ile DVM parametre optimizasyonunda lokal optima problemine maruz kalmadan uygun parametreleri belirlemiştir. Çalışmalar UCI veri setlerinden altı tanesi üzerinde yapılmıştır. Deneyler GA ve PSO ile karşılaştırılmıştır.

1.2.1. Kaynak araştırması değerlendirme

Kaynak araştırması göstermektedir ki DVM parametreleri yöntemin performansına büyük ölçüde etki etmektedir. Parametrelerin belirlenmesinde GS, PSO ve GA yaygın olarak kullanılan yöntemlerdir. Ancak GS'nin hesaplama karmaşıklığı yüksek ve zaman alıcıdır ve PSO ile GA'nın lokal optimumlara kolayca takıldığı görülebilmektedir. Bu sorunların üstesinden gelebilmek amaçlı birçok meta-sezgisel algoritma DVM parametre optimizasyonunda kullanılmıştır ve klasik yöntemlerden daha başarılı sonuçlar almışlardır. Her yeni meta-sezgisel algoritmanın bu konuda çeşitli

üstünlükler gösterdiği görülmüştür. Bu çalışmada, literatürde DVM parametrelerinin belirlenmesi konusunda daha önce kullanıldığı görülmeyen iki tane meta-sezgisel algoritma (KAA, SSA) kullanılmıştır.

DVM algoritması çekirdek fonksiyonu denilen ve arama uzayını çok boyutlu özellik uzayına dönüştüren yapılara sahiptir. Çalışmalarda bu çekirdek fonksiyonlarından en çok RBF ve polinom çekirdek fonksiyonlarının kullanıldığı görülmüştür. Bu çekirdek fonksiyonlarının kendi parametreleri ve C parametresinin ki eğitim hata oranının düşürmek ve model karmaşıklığını artırmak arasındaki noktayı belirler ilgilenmemiz gereken değerler olduğu anlaşılmıştır.

1.3. Tezin Organizasyonu

Tez çalışmasında Destek Vektör Makineleri (DVM) parametrelerinin uygun bir şekilde belirlenmesi için meta-sezgisel optimizasyon algoritmaları önerilmiştir. Çalışma 5 bölümden oluşmaktadır.

Birinci bölümde çalışmaya genel bir giriş yapılmıştır. Çalışmada kullanılan teknolojiler hakkında genel bilgiler verilmiştir. Çalışmanın amacı ve önemi üzerinde durulmuştur. Ayrıca literatürde geçmiş dönemlerde yapılmış benzer çalışmalar incelenmiş ve sıralanmıştır.

İkinci bölümde meta-sezgisel optimizasyon algoritmalarının amaçlarından, öneminden ve taşıdığı özelliklerden bahsedilmiştir. Çalışmada kullanılan iki algoritma Karga Arama Algoritması (KAA) ve Salp Sürüsü Algoritması (SSA)'nın ilham noktaları, matematiksel modelleri ve bu modellerin tek çıkışlı optimizasyon amacıyla kullanılması hakkında bilgiler verilmiştir. Bu algoritmaların meta-sezgisel algoritmaların avantajlarını taşıdıkları gösterilmeye çalışılmıştır.

Üçüncü bölüm Destek Vektör Makineleri (DVM) hakkında genel bilgiler verilmiştir. DVM'lerin sınıflandırma konusunda en onde gelen algoritmaların biri olmasını sağlayan gerek mantıksal gerek matematiksel avantajları gösterilmeye çalışılmıştır. DVM'lerin lineer olmayan özellik uzaylarında başarılı olmak için gerçekleştirdikleri hileler ortaya konulmuştur.

Dördüncü bölümde KAA ve SSA algoritmalarının DVM'lerin optimizasyonunda kullanılması için modeller oluşturulmuştur. Bu modeller UCI veri tabanından elde edilen veri setleri üzerinde uygulanmış ve başarıları ölçülmüştür. Kullanılan veri setleri DVM optimizasyonunda literatürde daha önce defalarca kullanılmış veri setlerinden seçilmiştir.

Elde edilen sonuçları değerlendirmek adına doğruluk, duyarlılık, seçicilik ve AUC değerleri verilmiş, ROC eğrileri çizilmiştir. Sonuçların dağılımını göstermek amacıyla kutu grafikleri çizilmiştir.

Beşinci bölümde çalışmada elde edilen sonuçlar tartışılmıştır. Gelecekte benzer konular üzerinde çalışma yapmak isteyenler için önerilerde bulunulmuştur.

2. OPTİMİZASYON

Bir problemin değişkenlerinin en iyi değerlerinin hedef fonksiyonunu maksimize veya minimize etmek amacıyla belirlenmesine optimizasyon denir. Optimizasyon problemleri farklı çalışma alanlarında uygulanmaktadır. Optimizasyon problemleri birkaç adımdan oluşurlar. Öncelikle problemin parametreleri belirlenmelidir. Bu aşamamada problem parametrelerin durumuna göre sürekli veya ayrık olarak tanımlanabilir. İkinci olarak, problem parametrelerinin kısıtları belirlenmelidir. Kısıtlar optimizasyon algoritmalarını kısıtlı ve kısıtsız kategorilerine ayırmaktadır. Son olarak optimizasyon probleminin amaçları belirlenmelidir. Optimizasyon problemleri amaçlarına göre tek amaçlı veya çok amaçlı olabilirler.

Matematiksel optimizasyon ilgili fonksiyonları meyil tabanlı (gradient descent) bilgilere dayanarak optimal çözümü bulmaya çalışır. Ancak meyil tabanlı algoritmaların bazı dezavantajları vardır. Matematiksel optimizasyon algoritmaları lokal optimuma takılma sorununa sahiptirler. Bu sorun algoritmanın elde ettiği bir lokal optimumu global optimum varsayıması ve gerçek global optimuma ulaşamaması olarak tanımlanabilir. Ayrıca matematiksel optimizasyon problemleri aşırı türev işlemleri sebebiyle verimsiz hale gelebilirler.

Optimizasyonun matematiksel formülü aşağıdaki gibi verilebilir:

$$\text{minimize } X \in R^n \quad f_i(x), \quad (i = 1, 2, \dots, M) \quad (2.1)$$

$$h_j(x) = 0, \quad (j = 1, 2, \dots, J) \quad (2.2)$$

$$g_k(x) \leq 0, \quad (k = 1, 2, \dots, K) \quad (2.3)$$

$f_i(x)$, $h_j(x)$ ve $g_k(x)$ karar vektörü $x = (x_1, x_2, \dots, x_n)^T$ 'nin fonksiyonlarıdır. Burada x_i 'ler karar değişkenleri olarak yer alırlar. $f_i(x)$ fonksiyonları amaç fonksiyonlarını ifade etmektedir ve M değeri 1 olduğunda sadece tek amaç vardır. Karar değişkenlerinin oluşturduğu uzaya arama uzayı, amaç fonksiyonu değerlerinin oluşturduğu uzaya ise çözüm uzayı denilir. $h_j(x)$ eşitliği ve $g_k(x)$ eşitsizliği kısıtları ifade ederler. $g_k(x)$ eşitsizliğini " \geq " şeklinde yazarak amaçlarımızı maksimize problemi olarak da formüle edebiliriz.

Optimizasyon problemlerini amaç sayısına göre sınıflandıracak olursak, iki kategori bulunmaktadır: tek amaçlı optimizasyon ($M = 1$) ve çok amaçlı optimizasyon

($M > 1$). Gerçek hayatı optimizasyon problemlerinin çoğu çok amaçlıdır. Bu çalışmada yer alan optimizasyon algoritmalarının çok amaçlı versiyonları elde edilebilir olsa da tek amaçlı versiyonları üzerinde çalışılmıştır.

Optimizasyon problemleri kısıtlara göre de sınıflandırılabilir. Eğer hiç kısıt yoksa ($J = K = 0$), kısıtsız optimizasyon problemi olarak isimlendirilir. Eğer $K = 0$ ve $J \geq 1$ ise eşitlik-kısıtlı optimizasyon olur. Eğer $J = 0$ ve $K \geq 1$ ise eşitsizlik-kısıtlı optimizasyon olur. Bazı formüllerde sadece eşitsizlikler yer alabilir, eşitliklere hiç yer verilmez. Çünkü herhangi bir eşitlik, iki eşitsizlik olarak yazılabilir.

2.1. Sezgisel ve Meta-sezgisel Kavramları

Olasılıksal (stochastic) algoritmalar sezgisel ve meta-sezgisel olmak üzere iki kısımda ele alınabilirler ve aralarındaki fark çok küçütür. “Sezgisel” kelimesi “bulmak” veya “deneme yanılma yoluyla keşfetmek” anımlarına gelir. Sezgisel algoritmalarla çözüm makul bir sürede bulunabilir ancak bulunan çözümün optimum olduğu garanti edilmez. Bu tarz algoritmalar çoğu zaman başarılı sonuçlar verir. Sezgisel algoritmalar en iyi çözüme ulaşmak yerine, yeterince iyi sayılabilecek ve kolayca ulaşılabilecek çözümlere erişilmek istediğiinde kullanılırlar.

Zor optimizasyon problemlerini makul bir zaman diliminde, kesin olarak tercih edilecek bir yöntemi olmayan problemler olarak tanımlayabiliriz. Bu tarz problemlerin çözümü için meta-sezgisel optimizasyon algoritmalarını kullanabiliriz. Meta-sezgisel algoritmalar zor optimizasyon problemlerini her problem için derinlemesine özelleşmeden çözmek için geliştirilmişlerdir. Eski Yunan dilinden gelen “meta” kelimesi “yüksek seviye” anlamına gelmektedir ve probleme-özelleşmiş optimizasyon algoritmalarına göre meta-sezgisel yöntemlerin çözüme daha bir üst seviyeden baktıklarını ifade eder. Meta-sezgisel algoritmalar genellikle spesifik bir yöntem tarafından tatmin edici sonuçların alınamadığı problemlere uygulanırlar. Neredeyse her meta-sezgisel optimizasyon algoritması aşağıdaki özelliklere sahiptirler (Boussaïd ve ark., 2013):

- Doğadan esinlenmelidir.
- Rastgele değişkenler içerirler.
- Hedef fonksiyonunun gradient veya hessian matrislerini kullanmazlar.

- Eldeki probleme uydurulması gereken bazı parametreleri vardır.

Bir meta-sezgisel algoritma eğer geçerli problem için çeşitlendirme (diversification) ve yoğunlaştırma (intensification) arasında dengeyi sağlayabilirse başarılı olur. Mevcut meta-sezgisel algoritmaları birbirinde ayıran en önemli fark bu dengeyi sağlamak için buldukları yöntemlerdir (Birattari ve ark., 2001).

Meta-sezgisel algoritmaların çoğunluğunu doğadan esinlenen algoritmalar oluşturmaktadır. Doğadan esinlenen algoritmalar genellikle biyolojiden esinlense de kimya ve fizik alanlarından esinlenen yöntemlerde mevcuttur. Biyolojik algoritmaların geniş bir kısmını sürü zekâsı algoritmaları oluşturmaktadır. Sürü zekâsı yöntemleri çeşitli canlıların merkezi bir kontrol olmadan, kolektif olarak gerçekleştirdikleri özellikle hareket etme ve besin arama yöntemlerinden esinlenerek geliştirilmiştirlerdir. Bu çalışmada kullanılan algoritmaların Karga Arama Algoritması (KAA) biyolojiden esinlenen bir algoritmadır ancak sürü zekâsı temelli değildir. Çalışmada kullanılan diğer algoritma olan Salp Sürüsü Algoritması ise okyanus derinliklerinde yaşayan Salp canlılarının Salp zinciri olarak bilinen sürü zekâsı davranışlarına dayanmaktadır.

2.2. Karga Arama Algoritması (Crow Search Algorithm)

Karga Arama Algoritması (KAA), Askarzadeh (2016) tarafından geliştirilmiştir. Algoritma kargaların zeki davranışlarını dikkate alarak geliştirilen popülasyon tabanlı bir yöntemdir. Kargalar dünyanın en zeki hayvanlarından sayılmaktadır (Rincon, 2005; Wikipedia contributors, 2018).

Bir karga sürüsünde optimizasyon süreçleri ile ilişkili bir davranış biçimi vardır. Bu davranış biçimine göre, kargalar fazla yiyeceklerini bazı gizli noktalarda saklarlar ve ihtiyaç duyuklarında kullanırlar. Ayrıca kargalar daha iyi besin kaynakları bulmak için birbirlerini takip ederler. Bir karganın gizli yiyecek deposunu bulmak zordur çünkü kargalar takip edildiklerini anırlar ise başka bir noktaya giderek şaşırtmaca yaparlar. Optimizasyon bakış açısı ile kargalar, arayıcıları; çevre, arama uzayını; çevredeki her pozisyon uygun bir çözümü, besin kaynağının kalitesi, uygunluk fonksiyonunu ve en iyi besin kaynağı global çözümü ifade eder. Bu benzerlik üzerinden KAA optimizasyon problemlerindeki en iyi çözümü bulmaya çalışmaktadır. KAA'nın dayandığı prensipler aşağıdaki gibidir.

- Kargalar sürü halinde yaşarlar.
- Kargalar yiyecek saklama yerlerinin pozisyonlarını hatırlarlar.
- Kargalar hırsızlık yapmak amacıyla birbirlerini takip ederler.
- Kargalar bir olasılık değerine göre yiyecek saklama yerlerini hırsızlığa karşı korurlar.

2.2.1. KAA matematiksel modeli

Bir grup kargadan oluşan d -boyutlu bir ortam olduğu varsayılmaktadır. N karga sayısını (flock size) ifade eder. Ortamdaki her bir i kargasının her iterasyondaki (itr) pozisyonu bir vektör ile belirlenir (Denklem (2.4)-(2.5)). itr_{max} maximum iterasyon sayısıdır.

$$x^{i,itr} \quad (i = 1, 2, \dots, N; itr = 1, 2, \dots, itr_{max}) \quad (2.4)$$

$$x^{i,itr} = [x_1^{i,itr}, x_2^{i,itr}, \dots, \dots, x_d^{i,itr}] \quad (2.5)$$

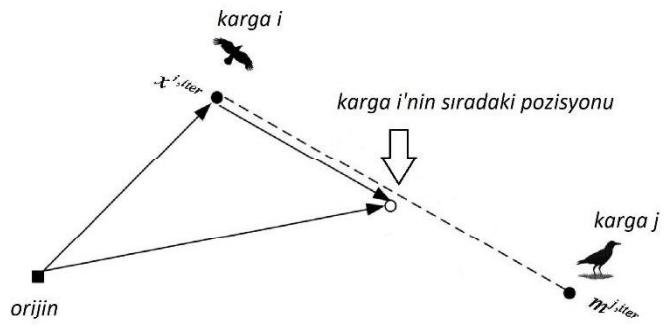
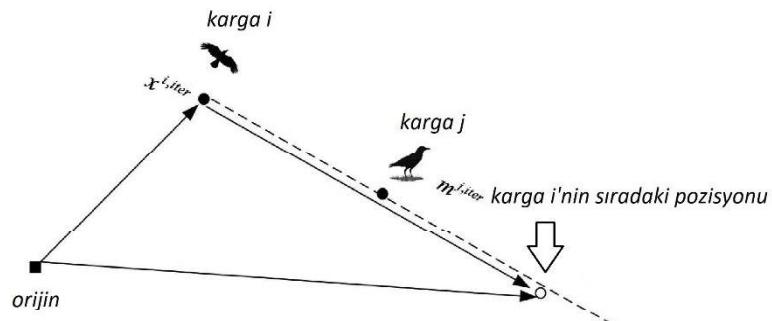
Her itr iterasyonunda i .karganın en iyi saklama yeri $m^{i,itr}$ ile ifade edilir. Bu değer i kargasının o iterasyona kadarki en iyi saklama yeridir.

Varsayıyalım ki itr iterasyonunda karga j kendisinin saklama yeri olan $m^{j,itr}$ 'e gitmek istiyor. Bu iterasyonda karga i , karga j 'yi takip etmek istiyor. Bu ortamda iki durum ortaya çıkacaktır:

Durum 1: Karga j , karga i 'nin kendisini takip ettiğini bilmiyor. Bu durumda karga i , karga j 'nin saklama yerine yaklaşır ve karga i 'nin yeni pozisyonu Denklem (2.6)'daki gibi oluşturulur.

$$x^{i,itr+1} = x^{i,itr} + r_i \times uu^{i,itr} \times (m^{j,itr} - x^{i,itr}) \quad (2.6)$$

Denklem (2.6)'te r_i 0-1 arasında rastgele bir sayı, $uu^{i,itr}$ ise karga i 'nin itr iterasyonundaki *uçus uzaklığım* ifade eder.

(a) $uu < 1$ (b) $uu > 1$

Şekil 2.1. KAA'nın Durum 1 için akış şeması (a) $uu < 1$ için (b) $uu > 1$ için. Karga i kesik çizgili alanın her yerine hareket edebilir. (Askarzadeh, 2016)

Şekil 2.1'de karga takip konseptinin akış şeması verilmiştir. Bu şemadan uu 'nun arama yeteneğine etkisi görülebilir. uu 'nun küçük değerleri lokal arama ile, büyük değerleri ise global arama ile sonuçlanmaktadır. Şekil 2.1'de görüldüğü gibi uu 'nun 1'den küçük seçilmesi durumunda karga i , $x^{i,itr}$ ile $m^{j,itr}$ arasında hareket edebilir. uu 'nun 1'den büyük seçilmesi durumunda ise karga i , $m^{j,itr}$ 'i aşabilen kesik çizgili alanda hareket edebilir.

Durum 2: Karga j , karga i 'nin kendisini takip ettiğini biliyor. Bu yüzden stokunun calınmaması için karga i 'yi aldatma yoluna gidiyor ve arama uzayında farklı bir pozisyonu gidiyor.

$$x^{i,itr+1} = \begin{cases} x^{i,itr+1} = x^{i,itr} + r_i \times uu^{i,itr} \times (m^{j,itr} - x^{i,itr}) & r_j \geq FO^{j,itr} \\ \text{rastgele bir pozisyon} & \text{diğer} \end{cases} \quad (2.7)$$

Denklem (2.7)'te Durum 1 ve 2'nin ifadesi beraberce verilmiştir. r_j , 0-1 aralığında rastgele bir değeri ve $FO^{j,itr}$ ise karga j 'nin itr iterasyonundaki *farkındalık olasılığını* (awareness probability) ifade etmektedir.

Farkındalık olasılığı karga algoritmasının yoğunlaştırma (intensification) ve çeşitlendirme (diversification) özellikleri ile ilişkilidir. FO değerinin düşük tutulması KAA algoritmasının mevcut iyi çözümün bulunduğu lokal bölgeye yoğunlaşmasını sağlar. FO değerinin yüksek tutulması ise bu ihtimali azaltır ve KAA iyi çözümü global seviyede arama eğilimi gösterir.

2.2.2. KAA optimizasyon modeli

KAA algoritmasının sözde kodu Şekil 2.2.'de verilmiştir. Adım adım KAA uygulaması aşağıdaki gibidir.

Adım 1: Problemi ve ayarlanabilir parametreleri başlatma

Optimizasyon problemi, karar değişkenleri ve kısıtların tanımlanması. Ayarlanabilir değerler olan karga sayısı (N), maksimum iterasyon sayısı (itr_{max}), uçuş uzaklığı (uu) ve farkındalık olasılığının (FO) değerlerinin verilmesi.

Adım 2: Karga pozisyonları ve hafızalarının başlatılması.

N adet karga d -boyutlu bir arama uzayında rastgele pozisyonlara yerleştirilir. d karar değişkenlerini ifade eder ve her bir pozisyon uygun (feasible) bir çözümü simgelemektedir. Başlangıçta kargaların tecrübeşi olmadığı için besinlerini başlangıç pozisyonlarına sakladıkları kabul edilir.

Adım 3: Uygunluk fonksiyonu değerlendirmesi

Her bir karganın pozisyonunun kalitesi karar değişkenleri uygunluk fonksiyonuna verilerek değerlendirilir.

Adım 4: Yeni pozisyonlar oluşturulması

Her bir karga takip etmek için rastgele başka bir karga seçer. Farz edelim karga i , karga j 'yi rastgele seçer ve bu karga tarafından saklanan yiyecekleri (m^j) bulmak için takip eder. Karganın yeni pozisyonu Denklem (2.6)'teki gibi oluşturulur.

Karga Arama Algoritması

*N bireyden oluşan karga sürüsünü arama uzayında rastgele oluştur
 Karga pozisyonlarının uygunluk değerlerini hesapla
 Her bir karganın hafızalarını oluştur
 while $itr < itr_{max}$
 for $i = 1 : N$
 Rastgele takip edilecek bir karga seç (j)
 Farkındalık olasılığı belirle
 if $r_j \geq FO^{i,itr}$
 $x^{i,itr+1} = x^{i,itr} + r_i \times uu^{i,itr} \times (m^{j,itr} - x^{i,itr})$
 else
 $x^{i,itr+1} = \text{arama uzayında rastgele bir pozisyon}$
 end if
 end for
 Yeni pozisyonların uygunluğunu kontrol et
 Yeni pozisyonları değerlendir
 Karga hafızalarını güncelle
 end while*

Şekil 2.2. Karga Arama Algoritması sözde kodu

Adım 5: Yeni pozisyonların uygunluğunun (feasibility) kontrol edilmesi

Yeni oluşturululan pozisyonun uygunluğu kontrol edilir. Eğer yeni pozisyon uygun ise karga yeni pozisyonu hareket eder. Aksi taktirde pozisyonunu değiştirmez.

Adım 6: Yeni pozisyonların uygunluk fonksiyonu değerlendirirmesi

Her yeni pozisyonun uygunluk değerleri hesaplanır.

