



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ROTASYONEL ORMAN İLE BİYOMEDİKAL ÖRÜNTÜ SINIFLAMA

Hasan KOYUNCU

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

+

Aralık-2013
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

ROTASYONEL ORMAN İLE BİYOMEDİKAL ÖRÜNTÜ SINIFLAMA

Hasan KOYUNCU

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Rahime CEYLAN

2013, 91 Sayfa

Jüri
Danışman: Yrd. Doç. Dr. Rahime CEYLAN
Doç. Dr. Yüksel ÖZBAY
Yrd. Doç. Dr. Gülay TEZEL

Örüntü sınıflandırma, biyomedikalde, tanı ve teşhis aşamalarında bilim insanlarına yardımcı olmaktadır. Literatürde bu amaçla birçok sınıflandırıcı sistem tasarımı gerçekleştirilmiştir. Bu metotlardan biri de yapay sinir ağları (YSA)' dır.

Ayrıca, çeşitli optimizasyon teknikleri, YSA yapısındaki ağırlık-bias değerlerinin ayarlanması ve performansın artırılması için güncelleme bölümüne entegre edilmektedir. Tez çalışmasının ilk aşamasında, YSA' daki ağırlık-bias değerlerinin güncellenmesi için, YSA güncelleme bölümü Parçacık Sürü Optimizasyonu (PSO) temelli oluşturulmuştur. Bu sayede sınıflandırma doğruluğu artırılmıştır. İkinci aşamada, içerisinde birden fazla hibrit PSO-YSA sınıflandırıcı birimi bulunduran Rotasyonel Orman (RO (hibrit PSO-YSA)) yapısının tasarımı gerçekleştirilmiştir. Böylece tek bir temel sınıflandırıcı (YSA) veya bu temel sınıflandırıcıdan daha iyi performansa sahip olan hibrit bir temel sınıflandırıcı (hibrit PSO-YSA) kullanmak yerine, RO (hibrit PSO-YSA) sınıflandırıcı topluluğunu kullanmanın daha uygun olacağı tespit edilmiştir. Bunun yanısıra hibrit yapı içerisindeki ağırlık ve bias değelerinin konum-hız sınırlamalarında, limit aralıklarının eşit alınmaması veya serbest bırakılmaması gerektiği, her bir durum için (ağırlık konum, ağırlık hız, bias konum ve bias hız limitleri) optimum sınırların tespit edilmesi gerektiği görülmüştür. Böylece, literatürdeki hibrit PSO-YSA yapılarından farklı optimize edilmiş bir ağ yapısı sunulmuştur. Tez çalışmasında sunulan yapılar (hibrit PSO-YSA ve RO (hibrit PSO-YSA)) göğüs kanseri verileri üzerinde test edilmiştir. Sonuçlar literatürde elde edilen sonuçlarla karşılaştırılarak değerlendirmeler sunulmuştur.

Anahtar Kelimeler: Örüntü Sınıflandırma, Parçacık Sürü Optimizasyonu, Rotasyonel Orman, Temel Bileşen Analizi, Yapay Sinir Ağları.

ABSTRACT

MS THESIS

BIOMEDICAL PATTERN CLASSIFICATION WITH ROTATION FOREST

Hasan KOYUNCU

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL AND ELECTRONICS ENGINEERING

Advisor: Asist. Prof. Dr. Rahime CEYLAN

2013, 91 Pages

Jury
Advisor: Asist. Prof. Dr. Rahime CEYLAN
Assoc. Prof. Dr. Yüksel ÖZBAY
Asist. Prof. Dr. Gülay TEZEL

Pattern classification helps to scientists on diagnosis and diagnostic in biomedical. For this purpose, there are lots of studies about classifier design in literature. One of these methods is the Neural Networks (NN). On the other hand, various optimization algoritms are integrated to update part of NN in order to tune the weight-bias values and in order to improve the performance of system. At the first part of thesis, update part of NN has been formed based on Particle Swarm Optimization (PSO) for adjusting the weight-bias values of NN and. Thus, classification accuracy of NN has been increased by PSO. At the second part of thesis, RF (hybrid PSO-NN), which contains multiple classification units (hybrid PSO-NN structures), has been designed. So it has been seen that usage of RF (hybrid PSO-NN) ensemble classifier is more convenient than the usage of NN and hybrid PSO-NN. Moreover, it has been seen that velocity and position boundaires of weight and bias in hybrid structure mustn't be chosen equal or mustn't be set free in space. But these boundaries (weight position, weight velocity, bias position and bias velocity boundaries) must be examined individually for finding their optimal intervals. So a new optimized network which is different from hybrid PSO-NN structures in literature has been presented. Shortly, it has been seen that interval limitations of hybrid structure must be adjusted individually in order to achieve higher classification performance. The structures presented in thesis study (hybrid PSO-NN and RF (hybrid PSO-NN)) have been tested on breast cancer dataset. The evaluation has been presented by comparing the results with others obtained in literature.

Keywords: Artificial Neural Networks, Particle Swarm Optimization, Pattern Classification, Principle Component Analysis, Rotation Forest.

ÖNSÖZ

Bütün çalışmalarımda değerli bilgi ve tecrübeleriyle bana yol gösteren, gerekli araştırma ve geliştirme çabalarımda yardımlarını esirgemeyen danışmanım Selçuk Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği öğretim üyesi Yrd. Doç. Dr. Rahime CEYLAN' a ve bölümümüzün değerli tüm öğretim elemanlarına, her türlü maddi manevi katkılarını esirgemeyen aileme ve öğrenci arkadaşlarıma teşekkür ederim.

Hasan KOYUNCU
KONYA-2013

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. MATERYAL VE YÖNTEM **8**

3.1. Topluluk Sınıflandırıcısı (Ensemble Classifier) 8
3.1.1. Topluluk sınıflandırıcısı tanımı ve çalışma alanları 8
3.1.2. Topluluk sınıflandırıcısı çeşitleri 10
3.1.3. Rotasyonel orman (*Rotation Forest*) topluluk sınıflandırıcısı 11
3.1.3.1. Özellik vektörünün K adet alt kümeye bölünmesinin amacı 12
3.1.3.2. Çoklu sınıflandırıcı sistem yapısı 13
3.1.3.3. Rotasyonel orman tekniğinde sınıflandırıcı kümesi 14
3.1.3.4. Rotasyonel orman tekniğinde algoritma 14
3.2. Optimizasyon 20
3.2.1. Optimizasyon nedir? 20
3.2.2. Optimizasyon çeşitleri 21
3.2.3. Optimizasyon ve sınıflandırma metotları ile kullanımı 22
3.2.4. Parçacık sürü optimizasyonu (PSO) 23
3.2.4.1. Parçacık sürü optimizasyonu algoritması 23
3.2.4.2. Parçacık sürü optimizasyonu parametreleri 26
3.3. Yapay Sinir Ağları (YSA) 27
3.3.1. Yapay sinir ağlarında temel öğrenme teknikleri 27
3.3.2. Yapay sinir ağlarında yapısal öğrenme kuralları 28
3.3.2.1. Perseptron yapısı 29
3.3.2.2. Çok katmanlı perseptron (ÇKP) 31

4. GERÇEKLEŞTİRİLEN HİBRİT SINIFLANDIRICI YAPILARI **35**

4.1. Hibrit PSO-YSA Yapısı 35
4.2. RO (Hibrit PSO-YSA) Yapısı 39
4.3. Elde Edilen Sonuçlar 41
4.3.1. Hibrit PSO-YSA yapısı ile elde edilen sonuçlar 43
4.3.2. RO (Hibrit PSO-YSA) yapısı ile elde edilen sonuçlar 56
4.3.3. Sınıflandırıcı yapıların karşılaştırılması 59

5. SONUÇLAR VE ÖNERİLER **62**

5.1 Sonuçlar 62

5.2 Öneriler 64

KAYNAKLAR 65

ÖZGEÇMİŞ.....78

SİMGELER VE KISALTMALAR

Simgeler

$pbest$: Parçacığın en iyi konum değeri

$gbest$: Sürüdeki en iyi konum değeri

X_i : PSO tekniği içerisindeki konum vektörü

V_i : PSO tekniği içerisindeki hız vektörü

C_1 : PSO tekniği içerisindeki birinci öğrenme faktörü

C_2 : PSO tekniği içerisindeki ikinci öğrenme faktörü

V_i^k : PSO tekniği içerisindeki k. parçacığın şimdiki hız değeri

V_i^{k+1} : PSO tekniği içerisindeki k. parçacığın yeni hız değeri

X_i^k : PSO tekniği içerisindeki k. parçacığın şimdiki konum değeri

X_i^{k+1} : PSO tekniği içerisindeki k. parçacığın yeni konum değeri

$f(x)$: Maliyet fonksiyonu

$Vmax$: Bir parçacıkta meydana gelebilecek maksimum hız

$Rand$: 0 ve 1 sayıları arasında rasgele bir sayı değeri

i : PSO tekniği içerisindeki parçacık numarası

k : PSO tekniği içerisindeki iterasyon sayısı

x : YSA sisteminin girişleri

w : YSA sistemindeki ağırlık değeri

n : YSA sistemi içerisindeki hücrelerin giriş sayısı

NET : YSA sistemi içerisindeki nöronların çıkış değerleri

s : Perseptron yapısı içerisinde işlemci elemana gelen net girdi

λ : YSA sistemi içerisindeki öğrenme faktörü

W_o : YSA yapısındaki eski ağırlık vektörü

W_n : YSA yapısındaki yeni ağırlık vektörü

E^p : YSA yapısındaki hata değeri

X : PSO sistemi içerisindeki popülasyon matrisi

o_j : j. nöronun çıkış verisi

d_j : j. nöronun istenen çıkış verisi

$\Delta^p W_{ji}$: YSA içerisinde hata fonksiyonunun gradyeni

η : YSA içerisinde en iyi öğrenme oranı

δ_j : j. düğümün hata terimi

α : Momentum katsayısı

$W_{ij}(t)$: i düğümü ile j düğümü arasındaki ağırlık değeri

$W_{ij}(t+1)$: i düğümü ile j düğümü arasındaki bir sonraki ağırlık değeri

$W_{ij}(t-1)$: i düğümü ile j düğümü arasındaki bir önceki ağırlık değeri

i_i : i düğümünün sonuç değeri

W_{jk} : j düğümü ile k düğümü arasındaki ağırlık değeri

p : YSA içerisindeki örüntü sayısı

o_j^p : j. düğümde p. örüntüye ait çıkış değeri

d_j^p : j. düğümde p. örüntüye ait istenen çıkış değeri

L : RF sistemindeki temel sınıflandırıcı

D : RF sistemindeki sınıflandırıcı kümesi

F : RF sistemindeki özellik dizisi

K : RF sistemindeki alt küme sayısı

n : RF sistemindeki özellik sayısı

M : RF sistemi içerisinde bulunan her bir alt küme içindeki veri sayısı

T : RF sisteminde uygulanan bootstrap örneği sayısı (iterasyon sayısı)

R_i : RF sistemi içerisindeki rotasyon matrisi

$\mathcal{L}$: RF sistemi içerisindeki eğitim dizisi

$\mathcal{W}$: RF sistemi içerisindeki temel öğrenme algoritması

C_t : RF sistemi içerisindeki t. Sınıflandırıcı

R_t^a : Temel rotasyon matrisi (rotasyon matrislerinin birleşimi)

$X_{t,k}$: RF sistemi içerisinde tanımlı alt matris

$D_{t,k}$: RF sistemi içerisinde sınıflandırıcı kümesinin alt matrisi

$C'(x)$: RF sistemi içerisindeki grup sınıflandırıcısı

$I(.)$: RF sistemi içerisindeki indikatör fonksiyonu

S : RF sistemi içerisindeki orijinal eğitim dizisi

$S_{i,j}$: RF sistemi içerisindeki orijinal eğitim dizisinin alt kümesi

$S_{i,j}'$: RF sistemi içerisinde $S_{i,j}$ matrisine bootstrap uygulanması ile elde edilen matris

$C_{i,j}$: RF sistemi içerisindeki sınıflandırıcı alt matrisi

R_i : RF sistemi içerisindeki rotasyon matrisi

ω : RF sistemi içerisindeki sınıf etiketi

μ_j : RF sistemi içerisindeki her bir ω_j sınıfı için uygunluk değeri

x : Ağırlık konum limit aralığı

v : Ağırlık hız limit aralığı

bx : Bias konum limit aralığı

bv : Bias hız limit aralığı

H : YSA yapısındaki gizli düğüm sayısı

Kısaltmalar

BBA	: Bağımsız Bileşen Analizi
ÇKP	: Çok Katmanlı Perseptron
DE	: Diferansiyel Evrim Tekniği
KA	: Karar Ağacı Metodu
MLP	: Multi Layer Perceptron
PSO	: Parçacık Sürü Optimizasyonu
RF	: Rotation Forest (Ensemble Classifier)
Rİ	: Rasgele İzdüşüm Tekniği
RO	: Rotasyonel Orman
TBA	: Temel Bileşen Analizi
YSA	: Yapay Sinir Ağları

1. GİRİŞ

Tanı ve teşhis aşamasında, biyomedikal verilerin işlenmesi çok önemlidir. Bu nedenle, veri işleme için literatürde pek çok sınıflandırma ve tanıma metodu önerilmektedir.