Adım 7: Hafızayı güncelleme

Kargalar hafızalarını Denklem (2.8)'e göre güncellerler.

$$m^{i,itr+1} = \begin{cases} x^{i,itr+1} & f(x^{i,itr+1}) \text{ daha iyi ise } f(m^{i,itr}) \\ m^{i,itr} & \text{diğer} \end{cases} \quad (2.8)$$

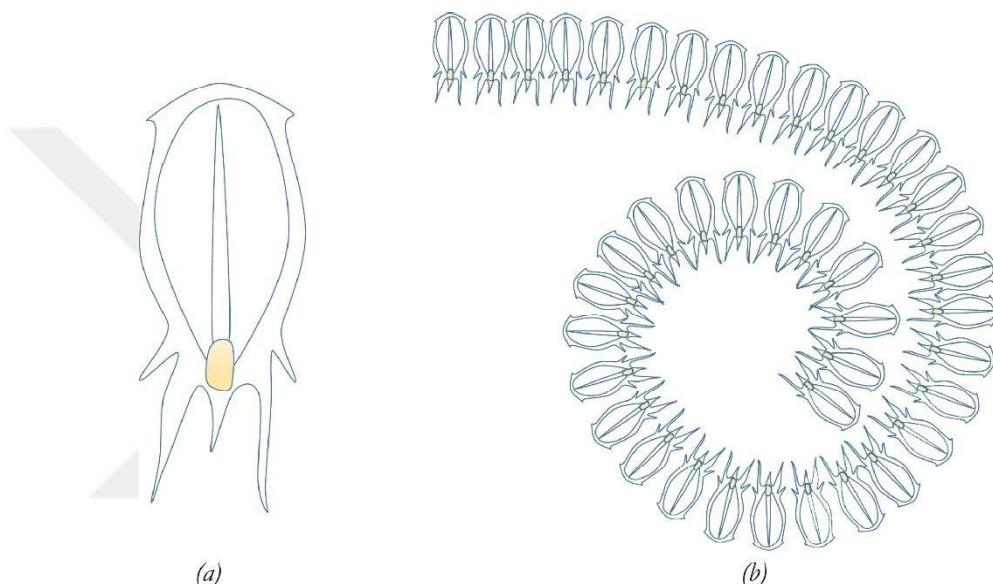
Yukarıdaki denklemde $f(\cdot)$ uygunluk fonksiyonunu ifade eder. Göründüğü üzere yeni pozisyonun uygunluk değeri, hafızaya alınmış uygunluk değerinden daha iyi ise güncelleme yapılır.

Adım 8: Bitirme kriterinin kontrol edilmesi

Adım 4-7 arası itr_{max} 'a ulaşana kadar tekrar edilir. Bitirme kriterine ulaşıldığında, uygunluk değeri kriterine göre hafızadaki en iyi pozisyon optimizasyon probleminin çözümü olarak verilir.

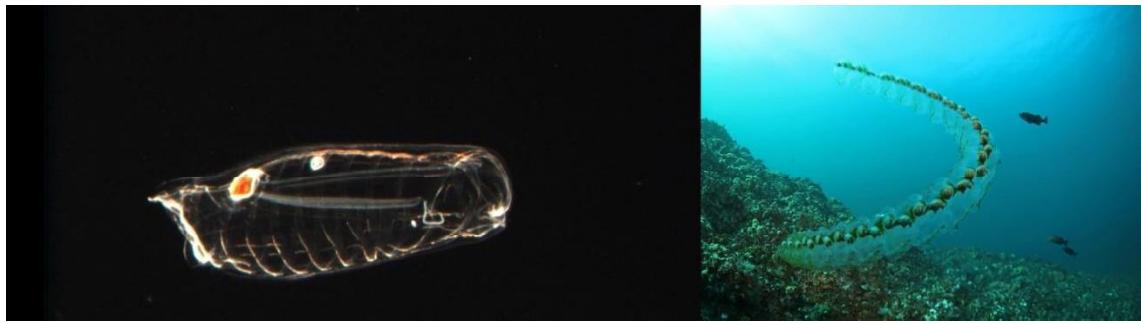
2.3. Salp Sürüsü Algoritması (Salp Swarm Algorithm)

Salp Sürüsü Algoritması (SSA) Mirjalili ve ark. (2017) tarafından geliştirilmiştir. Algoritmanın ana ilham kaynağı salp canlılarının okyanustaki hareket ve besin arama yöntemleridir. Salp canlıları Salpidae ailesine mensupturlar ve fıcı şeklinde vücutları vardır. Dokuları ve hareket yetenekleri deniz anaları ile benzerlik göstermektedir. Vücutları boyunca su pompalayarak itme kuvveti oluştururlar ve böylece hareket ederler (Madin, 1990). Salp canlısının yapısı Şekil 2.2 (a)'da görülmektedir.



Şekil 2.2. (a) Tekil Salp canlısı (b) Salp Sürüsü (Salp zinciri) (Mirjalili ve ark., 2017)

Salp canlıları üzerindeki bilimsel çalışmalar halen başlangıç seviyesindedir. Bunun sebebi yaşadıkları ortamın derin okyanuslar olması ve laboratuvar ortamında saklanmalarının zor olmasıdır. Salp canlılarının ilginç özelliklerinden birisi oluşturdukları sürü özelliğiidir. Salp canlıları derin okyanuslarda genellikle salp zinciri denilen bir sürü özelliği gösterirler. Bu salp zinciri Şekil 2.2.(b)'de görülebilir. Salp zincirinin temel hedefi tam olarak bilinmese de araştırmacılar bunun hızlı ve koordinasyonlu değişiklikler sayesinde daha iyi bir hareket kabiliyeti (locomotion) kazanmak olduğunu düşünmektedirler (Anderson ve Bone, 1980). Şekil 2.3.'de salp canlısının ve salp zincirinin fotoğrafları görülebilir.



Şekil 2.3. (sol resim) Tekil Salp Canlısı (NEFSC, 2013) - (sağ resim) Salp Zinciri (Wikimedia Commons contributors, 2016)

2.3.1. SSA matematiksel modeli

SSA matematiksel modelinde popülasyon iki gruba ayrılmıştır: lider ve takipçiler. Lider salp zincirin en başında yer alır ve zinciri yönlendirir. Diğer salp canlıları takipçi olarak kabul edilirler ve lideri takip ederler.

Diğer optimizasyon algoritmalarında olduğu gibi salp pozisyonları n -boyutlu bir arama uzayında kabul edilir ve n verilen problemin karar değişkenleri sayısını ifade eder. Bu yüzden salp pozisyonları iki boyutlu bir vektör olan x ile ifade edilir. Ayrıca B ile ifade edilen bir besin kaynağı bulunduğu farz edilmektedir.

Liderin pozisyonu aşağıdaki denklem ile güncellenmektedir:

$$x_j^1 = \begin{cases} B_j + c_1((ub_j - lb_j)c_2 + lb_j), & c_3 \geq 0 \\ B_j - c_1((ub_j - lb_j)c_2 + lb_j), & c_3 < 0 \end{cases} \quad (2.9)$$

Denklem (2.9)'da x_j^1 , j .boyuttaki lider salp canlısının pozisyonunu göstermektedir. B_j j .boyuttaki besin kaynağını, ub_j j .boyut için üst sınırı, lb_j j .boyut için alt sınırı göstermektedir. c_1, c_2, c_3 rastgele sayılardır.

Denklem (2.9)'dan görüldüğü gibi lider salp pozisyonunu besin kaynağına göre güncellemektedir. Denklem (2.10)'deki c_1 , SSA içerisindeki en önemli katsayıdır çünkü global arama (exploration) ve lokal arama (exploitation) arasındaki dengeyi belirler. Denklem (2.10)'de l şimdiki iterasyonu, L ise maximum iterasyon sayısını ifade etmektedir.

$$c_1 = 2e^{-\frac{4l}{L}^2} \quad (2.10)$$

c_2 ve c_3 değerleri $(0,1)$ arasında rastgele değerlerdir ve bir sonraki pozisyonun negatif sonsuza mı yoksa pozitif sonsuza doğru mu olacağını ve adım büyüklüğünü belirlerler.

Takipçi salp canlılarının sonraki pozisyonunu belirlemek için Denklem (2.11)'teki Newton'un hareket kanunundan yararlanılmıştır.

$$x_j^i = \frac{1}{2}at^2 + v_o t \quad (2.11)$$

Denklem (2.11)'de $i \geq 2$ için, x_j^i i . takipçinin j . boyuttaki pozisyonunu, t zamanı, v_o başlangıç hızını ifade eder. $a = \frac{v_{son}}{v_o}$ olarak belirlenir. Buradaki v şöyledir: $v = \frac{x - x_0}{t}$

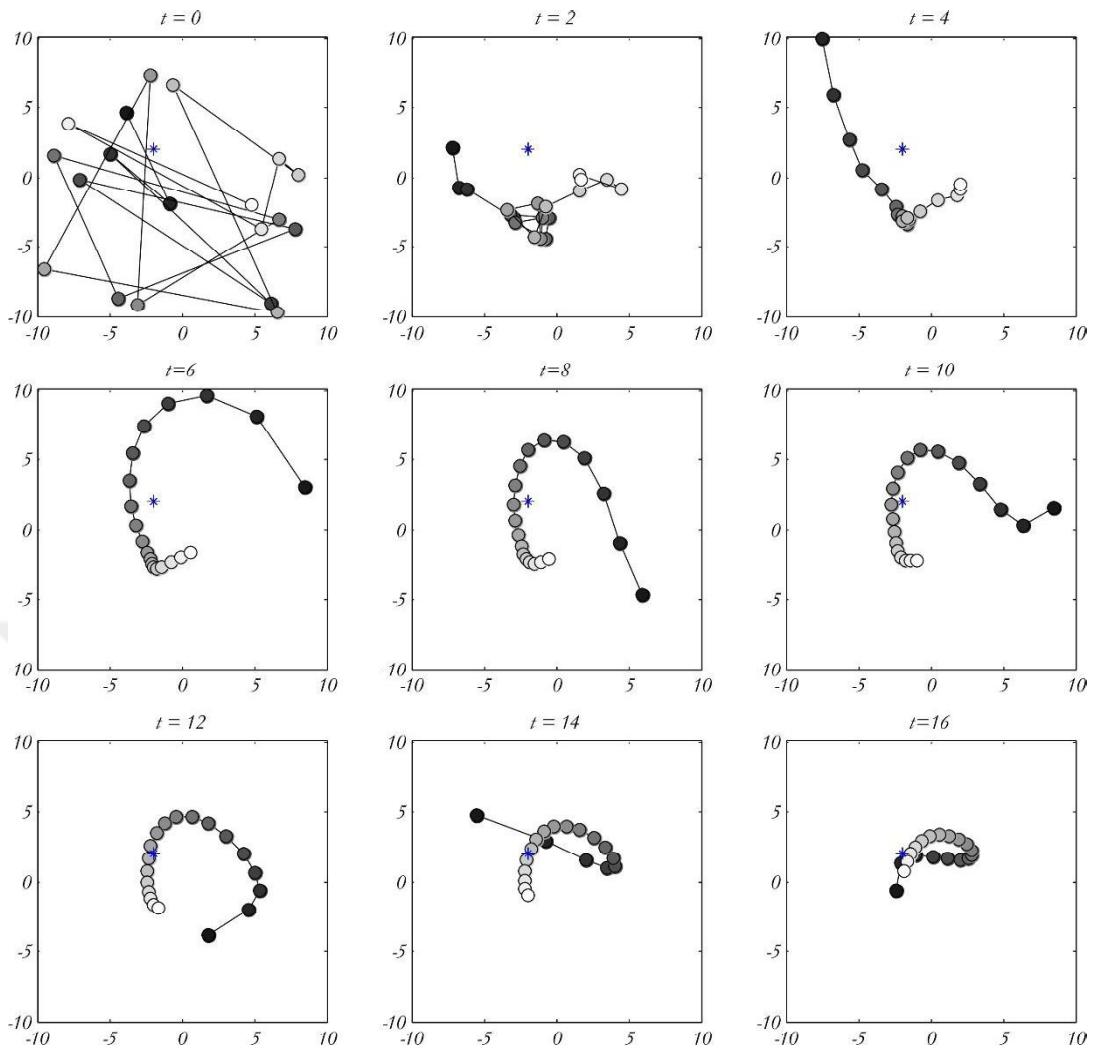
Optimizasyonda zaman *iterasyon* olduğu için ve iterasyonlar arasındaki aralık 1 olduğundan ve $v_o = 0$ olduğundan Denklem (2.11) aşağıdaki gibi ifade edilebilir:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (2.12)$$

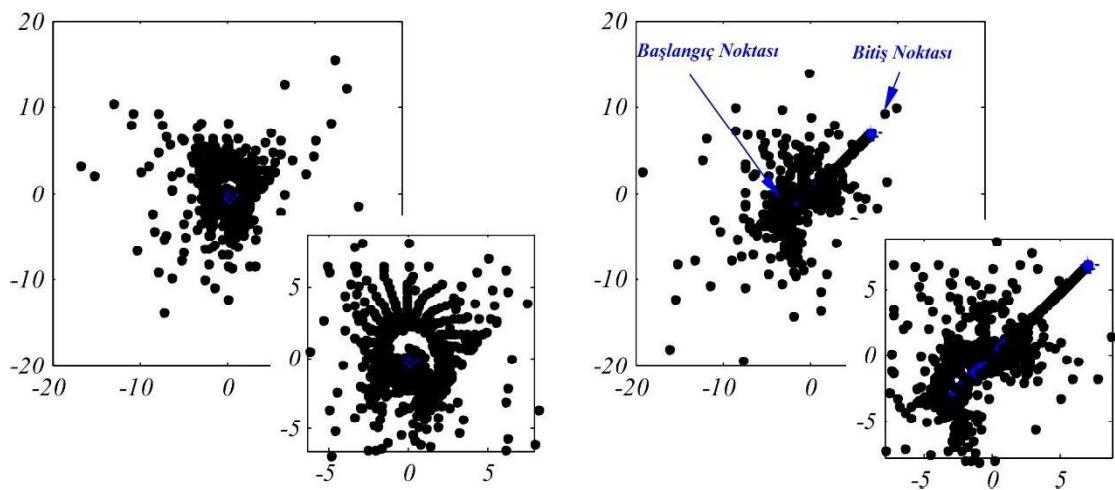
Yukarıdaki denklemde $i \geq 2$ için, x_j^i i . takipçinin j . boyuttaki pozisyonunu ifade eder.

2.3.2. SSA simülasyonları

Önerilen matematiksel modelin etkilerini görebilmek için Mirjalili ve ark. (2017) bazı simülasyonlar gerçekleştirmiştir. Salp zincirlerinin pozisyonları ve her salpın geçmişi Şekil 2.4. - 2.7. arasında gösterilmiştir. Mavi nokta başlangıç noktasını, en siyah nokta ise lider salrı belirtmektedir. Takipçi salplar sıralarına göre grının tonlarında gösterilmiştir. Şekil 2.4. ve Şekil 2.6. incelenirse salp zincirinin ilk iterasyondan sonra ortaya çıktıği ve etkili bir şekilde hareket ettiği görülmektedir. Ayrıca lider salpın besin kaynağı etrafında hareket ettiği ve takipçilerin onu kademeli olarak takip ettiği görülmektedir. Önerilen modelin çok boyutlu uzayda da benzer davranışları gösterdiği görülmektedir.



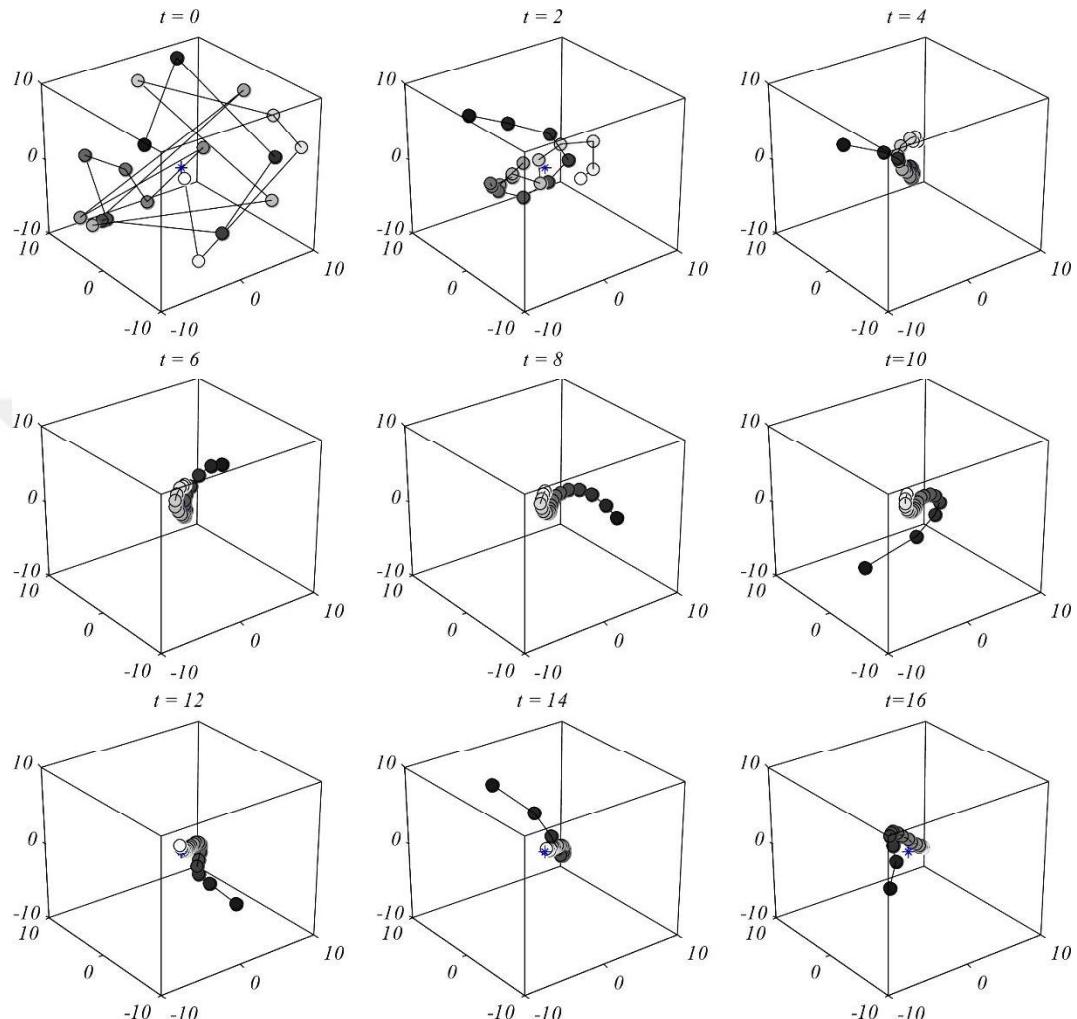
Şekil 2.4. Salp zincirinin sabit besin kaynağı etrafında 2-boyutlu uzayda hareketi (Mirjalili ve ark., 2017)



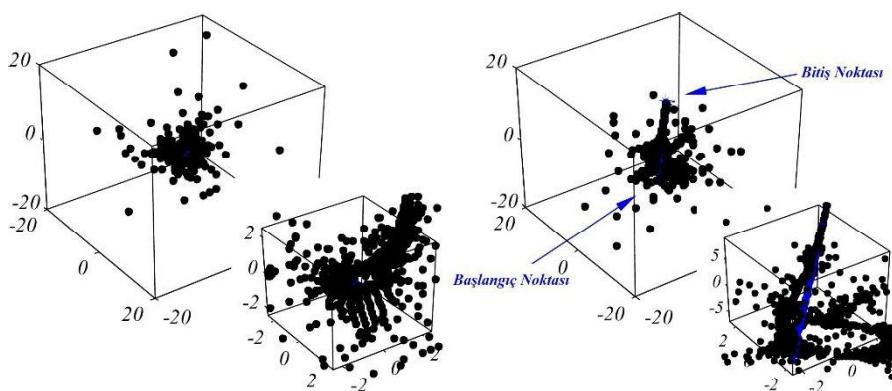
Şekil 2.5. 100 iterasyon sonunda, 2-boyutlu uzayda sabit ve mobil besin kaynakları etrafındaki arama geçmişi (Mirjalili ve ark., 2017)

Şekil 2.5. ve Şekil 2.7.'de 100 iterasyon sonrası 2-boyutlu ve 3-boyutlu uzayda salp pozisyonlarının geçmişi verilmiştir. Salp sürüsünün besin kaynağı etrafında etkili bir

şekilde keşif (exploration) ve sömürme (exploitation) yapabildiği görülmektedir. Ayrıca noktaların dağılımının başlangıç noktasında, bitiş noktasına göre daha yoğun olduğu görülmektedir. Bunun c_1 parametresi sayesinde oluşturduğu anlaşılmaktadır.



Şekil 2.6. Salp zincirinin sabit besin kaynağı etrafında 3-boyutlu uzayda hareketi (Mirjalili ve ark., 2017)



Şekil 2.7. 100 iterasyon sonunda, 3-boyutlu uzayda sabit ve mobil besin kaynakları etrafındaki arama geçmişimi (Mirjalili ve ark., 2017)

2.2.3. SSA optimizasyon modeli

SSA'nın matematiksel modelinin uygulanması için SSA geliştiricileri bazı değişiklikler yapılması gerektiğini belirtmişlerdir. Bir tek amaçlı optimizasyon algoritmasının temel amacı global optimum değeri bulmaktır. SSA modelinde takipçi salplar, lider salpı takip ederler. Lider salp ise besin kaynağına doğru hareket eder. Eğer besin kaynağı global optimum olarak belirlenirse salp zinciri global optimuma doğru hareket edecektir. Ancak optimizasyon problemlerinin global optimum değeri bilinmez. Bu durumda en son elde edilen değer global optimum olarak kabul edilmiş ve bu değerin salp zincirinin takip ettiği besin kaynağı olduğu varsayılmıştır. SSA tek amaçlı optimizasyon algoritmasının sözde kodu Şekil 2.8.'de verilmiştir.

Salp Arama Algoritması

```

Salp popülasyonunu başlat  $x_i$  ( $i = 1, 2, \dots, n$ ) kısıtlar ub ve lb
while (bitiş kriteri gerçekleşmediği sürece)
    Her arama ajanının (salp) uygunluk değerini hesapla
    B = en iyi arama ajanı
     $c_1$  değerini Denklem (2.7)'ye göre güncelle
    for her salp ( $x_i$ )
        if (  $i = 1$ )
            Lider salpin pozisyonunu güncelle Denklem (2.6)
        else
            Takipçi salpin pozisyonunu güncelle Denklem (2.9)
        end
    end
    Salpları üst ve alt kısıtlara göre düzelt
end
return B

```

Şekil 2.8. SSA algoritmasının sözde kodu

2.4. Bölüm Değerlendirmesi

Bu bölümde çalışmada kullanılan KAA ve SSA algoritmalarının ilham kaynakları, matematiksel modelleri ve bu modellerin optimizasyon amaçlı olarak nasıl kullanıldığı gösterilmiştir. Her iki algoritmanın sözde kodları verilmiştir. Bu süreç içerisinde KAA ve SSA algoritmalarının meta-sezgisel algoritmaların avantajlarını taşıdıkları gösterilmeye çalışılmıştır.

Meta-sezgisel algoritmalar çeşitlendirme (diversification) ve yoğunlaştırma (intensification) arasında iyi bir denge sağlamalıdır (Yang, 2011). KAA algoritmasında bu denge temel olarak *farkındalık olasılığı (FO)* parametresi ile sağlanmaktadır. Küçük FO değerleri yoğunlaştırmayı artırırken, yüksek FO değerleri çeşitlendirmeyi

artırmaktadır. SSA algoritmasında ise çeşitlendirme, yoğunlaştırma dengesi c_1 parametresi ile sağlanmaktadır. Şekil 2.5.'de 100 iterasyon sonunda salp canlılarının dağılımı gözükmemektedir. Salpların başlangıç noktasında bitiş noktasına göre daha yoğun olduğu görülmektedir. Bu c_1 parametresi ile sağlanmıştır.