Yapay sinir ağları (YSA) yapısı, ağırlık ve bias güncellemesi işlemini esas alan bir temel sınıflandırıcı tekniğidir. Teknik içerisindeki ağırlıkların güncellenmesi işlemi birçok metotla yapılabilmektedir. Ancak, kullanılan metotların ağırlık güncellemesi işlemini gerçekleştirmesi esnasında, istenmeyen bir durum söz konusu olmaktadır. Bu durum, sistem hata oranının global minimum yerine, lokal minimum değerlere takılmasıdır. Bu durum, YSA' da ağırlık güncellenmesi prosesinin bir optimizasyon tekniği ile yapılandırılması sonucu elimine edilebilmektedir.

Başka bir deyişle, hatada global minimum değerlerin elde edilebilmesi için, ağırlık güncellemesi, doğrudan optimizasyon teknikleri ile gerçeklenir. Böylece hibrit bir yapı elde edilir. Tez çalışmasında Hibrit PSO-YSA yapısı bu amaçla gerçeklenmiştir.

Ayrıca tez çalışmasında, Rotasyonel Orman Yapısını Temel Alan Hibrit Parçacık Sürü ve Yapay Sinir Ağı yapısı sunulmuştur. Böylece hibrit bir temel sınıflandırıcı (Rotasyonel Orman yapısı içerisinde) kullanılarak, çıkışta elde edilecek sınıflandırma doğruluğunun artırılması amaçlanmıştır.

Tez çalışmasının temel amacı, örüntü sınıflandırma için performansı yüksek bir sınıflandırıcı sunmaktır. Tez konusunun bir diğer amacı ise, hibrit PSO-YSA yapısı içerisindeki ağırlık konum, ağırlık hız, bias konum ve bias hız limit aralıklarındaki değişimlerin sistem çıkışına olan etkisini incelemektir. Bu amaçla, ağırlık ve bias değerlerinin konum-hız limit aralıkları değiştirilerek, bu limit sınırlamalarının, literatürdeki gibi sürekli eşit olarak seçilmesinin doğru olup olmadığı incelenmiştir.

Tez konusunun önemi ise, hibrit temel sınıflandırıcılı RO (hibrit PSO-YSA) sınıflandırıcı topluluğunun literatürde ilk kez gerçeklenmiş olmasıdır. Çünkü YSA tabanlı hibrit temel sınıflandırıcılara sahip olan RO yapısı, literatatürde henüz gerçeklenmemiştir. Bu sayede, örüntü sınıflandırma sistemlerinde hem YSA sistemindeki ağırlık ve biasların eğitimi stokastik bir şekilde gerçeklenmiş hem de YSA sistemine verilecek olan verilerin iyileştirilmesi (çeşitlilik) sağlanmıştır. Ayrıca sunulan

sınıflandırıcılarla göğüs kanseri veri seti üzerinde çalışmalar gerçekleştirilmiştir. Sonuçlar sunulan yapıların performansını kanıtlamaktadır.

2. KAYNAK ARAŞTIRMASI

Parçacık sürü optimizasyonu Kennedy ve Eberhart (1995) tarafından geliştirilen sezgisel bir algoritmadır (Kennedy ve Eberhart, 1995). Parçacık sürü optimizasyonunun temel esin kaynağı kuş ve balık sürülerinin hareketleridir. Parçacık sürü optimizasyonunun kökleri, sosyal davranışların bir yansıması niteliğindedir. Bu yansıma, bilgisayar grafikleri ve sürü psikolojisinden esinlenmiştir (Kennedy ve Eberhart, 1995).

Parçacık sürü optimizasyonunun ilk alanı bilgisayar grafikleridir. Bu konuda Reeves (1983) parçacık sistemleri üzerine bir çalışma yapmıştır. Bu parçacık sistemi, dinamik olan ve poligonlarla, yüzeylerle kolay bir şekilde ifade edilemeyen nesneleri modelleme işlemini gerçekleştirmektedir. Kolay bir şekilde ifade edilemeyen bu objelere ateş, duman, bulut ve su örnek olarak verilebilir. Bu sistemde söz konusu parçacıklar birbirlerinden bağımsız olarak hareket etmektedir. Aynı zamanda bu parçacıklar, bir dizi kurala göre konum değiştirmektedir. Birkaç yıl sonra Reynolds (1987), kuş sürülerinin kolektif davranışlarını bu parçacık sistemine adapte etmiştir.

Benzer bir çalışma da Heppner ve Grenader (1990) tarafından yapılmıştır. Çalışma içerisinde tasarlanan kuşların tüneyebileceği bir alan tanımlanmıştır. Bahsi geçen alan, ileride parçacık sürü optimizasyonu içerisinde kullanılacak olan kurallar dizisine göre oluşturulmuştur.

Nowak ve ark. (1990) yapmış oldukları çalışmada sosyal psikolojik bir araştırma konusu seçmişlerdir. Sürüdeki dinamik teorinin sosyal etkisini incelemişlerdir. Bu çalışma parçacık sürü optimizasyonunun ortaya çıkışındaki temel esin kaynaklarından biridir.

Problem arama uzayında parçacıkların hareket etmesini sağlayan kurallar bütünü, Kennedy ve Eberhart (1995)'ın çalışmalarında belirttikleri gibi, herhangi bir durumda bir insanın bireysel davranışı olarak görülebilir. Bu durumda insan, davranışlarını ayarlayacak ve akranlarına kendini kabul ettirmek için harekete geçecektir. Kennedy ve Eberhart (1995) çalışmalarında tam da bu noktaya değinmişlerdir. Böylece parçacık sürü optimizasyonu tekniğinin tasarımı gerçekleştirilmiştir.

Rotasyonel Orman tekniği Kuncheva ve Rodriguez (2006) tarafından üretilmiş, bir çeşit kümeleme sınıflandırıcısı algoritmasıdır. Önerildiği tarihten itibaren günümüze

RO tekniği ile ilgili önemli çalışmalar yapılmıştır. Bu çalışmalardan bazıları şu şekildedir.

Kuncheva ve Rodriguez (2006), yaptıkları çalışmada göğüs kanseri, iris gibi birçok veriyi kullanarak, Rotasyonel Orman, Bagging, Boosting ve Random Forest tekniklerini performans bakımından karşılaştırmışlardır. Yaptıkları çalışma neticesinde, Rotasyonel Orman kümelerin ürettiği bireysel sınıflandırıcıların, Adaboost ve Random Forest tekniklerinin oluşturduğu bireysel sınıflandırıcılardan daha yüksek doğruluğa sahip olduğu tespit edilmiştir. Aynı zamanda, RO tekniği tarafından oluşturulan bu sınıflandırıcıların Bagging yöntemi içerisindeki sınıflandırıcılardan daha çeşitli ve farklı olduğu ispatlanmıştır. Göğüs kanseri verileri üzerinde de RO tekniğinin, diğer tekniklere olan üstünlüğü açıkça görülmektedir.

Kuncheva ve Rodriguez (2007), yaptıkları bir diğer çalışmada Rotasyonel Orman tekniği ile, Bagging, Adaboost ve Random Forest tekniklerinin performans bakımından karşılaştırmalarını yapmışlardır. Sonuç olarak, RO tekniği, diğer tekniklerden daha üstün bir performans sergilemiştir. Aynı zamanda RO tekniği için kullanılan "Temel Bileşen Analizi" yöntemi ile "Parametrik Olmayan Diskriminant Analizi" ve "Rasgele İzdüşüm" teknikleri karşılaştırılmıştır. Bu çalışmada, RO yapılarında en başarılı özellik azaltma yönteminin "Temel Bileşen Analizi" olduğu doğrulanmıştır.

Liu ve Huang (2008) yapmış oldukları çalışmada göğüs kanseri ve prostat kanseri üzerine bir araştırma gerçekleştirmişlerdir. Bu çalışmada Rotasyonel Orman tekniği ile Bagging, Boosting gibi teknikler karşılaştırılmıştır. Aynı zamanda RO tekniği içerisinde özellik azaltma yöntemi olarak "Bağımsız Bileşen Analizi (BBA)" ve "Rasgele İzdüşüm Tekniği (Rİ)" ile "Temel Bileşen Analizi (TBA)" metotları performans bakımından karşılaştırılmıştır. Sonuç olarak, BBA tekniğinin, TBA ve Rİ tekniklerinden daha iyi sonuçlar verdiği görülmüş, RO tekniğinin diğer metotlara olan üstünlüğü gösterilmiştir.

Kotsiantis (2010), yapmış olduğu çalışmada Rotasyonel Orman, Bagging, Boosting, Random Subspace gibi tekniklerin performans karşılaştırmasını yapmıştır. Göğüs kanseri verileri üzerinde RO tekniğinin diğer tekniklere olan üstünlüğü açıkça görülmektedir.

Özçift ve Gülten (2011) yapmış oldukları çalışmada, Bilgisayar Destekli Teşhis Sistemlerinin (Computer-aided diagnosis system) tasarımı için Rotasyonel Orman yapısını önermişlerdir.

Novakovic ve Veljovic (2011) yapmış oldukları çalışmada, göğüs biyopsi sonuçlarının tahmini için Rotasyonel Orman tekniğini kullanmışlardır. Temel sınıflandırıcı olarak J48, Desicion Stump, Random Tree gibi sınıflandırıcı tekniklerini kullanarak, değişik sınıflandırıcılar için RO tekniğinde meydana gelen değişimleri incelemişlerdir. Çalışma içerisinde kullanılan bu temel sınıflandırıcı tekniklerinin her biri, karar ağacı sınıflandırıcısının değişik alt türleridir. Rotasyonel Orman algoritması için en iyi sınıflandırıcının J48, graftJ48 ve SimpleChart olduğu görülmüştür.

Bock ve Poel (2011) yaptıkları çalışmada, RotBoost ve Rotasyonel Orman tekniklerinin performans bakımından karşılaştırmasını yapmışlardır. Aynı zamanda bu teknikler için kullanılan özellik azaltma yöntemlerinin, tekniklerin çıkışlarına olan etkisi gözlemlenmiştir. Kullanılan özellik azaltma yöntemleri TBA, BBA ve Rİ teknikleridir. Çalışma içerisinde genel olarak, RotBoost tekniğinin RO tekniğinden daha üstün olduğu dile getirilmiştir.

Marques ve ark. (2012) yapmış oldukları çalışmada, kredi risk değerlendirmesi için Bagging, Boosting, Rotasyonel Orman ve Random Subspace metotlarını kombine ikili çiftler halinde birleştirmişlerdir. Bunun neticesinde, en iyi sonuçları veren ikili Bagging ve Rotasyonel Orman tekniklerinin birleşimi ile elde edilmiştir.

Han ve ark. (2012) tarafından yapılan çalışmada Rotasyonel Orman ve Decorate tekniğini, YSA tekniğini kullanarak birleştirmiştir. Temel sınıflandırıcı olarak yapay sinir ağları tekniği kullanılmıştır. Göğüs kanseri gibi birkaç veri üzerinde yapılan çalışmalar neticesinde Rot-Decorate tekniğinin, tek başına RO ve Decorate tekniklerinden üstün olduğu ispatlanmıştır. Bu sonuç göğüs kanseri verilerinin karşılaştırılmasında da açıkça görülmektedir. Bütün incelenen yapılarda, temel sınıflandırıcı olarak yapay sinir ağları kullanılmıştır.

Xu ve ark. (2007), yapmış oldukları çalışma içerisinde Diferansiyel Evrim algoritması, PSO algoritması ve Diferansiyel Evrim-PSO hibrit algoritmasını YSA sistemini eğitmede kullanmışlardır. Çalışmada üzerinde araştırma yapılan veriler gen düzenleyici ağlar şeklinde olup, en iyi performans sonuçlarını Diferansiyel Evrim-PSO hibrit algoritmasının bulunduğu YSA sistemi vermiştir.

Melgani ve Bazi (2008), yaptıkları çalışmada EKG aritmi verileri üzerinde çalışma yapmışlardır. Çalışma içerisinde k-ortalama komşu sınıflandırıcı, destek vektör makinesi ve radyal tabanlı YSA teknikleri performans bakımından karşılaştırılmıştır. Destek vektör makinesi sistemine PSO tekniği eklenmiştir. Bu suretle destek vektör

makinesinin doğruluk derecesinin artırılması amaçlanmıştır. Sonuç olarak DVM-PSO sisteminin diğer sistemlerden daha üstün olduğunu kanıtlamıştır.

Neto ve ark. (2010), yapmış oldukları çalışmada EKG sinyali ile aritmi tespiti yapmışlardır. Bu amaçla, dalgacık katsayılarını YSA yapısına giriş olarak vermiş, YSA içerisindeki ağırlıkların eğitimini ise PSO tekniği ile gerçekleştirmişlerdir. Doğruluk oranı olarak % 97.03 başarı elde etmişlerdir.

Fei (2010) yapmış olduğu çalışmada kalp ritim bozukluğu (arrhythmia cordis) teşhisi üzerinde bir araştırma gerçekleştirmiştir. Tasarlanan sistemde destek vektör makinesi ve PSO hibrit sistemi teşhis işlemini yerine getirmektedir. Çalışma neticesinde bu hibrit sistemin YSA yapısından daha iyi sonuçlar ortaya koyduğu ispatlanmıştır. Söz konusu karma sistem içerisinde PSO, destek vektör makinesi sistemindeki parametrelerin kontrolünü yapmaktadır. Tasarlanan sistem sonuçları, literatürdeki radyal tabanlı YSA ve geri yayılımlı YSA tekniklerinden elde edilen sonuçlara üstünlük sağlamaktadır.

Korürek ve Doğan (2010) yapmış oldukları çalışmada erken ventriküler kasılma (Premature Ventricular Contraction), normal ve ventriküler atışın birleşimi, atriyal erken atım (Atrial Premature Beat), sağ dal bloğu hızlı atımı, hızlı ve normal atım verileri üzerinde araştırmalar yapmışlardır. Bu amaçla kohonen YSA, k-ortalama YSA, PSO-YSA ve k-YSA tekniklerinin çalışma performansları karşılaştırılmıştır. Sonuç olarak PSO-YSA metodunun hassasiyet bakımından diğer tekniklerden üstün olduğu açıkça görülmektedir.

Yadav ve Mandal (2011) yaptıkları çalışmada konuşmacıyı tanıyan bir sistem tasarlamayı amaçlamışlardır. Bu amaçla PSO-YSA hibrit sistemini tasarlayarak, YSA sisteminin ağırlık atamalarını PSO tekniğinden faydalanarak gerçekleştirmişlerdir. Tasarlanan PSO-YSA hibrit sisteminin, YSA sistemine olan üstünlüğünü göstermişlerdir.

Bansal ve ark. (2011), yapmış oldukları çalışmada PSO tekniği içerisinde bulunan atalet ağırlığının hesaplanması üzerine bir çalışma yapmışlardır. Çalışma içerisinde değişik atalet ağırlığı formülleri, değişik maliyet fonksiyonları ile çalıştırılarak sistem çıkışı incelenmiştir. 15 adet farklı atalet formülü ve 5 adet maliyet fonksiyonu ile çalışma gerçekleştirilmiştir.

Shi ve ark. (2011) yapmış oldukları çalışmada çeşitli PSO tekniklerinin performans bakımından karşılaştırılmasını gerçekleştirmişlerdir. Bu teknikler; Hücresel PSO, hibrid hücresel otomat ve PSO şeklindedir. Aynı zamanda, bu tekniklerin

çalışmalarını incelemek için birçok maliyet fonksiyonu ile deneme yapmışlardır. Çalışma sonucunda popülasyon sayısı ile verimliliğin ve doğruluğun ters orantılı olduğunu tespit etmişlerdir. Tasarlanan sistemlerden hücresel PSO çeşidi olan PSO-outer tekniği genel olarak diğer tekniklerden daha iyi sonuç vermiştir.

Dou ve Gao (2012), yapmış oldukları çalışmada PSO tekniği ile karar ağaçları, YSA, destek vektör makinesi ve bayes sınıflandırıcılarını eğitmiştir. Çalışma içerisinde migren, uykusuzluk, hepatit gibi hastalıkların teşhisi gerçekleştirilmiştir. Sistemde genel olarak en iyi sonucu, önerilen Genişletme Sınıflandırıcısı (Extension Classifier) tekniği vermiştir.

Mandal ve Sairam (2012) yapmış oldukları çalışmada lenfografi ve bel ağrısı verileri üzerine bir araştırma gerçekleştirmişlerdir. Yapılan çalışma içerisinde Bayes sınıflandırıcısı, YSA sınıflandırıcısı, destek vektör makinesi gibi sınıflandırıcılar bulunmaktadır. Sistem içerisinde Rotasyonel Orman tekniği kullanılarak söz konusu sınıflandırıcıların performansları artırılmaya çalışılmıştır. Önerilen yapıda kapa hatası, karesel ortalama hata ve ROC kriterlerine göre hata oranları ölçülmüş, RO tekniğinin performansı olumlu yönde etkilediği tespit edilmiştir. Sistem içerisinde destek vektör makinesi-PSO hibrit sistemi bulunmaktadır. Rotasyonel Orman tekniği topluluk sınıflandırıcısı olarak katıldığı sistem içerisinde, hibrit sistemi yapılandırarak performans oranını artırmıştır.

Tez çalışmasında önerilen sınıflandırıcı, Hibrit Temel Sınıflandırıcılı Rotasyonel Orman yapısıdır. Kaynak araştırmasından da anlaşılacağı üzere, RO tekniği, daha önce hibrit YSA tabanlı bir temel sınıflandırıcı ile gerçeklenmemiştir. Bu yönüyle tez çalışmasında sunulan sonuçlar bir ilk niteliği taşımaktadır.

3. MATERYAL VE YÖNTEM

3.1. Topluluk Sınıflandırıcısı (Ensemble Classifier)

Bir örüntü tanıma sistemi temelde üç kısımdan oluşmaktadır. Bu kısımlar şu şekildedir:

- Söz konusu verileri elde etme
- Ön işleme (Veriyi temsil edecek olan özniteliklerin elde edilmesi)
- Sınıflandırma

Topluluk sınıflandırıcıları; isminden de anlaşılacağı üzere, birden fazla sınıflandırıcı tekniğini içerisinde barındıran topluluk olarak tanımlanabilir. Belirtilen örüntü tanıma sisteminde sınıflandırma işleminin, yüksek performansla gerçekleştirilmesi için önerilen yaklaşımlardan biri de topluluk sınıflandırıcısıdır. Örüntü sınıflandırmada topluluk sınıflandırıcılarının kullanılması, tek bir sınıflandırıcı tekniğinin kullanılmasından daha yüksek çeşitlilik ve kararlılık sağlamaktadır.

3.1.1. Topluluk sınıflandırıcısı tanımı ve çalışma alanları

Topluluk sınıflandırıcılarının temel mantığı, bireysel olarak çalışan temel sınıflandırıcıların bir araya getirilmesidir. Bu sayede elde edilecek olan sistem, *Topluluk Sınıflandırıcısı* olarak adlandırılır. Bu sınıflandırıcı tekniği, bireysel olarak çalışan temel bir sınıflandırıcıya üstün gelecektir. Çünkü söz konusu topluluk sınıflandırıcı tekniği, içerisinde birden fazla temel sınıflandırıcı (karar ağacı, yapay sinir ağı, destek vektör makinesi v.b. gibi) bulundurmaktadır.

Topluluk sınıflandırıcıları, kesin karara ulaşmadan önce birçok fikri kombine eder ve elde edilecek ortak karara göre çalışır. Bu yüzden topluluk sınıflandırıcı tekniği, çoklu sınıflandırıcı tekniklerine (*Multiple Classifier System*) örnek olarak verilebilir.

Topluluk sınıflandırıcı metodu ile ilgili birçok çalışma mevcuttur. Bu çalışma alanlarından bazıları aşağıdaki gibidir:

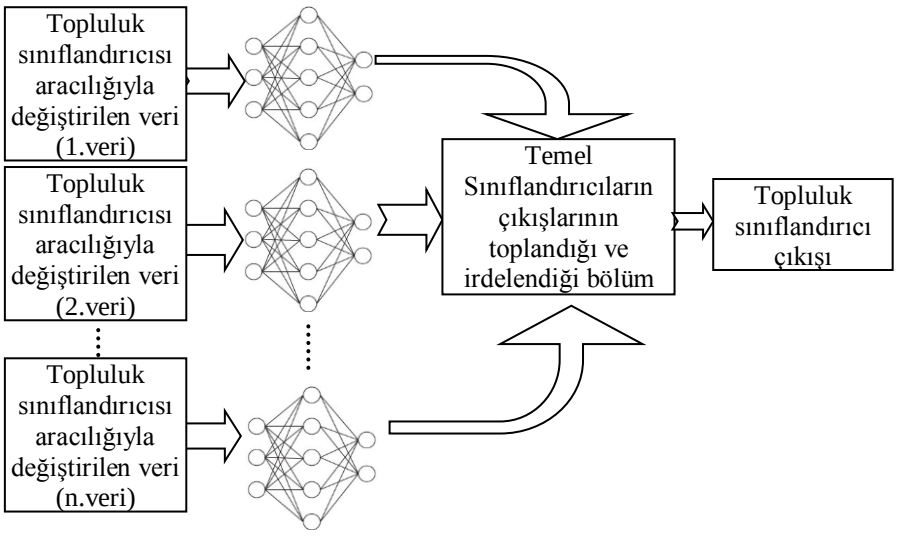
- Biyoloji bilgisi tabanlı çalışmalar
- Tıp

- Kimya bilgisi tabanlı çalışmalar
- İmalat
- Coğrafya
- Görüntü işleme

Topluluk sınıflandırıcılarının çalışma prensibine, bir patolojik veri üzerinde birden fazla doktorun teşhis koyması örnek olarak verilebilir. Söz konusu patolojik verinin yalnızca bir doktor tarafından teşhis edilmesi düşük doğruluğa sahip bir işlem iken, söz konusu verinin birden fazla doktor tarafından teşhis edilmesi yüksek doğruluğu sağlayacaktır. Ancak bu teori, pratikte her zaman optimum şekilde sonuç vermemekte, söz konusu topluluk sınıflandırıcısı için optimum parametre değerlerinin tespit edilmesi gerekmektedir. Topluluk sınıflandırıcılarında genel sınıflandırma işlemi Şekil 3.1' deki gibidir.