Her iki algoritmanın da matematiksel modellerinin anlaşılabilir yapıları olduğu gösterilmiştir. KAA ve SSA ile literatürde yapılan çalışmalar sınırlı sayıdadır. Özellikle SSA ile ilgili biyolojik çalışmalar dahi araştırma ortamlarının zorluğu yüzünden az sayıdadır.



3. DESTEK VEKTÖR MAKİNELERİ (SUPPORT VECTOR MACHINES)

Destek Vektör Makineleri (DVM) (Cortes ve Vapnik, 1995) sınıflandırma ve regresyon problemleri için hesaplama gücü yüksek, kernel (çekirdek fonksiyonu) tabanlı bir yöntemdir. Yapısal Risk Minimizasyonu (YRM) teorisine dayanarak, DVM lokal minimum ve yüksek boyut problemlerini çözmüştür. Bu yüzden diğer makine öğrenmesi yöntemleri ile karşılaştırıldığında DVM daha iyi genelleme özelliğine sahiptir. DVM sağlam bir teorik tabana sahiptir ve birçok uygulamada diğer algoritmalarla göre daha doğru sonuçlar verir. Birçok araştırmacıya göre DVM metin sınıflandırma alanında belki de en iyi yöntemdir (Nasraoui, 2008). Yapay sinir ağları ile birlikte DVM'ler verileri iyi ayıracak bir hiperdüzlem bulunmasının zor olduğu durumlarda birçok problem için uygulanabilecek en iyi yöntemlerdendir. Örneğin DVM'ler MNIST veri setinin (LeCun ve ark., 2018) sınıflandırılması konusunda uzun zaman boyunca en iyi yöntemdi. Bunun sebebi DVM'lerin yüksek - lineer olmayan dinamikleri bir matematiksel hile neticesinde kullanabiliyor olması ve bu hileyi algoritmada karmaşık değişiklikler yapmadan gerçekleştirmesidir.

DVM'ler mevcut danışmanlı (supervised) öğrenme algoritmaları arasında akla ilk gelen algoritmadır. DVM'ler *maximal margin ayırıcı* kullanırlar. Bu ayırıcı, örnek noktalara mümkün olan en uzak karar sınırlayıcısını temsil eder. DVM'ler lineer bir ayırıcı hiperdüzlem kullanırlar ama verileri çekirdek fonksiyonu hilesi (kernel trick) adı verilen bir yöntemle yüksek boyuta taşıyarak lineer olmayan verileri de sınıflandırabilirler. Genellikle orijinal girdi uzayında lineer olarak ayrılamayan veriler yüksek boyutlu özellik uzayında ayrılabilir olurlar. DVM'ler karmaşık problemleri temsil edecek düzeydedirler ve ezberleme (overfitting) karşı dirençlidirler.

3.1. Maximal Margin Ayırıcı

Bir p -boyutlu uzayda, $p - 1$ boyutlu düz, afin (orijinden geçmesi gerekmeyen) bir alt uzaya *hiperdüzlem* denir. Mesela iki boyutta, bir hiperdüzlem düz tek boyutlu bir altuzaydır yani bir çizgidir. Üç boyutta ise hiperdüzlem iki boyutlu düz bir altuzayı ifade eder ve buna düzlem/levha denir. Üç boyutun üstündeki hiperdüzlemleri ifade etmek zordur ama $p - 1$ boyutlu oldukları konsepti geçerlidir.

p -boyutlu bir ortamda hiperdüzlem Denklem (3.1)'deki gibi ifade edilir:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \quad (3.1)$$

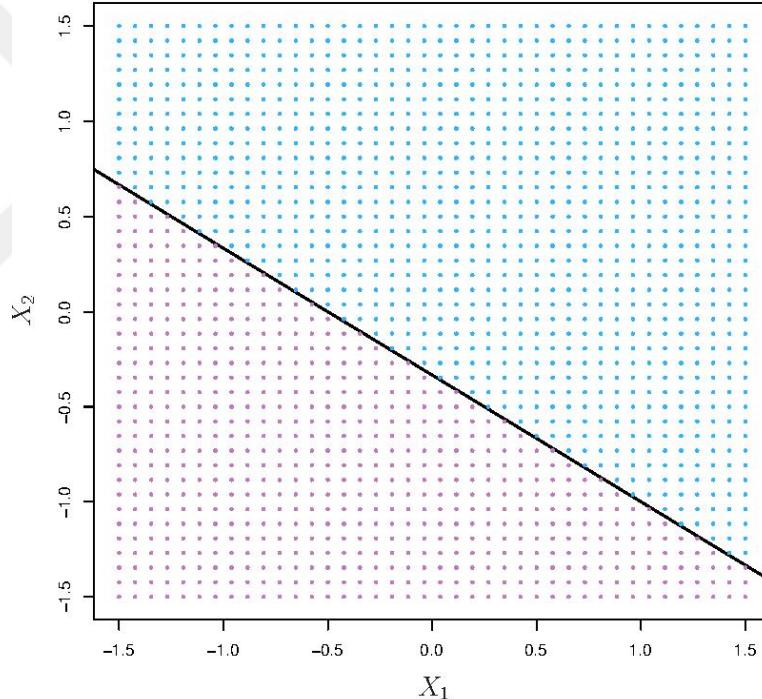
$\beta_0, \beta_1, \beta_2$ parametreleri ve $X = (X_1, X_2)^T$ (2 boyutlu bir uzayda) hiperdüzlemler üzerindeki bir noktayı ifade eder. Şekil 3.1.'de örnek 2-boyutlu hiperdüzlemler verilmiştir.

Hiperdüzlemin bir tarafı aşağıdaki Denklem (3.2) ile ifade edilir:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \quad (3.2)$$

Diğer tarafı ise Denklem (3.3) ile ifade edilir.

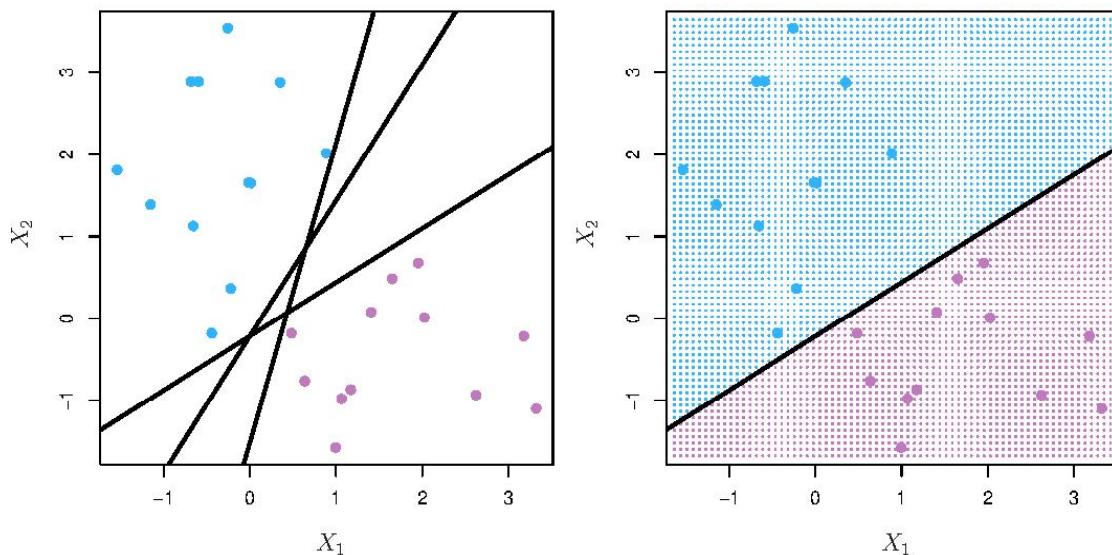
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \quad (3.3)$$



Şekil 3.1. Hiperdüzlemler $1 + 2X_1 + 3X_2 = 0$ görülmektedir. Mavi bölge $1 + 2X_1 + 3X_2 > 0$ 'ı ifade eder. Kırmızı bölge ise $1 + 2X_1 + 3X_2 < 0$ 'ı ifade eder. (James ve ark., 2014)

3.2. Maximal Margin Sınıflandırıcı

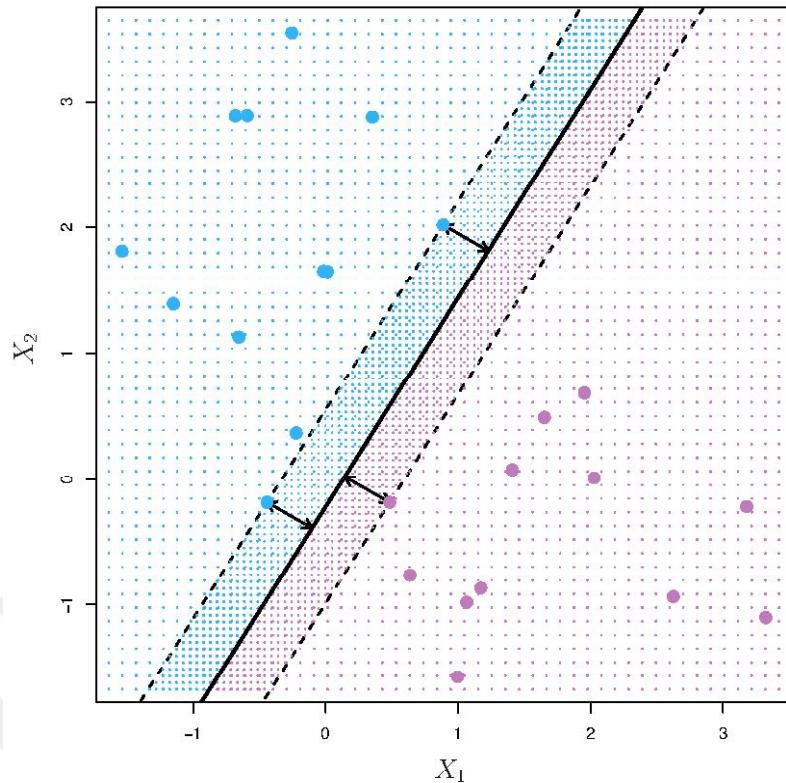
Eğer verimiz bir hiperdüzlem tarafından mükemmel bir şekilde ayrılabiliriyorsa, o zaman öyle bir hiperdüzlemden sonsuz sayıda var demektir. Çünkü elde edilen hiperdüzlem sağa, sola vb. çok küçük miktarlarda çevrilebilir ve hala verileri mükemmel bir şekilde ayırbilir. Şekil 3.2.'de sol panelde üç farklı hiperdüzlem gösterilmiştir. Burada önemli olan nokta o hiperdüzlemler arasından en iyisinin seçilmesidir ve bu seçim için mantıklı bir yolumuz olmalıdır.



Şekil 3.2. (sol) Üç farklı ayrıcı hiperdüzlem durumu (sağ) Eğer x^* test verisinin $f(x^*) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$ değeri 1'den büyükse pozitif sınıf, 1'den küçükse negatif sınıfına yerleştirilir. (James ve ark., 2014)

maximal margin hiperdüzlem (MMH) eğitim verilerinden en uzaktaki ayrııcı hiperdüzlemi ifade eder. Her eğitim örneğinden verilen hiperdüzleme dik uzaklıklar hesaplaysak, bu uzaklıkların en küçük olanına *margin* denir. MMH ise margin'in pozisyonuna göre olabilecek en geniş olduğu durumdur. Böylece MMH'nin pozisyonuna göre eğitim örneklerini sınıflandırabiliriz.

Şekil 3.3.'te, Şekil 3.2.'de ki verilerin MMH'si görülmektedir. Burda görülmektedir ki MMH gerçekten olabilecek en büyük margin'e sahiptir. MMH iki sınıf arasına sıkıştırabileceğimiz en geniş plaka olarak düşünülebilir.



Şekil 3.3. İki sınıf bulunmaktadır; mavi ve mor. Maximal margin hiperdüzlem koyu gösterilmiştir. Margin çizgili ısimların herhangibirinden hiperdüzleme kadar olan alandır. Çizgili kısımların üzerinde bulunan mavi noktalar destek vektörleridir. (James ve ark., 2014)

3.3. Esnek Margin Sınıflandırıcı

Verilerimizi maximal margin sınıflandırıcı ile ayıriyoruz. Fakat verilerden birisi veya birkaç hiperdüzlemimizi tamamen değiştirebilir. Şekil 3.4.'te görüldüğü gibi bir gözlemede yapılan değişiklik hiperdüzlemimizi dramatik olarak değiştirebilir. Bu durumda devreye Esnek Margin Sınıflandırıcı (EMS) girer. Esnek deniliyor çünkü EMS bazı verilerin hiperdüzlemi istismar etmesine izin verir. Böylece birkaç veri hatalı sınıflandırılsa da verilerin geneli için sağlam bir yapı oluşturulmuş olur.

EMS hiperdüzlemi Denklem (3.4 – 3.7) arasında verilen optimizasyon probleminin çözümüdür:

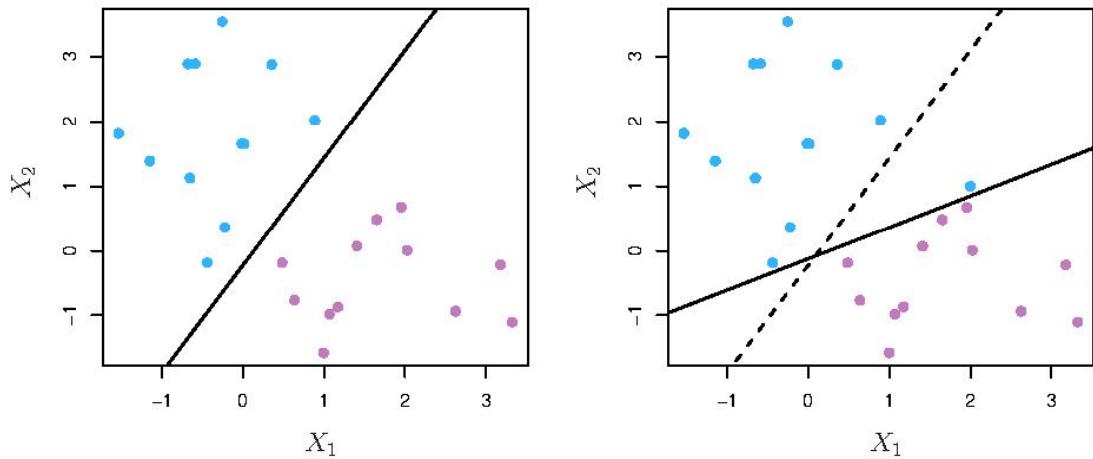
$$\text{maximize } M(\beta_0 \dots \beta_p)(\epsilon_0 \dots \epsilon_p) \quad (3.4)$$

$$\sum_{j=1}^p \beta_j^2 = 1 \quad (3.5)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \quad (3.6)$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C \quad (3.7)$$

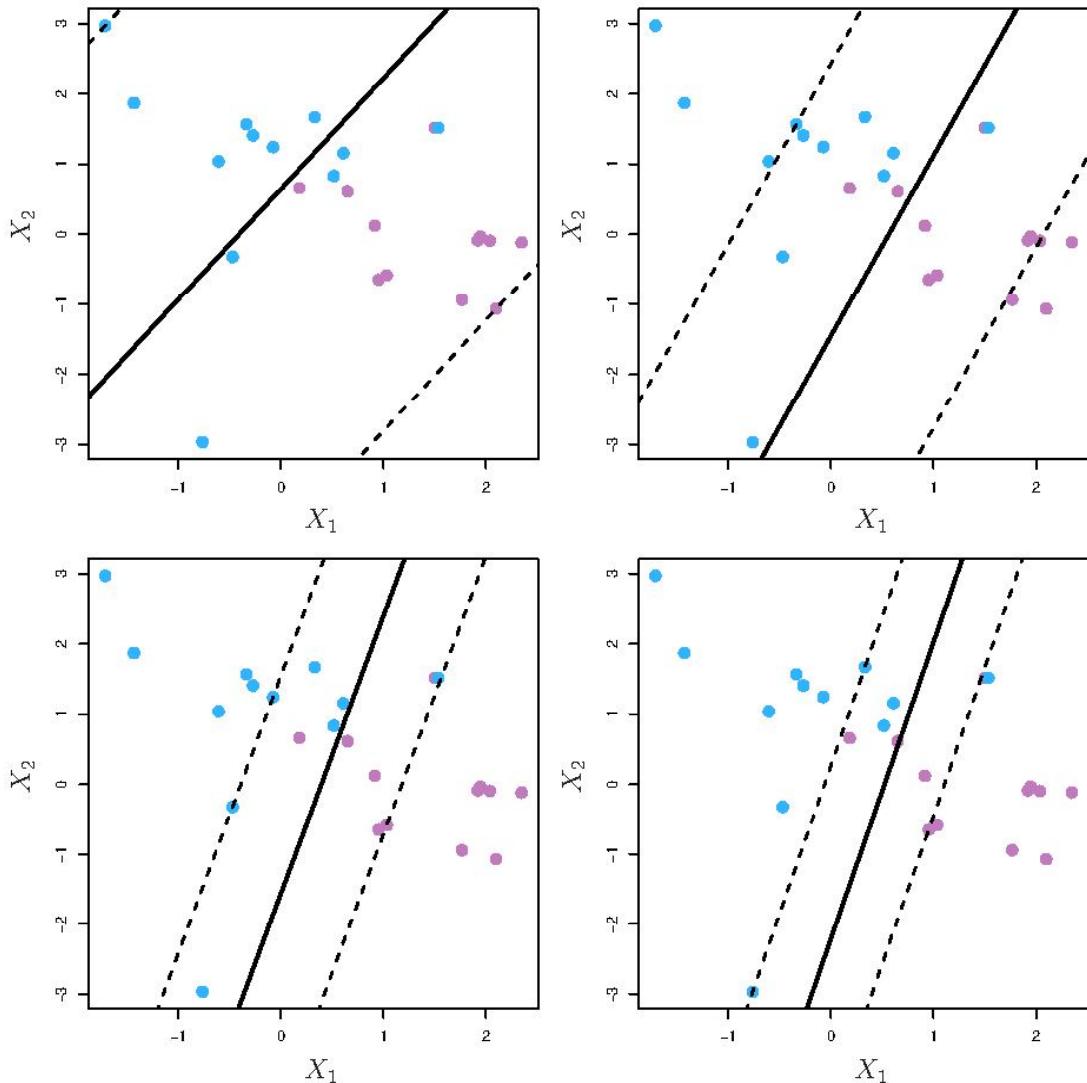
M margin'ın genişliğidir. $\epsilon_1 \dots \epsilon_n$ gevşek (slack) değişkenlerdir ve bazı verilerin margin veya hiperdüzlemin ötesine geçmesine müsaade ederler. C ayarlama (tuning) parametresidir.



Şekil 3.4. Bir gözlemin değişmesinin hiperdüzleme etkisi (James ve ark., 2014)

C değeri teknikin bias-varyans dengesini sağlamaktadır. C küçük değer alırsa çok nadiren istismar edilen dar marginler ortaya çıkar; sınıflandırıcı düşük bias, yüksek varyans elde eder. Diğer durumda çok geniş marginler ortaya çıkar ve sınıflandırıcı yüksek bias, düşük varyans elde eder. C parametresinin etkisi Şekil 3.5.'de görülebilir. Solt üst köşede en yüksek C değeri kullanılmıştır ve margin genişir. Sağ alt köşede en düşük C değeri kullanılmıştır ve margin dardır.

Sınıflandırıcının önemli bir özelliği sadece margin üzerinde olan veya margini geçen verilerin hiperdüzlem üzerinde etkisi olmasıdır. Diğer bir değişle açıkça bir marginin diğer tarafında yer alan bir gözlem sınıflandırıcıyı etkilemez. O gözlemin pozisyonunu değiştirmek, marginin doğru tarafında kaldığı müddetçe sınıflandırıcıyı değiştirmez. Marginin üzerinde veya hatalı tarafında bulunan gözlemlere *destek vektörleri* adı verilir. Bu gözlemler sınıflandırıcıyı etkilerler.



Şekil 3.5. C parametresinin hiperdüzlem üzerine etkisi (James ve ark., 2014)

3.4. Çekirdek Fonksiyonlarının Etkisi

Destek vektör sınıflandırıcısı lineer olarak ayrılabilen uzaylarda iyi çalışan bir sınıflandırıcıdır. Ancak pratikte lineer olmayan gözlem uzayları ile karşılaşabiliyoruz. Böyle durumlarda destek vektör sınıflandırıcısı işe yaramaz hale gelebilir.

DVM'ler destek vektör sınıflandırıcısının bir uzantısıdır ve özellik uzayıını çekirdek fonksiyonu (kernel) adı verilen modeller kullanarak genişletir.

Lineer destek vektör sınıflandırıcısı Denklem (3.8)-(3.9) arasındaki gibi verilebilir:

$$f(x) = \beta_0 + \sum_{i=1}^n \alpha_i \langle x, x_i \rangle \quad (3.8)$$

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^p x_{ij} \cdot x_{i'j} \quad (3.9)$$

Denklem (3.8)'de β_0 ve α_i değerlerini bulabilmek için tüm gözlem çiftlerine ait $\langle x_i, x_{i'} \rangle$ skalar çarpımını hesaplamamız gereklidir. Bu hesaplama karmaşıklığından bizi çekirdek fonksiyonları (kernel) kurtarır.

Denklem (3.8)'de, her Denklem (3.9) değerinin hesaplanması gerektiğinde onu skalar çarpımın daha genel bir ifadesi olan Denklem (3.10) ile değiştiriyoruz.

$$K(x_i, x_{i'}) \quad (3.10)$$

Denklem (3.101)'daki K çekirdek fonksiyonu (kernel) adı verilen bir fonksiyondur. Çekirdek fonksiyonu iki gözlem arasındaki benzerliği derecelendiren bir fonksiyondur.

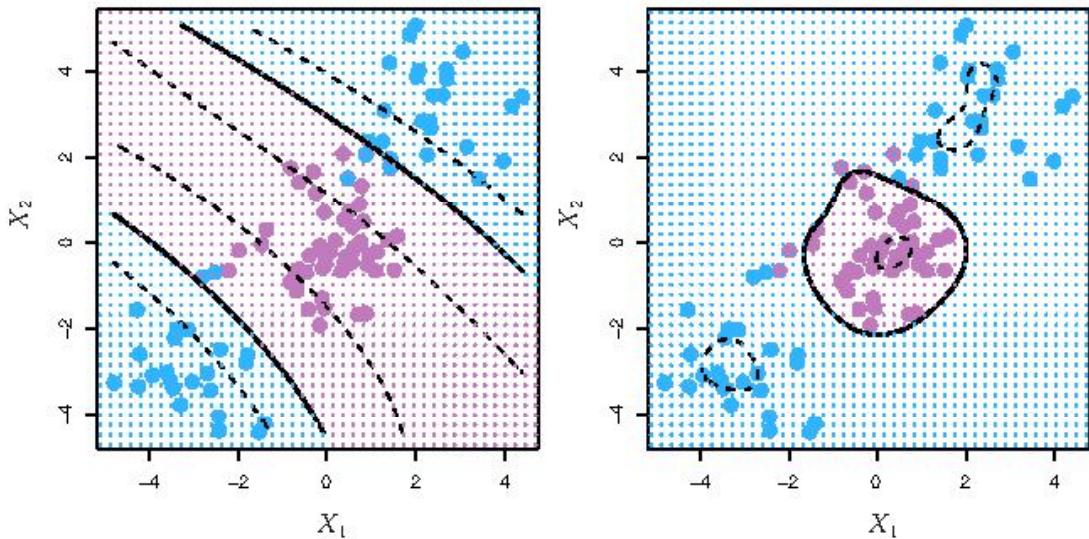
$$K(x_i, x_{i'}) = \sum_{j=1}^p x_{ij} \cdot x_{i'j} \quad (3.11)$$

Örneğin çekirdek fonksiyonu için Denklem (3.11)'i ele alırsak *lineer* çekirdek fonksiyonu elde etmiş oluruz. Ancak her $\sum_{j=1}^p x_{ij} \cdot x_{i'j}$ değerini Denklem (3.12) ile değiştirirsek *polinom* çekirdek fonksiyonunu elde etmiş oluruz.

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^p x_{ij} \cdot x_{i'j})^d \quad (3.12)$$

Diğer bir popüler çekirdek fonksiyonu seçimi ise Radial Basis Function (RBF)'tir. RBF çekirdek fonksiyonu Denklem (3.13)'de verilmiştir.

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2\right) \quad (3.13)$$



Şekil 3.6. (sol) polinom çekirdek fonksiyonu ($d=3$) uygulanmış örnekler. (sağ) RBF çekirdek fonksiyonu uygulanmış örnekler. (James ve ark., 2014)

Şekil 3.6'da görülen gözlemler aslında lineer olarak sınıflandırılamayan bir uzayı ifade etmektedir. Ancak polinom ve RBF çekirdek fonksiyonları kullanılarak her iki durumda da başarılı bir sınıflandırma durumu elde edilmiştir. Bu olay aslında *çekirdek fonksiyonu hilesi (kernel trick)* olarak adlandırılır. Başlangıç durumunda lineer olarak ayrılamayan gözlemler, çok boyutlu özellik uzayına dönüştürülmüştür (çekirdek fonksiyonları ile). Bu çok boyutlu özellik uzayında elde edilen lineer düzlem tekrar başlangıç uzayına uygulandığında Şekil 3.6.'daki gibi hiperdüzlemler elde edilmiştir.

3.5. Bölüm Değerlendirmesi

Bu bölümde sınıflandırma alanının en başarılı uygulamalarından olan DVM'ler incelenmiştir. DVM'ler sınıflandırma alanında geçmiş bilgiye sahip değilseniz ilk öğrenmeniz gereken algoritmadır.

Bölüm içerisinde ilgili algoritma Destek Vektör Makinesi adını alana kadar dört adımda incelenmiştir:

- Maximal margin ayırıcı
- Maximal margin sınıflandırıcı
- Esnek margin sınıflandırıcı
- Ve Destek Vektör Makineleri

DVM'lerin başarısı temel iki özelliğine dayanmaktadır. Bunlardan birincisi bazı örneklerin diğerlerinden çok daha önemli olmasıdır. Örnek uzayını lineer olarak ayıracak olan hiperdüzlemin belirlenmesinde sadece margin üzerinde veya margin'i geçmiş değerlerin etkisi vardır. Geri kalan örneklerin değişimsinin hiperdüzleme etkisi olmaz. DVM'lerin başarısını sağlayan ikinci özellik ise çekirdek fonksiyonu hilesi – (kernel trick) denilen konsepttir. Bu konsept sayesinde DVM'ler normalde lineer olarak ayıramayacakları sınıfları ufak bir matematiksel hile sayesinde ayırlabilirler. Bu işlem ise çok yüksek bir matematiksel yük getirmez.

DVM matematiksel ve mantıksal modeli incelediğinde kolayca anlaşılabilir. Matematiksel modeli yoğun işlemler içerde de karmaşık değildir. DVM modelinin karmaşık bir hale gelmeden birçok avantaj sağlama başarısının sebebi olarak düşünülebilir.

4. DVM PARAMETRELERİNİN OPTİMİZASYONU

Bu çalışmada meta-sezgisel algoritmaları kullanarak DVM parametrelerini belirlemeye yönelik modeller önerilmiştir. Bu modellerin DVM parametrelerini etkin bir şekilde belirleyebileceğini gösterilmiştir. Kullanılan algoritmalar Karga Arama Algoritması (KAA) ve Salp Sürüsü Algoritmalarıdır (SSA). Önerilen DVM modellerinin akış şeması Şekil 4.1.’de verilmiştir. Önerilen modeller (Chen ve ark., 2011a) tarafından yapılan çalışma örnek alınarak ana olarak iki aşamadan oluşmaktadır; iç parametre optimizasyonu ve dış sınıflandırma doğruluğu değerlendirmesi. İç parametre optimizasyonunda SVM parametreleri 5 katmanlı ÇD kullanarak optimizasyon algoritması tarafından dinamik olarak ayarlanmaktadır. Daha sonra elde edilen optimal parametreler dış değerlendirme沼guna verilmekte ve 10 katmalı ÇD kullanılarak değerlendirilmektedir.

Herhangi bir meta-sezgisel algoritmayı bir problemi optimize etmek için kullanırken dikkat edilmesi gereken iki faktör vardır; birey/popülasyon temsili ve uygunluk fonksiyonu.

Birey/popülasyon temsili gerçekleştirilirken hangi karar değişkenlerinin bireyi temsil edeceği belirlenmelidir. Bu çalışmada karar değişkenleri iki temel DVM parametresidir. Bu nedenle modelimizde her bir birey/çözüm iki DVM parametresini temsil etmektedir. Bu parametreler C (cost-maliyet) ve γ (gamma) olarak belirlenmiştir.

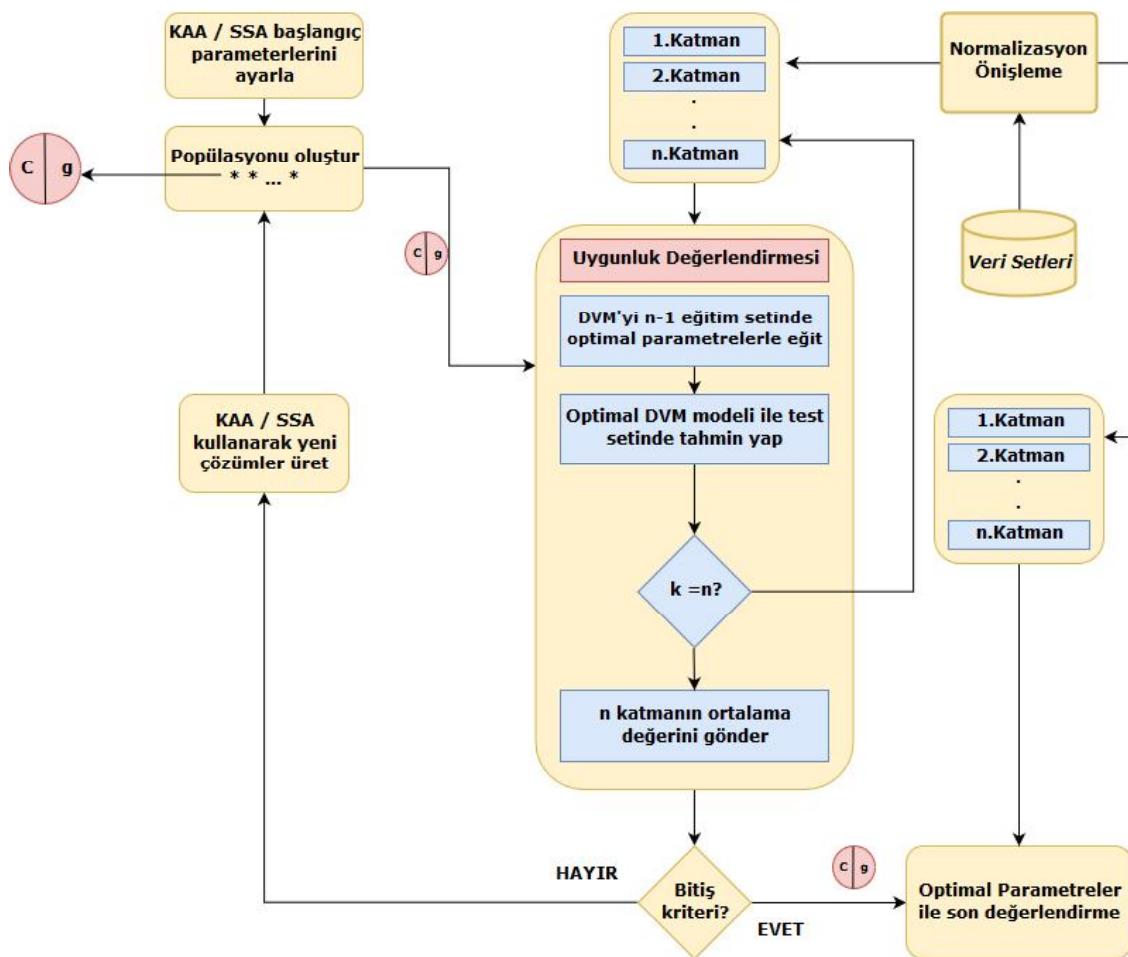
Her yeni bireyin kalitesini belirlemek için bir uygunluk fonksiyonu tanımlanmalıdır. Denklem (4.1) ile gösterilen sınıflandırma doğruluğu uygunluk fonksiyonu olarak ele alınmaktadır. Bu denklemdeki ortalamaDoğruluk (f), iç parametre optimizasyonu sırasında DVM sınıflandırıcısının 5 katmanlı ÇD ile elde ettiği test doğruluğunu ifade etmektedir.

$$f = \text{ortalamaDoğruluk} = \left(\sum_{i=1}^k \text{testDoğruluğu} \right) / k \quad (4.1)$$

4.1. Veri Setleri

Önerilen modelleri değerlendirmek için ana deneylerde 4 veri seti kullanılmıştır. Kullanılan veri setleri UCI (University of California, Irvine machine learning repository) veri deposundan alınan Wisconsin göğüs kanseri veri seti, Pima Kızılderilileri diyabet

veri seti, parkinson veri seti ve sonar veri setidir (Dua, 2017). Veri setlerinin örnek, öznitelik, sınıf sayıları ve eksik veri bilgileri Çizelge 4.1.'de verilmiştir.



Şekil 4.1. KAA-DVM ve SSA-DVM modelleri akış şeması

Veri setlerini deneylerde kullanmadan önce bazı ön işleme operasyonları yapılmıştır. Tüm veri setlerinin değerleri (-1,1) aralığında normalize edilmiştir. Böylece hem hesaplamalarda kolaylığa gidilmiş hem de tüm özelliklerin probleme katkı seviyesi eşitlenmiştir.

Çizelge 4.1. Deneylerde kullanılan veri setleri

N	Veri	Sınıf Say.	Örnek Say.	Öz nitelik Say.	Eksik veri
1	Wisconsin Göğüs Kanseri	2	699	9	Evet
2	Pima Kızılderilileri Diyabet	2	768	8	Hayır
3	Parkinson	2	195	22	Hayır
4	Sonar	2	208	60	Hayır
5	Şarap	3	178	13	Hayır
6	Cam	6	124	10	Hayır
7	Ünlü Harf	11	528	10	Hayır

Göğüs kanseri seti eksik değerler barındırmakta idi. Bu veri setinde eksik verilerin yerine; kategorik veriler için özniteliklerin tepe değeri (mode) ve sürekli değerler için ise ortalaması (mean) kullanılmıştır.

4.2. Deney Ortamı ve Başlangıç Değerleri

Önerilen modelin uygulaması Matlab R2017b ortamında gerçekleştirilmiştir. DVM için Chang ve Lin (2011) tarafından geliştirilen LIBSVM ortamından yararlanılmıştır. Deneylerin değerlendirilmesi aşağıdaki özellikleri taşıyan bir bilgisayar üzerinde yapılmıştır:

- * Intel Core i7 – 6700k, 4 GHz işlemci
- * 16 GB RAM
- * Windows 10 işletim sistemi

Çalışmadaki tüm deneyler KAA ve SSA üzerinde gerçekleştirilmiştir ve birbirleri ile karşılaştırılmıştır. Özellikle PSO (Parçacık Sürüsü Optimizasyonu) ve GA (Genetik Algoritma) gibi literatürde sıkılıkla değerlendirilen algoritmalar ile karşılaştırma Aljarah ve ark. (2018) tarafından yapılan bir çalışma üzerinden gerçekleştirilmiştir. Aljarah ve ark. (2018) tarafından yapılan çalışma, bu tezdeki ile uyumlu bir metodoloji kullanmaktadır ve aynı veri setleri üzerinde deneyleri mevcuttur. Çalışmada kullanılan tüm algoritmaların başlangıç değerleri Çizelge 4.2.'de verilmiştir.

Çizelge 4.2. Algoritmaların başlangıç değerleri

Algoritma	Parametreler	Değerler
GOA (Çekirge Opt. Alg.)	cMin	0.00001
	cMax	1
	Arama ajansı sayısı	50
	İterasyon sayısı	100
GA (Genetik Algoritma)	Çaprazlama oranı	0.9
	Mutasyon oranı	0.1
	Seleksiyon mekanizması	Rulet Tekerleği
	Popülasyon boyutu	50
	Nesiller	100
PSO (Parçacık Sürüsü Opt.)	Hız sabitleri	2,1-2,1
	Süredurum	0,9-0,6
	Parçacık Sayısı	50
	Nesiller	100
MVA (Çoklu Evren Opt.)	Asgari solucan deliği var olma oranı	0.1
	Azami solucan deliği var olma oranı	1
	Evrenler	50
	İterasyonlar	100
BA (Yarasa Algoritması)	Gürültü	0.5
	Vuruş miktarı	0.5
	Asgari frekans	0
	Azami frekans	2
FOA (Ateşböceği Algoritması)	Alfa	0.5
	Beta	0,2
	Gama	1.0
CSO (Guguk Kuşu Opt.)	Keşif Oranı ρ	0.25
	Yuva sayısı	50
	Nesiller	100
GWO (Gri Kurt Opt.)	α	0,2
KAA (Karga Arama Alg.)	Sürü boyutu	50
	Farkındalık olasılığı	0.1
	Uçuş uzaklığı	2
	İterasyon sayısı	100
SSA (Salp Sürüsü Alg.)	Sürü boyutu	50
	İterasyon sayısı	100

4.3. Performans Değerlendirme Kriterleri

Önerilen modelin performans değerlendirmesi için dört farklı değer verilmiştir. Bunlar, doğruluk (Denklem (4.2)), duyarlılık (Denklem (4.3)), seçicilik (Denklem (4.4)) ve AUC (area under the ROC curve) değerleridir. Doğruluk değerleri tüm deneylerde yüzde olarak verilmiştir.

$$\text{Doğruluk} = \frac{(DP + DN)}{\text{Toplam}} \quad (4.2)$$

$$\text{Duyarlılık} = \frac{DP}{DP + HN} \times 100\% \quad (4.3)$$

$$\text{Seçicilik} = \frac{DN}{HP + DN} \times 100\% \quad (4.4)$$

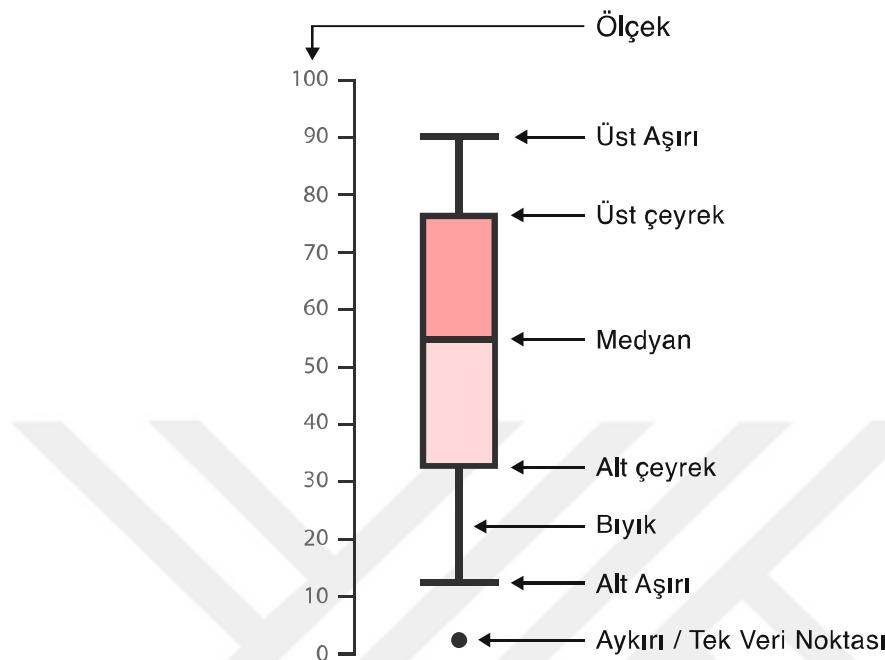
Yukarıdaki denklemlerde DP doğru pozitif sayısını, DN doğru negatif sayısını, HP hatalı pozitif sayısını ve HN hatalı negatif sayısını ifade etmektedir. DP ‘pozitif’ sınıfı doğrular kategorize edilen değerlerdir. DN ‘negatif’ sınıfı doğrular kategorize edilen değerlerdir. HP ‘pozitif’ sınıfı hatalı kategorize edilen değerlerdir. HN ‘negatif’ sınıfı hatalı kategorize edilen değerlerdir. AUC, ROC (Receiver Operating Characteristics) eğrisinin altındaki alanı ifade etmektedir. AUC iki sınıflı veri setleri üzerinde çalışan sınıflandırıcıları karşılaştırmak için kullanılan en iyi yöntemlerdendir. Fawcett (2004) tarafından yapılan çalışmadaki yöntem AUC değerini hesaplamakta kullanılmıştır. AUC değerinin başarı göstergesi aşağıdaki gibi gösterilebilir:

- $0.9 - 1 \rightarrow$ Çok iyi
- $0.8 - 0.9 \rightarrow$ İyi
- $0.7 - 0.8 \rightarrow$ Uygun
- $0.6 - 0.7 \rightarrow$ Zayıf
- $0.5 - 0.6 \rightarrow$ Başarisız

Çalışmada ayrıca KAA-DVM ve SSA-DVM değerlendirme sonuçlarının ROC eğrileri çizdirilmiştir. Her iki modelin farklı çekirdek fonksiyonları aynı grafiklerde gösterilmiştir. ROC eğrileri sınıflandırma algoritmalarını görselleştirmek ve aralarında seçim yapmak için kullanılan bir yöntemdir. ROC eğrilerinin makine öğrenmesi alanındaki en erken kullanımlarından birisi Spackman (1989) tarafından yapılan çalışmadır. Bu çalışmada ROC eğrilerinin algoritmaları karşılaştırmadaki değeri gösterilmiştir. ROC eğrilerinde X ekseni HP değerlerini, Y ekseni ise DP değerlerini ifade eder. ROC eğrilerinde bazı önemli noktalar vardır. $(0,0)$ noktası hiç pozitif sınıflandırma olmayan noktadır. $(1,1)$ ise daima pozitif sınıflandırma olan noktadır. $(0,1)$ noktası mükemmel sınıflandırmayı ifade eder.

Çalışmada kullanılan KAA-DVM ve SSA-DVM modellerinin değerlendirme aşamasındaki kutu grafikleri (boxplot) çizdirilmiştir. Kutu grafikleri bir grup verinin dağılım özelliklerini gözlemlemek için ideal yapılardır. Şekil 4.2'de bir kutu grafiğinin yapısı görülmektedir. Burada medyan, verilerin orta noktasını göstermektedir. Üst çeyrek ve alt çeyrek sırasıyla verilerin %75 ve %25'lik kısmını göstermektedir. Ortadaki kutu Interquartile Range (IQR) olarak isimlendirilir ve tüm verilerin ortasındaki %50'lik kısmı

ifade eder. Büyüklar IQR dışındaki değerlerin kutudan ne kadar uzadıklarını göstermektedir.



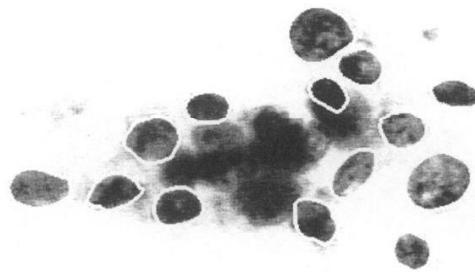
Şekil 4.2. Kutu grafiği yapısı (Veri Görselleştirme Kataloğu, 2018)

4.4. Deneyler

Karşılaştırmalı deneyler KAA-DVM ve SSA-DVM arasında yapılmıştır. Literatürde yer alan benzer bir çalışmadan (Aljarah ve ark., 2018) yararlanarak GOA, GA, PSO, MVO, BA, FF, CS ve GWO ile de doğruluk karşılaştırmaları yapılmıştır. Önerilen modellerin performansı 4 farklı veri seti üzerinde gösterilmiştir. Ayrıca çok sınıflı veri setleri üzerinde de 3 farklı veri seti ile doğruluk değerleri hesaplanmıştır.

4.4.1. Göğüs kanseri veri seti deneyleri

Wisconsin göğüs kanseri veri seti 699 örnekten ve 9 öznitelikten oluşmaktadır. Değerler göğüs kütlesindeki aspirasyon iğnelerinin dijitalleştirilmiş görüntülerinden elde edilmiştir (H Wolberg ve L Mangasarian, 1991; Mangasarian ve ark., 1995). Örnek bir görüntü Şekil 4.3.' de görülebilir.



Şekil 4.3. Bir kötü huylu göğüs aspirasyon iğnesi görüntüsü (Borges, 2015)

Veri setinde sınıflandırma iyi huylu ve kötü huylu hastalık olarak yapılmaktadır. İyi huylu niteliği 2 ile kötü huylu niteliği 4 ile gösterilmiştir. Her öznitelik 1-10 arasında değerler almıştır ve 10 en olağan dışı durumu belirtmektedir. Veri setinin 9 niteliği Çizelge 4.3.'de görülebilir.