Topluluk sınıflandırıcıları temel olarak 4 gruba ayrılmaktadır:

- Bagging
- Boosting
- Rasgele Orman (*Random Forest*)
- Rotasyonel Orman (*Rotation Forest*) şeklindedir.



Şekil 3.1. n adet temel sınıflandırıcılı Topluluk Sınıflandırıcılarında genel sınıflandırma işlemi.

Şekil 3.1' de topluluk sınıflandırıcı yapısı için genel bir gösterim verilmiştir. Şekil 3.1'de görüldüğü gibi, topluluk sınıflandırıcı sistemleri öncelikle veri üzerinde işlem yapmaktadırlar. Her topluluk sınıflandırıcısının kendine has veri kümeleme

sistemleri bulunmaktadır. Bir topluluk sınıflandırıcısı rasgelelik kuramına göre veriyi atarken, bir başka topluluk sınıflandırıcısı belirli bir düzen ihtiva edecek şekilde verileri sıralayabilmektedir. Sıralanan bu veriler YSA, Karar Ağacı, Destek Vektör Makinesi gibi sınıflandırıcıların girişine sunulacaktır. Bu sayede Topluluk Sınıflandırıcı sistem içerisinde, değiştirilmiş (iyileştirilmiş) olan birden fazla veri ile birçok temel sınıflandırıcı çalıştırılarak hem sistem performansı hem de sistem çıkışında çeşitliliğin sağlanması amaçlanır.

3.1.2. Topluluk sınıflandırıcısı çeşitleri

Farklı sonuçlara sahip topluluk sınıflanıdırıcılarının üretiminde, homojen (aynı tür temel sınıflandırıcılardan meydana gelmiş) ve heterojen (farklı temel sınıflandırıcılardan meydana gelmiş) sınıflandırıcı toplulukları kullanılmaktadır.

Homojen toplulukların üretilmesi iki şekilde olmaktadır. Bunlardan birincisi, aynı temel sınıflandırıcının farklı parametrelerle, aynı eğitim kümesi üzerinde işlem yapması ile elde edilir. İkinci yöntem ise aynı temel sınıflandırıcının farklı eğitim kümeleri üzerinde işlem yapması ile elde edilir. Literatürde birbirinden farklı eğitim kümeleri üretebilmek amacıyla çeşitli teknikler ileri sürülmüştür. Bu yöntemler temel olarak Bagging, Boosting, Rasgele Orman (*Random Forest*) ve Rotasyonel Orman (*Rotation Forest*) şeklinde sıralanır.

Bagging: Breiman tarafından gerçekleştirilmiştir. Bagging'de N adet örnekten oluşan eğitim setinden yine N örnekli bir eğitim seti, yerine koymalı rastgele seçimle üretilir. Bu durumda, bazı eğitim örnekleri yeni eğitim kümesinde bulunmazken, bazıları birden fazla kez yer alırlar. Topluluktaki her bir temel sınıflandırıcı, bu şekilde üretilmiş birbirinden farklı örnekler içeren eğitim kümeleriyle eğitilirler ve sonuçları çoğunluk oylaması ile gerçekleştirilir (Amasyalı ve Ersoy, 2011).

Boosting: Bu algoritmada ise her bir temel sınıflandırıcı, önceki temel sınıflandırıcının doğru sınıflandıramadığı örnekler ile eğitilmektedir. Temel sınıflandırıcıların kararları kendi eğitim kümeleri üzerindeki başarılarıyla ağırlıklandırılarak gerçeklenmektedir (Amasyalı ve Ersoy, 2011).

Rasgele Orman (Random Forest): Breiman tarafından gerçeklenmiştir. Temel sınıflandırıcıları karar ağaçlarıdır. Bu yöntemde temel sınıflandırıcılar, yine Bagging ile üretilmiş eğitim örnekleriyle eğitilirler. Ancak temel sınıflandırıcıların (karar ağaçlarının) her bir düğümünde veriyi bölerken, tüm özelliklerin incelenmesi yerine özelliklerin rastgele bir alt kümesi incelenir. Bu sayede hem karar ağacının üretim süresi azalmakta hem de ağaçların kararları arasındaki farklar yeni bir rastgelelikle arttırılmaktadır. Temel öğrenicilerin sonuçları yine çoğunluk oylaması ile gerçeklenmektedir (Amasyalı ve Ersoy, 2011).

Rotasyonel Orman (Rotation Forest): Rotasyonel Orman (RO) tekniği, doğru ve çeşitli sınıflandırıcılar üretmeyi amaçlar. RO tekniği Bagging yönteminde olduğu gibi, bireysel sınıflandırıcılar için eğitim seti olarak bootstrap (parametrik olmayan istatistiksel bir yaklaşım, örneklerden örnekler tahmin etmeyi sağlayan sistem) örneklerini kullanır. Kümedeki her bir sınıflandırıcı için dolu bir özellik dizisinin, ardışıl olarak oluşturulması ve özellik azaltma tekniğinin uygulanması temel amaçtır. Bu işlemi gerçekleştirmek için özellik dizisi (F), rasgele bir şekilde K adet alt kümeye ayrılır. Her bir alt küme üzerine TBA metodu ayrı ayrı uygulanır. Bu işlemin ardından n adet doğrusal olarak azaltılmış özelliklerin yeni dizisi, bütün temel bileşenlerin bir havuzda toplanması ile elde edilir. Söz konusu veri, doğrusal bir şekilde yeni özellik uzayına taşınır. D_i sınıflandırıcısı bu veri seti ile eğitilir. Özellik dizisinin farklı bölümleri, farklı özellik azaltımına sebep olacaktır. Böylece bootstrap tekniği ile çeşitlilik artırılmış olur (Kuncheva ve Rodriguez, 2007).

3.1.3. Rotasyonel orman (*Rotation Forest*) topluluk sınıflandırıcısı

RO tekniği, doğru ve çeşitli sınıflandırıcılar üretmeyi amaçlayan bir küme üretim tekniğidir. RO sistemindeki temel prensip, küme içerisindeki her bir sınıflandırıcıya uygulanacak olan yeni bir özellik dizisinin üretilmesi işlemini gerçekleştirmek amacıyla, özelliklerin alt kümelerine TBA metodunun uygulanması ve özelliklerin farklı düzlemlerde ifade edilmesinin sağlanmasıdır. RO kümeleri, Bagging tekniğinin ürettiği temel sınıflandırıcılardan daha çeşitli, Adaboost ve Random Forest tekniklerinin ürettikleri temel sınıflandırıcılardan daha yüksek doğruluğa sahip temel sınıflandırıcılar üretme eğilimindedir. RO tekniği içerisinde Karar Ağaçları, özellik eksenlerinin dönüşüne olan hassasiyetleri sebebiyle temel sınıflandırıcı olarak seçilirler.

Bu özelliğinin yanı sıra, bu hassas işlemi gerçekleştirirken doğruluk oranları da yüksek seviyede kalmaktadır. RO tekniği özellik çıkarma işleminde Temel Bileşen Analizi metodunu kullanmaktadır (Rokach, 2010).

RO tekniği, her bir ağaç için azaltılmış farklı özellik seti kullanarak, bağımsız bir şekilde *L* adet karar ağacını eğiten bir topluluk metodudur. *X*, *n* adet özellikli, *N x n* boyutlu eğitim örneklerini içeren bir matris olsun ve $X= [x_1, x_2, \ldots, x_n]^T$ şeklinde ifade edilsin. Bütün eğitim örneklerinin doğru sınıf etiketlerinin gerçeklenmiş olduğu varsayılsın. *D* ise, *L* sınıflandırıcılarının kümesi olarak tayin edilsin ve $D=\{D_1,D_2,\ldots,D_L\}$ ile gösterilsin. *F* ise özellik vektörü olsun. Bu şekilde RO sisteminin, temel karar organlarının yapısal iskeleti gerçekleştirilmiş olur (Kuncheva ve Rodriguez, 2007).

RO algoritmasında özellik dizisi (*F*), rasgele bir şekilde *K* adet alt kümeye ayrılır. Her bir alt küme üzerine TBA metodu ayrı ayrı uygulanır. Bu işlemin ardından *n* adet, doğrusal olarak azaltılmış özelliklerin yeni dizisi, bütün temel bileşenlerin bir havuzda toplanması ile elde edilir. Söz konusu veri, doğrusal bir şekilde yeni özellik uzayına taşınır. D_i sınıflandırıcısı bu veri seti ile eğitilir. Özellik dizisinin farklı bölümleri, farklı özellik azaltımına sebep olacaktır. Böylece *bootstrap* tekniği ile çeşitlilik artırılmış olur (Kuncheva ve Rodriguez, 2007).

3.1.3.1. Özellik vektörünün K adet alt kümeye bölünmesinin amacı

Özellik vektörü *X*' in *K* adet alt kümeye bölünmesi işleminin temel amacı çeşitliliği sağlamaktır. Bu durum sonucunda yeni çıkarılmış özelliklerin her biri orijinal özelliklerin (*M*) doğrusal bir kombinasyonudur. *M*, herbir altkümedeki özellik sayısı olup, (3.1) eşitliği ile gösterilir. Buradaki *n*, özellik sayısını ifade etmektedir (Kuncheva ve Rodriguez, 2007).

$$M= [n / K] \tag{3.1}$$

Bu aşamanın ardından *R* ile ifade edilen rotasyon matrisi, orijinal eğitim dizisinden alınan *T* adet bootstrap örneği ile çarpılarak, yeni çıkarılmış ayrık bir eğitim dizisi olan *T'* matrisi elde edilir. Bu yeni çıkarılmış eğitim dizisi (*T'*), (3.2) eşitliği ile ifade edilir (Kuncheva ve Rodriguez, 2007).

$$T' = T.R \quad (3.2)$$

Şimdi ayrık bir rotasyon matrisini ayrık olmayan bir rotasyon matrisi ile karşılaştıralım. Bu durumda iki adet yaklaşımda bulunmamız gerekir.

Bu yaklaşımlardan birincisi "Rasgele İzdüşümler" olarak isimlendirilir. *L* adet rasgele (bozulmaya uğramamış), $R_1, R_2, ..., R_L$ ile ifade edilen *nxn* boyutlu dönüşüm matrisleri üretilir. Bu matris elemanları, standart dağılımdan örneklenerek elde edilirler *(~ N(0,1))*. R_i rotasyon matrisinin tersine çevrilebildiği ve orijinal uzay eski haline getirilebildiği gibi, dönüşüm içerisindeki bilgide herhangi bir kayıp söz konusu olmayacaktır. Ancak bu durumda, böyle bir dönüşüm, ayrımcı bilgiyi saptırabilir veya bu bilgiyi abartabilir. Diğer bir deyişle, bir rotasyon, her iki durumda da çok değişik boyutlarda karar ağaçlarının oluşmasına neden olarak, problemin sadeleşmesini veya daha karmaşık hale gelmesini sağlayabilir (Kuncheva ve Rodriguez, 2007).

"Ayrık Rasgele İzdüşüm" olarak isimlendirilen ikinci yaklaşım içerisindeki ayrık dönüşüm matrisleri, RO tekniği içerisindeki matrise benzetilerek üretilir. Bu matrislerin sıfır olmayan değerleri, standart bir dağılıma uymak kaydıyla rasgele olarak yeniden örneklenir. *L* matrisleri ise kümeyi oluşturmak amacıyla üretilir (Kuncheva ve Rodriguez, 2007).

Kuncheva ve Rodriguez (2007) yaptıkları çalışmalarında budanmış veya budanmamış ağaçların her birisi üzerinde, Ayrık Rasgele İzdüşümler tekniğinin Rasgele İzdüşümler tekniğinden daha iyi sonuçlar verdiğini ispatlamışlardır.