Çizelge 4.3. Wisconsin göğüs kanseri veri setinin öznitelikleri

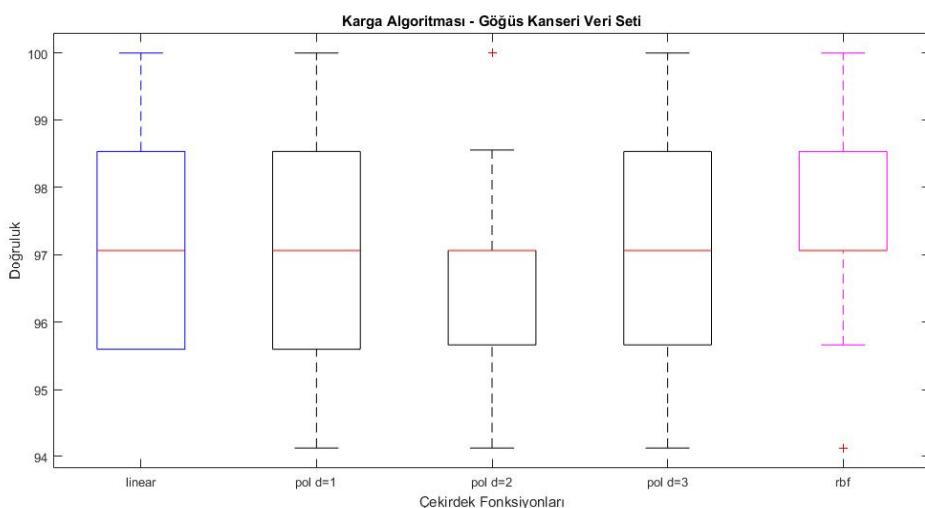
Öznitelik	Aralık
Yığın kalınlığı	1-10
Hücre büyülüğu tekdüzeligi	1-10
Hücre şekli tekdüzeligi	1-10
Marjinal adhezyon	1-10
Tek epitelyum hücre boyutu	1-10
Çıplak çekirdek	1-10
Mülaim kromatin	1-10
Normal çekirdek	1-10
Mitoz	1-10

Çizelge 4.4.'te göğüs kanseri veri seti üzerindeki KAA-DVM ve SSA-DVM sonuçları görülebilir. KAA-DVM modelinde RBF çekirdek fonksiyonu 97.22 başarı oranı ile en iyi sonucu vermiştir. SSA-DVM modelinde ise lineer çekirdek fonksiyonu 96.72 ile en iyi sonucu vermiştir. Bu değerler çizelge üzerinde sarı ile işaretlenmiştir. Tüm doğruluk değerlerinde KAA-DVM, SSA-DVM'ye göre daha düşük standart sapma değerleri vermiştir. Doğruluk kriterine göre KAA-DVM'nin 96.98 ortalama ile SSA-DVM'nin 94.89 ortalamasından daha iyi olduğu gözlemlenmiştir. KAA-DVM ve SSA-DVM'nin duyarlılık, seçicilik ve AUC değerleri bakımından ortalamalarının karşılaştırılması sırasıyla (KAA-SSA şeklinde) 0.9700-0.9638, 0.9682-0.9228 ve 0.9945-0.9888 belirlenmiştir. Her üç durumda da KAA-DVM daha başarılı sonuçlar vermiştir.

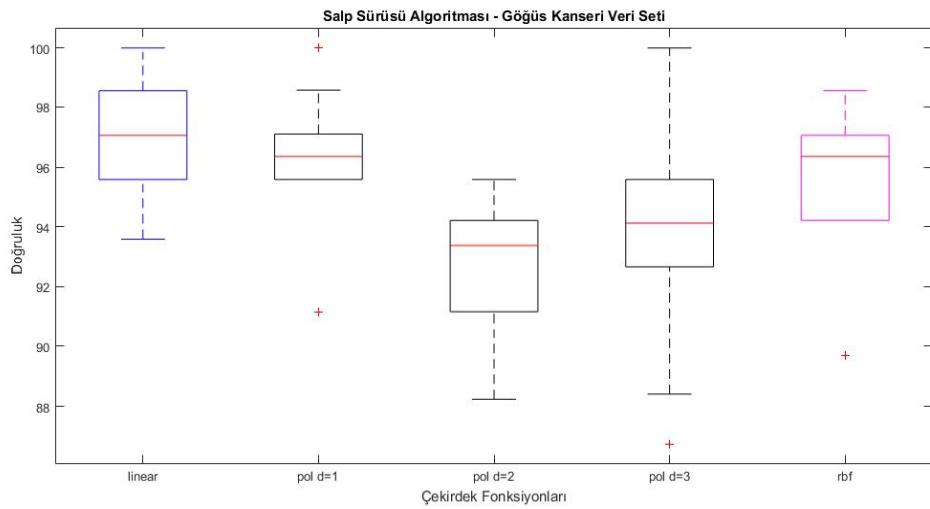
Çizelge 4.4. Wisconsin göğüs kanseri veri seti üzerinde detaylı sınıflandırma sonuçları

Cekirdek Fonksiyonu	Değerlendirme Kriteri	KAA-DVM	SSA-DVM
Lineer	Doğruluk	97.07±1.54	96.72±2.02
	Duyarlılık	0.9750±0.02	0.9727±0.01
	Seçicilik	0.9602±0.03	0.9603±0.04
	AUC	0.9948±0.004	0.9957±0.003
Polinom D=1	Doğruluk	96.69±1.76	96.34±2.34
	Duyarlılık	0.9750±0.02	0.9715±0.02
	Seçicilik	0.9562±0.03	0.9487±0.03
	AUC	0.9950±0.003	0.9949±0.004
Polinom D=2	Doğruluk	96.78±1.66	92.68±2.48
	Duyarlılık	0.9638±0.02	0.9495±0.03
	Seçicilik	0.9748±0.02	0.8820±0.05
	AUC	0.9948±0.003	0.9777±0.01
Polinom D=3	Doğruluk	96.92±1.87	93.56±3.85
	Duyarlılık	0.9634±0.02	0.9616±0.03
	Seçicilik	0.9789±0.02	0.8917±0.06
	AUC	0.9939±0.004	0.9867±0.005
RBF	Doğruluk	97.22±1.61	95.17±3.11
	Duyarlılık	0.9727±0.01	0.9635±0.02
	Seçicilik	0.9708±0.02	0.9311±0.06
	AUC	0.9942±0.004	0.9893±0.007

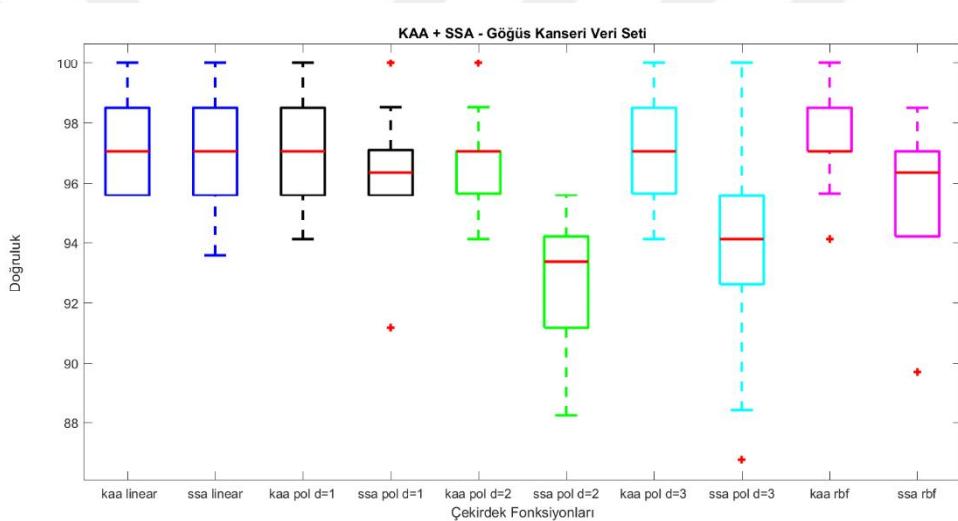
Şekil 4.4.'de KAA-DVM'nin ve Şekil 4.5'de SSA-DVM'nin test verileri üzerindeki kutu grafikleri verilmiştir. Şekil 4.6.'te ise KAA-DVM ve SSA-DVM'nin karşılaştırmalı kutu grafiği verilmiştir. KAA-DVM tüm çekirdek fonksiyonları için aynı medyan değerlerini (kırmızı çizgi) vermektedir. Bu KAA-DVM'nin farklı çekirdek fonksiyonları için benzer değerler etrafında toplandığını göstermektedir. SSA-DVM'de polinom ($d=2$) çekirdek fonksiyonu diğer fonksiyonlara nazaran grafikte aşağı seviyede kalmıştır. Şekil 4.6.'te görüldüğü üzere KAA-DVM polinom ($d=2$), polinom ($d=3$) ve RBF çekirdek fonksiyonlarında SSA-DVM'ye göre yukarı seviyede yer almaktadır.



Şekil 4.4. KAA-DVM'nin göğüs kanseri veri seti üzerindeki kutu grafiği

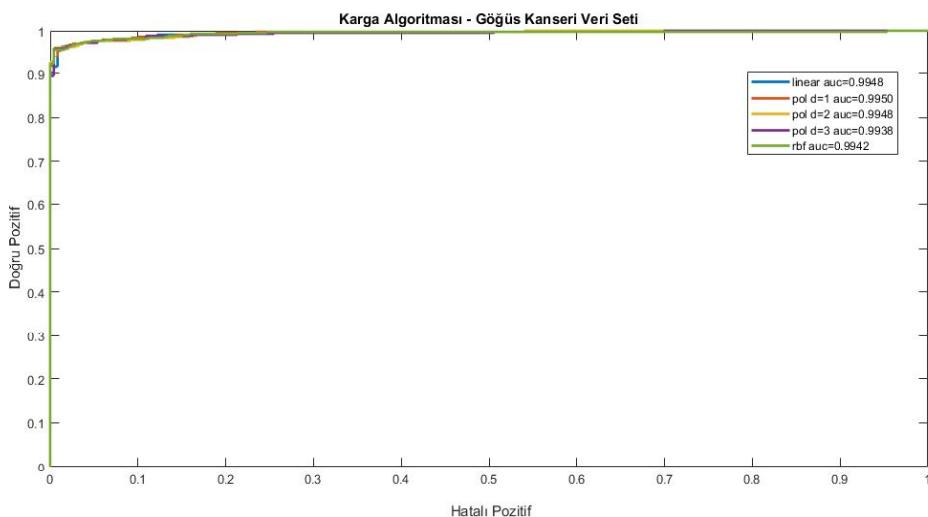


Şekil 4.5. SSA-DVM'ın göğüs kanseri veri seti üzerindeki kutu grafiği

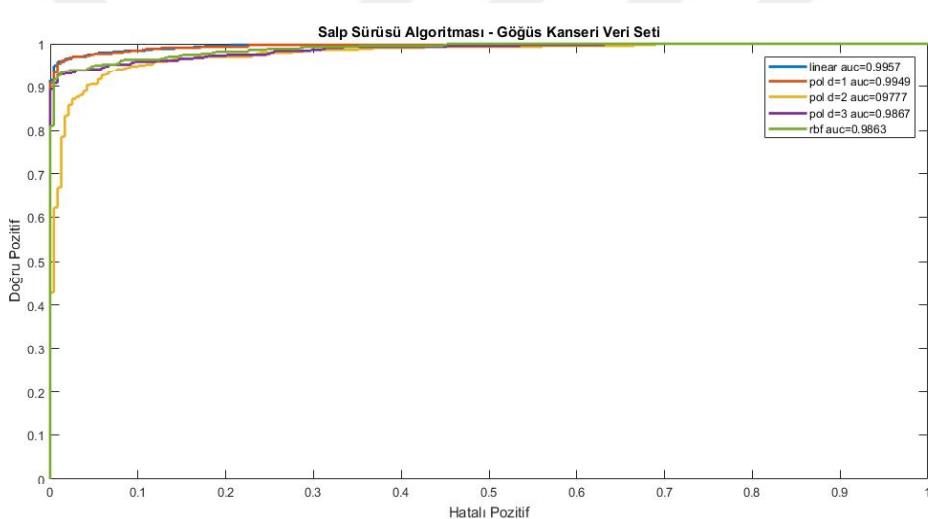


Şekil 4.6. KAA-DVM ve SSA-DVM'ın göğüs kanseri veri seti üzerindeki karşılaştırmalı kutu grafiği

Şekil 4.7.'da KAA-DVM'nin ve Şekil 4.8.'da SSA-DVM'nin test verileri üzerindeki ROC eğrileri verilmiştir. KAA-DVM'nin tüm çekirdek fonksiyonları için köşe noktasına çok yakın eğriler verdiği görülmüştür. SSA-DVM'de polinom çekirdek fonksiyonunun ($d=2$) diğerlerine nazaran köşe noktasından bir miktar uzaklaşlığı görülmektedir. Genel olarak her iki modelinde ROC eğrileri, eğri grafiğinin "çok iyi" sonuçlar ifade ettiği bölgede yer almaktadır.



Şekil 4.7. KAA-DVM göğüs kanseri veri seti test verileri üzerindeki ROC eğrisi



Şekil 4.8. SSA-DVM göğüs kanseri veri seti test verileri üzerindeki ROC eğrisi

Çizelge 4.5.'te KAA-DVM ve SSA-DVM modellerinin diğer algoritmalar ile doğruluk değeri açısından karşılaştırılmasına yer verilmiştir. KAA rakibi olan 9 algoritmanın 8'inden daha başarılı sonuç vermiştir. SSA ise rakibi olan 9 algoritmanın 4'ünden daha iyi sonuç vermiştir. SSA gerisinde kaldığı algoritmala da çok yakın sonuçlar vermiştir. KKA-DVM en iyi ikinci algoritma, SSA-DVM ise en iyi altıncı algoritma olmuştur.

Çizelge 4.5. KAA ve SSA'nın göğüs kanseri veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması

Algoritma	Göğüs Kanseri Veri Seti
Karga Arama Alg.	97.22 ± 1.61
Salp Sürüsü Alg.	96.72 ± 2.02
GOA	97.23 ± 1.88
MVO	96.92 ± 2.13
GA	96.78 ± 1.94
PSO	96.63 ± 1.15
GWO	96.64 ± 1.30
FF	96.64 ± 1.30
BA	96.94 ± 1.37
CS	96.49 ± 1.47

4.4.2. Diyabet veri seti deneyleri

Diyabet veri seti toplam 768 örnekleten oluşmaktadır. Bunlardan 500 tanesi sağlıklı, 268 tanesi diyabet hastasıdır. Veriler Pima Kızılderili soyundan olan kadınlarından alınmıştır. Veriler iki yolla elde edilmiştir. Bir otomatik elektronik kayıt cihazı ile ve elle elde edilen veriler bulunmaktadır. Elektronik kayıt cihazı verilere zaman etiketi eklemiştir. Elle elde edilen verilerde zaman kahvaltı, öğle yemeği, akşam yemeği gibi alınmıştır. Bunlar sabit zaman değerleri ile değiştirilmiştir. Sabah yemeği (8.00), öğle yemeği (12.00) gibi. Diyabet veri setinin 8 niteliği Çizelge 4.6.'da görülebilir.

Çizelge 4.6. Diyabet veri setinin öznitelikleri

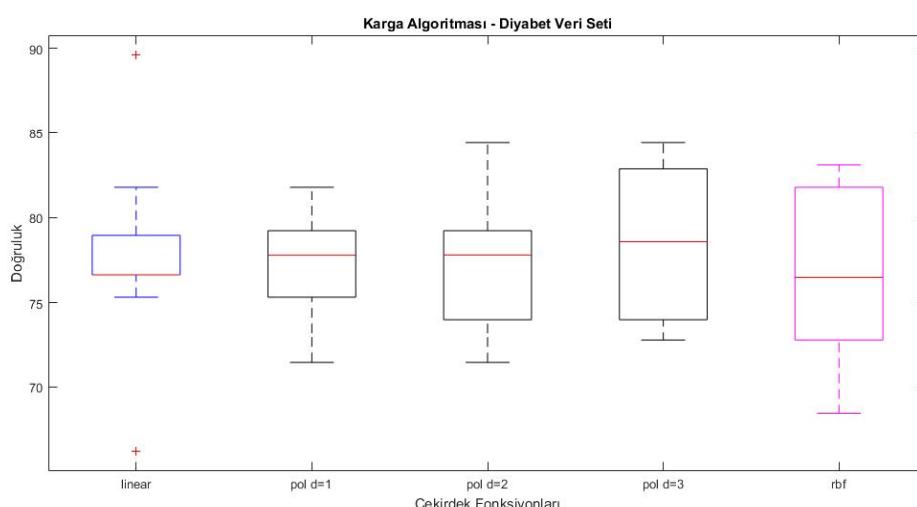
Öznitelik	Aralık
Hamile kalma sayısı	1-17
2 saat oral glikoz tolerans testi	0-199
Diyastolik tansiyon	0-122
Triceps deri kat kalınlığı	0-99
2 saat insülin serumu	0-846
Vücut kitle indexi	0-67.1
Diyabet soyağacı fonksiyonu	0.078-2.42
Yaş	21-81

Çizelge 4.7'de diyabet veri seti üzerinde detaylı sınıflandırma sonuçları verilmiştir. En başarılı doğruluk değerleri her iki algoritmada da polinom ($d=3$) çekirdek fonksiyonu ile elde edilmiştir. Tüm çekirdek fonksiyonları için KAA-DVM 77.34 ortalama doğruluk sonucu ile SSA-DVM'nin 76.74 değerinden daha başarılıdır. Modellerin duyarlılık, seçicilik ve AUC açısından elde ettikleri ortalama değerler KAA-DVM için 0.8895-0.5607-0.8291 ve SSA-DVM için 0.8793-0.5572-0.8211 hesaplanmıştır. Her üç değer içinde KAA-DVM daha başarılı sonuçlar vermiştir. Ayrıca KAA-DVM tüm değerlerin ortalama standart sapmalarında da üstünlük göstermektedir.

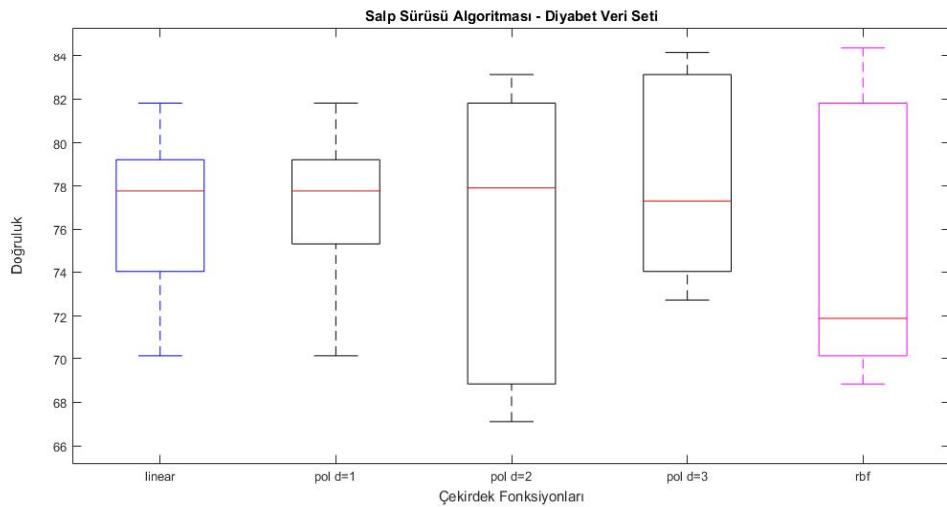
Çizelge 4.7. Pima Kızılderilileri diyabet veri seti üzerinde detaylı sınıflandırma sonuçları

Cekirdek Fonksiyonu	Değerlendirme Kriteri	KAA-DVM	SSA-DVM
Lineer	Doğruluk	77.60 ± 5.79	76.95 ± 3.98
	Duyarlılık	0.8847 ± 0.05	0.8794 ± 0.04
	Seçicilik	0.5843 ± 0.09	0.5626 ± 0.08
	AUC	0.8303 ± 0.02	0.8305 ± 0.02
Polinom D=1	Doğruluk	77.21 ± 3.65	76.95 ± 3.83
	Duyarlılık	0.8794 ± 0.04	0.8774 ± 0.04
	Seçicilik	0.5700 ± 0.07	0.5668 ± 0.08
	AUC	0.8302 ± 0.02	0.8303 ± 0.02
Polinom D=2	Doğruluk	77.21 ± 3.71	76.56 ± 6.38
	Duyarlılık	0.8975 ± 0.05	0.8905 ± 0.05
	Seçicilik	0.5506 ± 0.09	0.5321 ± 0.11
	AUC	0.8342 ± 0.02	0.8195 ± 0.02
Polinom D=3	Doğruluk	78.25 ± 4.51	77.99 ± 4.52
	Duyarlılık	0.8984 ± 0.04	0.8944 ± 0.04
	Seçicilik	0.5506 ± 0.09	0.5654 ± 0.10
	AUC	0.8291 ± 0.02	0.8286 ± 0.02
RBF	Doğruluk	76.43 ± 5.74	75.26 ± 6.37
	Duyarlılık	0.8877 ± 0.04	0.8549 ± 0.06
	Seçicilik	0.5331 ± 0.11	0.5593 ± 0.11
	AUC	0.8217 ± 0.04	0.7966 ± 0.05

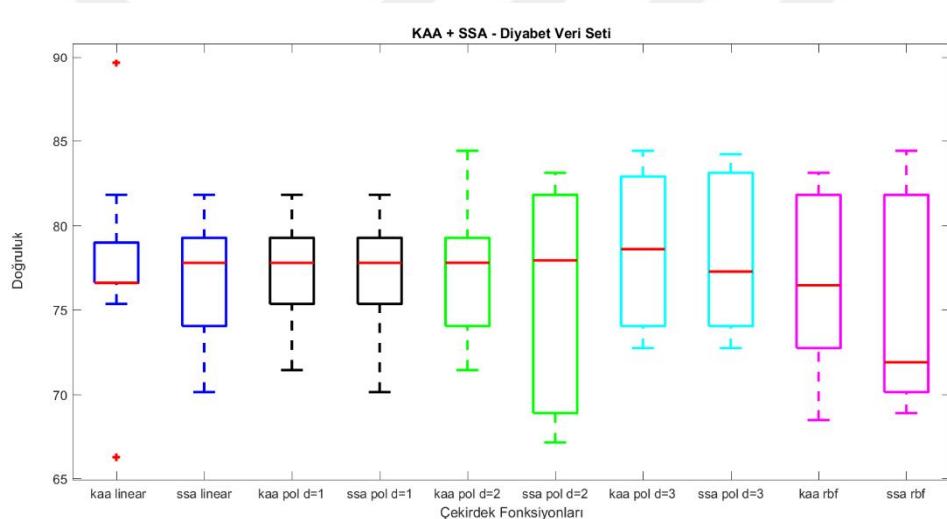
Şekil 4.9.'de KAA-DVM'nin diyabet veri seti üzerindeki kutu grafiği verilmiştir. Yaklaşık olarak eşit medyan seviyeleri vardır. Polinom (d=3) simetrik bir dağılım göstermiştir. Şekil 4.10'da SSA-DVM'nin diyabet veri seti üzerindeki kutu grafiği verilmiştir. Polinom (d=2) çekirdek fonksiyonunda kutu aşağı çeyreğe kaymıştır ve RBF çekirdek fonksiyonunda kutu yukarı çeyreğe kaymıştır. Şekil 4.11.'da KAA-DVM ve SSA-DVM'nin karşılaştırmalı kutu grafikleri verilmiştir. KAA-DVM lineer çekirdek fonksiyonunda daha başarılı bir kutu göstermektedir. RBF çekirdek fonksiyonu için SSA-DVM'ye göre KAA-DVM'nin üst çeyreği daha dar gözlemlenmiştir.



Şekil 4.9. KAA-DVM'nin diyabet veri seti üzerindeki kutu grafiği

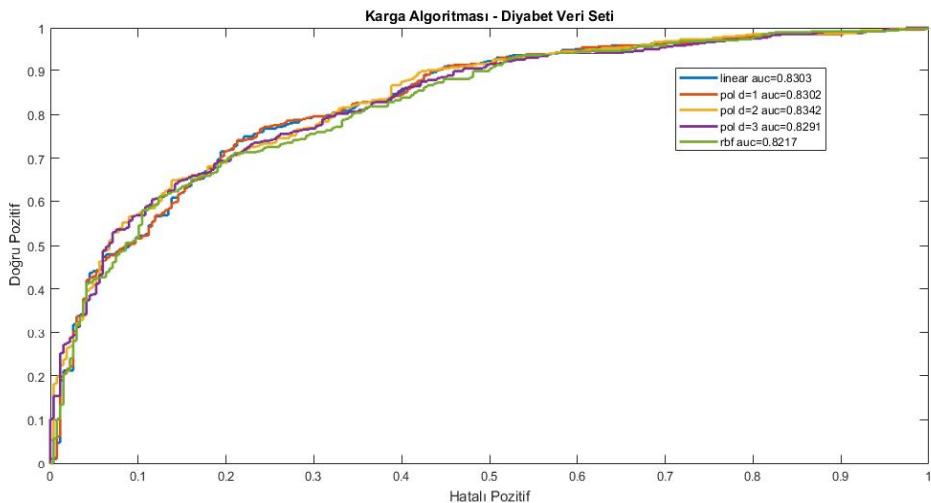


Şekil 4.10. SSA-DVM’nin diyabet veri seti üzerindeki kutu grafiği

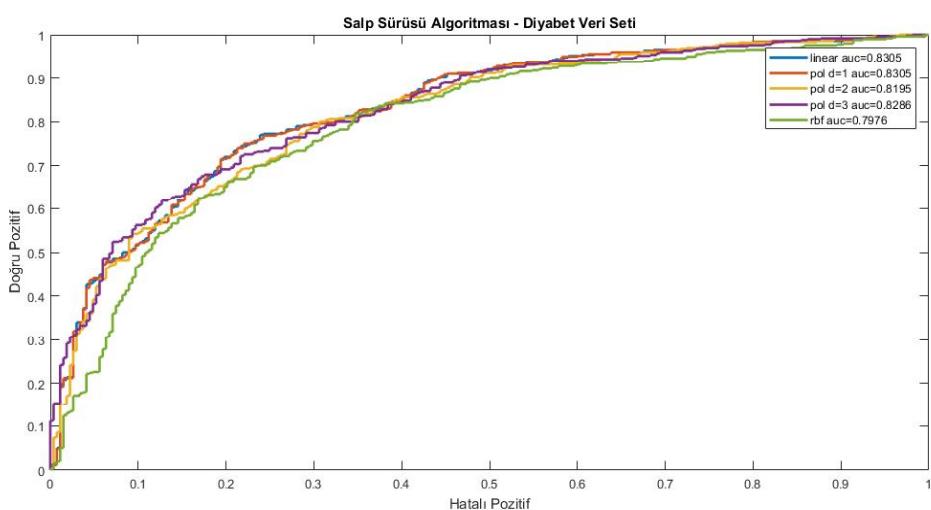


Şekil 4.11. KAA-DVM ve SSA-DVM’nin diyabet veri seti üzerindeki karşılaştırmalı kutu grafiği

Şekil 4.12’de KAA-DVM modelinin, Şekil 4.13.’de ise SSA-DVM modelinin diyabet veri seti üzerindeki ROC eğrileri verilmiştir. Her iki modelde de ROC eğrileri, eğri grafiğinin “iyi” başarısındaki bölgelerdir. Ancak SSA-DVM modelinde RBF çekirdek fonksiyonun eğrisi KAA-DVM modelinkine göre biraz daha y ekseniinden uzaklaşmıştır. Doğru pozitif değerlere yaklaşmanın daha önemli olduğu düşünüldüğünde KAA-DVM’nin RBF çekirdek fonksiyonunda daha başarılı olduğu söylenebilir.



Şekil 4.12. KAA-DVM diyabet veri seti üzerindeki ROC eğrisi



Şekil 4.13. SSA-DVM diyabet veri seti üzerindeki ROC eğrisi

Çizelge 4.8'de KAA ve SSA'nın diyabet veri seti üzerinde diğer optimizasyon algoritmaları ile karşılaştırması gösterilmektedir. KAA-DVM bu algoritmalar arasında 78.25 doğruluk oranı ile en iyi sonucu vererek birinci olmuştur. SSA-DVM ise 77.99 doğruluk oranı ile ikinci olmuştur. KAA ve SSA sadece iki algoritma tarafından standart sapma değerinde geride bırakılmıştır. Bunlar 3.72 ile GOA ve 3.81 ile MVO algoritmalarıdır.

Çizelge 4.8. KAA ve SSA'nın diyabet veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması

Algoritma	Diyabet Veri Seti
Karga Arama Alg.	78.25 ± 4.51
Salp Sürüsü Alg.	77.99 ± 4.52
GOA	76.05 ± 3.72
MVO	76.17 ± 3.81
GA	76.44 ± 6.43
PSO	76.83 ± 6.48
GWO	75.91 ± 4.55
FF	75.77 ± 4.83
BA	76.31 ± 6.50
CS	76.70 ± 3.54

4.4.3. Parkison veri seti deneyleri

Parkinson veri seti Oxford Üniversitesi'nden Max Little tarafından oluşturulmuştur (Little ve ark., 2007). Veri setinin amacı sağlıklı insanlarla Parkinson hastalığına sahip insanları ayırt etmektir. Toplamda 195 örnek ve 22 öznitelik içerir. Veri seti toplamda 31 hastanın (23 hasta, 8 sağlıklı) biyomedikal ses ölçümelerinden oluşur. Verileri alınan insanların tedavi zamanları 0-28 yıl arasında değişmektedir ve bu insanların yaş aralığı 46-85 arasındadır. Her satır bir ses kaydını ifade etmektedir. Her sütun bir ses ölçüme karşılık gelmektedir. Parkinson veri setinin 22 niteliği Çizelge 4.9'da görülebilir.

Çizelge 4.9. Parkinson veri setinin 22 niteliği

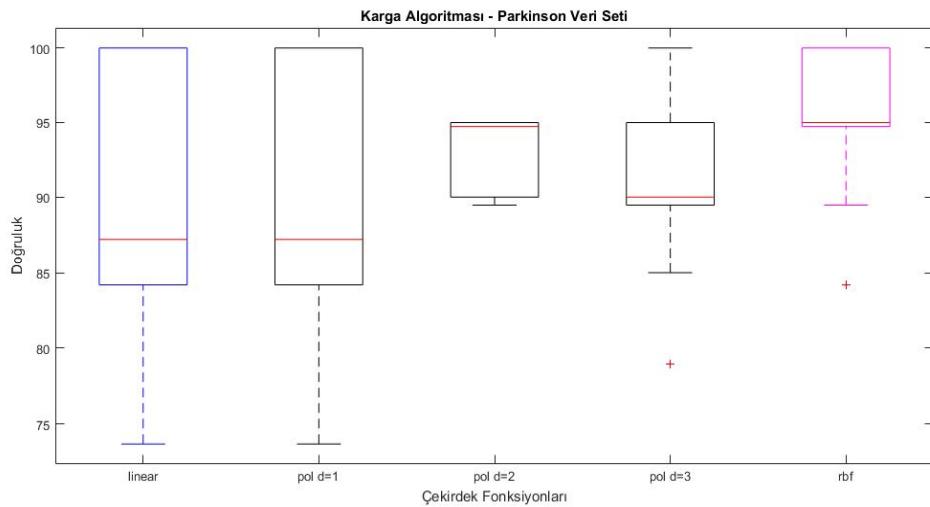
No	Öznitelik	Tanım
1	MDVP:Fo (Hz)	Ortalama vokal temel değer
2	MDVP:Fhi (Hz)	Maximum vokal temel frekans
3	MDVP:Flo (Hz)	Minimum vokal temel frekans
4	MDVP:Jitter (%)	Temel frekanstaki değişimlerin birkaç ölçümü
5	MDVP: Kitter (Abs)	
6	MDVP:RAP	
7	MDVP:PPQ	
8	Jitter:DDP	
9	MDVP:Shimmer	Genlikteki değişimlerin birkaç ölçümü
10	MDVP:Shimmerter (dB)	
11	Shimmer:APQ3	
12	Shimmer:APQ5	
13	MDVP:APQ	
14	Shimmer:DDA	
15	NHR	Sesteki tonal bileşenlerin gürültü oranlarının iki ölçümü
16	HNR	
17	RPDE	İki lineer olmayan dinamik karmaşıklık ölçümü
18	D2	
19	DFA	Sinyal fraktal ölçeklendirme üssü
20	Spread1	Üç lineer olmayan temel frekans değişimi ölçümü
21	Spred2	
22	PPE	

Çizelge 4.10'da Parkinson veri seti üzerinde detaylı sınıflandırma sonuçları görülebilir. Her iki algoritma için de RBF çekirdek fonksiyonu ile en iyi doğruluk değerleri elde edilmiştir. SSA-DVM 95.38 doğruluk değeri ile ve KAA-DVM 94.87 doğruluk değeri ile en iyi sonuçları vermişlerdir. KAA-DVM tüm çekirdek fonksiyonları doğruluk değeri ortalaması 91.38 ile SSA-DVM'nin ortalaması 90.97'den daha başarılı olmuştur. SSA-DVM modelinin tüm çekirdek fonksiyonları için ortalama standart sapması KAA-DVM'ye göre daha düşüktür. KAA-DVM ve SSA-DVM'nin duyarlılık, seçicilik ve AUC değerleri için ortalama değerleri sırasıyla 0.9488-0.9431, 0.8063-0.8120, 0.9364-0.9262 olarak hesaplanmıştır. KAA-DVM'nin seçicilik değeri haricinde daha başarılı sonuçlar verdiği görülmüştür.

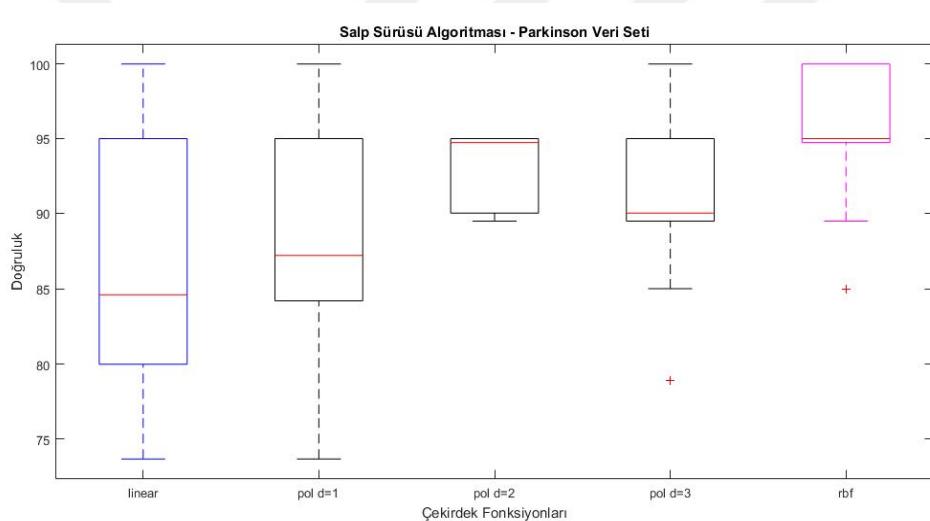
Çizelge 4.10. Parkinson veri seti üzerinde detaylı sınıflandırma sonuçları

Çekirdek Fonksiyonu	Değerlendirme Kriteri	KAA-DVM	SSA-DVM
Lineer	Doğruluk	88.72 ± 9.06	87.18 ± 8.94
	Duyarlılık	0.9546 ± 0.07	0.9350 ± 0.07
	Seçicilik	0.6850 ± 0.24	0.6850 ± 0.24
	AUC	0.8991 ± 0.05	0.8726 ± 0.08
Polinom D=1	Doğruluk	88.72 ± 9.06	87.69 ± 7.83
	Duyarlılık	0.9546 ± 0.07	0.9428 ± 0.06
	Seçicilik	0.6850 ± 0.24	0.6850 ± 0.24
	AUC	0.8989 ± 0.05	0.8734 ± 0.08
Polinom D=2	Doğruluk	93.33 ± 2.54	93.33 ± 2.54
	Duyarlılık	0.9399 ± 0.04	0.9399 ± 0.04
	Seçicilik	0.9067 ± 0.13	0.9067 ± 0.13
	AUC	0.9528 ± 0.03	0.9528 ± 0.03
Polinom D=3	Doğruluk	91.28 ± 6.49	91.28 ± 6.49
	Duyarlılık	0.9266 ± 0.06	0.9266 ± 0.06
	Seçicilik	0.8650 ± 0.18	0.8650 ± 0.18
	AUC	0.9425 ± 0.03	0.9425 ± 0.03
RBF	Doğruluk	94.87 ± 4.96	95.38 ± 5.04
	Duyarlılık	0.9682 ± 0.06	0.9714 ± 0.04
	Seçicilik	0.8900 ± 0.17	0.9181 ± 0.13
	AUC	0.9889 ± 0.0009	0.9896 ± 0.0007

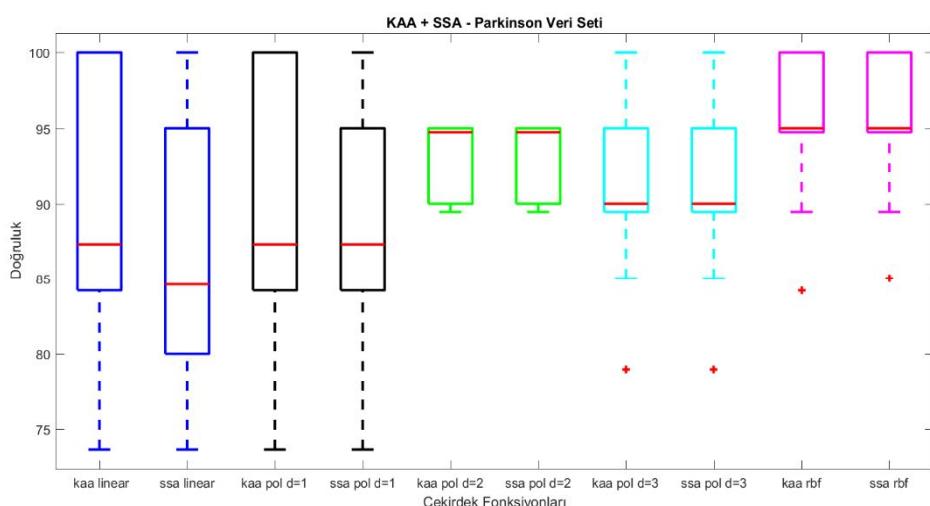
Şekil 4.14.'de KAA-DVM'nin ve Şekil 4.15.'de SSA-DVM'nin Parkinson veri seti üzerindeki kutu grafikleri verilmiştir. Her iki algoritmada da birbirine benzer kutular elde edilmiştir. Lineer ve polinom ($d=1$) çekirdek fonksiyonları her iki algoritma içinde çok yakın dağılım göstermiştir. Polinom ($d=2$)'de aşağı çeyreğe, polinom ($d=3$) ve RBF çekirdek fonksiyonlarında ise yukarı çeyreğe kayma görülmektedir. Şekil 4.16'de KAA-DVM ve SSA-DVM'nin karşılaştırmalı kutu grafikleri verilmiştir. Bu grafiklerden RBF çekirdek fonksiyonun en iyi dağılım kutusuna sahip olduğu görülebilir.



Şekil 4.14. KAA-DVM parkinson veri seti üzerindeki kutu grafiği

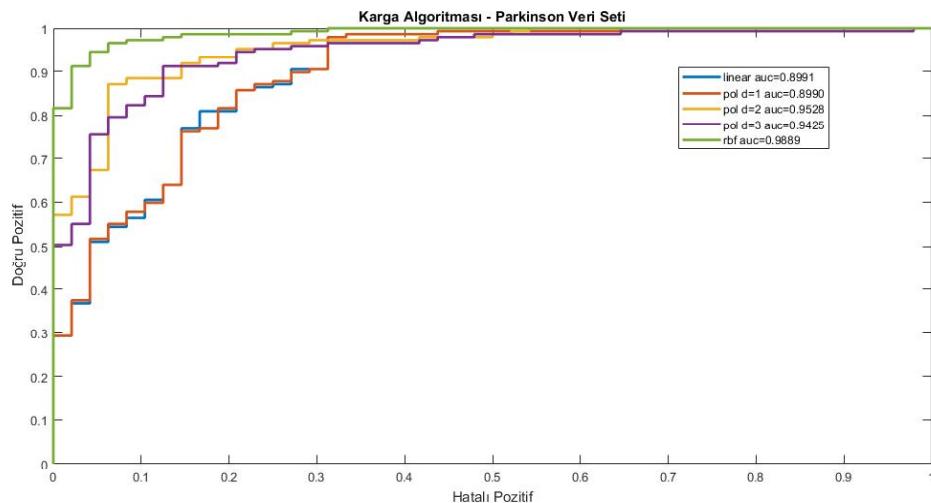


Şekil 4.15. SSA-DVM parkinson veri seti üzerindeki kutu grafiği

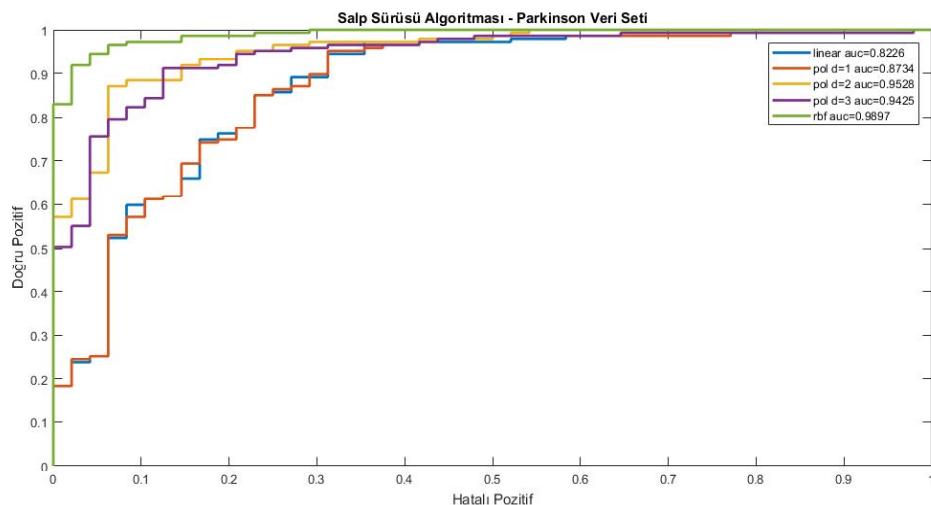


Şekil 4.16. KAA-DVM ve SSA-DVM'ın parkinson veri seti üzerindeki karşılaştırmalı kutu grafiği

Şekil 4.17.'da KAA-DVM modelinin ve Şekil 4.18'de SSA-DVM modelinin Parkinson veri seti üzerindeki ROC eğrileri verilmiştir. Her iki modelde de RBF çekirdek fonksiyonu en iyi eğriyi vermiştir ve “çok iyi” bölgesinde yer almaktadır. Polinom ($d=2$) ve polinom ($d=3$) çekirdek fonksiyonları da “çok iyi” bölgesinde yer almaktadır. Polinom ($d=1$) ve lineer çekirdek fonksiyonları ise “iyi” bölgesinde yer almaktadır.



Şekil 4.17. KAA-DVM parkinson veri seti üzerindeki ROC eğrisi



Şekil 4.18. SSA-DVM parkinson veri seti üzerindeki ROC eğrisi

Çizelge 4.11'de KAA ve SSA'nın Parkinson veri seti üzerinde diğer algoritmalarla karşılaştırmasına yer verilmiştir. SSA algoritması 95.38 doğruluk değeri ile en iyi sonucu vermektedir. KAA algoritması ise 94.87 doğruluk değeri ile altıncı sıraya yerleşmektedir. Ancak KAA algoritmasının doğruluk oranı rakiplerine çok yakındır.

Çizelge 4.11. KAA ve SSA'nın parkinson veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması

Algoritma	Parkinson Veri Seti
Karga Arama Alg.	94.87 ± 4.96
Salp Sürüsü Alg.	95.38 ± 5.04
GOA	94.95 ± 5.00
MVO	94.89 ± 5.05
GA	94.92 ± 4.47
PSO	93.37 ± 4.52
GWO	94.89 ± 5.62
FF	93.89 ± 4.93
BA	92.79 ± 6.63
CS	93.82 ± 5.55

4.4.4. Sonar veri seti deneyleri

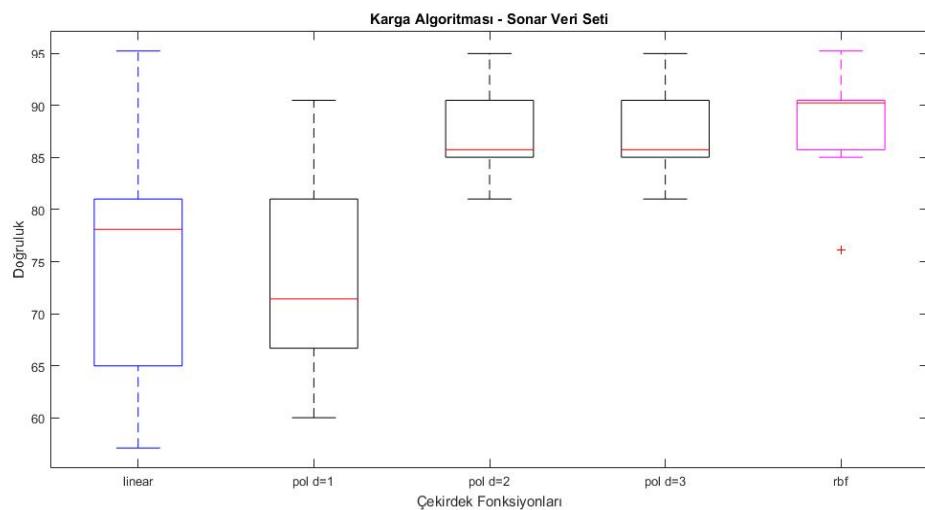
Veri setinin amacı bir metal silindir ile kabaca silindir şeklinde olan bir kaya parçasından yansıyan sonar sinyalleri birbirinden ayırt etmektir. Veri seti literatüre Gorman ve Sejnowski (1988) tarafından kazandırılmıştır. Veri setinde 111 tane sinyal metal silindirlerden ve 97 tane sinyal kayalardan yansıyan sonar verisinden elde edilmiştir. Her desen 0.0-1.0 aralığından 60 sayıdan oluşmaktadır. Her sayı bir zaman aralığında, belli bir frekans bandının enerjisini ifade eder.