3.1.3.2. Çoklu sınıflandırıcı sistem yapısı

Çoklu sınıflandırıcı sistem (*Multiple Classifier System*), örüntü tanıma ve makine öğrenimi alanlarında etkin bir araştırma alanıdır. Etkili bir çoklu sınıflandırıcı sistem, doğruluğu yüksek ve çeşitlilik sağlayan sınıflandırıcılardan meydana gelir. Güçlü bir çoklu sınıflandırıcı sistem içerisinde çalışan temel sınıflandırıcılar, yüksek sınıflandırma doğruluğuna sahip olmalıdır. Bunun yanı sıra bu sistem, tesadüfî hataları önleyici şekilde tasarlanmalıdır. Bu nedenle, bir temel sınıflandırıcı tarafından yanlış sınıflandırılan bir örnek, diğer sınıflandırıcılar tarafından doğru sınıflandırılarak yüksek sınıflandırma doğruluğu elde edilir. Ayrıca sistem çıkışında birleştirilen veriler, birçok sınıflandırıcının kullanılması sayesinde, sistem içerisinde en iyi çözümleri üreten bireysel sınıflandırıcının veri hakkında vereceği karardan daha kesin ve doğru bir karara

sahip olacaktır. Genelde, çoklu sınıflandırıcı sistem içerisindeki her bir sınıflandırıcı için doğruluk ve çeşitlilik ters orantılı olarak değişir. Kısaca, yüksek doğruluğa sahip olan temel bir sınıflandırıcı, düşük çeşitliliğe sahip olacaktır. Çoklu sınıflandırıcı sistemler içerisindeki bu dezavantajı bertaraf etmek amacıyla birçok metot geliştirilmiştir. İşte bu metotlardan bir tanesi de RO tekniğidir (Liu ve Huang, 2008).

3.1.3.3. Rotasyonel orman tekniğinde sınıflandırıcı kümesi

Sınıflandırıcı toplulukları genellikle basit ve temel bir sınıflandırıcıdan daha yüksek doğruluklu sonuçlar elde etmektedir. Makine öğrenimi literatüründe, bagging ve boosting gibi çeşitli topluluk sınıflandırıcı modelleri mevcuttur. Bagging yöntemi içerisinde, sınıflandırıcılar birbirlerinden bağımsız olarak tasarlanır. Bu sayede çeşitlilik elde edilir (Özçift ve Gülten, 2011). Ayrıca bu işlem esnasında rasgelelik kuramı kullanılır. Bagging metodunda çeşitlilik, "Rasgele Orman (*Random Forest*)" grup modeline benzer şekilde, yüksek rasgelelik kuramına göre gerçekleştirilir. Bu grup modeli içerisinde karar ağaçları sınıflandırıcısı kullanılmaktadır. Karar ağaçları, veri setine bootstrap işlemi uygulanarak elde edilen verileri sınıflandırır. Ayrıca bu grup modeli, ağaçların üretimi esnasında her bir düğümdeki özellik seçimini rasgele yaparak, çeşitliliği artırmaktadır. RO tekniği, rasgele orman yöntemine benzer şekilde, birbirinden bağımsız karar ağaçları üzerine inşa edilmiştir. Ancak RO tekniğindeki her bir ağacın eğitim işlemi, döndürülen özellik uzayında bütün bir eğitim dizisi ile gerçekleştirilir. RO sınıflandırıcıları üretilirken, özellik eksenlerine paralel alt düzlemleri kullanılır. Kısaca, özellik eksenindeki ufak bir değişim, üretilen ağaçların çeşitliliğini sağlayacaktır (Özçift ve Gülten, 2011).

3.1.3.4. Rotasyonel orman tekniğinde algoritma

Zhang ve Zhang (2008) yaptıkları çalışmada aşağıdaki RO algoritmasını önermişlerdir.

Giriş Bölümü

- $\mathcal{L}$ bir eğitim dizisidir. $\mathcal{L}$, (3.3) eşitliğinde ifade edildiği gibidir.

$$\mathcal{L} = \{(x_i, y_i)\} = [X\ Y] \tag{3.3}$$

Bu eşitlikteki *X*, *Nxp* boyutlu bir matristir. Bu matris giriş özellik değerlerini tutmaktadır. *Y* (*Hedef)* ise *N* boyutlu bir sütun vektörü olup, sınıf etiketlerini içermektedir.

- *K*: özellik alt kümelerinin sayısı (veya *M*: her bir alt küme içerisine yerleştirilmiş olan özelliklerin sayısıdır).
- $\mathcal{W}$: temel bir öğrenme algoritması
- *T* : iterasyon sayısı
- *x*: sınıflandırılacak olan veri noktası

Eğitim Bölümü

For t=1,2,…,*T*

- t. sınıflandırıcı olan C_t için R_t^a rotasyon matrisini hesapla.

  1.Özellik dizisi olan *X*' i *K* adet alt kümeye, rasgele bir şekilde ayır.

  2. For *k*=1,2,…,*K*

  (a) $X_{t,k}$ alt matrisini üretmek için $F_{t,k}$ matrisi içerisindeki özelliklere uyan *X* satırlarını seç.

  (b) $X_{t,k}$ içerisinden bootstrap tekniği kullanarak $X_{t,k}$' matrisini oluştur ($X_{t,k}$' matrisinin boyutu $X_{t,k}$ matrisinin boyutundan küçük olmalıdır).

  (c) $D_{t,k}$ matrisinin elde edilmesi için $X_{t,k}$' matrisine temel bileşen analizi uygula (Burada $D_{t,k}$ matrisinin *i*. sütunu, *i*. temel bileşen katsayısını tutmalıdır).

  3.EndFor

  4. $D_{t,k}$ (k=1,2,…,*K*) matrisini, R_t köşegen blok matrisine yerleştir.

  5. R_t matrisinin satırlarını yeniden düzenleyerek R_t^a rotasyon matrisini oluştur (Böylece R_t^a matrisi, orijinal özellikler dizisi olan *F* matrisine cevap verecek hale gelmiş olacaktır).

- Bir C_t sınıflandırıcısını oluşturmak için, [*X* R_t^a *Y*] dizisini, $\mathcal{W}$ temel sınıflandırıcısının girişi olarak ata.

EndFor

Çıkış Bölümü

- *C'* grup sınıflandırıcısı tarafından, *x* verisinin tahmin edilen sınıf etiketi (3.4) eşitliğinde gösterildiği gibidir.

$$C'(x) = \underset{y \in \Phi}{argmax} \sum_{t=1}^{T} I(C_t.(x.R_t^a) = y), \quad (3.4)$$

Buradaki *I(.)*, sınıflandırıcı çıkışları doğrultusunda karar veren fonksiyondur.

Roach tarafından 2010 yılında oluşturulan Rotasyonel Orman algoritması, Zhang ve Zhang (2008)' ın önerdiklerinden biraz farklıdır. Roach' ın algoritması şu şekildedir:

S, orijinal eğitim dizisini ifade etsin. *T*, iterasyon sayısı olsun. *K* ise alt kümelerin sayısını ifade etsin.

1. For i=1 : *T*
2. $F_{i,j}$ matrisini oluşturacak şekilde, özellik dizisini *K* adet alt kümeye böl.
3. For j=1 : *K*
4. $F_{i,j}$ özellikler için, $S_{i,j}$ matrisini *S* matrisine uygun bir matris olarak tanımla.
5. $S_{i,j}$ matrisinden sınıf alt kümelerini rasgele çıkart.
6. $S_{i,j}$ matrisindeki örüntü sayısının %75' i kadar $S_{i,j}$ matrisinden bootstrap tekniği ile örnek al. Bu yeni diziyi $S_{i,j}'$ ile ifade et.
7. $C_{i,j}$ matrisi içerisindeki katsayıları elde etmek için $S_{i,j}'$ matrisine temel bileşen analizi uygula.
8. EndFor
9. (3.5)' te olduğu gibi, $C_{i,j}$ matrisini düzenle (1'den *K*'ya kadar bir rotasyon matrisi).

$$R_i = \begin{pmatrix} a_{i,1}^{(1)}, a_{i,1}^{(2)},\ldots, a_{i,1}^{(M_1)} & [0] & \ldots & [0] \\ [0] & a_{i,2}^{(1)}, a_{i,2}^{(2)},\ldots, a_{i,2}^{(M_2)} & \ldots & [0] \\ \ldots & \ldots & \ldots & \ldots \\ [0] & [0] & \ldots & a_{i,1}^{(1)}, a_{i,1}^{(2)},\ldots, a_{i,1}^{(M_1)} \end{pmatrix} \quad (3.5)$$

10. R_i matrisinin sütunlarını, R_i^a matrisini yeniden düzenlemek için yeniden yapılandır (Bu sayede F özellik dizisi içerisindeki özelliklerin düzeni ile karşılaştırma yapılacaktır).
11. EndFor
12. $(S\ R_i^a, X)$ dizisini eğitim dizisi olarak kullanıp, M_i sınıflandırıcısını oluştur.

Rotasyonel Orman tekniğinin bir değişik versiyonu da Kuncheva ve Rodriguez (2006) tarafından ortaya atılmıştır. Kuncheva ve Rodriguez tarafından önerilen algoritma şu şekildedir.

Eğitim Bölümü

- X: eğitim dizisindeki veriler (Nxn boyutlu bir matris)
- Y: eğitim dizisinin etiketleri ($Nx1$ boyutlu bir matris)
- L: grup içerisindeki sınıflandırıcıların sayısı
- K: alt küme sayısı
- $\{w_1, w_2,\ldots, w_c\}$: sınıf etiketlerinin dizisi

For i=1 : L

- R_i^a ile ifade edilen rotasyon matrisinin hazırlanması:

  -Özellik dizisi olan F matrisini K adet alt kümeye ayır: $F_{i,j}$ (for j=1:K).

  -For j=1:K

  1. $F_{i,j}$ özellik matrisi içindeki özellikler için X veri setini, $X_{i,j}$ ile ifade et.
  2. $X_{i,j}$ matrisinden sınıfların bir alt kümesini rasgele çıkart.
  3. $X_{i,j}$ matrisi içerisinden, $X_{i,j}$ matrisinin veri sayısının %75' i kadar veriyi, bootstrap tekniğinin uygulanması için seç. Bu yeni matrisi $X_{i,j}$' olarak ifade et.

4. $X_{i,j}$' matrisine, $C_{i,j}$ matrisindeki katsayıların elde edilmesi için temel bileşen analizi tekniğini uygula.

-$C_{i,j}$ matrisini düzenle.

-R_i^a matrisini, R_i matrisinin sütunlarını yeniden düzenleyerek oluştur (Bu sayede özellik dizisi olan F matrisindeki özellikler ile karşılaştırma işlemi yapılacaktır).

- (X R_i^a, Y) eğitim dizisini kullanarak D_i sınıflandırıcısını inşa et.

Sınıflandırma Bölümü

- $d_{i,j}(xR_i^a)$ ifadesini, ω_j sınıfından bir "x" değerine göre, $D_{i,j}$ sınıflandırıcısı tarafından belirtilen olasılık olarak tanımlayalım. Ortalama kombinasyon metodunu kullanarak, her bir ω_j sınıfı için uygunluk değerini hesapla.

$$\mu_j(x) = (1/L) \sum_{i=1}^{L} d_{i,j}(xR_i^a) \text{ , } j=1,...,c. \qquad (3.6)$$

- x verisini en büyük uygunluk değerini sağlayan sınıfa ata.

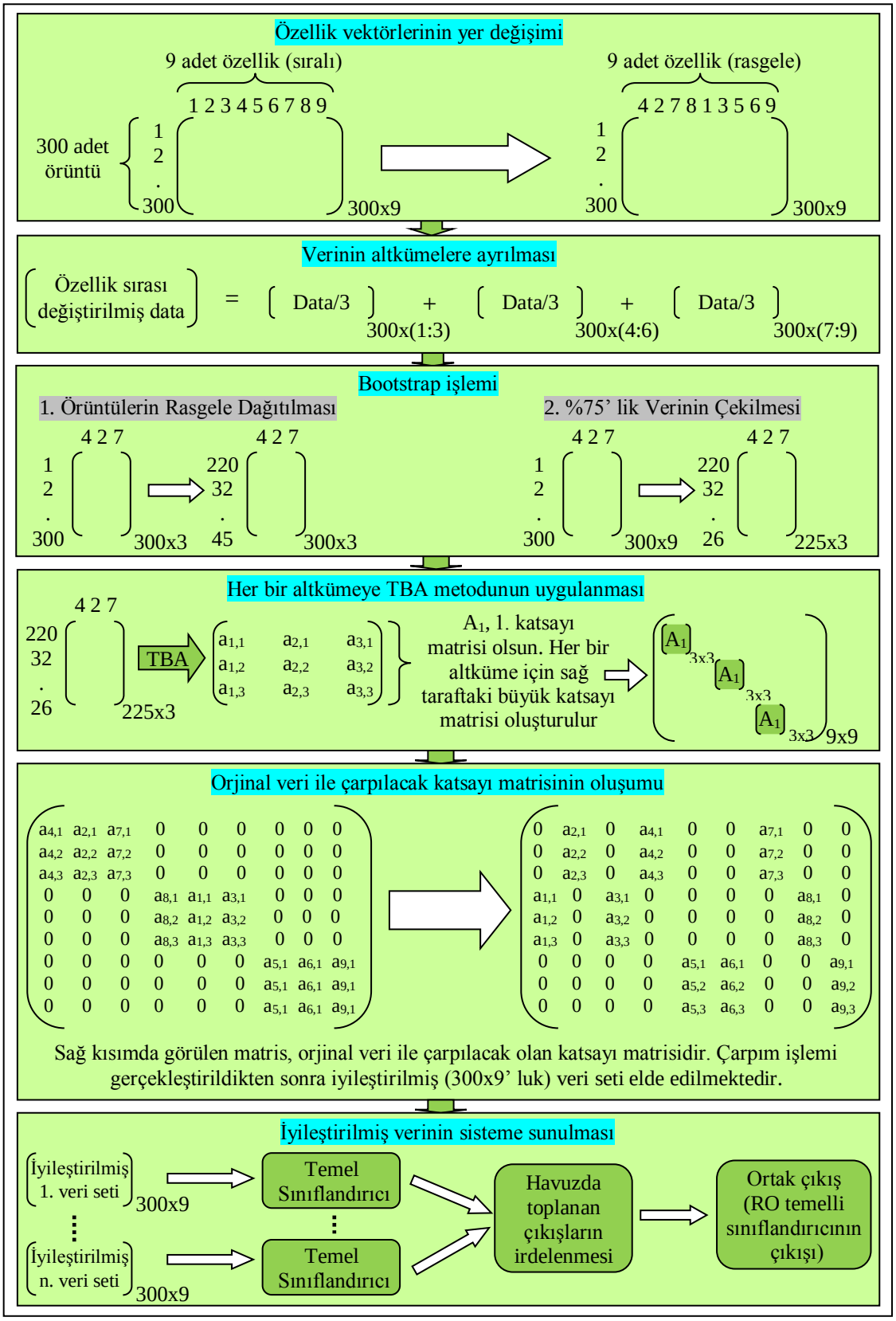
Örneğin, elimizde 300x9' luk bir veri matrisi olsun. Her bir satır bir örüntüyü, her bir sütun ise bir özelliği temsil etsin. Şekil 3.2' de, bu veri matrisi üzerinden, iyileştirilmiş olan veri setlerinin elde edilişi gösterilmektedir.

RO algoritması içerisindeki iyileştirilmiş verilerin elde edilme sürecinde, ilk aşamada özellik vektörlerinin yer değişimi işlemi gerçekleştirilmektedir. Yani, 9 adet özellik vektörleri (sütunlar) rasgele yer değiştirilir.

Bir sonraki adımda, özellik vektörleri rasgele yer değiştirilmiş olan veri seti, kullanıcı tarafından belirlenen sayıda alt kümeye ayrılır.

Alt kümelerin her birisine bootstrap işlemi uygulanarak, TBA matodu uygulanacak alt matrisler elde edilir. Bootstrap işlemi iki aşamadan meydana gelmektedir. İlk aşamada her bir örüntü rasgele olarak yer değiştirilir. İkinci aşamada ise, ilk %75' lik veri kısmı çekilerek bootstrap işlemi gerçekleştirilmiş olur. Bootstrap işlemi uygulanan her bir alt kümeye, diğer aşamada TBA metodu uygulanmaktadır.

Böylece her bir alt küme içerisindeki özellikleri temsil eden, küçük özvektör matrisleri elde edilir.



Şekil 3.2. İyileştirilmiş veri setlerinin elde edilişi

Bu küçük katsayı matrisleri diagonal bir biçimde büyük katsayı matrisi içerisine yerleştirilmektedir. Son aşamada, büyük katsayı matrisi içerisindeki her bir özvektör, orjinal veri içerisindeki özellik sırası sağlanacak şekilde yeniden sıralanmaktadır. Bu sayede orijinal veri ile çarpılacak olan katsayı matrisi elde edilmiş olur. Katsayı matrisi ile orjinal veri çarpılarak iyileştirilmiş olan veri seti elde edilmektedir. Bu işlemler RO yapısı içerisindeki her bir temel sınıflandırıcı için, bireysel olarak tekrarlanmaktadır. Özellik vektörlerinin yer değişimi ve bootstrap işlemlerindeki çeşitliliğin sağlanabilmesi için, bir önceki kombinasyon gerçekleşmeyecek şekilde rasgele atama işlemleri gerçekleştirilmelidir.

3.2. Optimizasyon

3.2.1. Optimizasyon nedir?

En temel anlamı ile optimizasyon, elimizdeki sınırlı kaynakları en iyi şekilde kullanmak olarak tanımlanabilir (Anonim, 2012). Matematiksel olarak ifade etmek gerekirse, optimizasyon bir fonksiyonun minimizasyonunun veya maksimizasyonun gerçeklenmesi olarak tanımlanabilir. Diğer bir deyişle optimizasyon "elde edilebilecek optimum amaç için, en iyi değeri veren durumlarda değişken değerlerinin bulunmasıdır" (Anonim, 2012). Başka bir tanımlama ile "belirli amaçları elde edebilmek için en iyi kararları verme yolu" veya "sınırlı durumlar altında herhangi bir işlemi en iyi şekilde gerçekleme" olarak da tanımlanan optimizasyon, "en iyi sonuçları elde eden işlemler dizisidir" (Anonim, 2012).

Değişen teknolojilerin, sınırlı kaynakların, artan rekabetin, karmaşık hale gelen sistemlerin doğurduğu problemlerin klasik yöntemlerle (matematiksel veya matematiksel olmayan, analitik veya sayısal) çözümünün güçleşmesi, optimizasyon kavramını güncelleştiren en önemli sebeptir. Bu yönüyle optimizasyonun kullanılmadığı bir bilim dalı hemen hemen yok gibidir (Kara, 1986).

İnsanlar hayatları boyunca karşılaştıkları sorunları çözmek amacıyla, bu çözümleri modeller üzerinde arama yaklaşımına girmişlerdir. Matematik ve bilgisayar sistemleri alanındaki gelişmeler, dış dünyanın problemlerini matematiksel olarak gerçekleyerek, bu çözümleri modelleyip gerçek hayata yansıtma olanağı vermiştir (Yılmaz, 2012). Matematiksel modelleme tekniği, öncelikle doğrusal ve az sayıda olan değişkenlerin kullanılmasıyla başlamıştır. Bir süre sonra doğrusallık varsayımının her

problem için geçerli olmadığı anlaşılmıştır. Bu durumda doğrusal olmayan modellemeye gidilmiştir. Ancak doğrusal olmayan modellerin kendine özgü çözümleri, uygulamada birçok sorunu beraberinde getirmiştir. Zamanla geliştirilen bazı yöntemlerle doğrusal olmayan modellerin hızla çözümlenmesi sağlanmış ve bu durum optimizasyon teorisini geliştirmiştir (Yılmaz, 2012).

Bir işin yapılmış olması demek, o işin en iyi şekilde yapıldığı anlamına gelmez. Optimizasyon teknikleri, yapılmış veya yapılmakta olan işin en iyi çözümünü ortaya koymak için kullanılır. Bu teknikler kullanılarak ortaya konulmuş olan çözüm, en iyi çözüm olarak adlandırılır. Hedef her zaman için bu en iyi çözümü yakalayabilmektir. Optimizasyon, anlamından da anlaşılacağı gibi her alanda kullanılmaktadır. Yapılacak olan bir inşaattan tutun bir web sitesine kadar her alanda bu tekniklere ihtiyaç duyulur (Yılmaz, 2012).

3.2.2. Optimizasyon çeşitleri

Optimizasyon, farklı alternatifler içerisinden en iyi olanını seçme işlemidir. Sezgisel algoritmalar, büyük boyutlu işlemlerde, kısa sürede optimuma yakın sonuçlar üretebilen algoritmalardır. Sezgisel optimizasyon algoritmaları; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı gruba ayrılmaktadır. Sürü zekâsı tabanlı optimizasyon algoritmaları oluşturulurken kuş, balık ve arı gibi canlı sürülerinin hareketlerinden esinlenilmiştir (Akyol ve Alataş, 2012).

Optimizasyonun ilk kez ortaya çıkışından günümüze, çeşitli problemlerin çözümünü gerçeklemek ve çözmek amacıyla birçok optimizasyon yöntemi geliştirilmiştir. Bu optimizasyon tekniklerinden en çok kullanılanları aşağıdaki gibi sıralanmıştır:

- Benzetilmiş Tavlama (*Simulated Annealing*) Tekniği
- Diferansiyel Evrim Algoritması Tekniği
- Karınca Kolonisi Optimizasyonu
- Parçacık Sürü Optimizasyonu
- Yapay Bağışıklık Sistemi Optimizasyonu
- Temel Tabu Arama Algoritması

3.2.3. Optimizasyon ve sınıflandırma metotları ile kullanımı

Optimizasyon teknikleri en iyiyi gerçekleme amacını taşıyan, rasgelelik kuramına dayalı algoritmalardır. Temel sınıflandırıcı teknikleri olan karar ağaçları, YSA, çok katmanlı ağlar, destek vektör makinesi gibi tekniklerin daha iyi sonuç vermeleri amacı ile, bu sistemlere çeşitli optimizasyon teknikleri eklenmektedir.

Optimizasyon teknikleri genel olarak biyolojik veya ekolojik sistemlerden esinlenerek geliştirilmişlerdir. Parçacık sürü optimizasyonu kuş ve balık sürülerinin hareketlerinden esinlenmiştir. Arı kolonisi optimizasyonu arıların yemek bulma ve bu yemeği kovana taşıma işlerinden yola çıkılarak keşfedilmiştir. Karınca kolonisi optimizasyonu ise, karıncaların yiyecek bulma ve bu yiyeceğe diğer karıncaların da ulaşabilmesi için yürüttükleri faaliyetlerden faydalanılarak tasarlanmıştır. Bu şekilde tasarlanan optimizasyon teknikleri "Zeki Optimizasyon Teknikleri" olarak da bilinirler.

Bu tekniklerin temel sınıflandırıcı sistemlerle kullanılması sonucu, temel sınıflandırıcıların doğruluk ve performans değerlerinin artırılması hedeflenir. Daha önce gerçekleştirilen yüksek lisans seminer çalışmasında kaskat PSO-YSA sistemi tasarlanmıştır. Buradaki asıl amaç sınıflandırma işlemini daha yüksek doğruluk oranı ile gerçekleştirebilmektir. Önerilen yapıda, başlangıçta rasgele üretilen ağırlık değerleri, PSO tekniğiyle üretilmiştir. Böylece sınıflandırma doğruluğunun artırılması sağlanmıştır.