Şekil 4.12.'se sonar veri seti için detaylı sınıflandırma sonuçları verilmiştir. KAA-DVM modelinde 88.46 doğruluk değeri ile RBF çekirdek fonksiyonu, SSA-DVM modelinde ise 88.50 doğruluk değeri ile yine RBF çekirdek fonksiyonu en iyi değeri vermiştir. KAA-DVM'nin tüm çekirdek fonksiyonları üzerindeki ortalama doğruluk değeri 82.40 olarak belirlenmiştir. SSA-DVM ise biraz daha düşük 82.22 ortalama doğruluk değeri elde etmiştir. KAA-DVM ve SSA-DVM'nin duyarlılık, seçicilik ve AUC ortalamaları sırasıyla 0.8007-0.7914, 0.8557-0.8516, 0.8756-0.8731 olarak belirlenmiştir. Tüm ortalama durumlarında KAA-DVM daha iyi sonuçlar vermiştir.

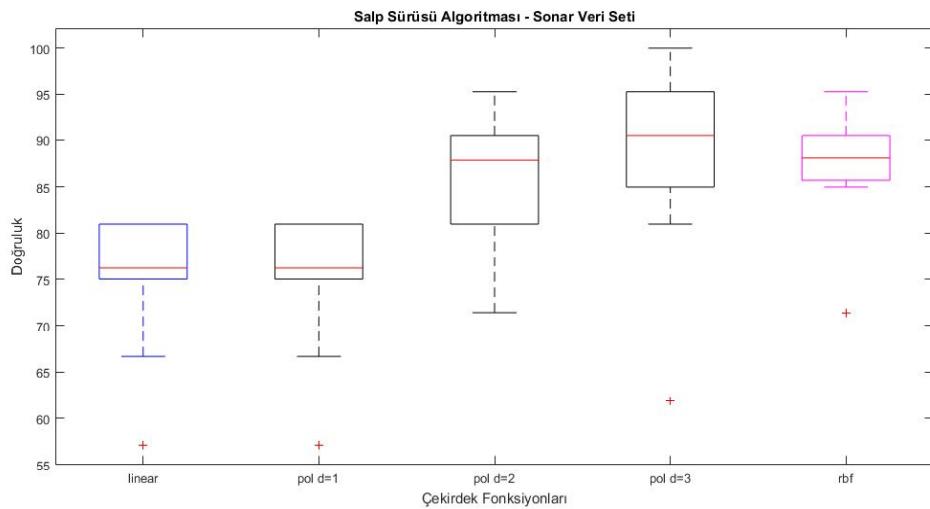
Çizelge 4.12. Sonar veri seti üzerinde detaylı sınıflandırma sonuçları

Cekirdek Fonksiyonu	Değerlendirme Kriteri	KAA-DVM	SSA-DVM
Lineer	Doğruluk	75.48 ± 11.65	75.00 ± 7.62
	Duyarlılık	0.7350 ± 0.17	0.7196 ± 0.14
	Seçicilik	0.7740 ± 0.10	0.7765 ± 0.09
	AUC	0.7968 ± 0.03	0.7865 ± 0.03
Polinom D=1	Doğruluk	74.04 ± 9.32	75.00 ± 7.63
	Duyarlılık	0.7168 ± 0.18	0.7196 ± 0.14
	Seçicilik	0.7682 ± 0.12	0.7765 ± 0.09
	AUC	0.7990 ± 0.04	0.7865 ± 0.03
Polinom D=2	Doğruluk	87.02 ± 4.50	85.10 ± 7.72
	Duyarlılık	0.8484 ± 0.10	0.8342 ± 0.10
	Seçicilik	0.9068 ± 0.08	0.8766 ± 0.10
	AUC	0.9116 ± 0.06	0.9116 ± 0.06
Polinom D=3	Doğruluk	87.02 ± 4.50	87.50 ± 10.59
	Duyarlılık	0.8484 ± 0.10	0.8496 ± 0.15
	Seçicilik	0.9068 ± 0.08	0.9075 ± 0.09
	AUC	0.9116 ± 0.06	0.9284 ± 0.06
RBF	Doğruluk	88.46 ± 5.62	88.50 ± 6.80
	Duyarlılık	0.8550 ± 0.10	0.8230 ± 0.14
	Seçicilik	0.9229 ± 0.06	0.9320 ± 0.07
	AUC	0.9591 ± 0.06	0.9524 ± 0.04

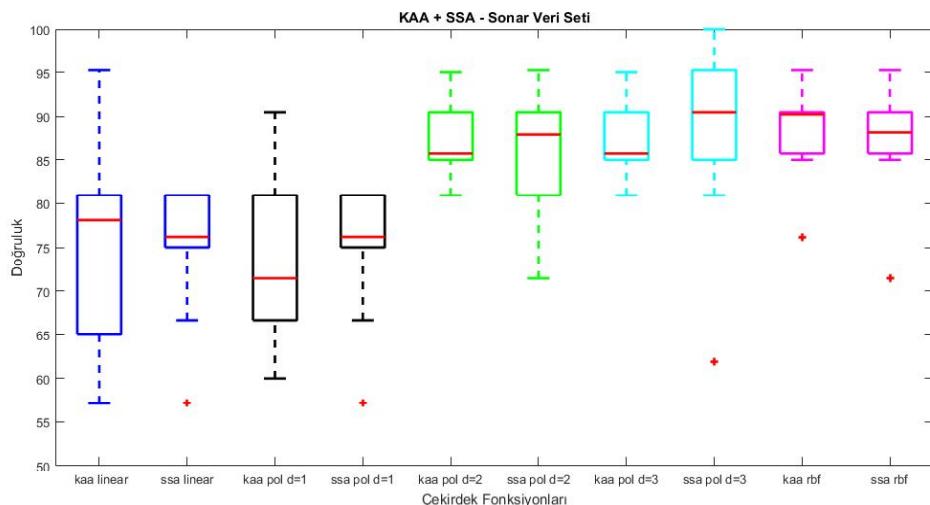
Şekil 4.19'da KAA-DVM'nin ve Şekil 4.20'de SSA-DVM'nin kutu grafikleri görülmektedir. Şekil 4.21.'de ise KAA-DVM ve SSA-DVM'nin karşılaştırmalı kutu grafikleri görülmektedir. Her iki modelde de polinom ($d=1$) ve lineer çekirdek fonksiyonları diğer çekirdek fonksiyonlarına kıyasla daha kötü dağılım göstermiştir. İki modelin birbirine yakın dağılımlar gösterdiği gözlenmiştir.



Şekil 4.19. KAA-DVM sonar veri seti üzerindeki kutu grafiği



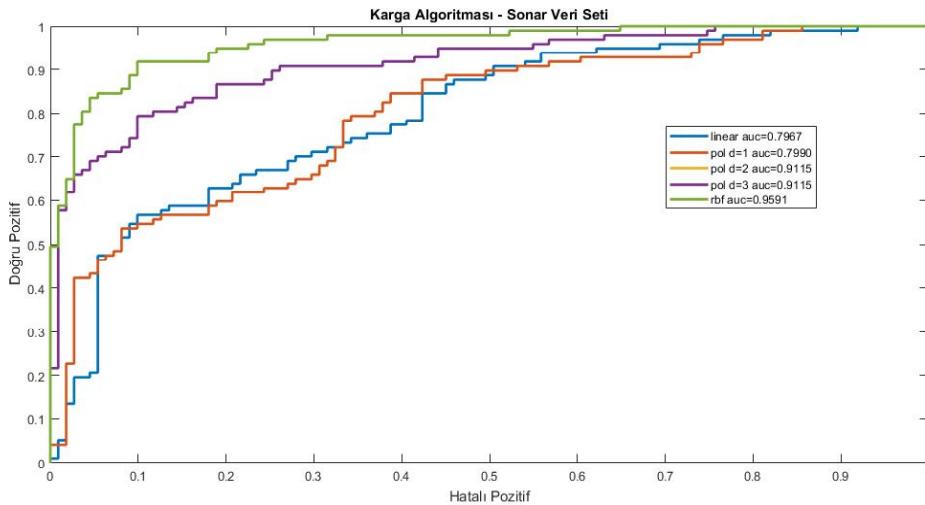
Şekil 4.20. SSA-DVM sonar veri seti üzerindeki kutu grafiği



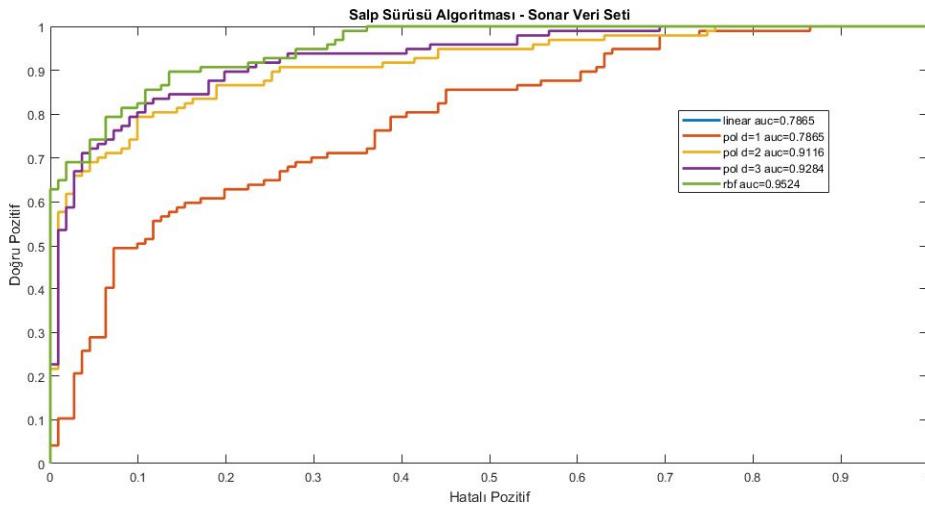
Şekil 4.21. KAA-DVM ve SSA-DVM'ın sonar veri seti üzerindeki karşılaştırmalı kutu grafiği

Şekil 4.22'de KAA-DVM modelinin ROC eğrisi görülmektedir. RBF çekidek fonksiyonu en başarılı eğriyi vermiştir. RBF, polinom ($d=2,3$) çekirdek fonksiyonlarının ROC "çok iyi" bölümünde oldukları, lineer ve polinom ($d=1$)'in ise "uygun" bölümünde yer aldıkları görülmüştür.

Şekil 4.23'de SSA-DVM modelinin ROC eğrisi görülmektedir. RBF çekidek fonksiyonu en başarılı eğriyi vermiştir. RBF, polinom ($d=2,3$) çekirdek fonksiyonlarının ROC "çok iyi" bölümünde oldukları, lineer ve polinom ($d=1$)'in ise "uygun" bölümünde yer aldıkları görülmüştür.



Şekil 4.22. KAA-DVM sonar veri seti üzerindeki ROC eğrisi



Şekil 4.23. SSA-DVM sonar veri seti üzerindeki ROC eğrisi

Çizelge 4.13'de KAA ve SSA modellerinin diğer algoritmalar ile doğruluk değeri üzerinden karşılaştırması verilmiştir. Her iki algoritmanın da başarılı oldukları ve diğer algoritmalar ile çok yakın değerler verdikleri görülmektedir.

Çizelge 4.13. KAA ve SSA'nın sonar veri seti üzerinde diğer algoritmalarla doğruluk karşılaştırması

Algoritma	Parkinson Veri Seti
Karga Arama Alg.	88.46 ± 5.62
Salp Sürüsü Alg.	88.50 ± 6.80
GOA	88.55 ± 8.01
MVO	88.50 ± 6.14
GA	87.52 ± 8.57
PSO	87.98 ± 5.34
GWO	88.50 ± 5.93
FA	88.45 ± 3.85
BA	88.02 ± 7.48
CSA	85.10 ± 6.88

4.5. Çok Sınıflı Veri Setleri ve Literatür Karşılaştırması

Modellerimizin çok sınıflı veri setlerindeki performansını da gözlemlemek için UCI veri tabanında üç tane çok sınıflı veri seti seçtik. Şarap, cam ve ünlü harf veri setlerinin öznitelik ve örnek sayıları Çizelge 4.1'de verilmiştir.

Şarap veri seti (Aeberhard ve ark., 1994) İtalya'da üç farklı türde üretilen şarapların analiz verilerini içerir. 13 farklı bileşen miktarı kaydedilmiştir. Cam veri seti (Evett ve Spiehler, 1987) Amerika Birleşik Devletleri (ABD) adli bilimler ofisinden alınmıştır. Kriminolojik vakalarda, suç ortamında bulunan camların ayırt edilebilmesi veri setinin temel amacıdır. Oksit miktarlarına göre 6 tip cam türü içerir. Ünlü harf veri setinin (Niranjan ve Fallside, 1990) amacı İngiliz İngilizcesindeki bazı ünlü sesleri konuşmacıdan bağımsız olarak tespit edebilmektir. Ünlüler 0-10 arasında indekslenmiştir. Konuşmacılar 0-89 arasında indekslenmiştir. Aslında her sesi 6 kere tekrarlayan 15 konuşmacı vardır.

Çizelge 4.14. KAA ve SSA model performanslarının literatür karşılaştırması

GÖĞÜS KANSERİ	DİYABET	PARKİNSON	SONAR	ŞARAP	CAM	ÜNLÜ HARF
KAA	97.22	78.25	94.87	88.46	97.78	67.77
SSA	96.72	77.99	95.38	88.50	97.19	66.64
GA	96.19	81.50	-	98.00	-	-
GS	95.30	77.3	-	87	-	-
PSO	97.95	80.19	-	88.32	-	-
GOA	97.23	76.0	94.95	88.55	97.77	70.06
BA	96.80	83.50	-	96.30	97.60	83.80
FOA	96.90	77.46	96.90	-	-	-
SSO	-	71.12	-	69.42	-	59.99
IACO	96.93	86.98	-	-	92.70	63.80

Çizelge 4.14.'te farklı algoritmaların farklı çalışmalardan elde edilmiş DVM parametre optimizasyonu doğruluk oranı başarıları verilmiştir. Çizelgede görülmektedir ki KAA-DVM ve SSA-DVM modelleri literatürdeki diğer çalışmalara yakın sonuçlar vermektedir. Ayrıca çok sınıflı veri setlerinde de başarılı sonuçlar elde edilmiştir. Çizelge 4.15.'te karşılaştırma yapılan çalışmaların referansları verilmiştir. Ayrıca Şekil 4.24'te literatür karşılaştırması grafiksel olarak verilmiştir. Şekil 4.24.'te, Çizelge 4.14'te gösterilen boş değerler her veri setinin en küçük değeri ile doldurulmuştur. Böylece KAA

ve SSA'nın diğer uygulamalar arasındaki yeri gösterilmeye çalışılmıştır. KAA ve SSA algoritmaların tüm veri setlerinde diğer algoritmala yakın seviyelerde yer aldığı görülmektedir. Sonar veri setinde KAA 88.46, SSA ise 88.50 doğruluk değerleri ile GA algoritmasının 98.00 doğruluk oranından daha düşük sonuçlar vermişlerdir. Cam veri setinde BA algoritması 83.80 doğruluk oranı vermiştir. KAA ve SSA 67.77 ve 66.64 doğruluk oranları ile BA algoritmasının gerisinde kalmışlardır.

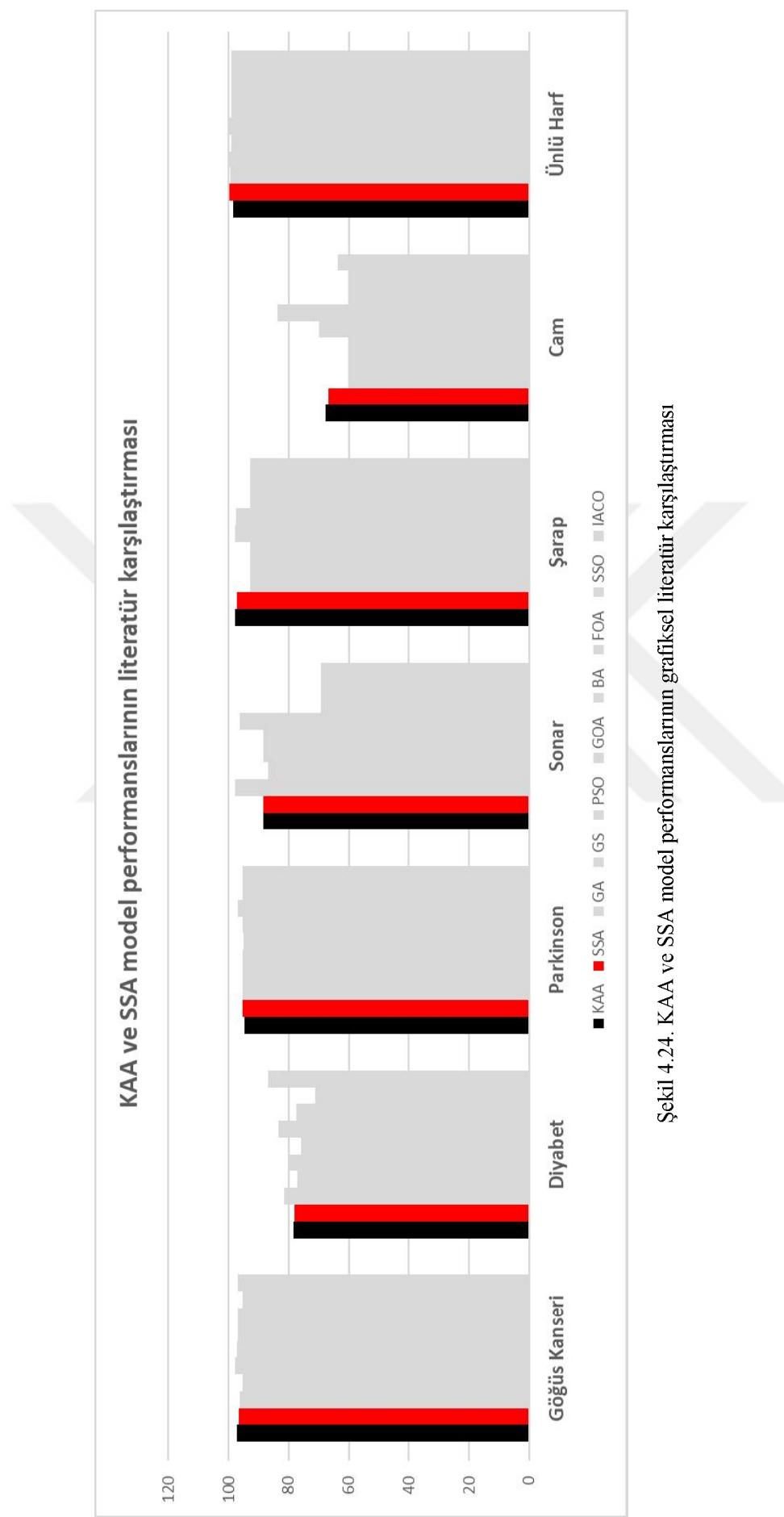
Çizelge 4.15. Literatür karşılaştırması yapılan çalışmalar

Çalışma İsmi	İterasyon Sayısı	Veri Seti	Referans
KAA-DVM	100	7	Bu çalışma
SSA-DVM	100	7	Bu çalışma
GA-DVM	600	11	(Huang ve Wang, 2006)
GS-DVM	600	11	(Huang ve Wang, 2006)
PSO-DVM	250	17	(Lin ve ark., 2008)
GOA-DVM	200	18	(Aljarah ve ark., 2018)
BA-DVM	20	9	(Tharwat ve ark., 2017)
FOA-DVM	250	4	(Shen ve ark., 2016)
SSO-DVM	200	10	(Pereira ve ark., 2014)
IACO-DVM	500	9	(Chen ve Tian, 2016)

4.6. Bölüm Değerlendirmesi

Bu bölüme DVM parametre optimizasyonu için oluşturulan modelin yapısı anlatılarak başlanmıştır. Modelin akış şeması verilmiştir. DVM parametre optimizasyonu için DVM'de kullanılacak olan uygunluk fonksiyonundan ve karar değişkenlerinden bahsedilmiştir. Kullanılan veri setlerinin örnek sayısı, öznitelik sayısı vb. belirtilmiştir. Veriler üzerindeki ön işlemeden bahsedilmiştir. Karşılaştırma yapılacak olan literatür algoritmaları ve algoritmaya spesifik parametre ayarları verilmiştir. Deneylerde kullanılan materyal ve deney ortamından bahsedilmiştir.

Deneyler her veri seti için ayrı bölümlerde yapılmıştır. Her bölümde kullanılan veri setinin açıklaması ve ayrıntılı özellikleri verilmiştir. Daha sonra kullanılan her iki optimizasyon algoritmasının doğruluk, duyarlılık, seçicilik ve AUC değerleri çizelgeler halinde verilmiştir. Bu değerler DVM çekirdek fonksiyonlarından lineer, polinom ($d=1,2,3$) ve RBF için ayrı ayrı verilmiştir. Deney sonunda optimize edilen parametrelerin DVM değerlendirmesi sonundaki ROC eğrileri çizdirilmiştir. Ayrıca sonuçların dağılımlarını gözlemleyebilmek için kutu grafikleri çizdirilmiştir. Her veri seti ile elde edilen doğruluk sonuçları literatürden seçilen bir çalışmada (Aljarah ve ark., 2018) yer alan diğer algoritmalar ile karşılaştırılmıştır. Modellerin çok sınıflı veri setlerindeki başarılarına ve literatürdeki farklı çalışmalar ile karşılaştırmalara da yer verilmiştir.



Sekil 4.24. KAA ve SSA model performanslarının grafiksel literatür karşılaştırması

5. SONUÇLAR VE ÖNERİLER

DVM parametrelerinin optimizasyonu konusunda kayda değer sayıda çalışma yapılmıştır. Ancak sürekli literatüre giren yeni optimizasyon algoritmaları bulunmaktadır. Bu yeni optimizasyon algoritmalarının DVM parametrelerinin belirlenmesindeki performansının ölçülmesi önem arz etmektedir. Yakın zamanda literatüre giren iki doğadan esinlenmeli meta-sezgisel algoritma olan Karga Arama Algoritması (KAA) ve Salp Sürüsü Algoritması (SSA) bu çalışmada kullanılmıştır. SSA sürü zekâsı mantığına dayanmaktadır.