Optimizasyonun sınıflandırıcılara doğru adapte edilmesi çok önemlidir. Optimizasyon tekniği içerisinde bulunan değişkenler ile temel sınıflandırıcı içerisinde bulunan değişkenlerin uyumlu olması, uyumlandırılması gereklidir. Uyumlama işlemi ile temel sınıflandırıcının gerekli değişkeninin optimizasyon tekniğine göre şekillenmesi, optimizasyon yönteminin temel amacını oluşturur.

Sınıflandırıcı tekniklerde yaşanan önemli bir problem, global minimum değerinin elde edilememesidir. Optimizasyon tekniklerinin amacı, global minimuma ulaşılmasını sağlamaktır. Söz konusu sistemlerin yerel minimum veya maksimum değerlere sapmasını önlemek, örneklerin sınırlı bir çerçeve ile kurallar dizisi içerisinde işlenmesini sağlamak ve global minimum-maksimum değerlerin elde edilmesi amaçları, optimizasyon tekniklerinin temel sınıflandırıcı sistemler içerisinde kullanılmasını zorunlu hale getirmektedir.

3.2.4. Parçacık sürü optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (*Particle Swarm Optimization*) 1995' te Dr. Eberhart ve Dr. Kennedy tarafından geliştirilmiş, popülasyon temelli, sezgisel bir optimizasyon tekniğidir (Sağ, 2009). Kuş ve balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir (Sağ, 2009).

PSO, Genetik Algoritmalar gibi evrimsel hesaplama teknikleri ile birçok benzerlikler gösterir. Sistem rasgele çözümlerden oluşan bir popülâsyonla başlatılır ve en iyi çözüm için jenerasyonları güncelleyerek arama yapar. Buna karşın, Genetik Algoritmanın tersine, PSO' da çaprazlama ve mutasyon gibi evrimsel operatörler yoktur (Sağ, 2009). PSO' nun avantajları, genetik algoritma gibi karmaşık bir yapıda olmaması ve ayarlanması gereken çok az parametresinin olması şeklinde sıralanabilir. PSO' nun uygulama alanları; fonksiyon optimizasyonu, yapay sinir ağları eğitimi, bulanık sistem kontrolü ve genetik algoritmanın uygulanabildiği diğer alanlardır (Sağ, 2009). Biyolojik sistemlerden esinlenen birçok hesaplama tekniği mevcuttur. Örneğin yapay sinir ağları, insan beyninin basitleştirilmiş bir modelidir. Genetik algoritmalar biyolojideki evrimsel süreçten esinlenir. Burada ise ele alınan konu, biyolojik sistemlerin farklı bir türü olan sosyal sistemlerdir. Özellikle birbiriyle ve çevresiyle etkileşim içinde olan bireylerin, sürü psikolojileri incelenmektedir. Bu kavram parçacık zekâsı olarak da bilinir. PSO' da parçacık olarak isimlendirilen çözümler, en iyi çözümleri kontrol ederek problem uzayında arama yaparlar (Sağ, 2009).

3.2.4.1. Parçacık sürü optimizasyonu algoritması

Bir alanda rasgele yiyecek arayan bir kuş grubunun olduğunu ve arama alanında yalnızca bir parça yiyecek olduğunu varsayalım. Kuşlar yiyeceğin pozisyondan habersiz olsun. Fakat her bir iterasyon sonunda yiyeceğin ne kadar uzakta olduğunu bilsinler. Bu durumda optimum seçenek, yiyeceğe en yakın olan kuşu takip etmektir. PSO bu senaryoya göre çalışır ve optimizasyon problemlerini çözmek için kullanılır (Sağ, 2009). PSO' da her bir çözüm, arama uzayındaki bir kuştur ve bunlar bir "parçacık" olarak isimlendirilir. Tüm parçacıkların, optimize edilecek uygunluk fonksiyonu tarafından değerlendirilen bir uygunluk değeri ve uçuşlarını yönlendiren hız bilgileri vardır (Sağ, 2009). Parçacıklar, problem uzayında optimum parçacıkları takip ederek

hareket ederler. PSO bir grup rasgele üretilen değerlerle (parçacıkla) başlatılır ve jenerasyonlar güncellenerek en uygun değer araştırılır (Sağ, 2009).

Parçacıklar her iterasyonda iki "*en iyi*" değere göre güncellenir. Bunlardan birincisi bir parçacığın o ana kadar elde ettiği en optimum uygunluk değeridir. Ayrıca bu değer daha sonra kullanılmak üzere hafızada tutulur ve "*pbest"* (parçacığın en iyi değeri) olarak isimlendirilir. Diğer en iyi değer ise popülasyondaki herhangi bir parçacık tarafından o ana kadar elde edilmiş en optimum uygunluk değerine sahip çözümdür. Bu değer popülasyon için global en iyi değerdir ve *"gbest"* olarak isimlendirilir (Sağ, 2009).

D adet parametreden oluşan *n* adet parçacık için popülasyon matrisi, (3.7)' deki gibi ifade edilir.

$$\boldsymbol{X}=\begin{pmatrix} x_{11} & \cdot & \cdot & \cdot & x_{1D} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ x_{n1} & \cdot & \cdot & \cdot & x_{nD} \end{pmatrix} \tag{3.7}$$

Matrise göre; *i.* parçacık (3.8)' de ifade edildiği gibidir. Parçacığın en iyi konumu (3.9) ve sürüdeki en iyi konum (3.10) ile gösterilmektedir (Sağ, 2009).

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}] \tag{3.8}$$

$$pbest_i = [P_{i1}, P_{i2}, \ldots, P_{iD}] \tag{3.9}$$

$$gbest_i = [P_1, P_2, \ldots, P_D] \tag{3.10}$$

i. parçacığın her konumdaki değişim miktarını gösteren hız vektörü (3.11)' te ifade edildiği gibidir (Sağ, 2009).

$$V_I = [V_{i1}, V_{i2}, \ldots, V_{iD}] \tag{3.11}$$

Bu iki en iyi değer bulunduktan sonra; parçacık, hızını ve konumunu sırasıyla aşağıdaki (3.12) ve (3.13) eşitliklerine göre günceller (Sağ, 2009).

$$V_i^{k+1} = V_i^k + c_1 . rand_1^k .(pbest_i^k - x_i^k) + c_2 . rand_2 . (gbest^k - x_i^k) \quad (3.12)$$

$$x_I^{k+1} = x_i^k + V_i^{k+1} \quad (3.13)$$

Burada "*rand"* (0,1) arasında üretilen rasgele bir değeri, *i* parçacık numarasını, *k* ise iterasyon sayısını gösterir. c_1 ve c_2 öğrenme faktörleridir. Bunlar parçacıkları *pbest* ve *gbest* konumlarına doğru yönlendiren sabitlerdir. c_1 parçacığın kendi tecrübelerine göre, *c2* ise sürüdeki diğer parçacıkların tecrübelerine göre hareketi yönlendirir. Düşük değerler seçilmesi, parçacıkların hedef bölgeye ulaşmadan bu bölgeden uzak alanda tarama yapılmasını sağlar. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi hedefe ulaşmayı hızlandırırken, istenmeyen hareketlerin oluşmasına (hedefin es geçilmesine) sebep olabilir. Genellikle $c_1=c_2=2$ olarak almanın iyi sonuçlar verdiği belirtilmiştir (Sağ, 2009).

PSO tekniğinin algoritması Çizelge 3.1' de verilmiştir (Sağ, 2009).

Çizelge 3.1. PSO tekniğinin algoritması

For her parçacık için
Parçacığı başlangıç konumuna getir
End

Do
For her parçacık için
Uygunluk değerini hesapla
Eğer uygunluk değeri pbest ten daha iyi ise**,**
Şimdiki değeri yeni pbest olarak ayarla
End

Tüm parçacıkların bulduğu pbest değerlerinin en iyisini, tüm parçacıkların gbest'i olarak ayarla

For her parçacık için
Denklem (1)'e göre parçacık hızını hesapla
Denklem (2)'ye göre parçacık konumunu güncelle
End

While maksimum iterasyon sayısına veya minimum hata koşulu sağlanana kadar devam et (Tamer ve Karakuzu, 2006).

3.2.4.2. Parçacık sürü optimizasyonunun parametreleri

PSO' nun avantajlarından biri, reel sayılarla çalışıyor olmasıdır. Genetik algoritmalardaki gibi karmaşık hesaplamaların yapılmasına ya da zorunlu operatörlere ihtiyaç duyulmaz (Sağ, 2009).

Örneğin $f(x) = x_1^2 + x_2^2 + x_3^2$ fonksiyonu için çözümü bulmayı deneyelim. Burada 3 bilinmeyen olduğundan dolayı D=3 boyutludur ve parçacık (x_1 , x_2 , x_3) şeklinde ayarlanır. $f(x)$ fonksiyonu da uygunluk fonksiyonu olarak kullanılır. Sonlanma kriteri olarak, maksimum iterasyon sayısı veya minimum hata koşulunun sağlanması durumları göz önünde bulundurulur. PSO' da ihtiyaç duyulan çok az sayıda parametre vardır. Bu parametreler; parçacık sayısı, parçacık boyutu, parçacık aralığı, hız sınırları, öğrenme faktörleri ve durma koşuludur (Tamer ve Karakuzu, 2006).

1-) *Parçacık Sayısı:* Genellikle 20 ila 40 arasında alınır. Aslında çoğu problem için sayıyı 10 almak, iyi çözümler elde etmek için yeterlidir (Tamer ve Karakuzu, 2006).

2-) *Parçacık boyutu*: Optimize edilecek probleme göre değişmektedir (Tamer ve Karakuzu, 2006).

3-) *Parçacık aralığı*: Optimize edilecek probleme göre değişmekle birlikte farklı boyutlarda ve aralıklarda parçacıklar tanımlanabilir (Tamer ve Karakuzu, 2006).

4-) *Vmax:* Bir iterasyonda, bir parçacıkta meydana gelecek maksimum değişikliği hız belirler. Genellikle parçacık aralığına göre belirlenir. Örneğin X_1 parçacığı (-10,10) aralığında ise Vmax=20 şeklinde sınırlandırılabilir (Tamer ve Karakuzu, 2006).

5-) *Öğrenme Faktörleri*: c_1 ve c_2 genellikle 2 olarak seçilir. Fakat farklı da seçilebilir. Ancak değişkenlerden herbirinin [0,4] aralığında seçilmesi gereklidir (Tamer ve Karakuzu, 2006).

6-) *Durma Koşulu:* Maksimum iterasyon sayısına ulaşıldığında veya değer fonksiyonu istenilen seviyeye ulaştığında algoritma durdurulabilir (Tamer ve Karakuzu, 2006).

3.3. Yapay Sinir Ağları (YSA)

Yapay sinir ağları terimi, beynin çalışma mantığının bir bilgisayar sistemi üzerine adapte edilmesi fikri ile ortaya çıkmıştır (Anonim, 2012). Bu kapsamdaki ilk çalışmalar beyni oluşturan biyolojik hücrelerin, ya da diğer bir adıyla nöronların, matematiksel bir şekilde modellenmesi sayesinde gerçekleştirilmiştir. Her bir nöronun kendisine komşu olan diğer nöronlardan bilgiler aldığı ve bu bilgilerin biyolojik nöron sisteminin öngördüğü biçimde, bir çıktıya dönüştüğü sistem olarak da tanımlanabilmektedir (Anonim, 2012).

Yapay sinir ağları olarak adlandırılan sistem, birçok nöronun belirli koşullar çerçevesinde bir araya getirilip, bir amacın gerçekleştirilmesi üzere, yapısal olduğu kadar matematiksel ve bilimsel sorunlara da cevap arayan bir sistem olmuştur (Anonim, 2012). Yapay sinir ağları sistemleri, karar alma ve verme hızı bakımdan insan beyni ile rekabet edebilecek seviyeye henüz gelememiş olmalarına rağmen, kompleks eşleştirmelerin hassas bir biçimde gerçeklenebilmesi ve bunun yanı sıra yapısal yeterliliğe sahip olmaları nedeniyle, her geçen gün uygulama alanları büyümektedir (Anonim, 2012).