KAA 2016 yılında yayınlanmıştır ve meta-sezgisel bir algoritmadır. Bu algoritma kargaların diğer kargaların besin depoları çalmak için ortaya koydukları zeki davranışlara dayanmaktadır. Bir karga diğer bir karganın gizli deposunu bulmak için onu takip edebilir. Takip edilen karga ise bunun farkına varıp takipçisini yaniltma girişiminde bulunabilir. Bu süreçten ilham alarak modellenen KAA, çeşitlendirme ve yoğunlaştırma dengesini sağlayacak parametrelere sahiptir.

SSA ismiyle ortaya çıkan ve 2017 yılında yayınlanmış bir meta-sezgisel algoritmadır. Salp denilen okyanus canlısının derin sularda oluşturduğu sürü özelliklerinden esinlenmiştir. Salp canlıları derin okyanuslarda salp zinciri denilen yapılar oluştururlar. Bu yapıların amacı kesin olarak bilinmese de ortaklaşa daha hızlı hareket etmek için oluşturuldukları düşünülmektedir. Salp canlıları hakkında biyolojik çalışma dahi az iken, yazarlar bu alandan bir optimizasyon süreci çıkarmışlardır. Bu algoritma da çeşitlendirme ve yoğunlaştırma dengesini sağlayacak parametrelere sahiptir.

Deneysel çalışmalar yedi farklı UCI veri seti üzerinde üç farklı çekirdek fonksiyonu (lineer, polinom ($d=1,2,3$), RBF) kullanılarak uygulanmıştır. Çekirdek fonksiyonları DVM'lere lineer olmayan örnek uzaylarını sınıflandırma imkânı tanırlar. Çekirdek fonksiyonları hem DVM doğruluk performansını artırırlar hem de hesaplama karmaşıklığını azaltırlar. Polinom çekirdek fonksiyonu özellikle görüntü işleme alanında çok kullanılmaktadır. RBF çekirdek fonksiyonu ise ilk akla gelen ve en çok kullanılan çekirdek fonksiyonudur. Ancak bu çekirdek fonksiyonlarının birbirlerinden veya lineer çekirdek fonksiyonundan her zaman üstün oldukları söylenemez. Çalışmadaki deneyler göstermektedir ki RBF çekirdek fonksiyonu hem doğruluk hem AUC açısından veri setlerinin tamamı düşünüldüğünde daha stabil sonuçlar vermektedir. Lineer çekirdek fonksiyonu ve polinom ($d=1$) çekirdek fonksiyonu wisconsin göğüs kanseri veri seti haricinde diğerlerine nazaran kötü sonuçlar vermişlerdir. Ayrıca lineer ve polinom ($d=1$)

çekirdek fonksiyonları gerek rakamsal sonuçlarda gerekse grafiklerde çok benzer özellikler göstermektedir. Aslında bu iki çekirdek fonksiyonu matematiksel olarak aynı şeyi ifade etmektedir. Modellerin uygunluğu kontrol edebilmek için ikisi beraber deneylere dahil edilmiştir. Parkinson ve sonar veri setlerinin ROC eğrilerine bakıldığında RBF çekirdek fonksiyonunun üstünlüğü açıkça görülebilir. Bu veri setlerinde RBF çekirdek fonksiyonu eğrinin “çok iyi” kısmında yer alırken diğer çekirdek fonksiyonları “iyi” ve ”uygun” kısımlarında yer almışlardır. Bazı durumlarda polinom ($d=3$) RBF’e göre daha iyi değerler vermiştir. Ancak polinom çekirdek fonksiyonunun $d>2$ için *aşırı-uyuma* (overfit) sebep olabileceği unutulmamalıdır. Lineer çekirdek fonksiyonunun göğüs kanseri veri setinde yüksek doğruluk oranı (96.72) verdieneniğini görmekteyiz. Bu veri seti için yüksek boyutlu özellik uzayına ihtiyaç duyulmaması performans katkısı sağlayabilir. Lineer çekirdek fonksiyonunun örnek sayısının yüksek olduğu göğüs kanseri veri setinde iyi sonuç vermesi dikkate değerdir. Çünkü örnek uzayının genişlemesi daha iyi lineer ayırcı bulma ihtimalini artırmaktadır.

KAA ve SSA algoritmalarını karşılaştırdığımızda KAA algoritmasının daha başarılı olduğunu görmekteyiz. Doğruluk, duyarlılık, seçicilik ve AUC değerlerinde hem her çekirdek fonksiyonu için ayrı ayrı hem de ortalama olarak, çoğu durumda KAA üstünlük sağlamaktadır. Parkinson veri setinde SSA 95.38 doğruluk oranı ile KAA’nın 94.87 doğruluk oranından daha iyi sonuç vermiştir. Kutu grafikleri incelendiğinde KAA’nın SSA’ya göre biraz daha iyi bir dağılım gösterdiği söylenebilir. RBF çekirdek fonksiyonu deneylerinin kutu grafiklerinde diğer çekirdek fonksiyonlarına göre daha dar ve yukarıda olan kutular oluşturduğu görülmektedir.

Bu tez çalışması DVM parametre optimizasyonu modeli için iki farklı metasezgisel optimizasyon algoritması önermiştir. Literatür karşılaştırmaları göstermektedir ki KAA-DVM ve SSA-DVM parametre optimizasyonu için kararlı ve güvenilir alternatifler olarak sunulabilirler. Çalışmada önerilen modellerin DVM parametrelerini başarılı bir şekilde ayarlayabileceği görülmüştür. DVM parametre optimizasyonunu geliştirme kapasitesine sahip iki algoritma, DVM literatürüne kazandırılmıştır.

Gelecek çalışmalar için DVM’nin uzantıları olan diğer metodolojiler kullanılabilir. Örneğin İkiz-DVM (TWSVM) modeli her sınıf için paralel olmayan düzlemler oluşturmaktadır ve hesaplama karmaşıklığını büyük oranda azaltmaktadır. Kullanılan optimizasyon algoritmaları yeni oldukları için literatürde orijinal versiyonuna üstünlük sağlayan kayda değer çalışmalar yoktur. Bu optimizasyon algoritmalarının

yapısına müdahale edilerek geliştirilebilirler. SSA algoritmasının çok sınıflı optimizasyon uyarlaması algoritmanın yazarları tarafından halihazırda önerilmiştir.



KAYNAKLAR

- Aeberhard, S., Coomans, D. ve de Vel, O., 1994, Comparative analysis of statistical pattern recognition methods in high dimensional settings, *Pattern Recognition*, 27 (8), 1065-1077.
- Alatas, B., 2011, ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization, *Expert Systems with Applications*, 38 (10), 13170-13180.
- Ali, S. ve Smith-Miles, K., 2003, Automatic parameter selection for polynomial kernel, p.
- Aljarah, I., Al-Zoubi, A. M., Faris, H., Hassonah, M. A., Mirjalili, S. ve Saadeh, H., 2018, Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm, *Cognitive Computation*, 10 (3), 478-495.
- Anderson, P. A. ve Bone, Q., 1980, Communication between individuals in salp chains. II. Physiology, *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 210 (1181), 559.
- Ao, H., Junsheng, C., Yang, Y. ve Truong, T., 2013, The support vector machine parameter optimization method based on artificial chemical reaction optimization algorithm and its application to roller bearing fault diagnosis, p.
- Askarzadeh, A., 2016, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Computers & Structures*, 169, 1-12.
- Bamakan, S. M. H., Wang, H. ve Ravasan, A. Z., 2016, Parameters Optimization for Nonparallel Support Vector Machine by Particle Swarm Optimization, *Procedia Computer Science*, 91, 482-491.
- Bian, X.-Q., Zhang, Q., Zhang, L. ve Chen, J., 2017, A grey wolf optimizer-based support vector machine for the solubility of aromatic compounds in supercritical carbon dioxide, *Chemical Engineering Research and Design*, 123, 284-294.
- Birattari, M., Paquete, L., Stützle, T. ve Varrentrapp, K., 2001, Classification of metaheuristics and design of experiments for the analysis of components, *Teknik Rapor, AIDA-01-05*.
- Blum, C. ve Roli, A., 2003, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.*, 35 (3), 268-308.
- Borges, L., 2015, Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection, p.
- Boussaïd, I., Lepagnot, J. ve Siarry, P., 2013, A survey on optimization metaheuristics, *Information Sciences*, 237, 82-117.
- Chang, C.-C. ve Lin, C.-J., 2011, LIBSVM: A library for support vector machines, *ACM Trans. Intell. Syst. Technol.*, 2 (3), 1-27.
- Chapelle, O., Vapnik, V., Bousquet, O. ve Mukherjee, S., 2002, Choosing Multiple Parameters for Support Vector Machines, *Machine Learning*, 46 (1), 131-159.
- Chen, H.-L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.-J. ve Liu, D.-Y., 2011a, A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method, *Knowledge-Based Systems*, 24 (8), 1348-1359.
- Chen, J.-L., Liu, H.-B., Wu, W. ve Xie, D.-T., 2011b, Estimation of monthly solar radiation from measured temperatures using support vector machines – A case study, *Renewable Energy*, 36 (1), 413-420.
- Chen, W. ve Tian, Y., 2016, Parameter Optimization of SVM Based on Improved ACO for Data Classification.

- Chou, J.-S., Cheng, M.-Y., Wu, Y.-W. ve Pham, A.-D., 2014, Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification, *Expert Systems with Applications*, 41 (8), 3955-3964.
- Cortes, C. ve Vapnik, V., 1995, Support-Vector Networks, *Machine Learning*, 20 (3), 273-297.
- Ding, S., An, Y., Zhang, X., Wu, F. ve Xue, Y., 2017, Wavelet twin support vector machines based on glowworm swarm optimization, *Neurocomputing*, 225, 157-163.
- Dong, H. ve Jian, G., 2015, Parameter Selection of a Support Vector Machine, Based on a Chaotic Particle Swarm Optimization Algorithm, *Cybern. Inf. Technol.*, 15 (3), 140-149.
- Dorigo, M., Birattari, M. ve Stutzle, T., 2006, Ant colony optimization, *IEEE Computational Intelligence Magazine*, 1 (4), 28-39.
- Dua, D., Karra Taniskidou, E., 2017, UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science.
- Evett, I. W. ve Spiehler, E., 1987, Rule induction in forensic science, *KBS in Goverment*, 107-118.
- Fawcett, T., 2004, ROC graphs: Notes and practical considerations for researchers, *Machine Learning*, 31 (1), 1-38.
- Fister jr, I., Yang, X.-S., Fister, I., Brest, J. ve Fister, D., 2013, A Brief Review of Nature-Inspired Algorithms for Optimization, p.
- Friedrichs, F. ve Igel, C., 2005, Evolutionary tuning of multiple SVM parameters, *Neurocomput.*, 64, 107-117.
- Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, 13 (5), 533-549.
- Gorman, R. P. ve Sejnowski, T. J., 1988, Analysis of hidden units in a layered network trained to classify sonar targets, *Neural Networks*, 1 (1), 75-89.
- Gumus, E., Kilic, N., Sertbas, A. ve Ucan, O. N., 2010, Evaluation of face recognition techniques using PCA, wavelets and SVM, *Expert Systems with Applications*, 37 (9), 6404-6408.
- H Wolberg, W. ve L Mangasarian, O., 1991, Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology, p.
- Hefny, H. A. ve Azab, S. S., 2010, Chaotic particle swarm optimization, *2010 The 7th International Conference on Informatics and Systems (INFOS)*, 1-8.
- Holland, J. H., 1992, Adaptation in natural and artificial systems, MIT Press, p.
- Hu, J., Zeng, X. ve Xiao, J., 2010, Artificial Fish School Algorithm for Function Optimization, *2010 2nd International Conference on Information Engineering and Computer Science*, 1-4.
- Huang, C.-L. ve Wang, C.-J., 2006, A GA-based feature selection and parameters optimizationfor support vector machines, *Expert Systems with Applications*, 31 (2), 231-240.
- James, G., Witten, D., Hastie, T. ve Tibshirani, R., 2014, An Introduction to Statistical Learning: with Applications in R, Springer Publishing Company, Incorporated, p.
- Jean-Pierre, D., Florent, B., Hairong, X., Annick, P. ve Botao, F., 2007, Nonlinear SVM Approaches to QSPR/QSAR Studies and Drug Design, *Current Computer-Aided Drug Design*, 3 (4), 263-289.
- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *Neural Networks*, 1995. *Proceedings., IEEE International Conference on*, 1942-1948 vol.1944.

- Krishnanand, K. ve Ghose, D., 2005, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, *Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 84-91.
- LeCun, Y., Cortes, C. ve Burges, C. J. C., 2018, THE MNIST DATABASE of handwritten digits.
- Li, F. ve Aggarwal, R. K., 2000, Fast and accurate power dispatch using a relaxed genetic algorithm and a local gradient technique, *Expert Systems with Applications*, 19 (3), 159-165.
- Li, X. Z. ve Kong, J. M., 2014, Application of GA-SVM method with parameter optimization for landslide development prediction, *Nat. Hazards Earth Syst. Sci.*, 14 (3), 525-533.
- Lin, K.-C., Chen, S.-Y. ve Hung, J. C., 2015, Feature selection and parameter optimization of support vector machines based on modified artificial fish swarm algorithms, *Mathematical Problems in Engineering*, 2015.
- Lin, S.-W., Ying, K.-C., Chen, S.-C. ve Lee, Z.-J., 2008, Particle swarm optimization for parameter determination and feature selection of support vector machines, *Expert Systems with Applications*, 35 (4), 1817-1824.
- Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A. ve Moroz, I. M., 2007, Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection, *BioMedical Engineering OnLine*, 6 (1), 23.
- Madin, L. P., 1990, Aspects of jet propulsion in salps, *Canadian Journal of Zoology*, 68 (4), 765-777.
- Mangasarian, O. L., Street, W. N. ve Wolberg, W. H., 1995, Breast Cancer Diagnosis and Prognosis Via Linear Programming, *Operations Research*, 43 (4), 570-577.
- Miettinen, K., 1999, Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, GE, John Wiley & Sons, Inc., p.
- Mirjalili, S., Mirjalili, S. M. ve Lewis, A., 2014, Grey Wolf Optimizer, *Advances in Engineering Software*, 69, 46-61.
- Mirjalili, S., 2016, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Computing and Applications*, 27 (4), 1053-1073.
- Mirjalili, S. ve Lewis, A., 2016, The Whale Optimization Algorithm, *Advances in Engineering Software*, 95, 51-67.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. ve Mirjalili, S. M., 2017, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, 114, 163-191.
- Nasraoui, O., 2008, Web data mining: exploring hyperlinks, contents, and usage data, *SIGKDD Explor. Newsl.*, 10 (2), 23-25.
- NEFSC, 2013, "Greetings from the nightwatch." Field Fresh <https://nefsc.wordpress.com/2013/08/07/greetings-from-the-nightwatch/> 2018.
- Niranjan, M. ve Fallside, F., 1990, Neural networks and radial basis functions in classifying static speech patterns, *Computer Speech & Language*, 4 (3), 275-289.
- Nocedal, J. ve Wright, S. J., 2006, Numerical optimization 2nd, Springer.
- Pai, P.-F., Hsu, M.-F. ve Lin, L., 2014, Enhancing decisions with life cycle analysis for risk management, *Neural Computing and Applications*, 24 (7), 1717-1724.
- Pereira, D. R., Pazoti, M. A., Pereira, L. A. M. ve Papa, J. P., 2014, A social-spider optimization approach for support vector machines parameters tuning, *2014 IEEE Symposium on Swarm Intelligence*, 1-6.

- Pereira, D. R., Pazoti, M. A., Pereira, L. A. M., Rodrigues, D., Ramos, C. O., Souza, A. N. ve Papa, J. P., 2016, Social-Spider Optimization-based Support Vector Machines applied for energy theft detection, *Computers & Electrical Engineering*, 49, 25-38.
- Pereira, L., Papa, J. ve Souza, A., 2013, Harmony search applied for support vector machines training optimization, p.
- Rincon, P., 2005, Crows and jays top bird IQ scale <http://news.bbc.co.uk/2/hi/science/nature/4286965.stm>: [15.03.2018].
- Sai, L. ve Huajing, F., 2017, A WOA-based algorithm for parameter optimization of support vector regression and its application to condition prognostics, *2017 36th Chinese Control Conference (CCC)*, 7345-7350.
- Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., Yang, B. ve Liu, D., 2016, Evolving support vector machines using fruit fly optimization for medical data classification, *Knowledge-Based Systems*, 96, 61-75.
- Spackman, K. A., 1989, Signal detection theory: valuable tools for evaluating inductive learning. Proceedings of the sixth international workshop on Machine learning. Ithaca, New York, USA, Morgan Kaufmann Publishers Inc.: 160-163.
- Stadler, W., 2013, Multicriteria Optimization in Engineering and in the Sciences, Springer Science & Business Media, p.
- Subasi, A. ve Ismail Gursoy, M., 2010, EEG signal classification using PCA, ICA, LDA and support vector machines, *Expert Systems with Applications*, 37 (12), 8659-8666.
- Subasi, A., 2013, Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders, *Computers in Biology and Medicine*, 43 (5), 576-586.
- Tharwat, A., Hassanien, A. E. ve Elnaghi, B. E., 2017, A BA-based algorithm for parameter optimization of Support Vector Machine, *Pattern Recognition Letters*, 93, 13-22.
- Tharwat, A., Gabel, T. ve Hassanien, A. E., 2018, Parameter Optimization of Support Vector Machine Using Dragonfly Algorithm, *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017*, Cham, 309-319.
- Tharwat, A. ve Hassanien, A. E., 2018, Chaotic antlion algorithm for parameter optimization of support vector machine, *Applied Intelligence*, 48 (3), 670-686.
- Tuba, E., Mrkela, L. ve Tuba, M., 2016, Support vector machine parameter tuning using firefly algorithm, p.
- Tuba, E., Ribic, I., Capor-Hrosik, R. ve Tuba, M., 2017, Support Vector Machine Optimized by Elephant Herding Algorithm for Erythemato-Squamous Diseases Detection, *Procedia Computer Science*, 122, 916-923.
- Vatsa, M., Singh, R. ve Noore, A., 1997, Improving biometric recognition accuracy and robustness using DWT and SVM watermarking, p.
- Vatsa, M., Singh, R. ve Noore, A., 2005, Improving biometric recognition accuracy and robustness using DWT and SVM watermarking, *IEICE Electronic Express*, 2, 362-367.
- Veri Görselleştirme Kataloğu, 2018, Kutu-Biyik Grafikleri, https://datavizcatalogue.com/TR/yontemleri/kutu_biyiki_grafikleri.html: [23.07.2018].
- Wang, G. G., Deb, S. ve Coelho, L. d. S., 2015, Elephant Herding Optimization, *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, 1-5.

- Wikimedia Commons contributors, 2016, File:23_salpchain_frierson_odfw_(8253212250).jpg,
[https://commons.wikimedia.org/w/index.php?title=File:23_salpchain_frierson_odfw_\(8253212250\).jpg&oldid=187809772](https://commons.wikimedia.org/w/index.php?title=File:23_salpchain_frierson_odfw_(8253212250).jpg&oldid=187809772); [06.05.2018].
- Wikipedia contributors, 2018, "Corvus," Wikipedia, The Free Encyclopedia, <https://en.wikipedia.org/w/index.php?title=Corvus&oldid=846865293>; [15.04.2018].
- Wolpert, D. ve Macready, W., 1997, Macready, W.G.: No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67-82, p.
- Wu, C.-H., Tzeng, G.-H. ve Lin, R.-H., 2009, A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression, *Expert Systems with Applications*, 36 (3, Part 1), 4725-4735.
- Xian, G.-m., 2010, An identification method of malignant and benign liver tumors from ultrasonography based on GLCM texture features and fuzzy SVM, *Expert Systems with Applications*, 37 (10), 6737-6741.
- Yang, X.-S., 2009, Firefly Algorithms for Multimodal Optimization, *Stochastic Algorithms: Foundations and Applications*, Berlin, Heidelberg, 169-178.
- Yang, X.-S. ve Deb, S., 2009, Cuckoo Search via Lévy flights, *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210-214.
- Yang, X.-S., 2010a, A New Metaheuristic Bat-Inspired Algorithm, In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Eds: González, J. R., Pelta, D. A., Cruz, C., Terrazas, G. ve Krasnogor, N., Berlin, Heidelberg: Springer Berlin Heidelberg, p. 65-74.
- Yang, X.-S., 2010b, Firefly Algorithm, Stochastic Test Functions and Design Optimisation, p.
- Yang, X.-S., 2010c, Nature-Inspired Metaheuristic Algorithms, p.
- Yang, X.-S. ve Wiley, I., 2010, Engineering optimization : an introduction with metaheuristic applications.
- Yang, X.-S., 2011, Metaheuristic Optimization, *Scholarpedia*, 6, 11472.
- Yang, X.-S., Karamanoglu, M. ve He, X., 2014, Flower pollination algorithm: A novel approach for multiobjective optimization, *Engineering Optimization*, 46 (9), 1222-1237.
- Yu, J. J. Q. ve Li, V. O. K., 2015, A social spider algorithm for global optimization, *Applied Soft Computing*, 30, 614-627.
- Zhang, X., Chen, X. ve He, Z., 2010, An ACO-based algorithm for parameter optimization of support vector machines, *Expert Systems with Applications*, 37 (9), 6618-6628.
- Zong Woo, G., Joong Hoon, K. ve Loganathan, G. V., 2001, A New Heuristic Optimization Algorithm: Harmony Search, *SIMULATION*, 76 (2), 60-68.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:	Burak Tezcan
Uyruğu	:	TÜRKİYE
Doğum Yeri ve Tarihi	:	Nevşehir / 1990
Telefon	:	(539) 512 5739
Faks	:	
e-mail	:	btezcan@selcuk.edu.tr

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Nevşehir Anadolu Lisesi, Merkez, Nevşehir	2008
Üniversite	: Pamukkale Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Merkez, Denizli	2014
Yüksek Lisans :	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Bilimleri Anabilim Dalı, Merkez, Konya	2018

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2016-..	Selçuk Üniversitesi Teknoloji Fak.	Araştırma Görevlisi

UZMANLIK ALANI

Makine Öğrenmesi, Optimizasyon

YABANCI DİLLER

İngilizce

YAYINLAR

Tezcan B., Golcuk A., Tasdemir S., Balci M., Analysis Of A Metaheuristic Optimization Algorithm For Data Classification, International Conference, ICENTE, Page 25, Konya, Turkey, December 07-09, 2017.

(Yüksek lisans tezinden yapılmıştır)

Tezcan B., Tasdemir S., Golcuk A., Balci M., Optimizing Support Vector Machine Parameters, 7th International Conference, ICAT'18, Page 60, Antalya, Turkey, April 28-May 1, 2018.

(Yüksek lisans tezinden yapılmıştır)