3.3.1. Yapay sinir ağlarında temel öğrenme teknikleri

Yapay sinir ağları sisteminde bilgilerin tanıtılması örnekler yolu ile olur. Örnekler, bir olayın girdi ve çıktılarını temsil etmektedir. Girdi ve çıktılar arasındaki ilişkiyi öğrenen yapay sinir ağının, verilecek farklı girişler için çıkışları üretebilecek bilgi ve tecrübeyi kazandığı söylenebilir. Örneğin bir sistemin birim basamak girişe verdiği cevap biliniyorsa, birim basamak giriş ve çıkış değerleri ile yapay sinir ağının sistem karakteristiğini öğrenmesi sağlanabilir. Artık yapay sinir ağı, sistemin diğer girişleri için nasıl cevap vereceğini bilmektedir. Örnekler oluşturulurken mutlaka giriş ve çıkış değerlerinin birlikte kullanılması gerekmez. Bazı durumlarda sadece giriş değerleri ile örnekler oluşturulabilir. Örneklerin nasıl oluşturulduğuna bağlı olarak,

yapay sinir ağlarının üç temel öğrenme biçimi vardır. Bu öğrenme teknikleri aşağıda belirtildiği gibidir (Öğücü, 2006).

Öğretmenli (Supervised) Öğrenme: Örnekler girdi-çıktı kümesinin birlikte kullanılması ile oluşturulur. Sistem girişleri ve sistemin bu girişlere verdiği cevaplar, yapay sinir ağına tanıtılır ve yapay sinir ağının giriş-çıkış arasındaki ilişkileri öğrenmesi istenir (Öğücü, 2006).

Destekleyici (Reinforcement) Öğrenme: Örnekler sadece giriş değerleri ile oluşturulur. Çıkış değerlerini yapay sinir ağına tanıtmak yerine, yapay sinir ağının bir çıkış üretmesi beklenir ve üretilen bu çıkışın doğru ya da yanlış olmasına göre yapay sinir ağına, onaylayıcı ya da reddedici bir sinyal gönderilir (Öğücü, 2006).

Öğretmensiz (Unsupervised) Öğrenme: Yapay sinir ağlarının öğrenmesine yardımcı olacak herhangi bir öğretmen ya da destekleyici unsur yoktur. Sisteme sadece giriş değerleri tanıtılır. Bu değerler arasındaki ilişkileri, benzerlikleri ve farklılıkları yapay sinir ağının kendi kendisine öğrenmesi istenir. Fakat sistemin öğrenme süreci bittikten sonra çıktıların ne anlama geldiğini gösteren etiketlemenin, kullanıcı tarafından yapılması gerekmektedir. Bu öğrenme biçimi daha çok sınıflandırma problemleri için kullanılır (Öğücü, 2006).

Karma Öğrenme: Yukarıda bahsedilen öğrenme biçimlerinin birlikte kullanılmasıyla karma öğrenme gerçekleştirilebilir (Öğücü, 2006).

3.3.2. Yapay sinir ağlarında yapısal öğrenme kuralları

Söz konusu bilginin, kurallar şeklinde açıklandığı klasik uzman sistemlerin tersine YSA sistemi, gösterilen örnekten öğrenerek kendi kurallarını oluşturur. Öğrenme; giriş örneklerine veya bu girişlerin çıkışlarına bağlı olarak, ağın bağlantı ağırlıklarını değiştiren veya ayarlayan öğrenme kuralı ile gerçekleştirilir. Öğreticisiz öğrenmede giriş işareti için istenen çıkış sisteme tanıtılır. Bu işlemin ardından YSA giriş-çıkış ilişkisini gerçekleştirene kadar kendi kendini kademeli olarak ayarlar. Günümüzde kullanılan birçok öğrenme kuralı vardır.

En çok kullanılan öğrenme kuralları şunlardır (Özbay, 1999):

1-Rastlantısal (Hebb) öğrenme kuralı,
2-Performans (Widrow ve ADALİNE) öğrenme kuralı,
3-Kompetetif (Kohonen) öğrenme,
4-Filtreleme (Grossberg),
5-Spotitemporal öğrenme,
6-Genelleştirilmiş delta kuralı öğrenme.

Bu öğrenme tekniklerinden, seminer çalışması içerisinde kullanılan "Genelleştirilmiş Delta Kuralı Öğrenme" tekniği işlenecek olup, diğer öğrenme kurallarına değinilmeyecektir. Genelleştirilmiş delta kuralı öğrenme, perseptron yapısından başlanarak son haline kadar geçirdiği değişim süreci ile anlatılacaktır.

3.3.2.1. Perseptron yapısı

Perseptron, F.Rosenblatt tarafından 1950'li yılların sonlarında ortaya atılmış bir modeldir. İlk yapay sinir ağı modellerinden olan perseptronlar son derece sınırlı olmalarına karşın, daha sonra geliştirilen modellere temel olmaları ve mevcut algoritmalarla yakın ilişki içinde oldukları için oldukça önemlidirler (Baş, 2006). Perseptron mantığı, en basit şekilde bir yapay sinir hücresinin birden fazla girdiyi alarak, tek bir çıktı üretmesi temeline dayanmaktadır (Baş, 2006). Ağın çıktısı 1 veya 0 olmaktadır. Ağ çıkış değerinin hesaplanması ise, adım fonksiyonu kullanılarak yapılmaktadır. Perseptron, üzerine eğitme işlemi uygulanabilen bir adet işlemci elemandan oluşmaktadır (Baş, 2006). İşlemci elemana girdi değerleri ve bu değerlere karşılık gelen çıktı değerleri gösterilir. Bu doğrultuda, öğrenme kuralına göre ağın çıktı değeri hesaplanmaktadır. Olması gereken çıktı değerine ulaşılana kadar ağırlıklar ve eşik değerleri değiştirilmektedir (Baş, 2006). Perseptron modeli Şekil 3.3' de gösterilmiştir.

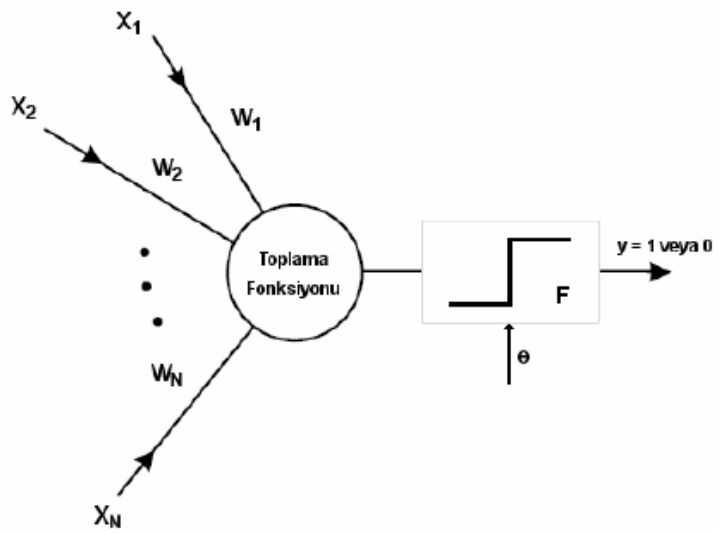
Perseptron öğrenme algoritması dört adımda açıklanabilmektedir;

Adım 1: Yapay sinir ağına girdi değerleri ve ona karşılık gelen çıktı değeri gösterilmektedir. Birden fazla girdi değeri olabilmektedir. Buna karşılık çıktı değeri 1 ve 0 değerlerinden birini almaktadır (Baş, 2006).

Adım 2: İşlemci elemana gelen net girdi (3.14)' de ifade edildiği gibidir (Baş, 2006).

$$S = \sum x_i \,.\, w_i \tag{3.14}$$

Adım 3: Bu adımda işlemci elemanın çıktısı hesaplanır. Net girdi, eşik değeri ile kıyaslanır ve eğer net girdi eşik değerinden büyükse çıktı 1, küçükse 0 değerini alır. Bu ifade (3.15) ile gösterilmiştir (Baş, 2006).



Şekil 3.3. Perseptron modeli (Baş, 2006)

$$y = \begin{cases} 1 & s > \theta \\ 0 & s < \theta \end{cases} \tag{3.15}$$

Eğer gerçekleşen çıktı ile beklenen çıktı aynı değeri alırlarsa, ağırlıklarda herhangi bir değişiklik olmaz. Eğer farklı bir çıktı üretilirse, şu iki durum söz konusu olur (Baş, 2006).

1. Durum: Eğer ağın beklenen çıktı değeri 0, ancak gerçekleşen çıktı değeri 1 ise, ağırlık değerleri azaltılır. Bunun için vektörel olarak (3.16) eşitliği kullanılır;

$$W_n = W_o - \lambda X \tag{3.16}$$

Burada λ öğrenme katsayısı olup, sabit bir değere sahiptir. W_n, yeni ağırlık vektörünü, W_o ise eski ağırlık vektörünü ifade etmektedir.

2. Durum: *Eğer* ağın beklenen çıktı değeri 1, ancak gerçekleşen çıktı değeri 0 ise, ağırlık değerleri artırılır. Bunun için kullanılan vektörel eşitlik ise (3.17)' de ifade edildiği gibidir.

$$W_n = W_o + \lambda X \tag{3.17}$$

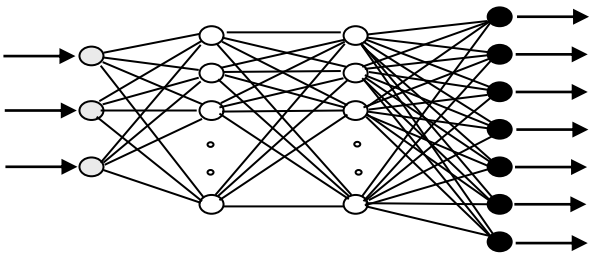
Adım 4: Yukarıdaki ilk 3 adım, bütün girdi setindeki örnekler için doğru sınıflandırmalar yapılıncaya kadar tekrarlanır.

Çizelge 3.2' de ise yapay sinir ağı içerisinde kullanılan aktivasyon fonksiyonları, fonksiyonların çıkışları ve şekilleri görülmektedir.

3.3.2.2. Çok katmanlı perseptron (ÇKP)

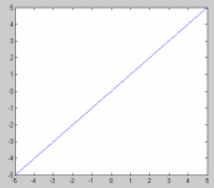
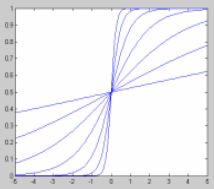
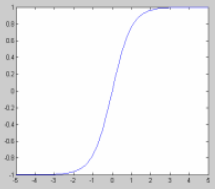
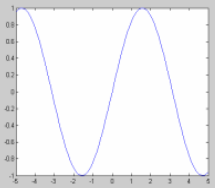
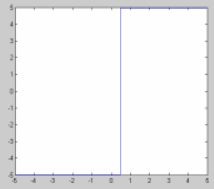
Günümüzde en çok bilinen ve yaygın biçimde kullanılan yapay sinir ağı sınıflandırma yöntemi, çok katmanlı perseptronlardır.

Genel amaçlı, esnek ve çoklu tabakalar halinde organize edilmiş birimlerden (nöron) oluşan doğrusal olmayan (nonlineer) modellerdir. ÇKP sinir ağı, yaklaşık olarak her fonksiyon sonucunu tahmin etme yeteneğine sahip, yaklaşım özellikli bir ağdır (Arslan ve ark., 2007). Birimler genellikle ileri beslemeli bağlantılara sahiptir. İleri beslemeli bağlantının anlamı; bağlantıların herhangi bir şekilde döngü oluşturmamasıdır. ÇKP sinir ağının üstünlüğü, nöronlarla doğrusal olmayan (*nonlineer*) bir ilişki kurabilme yeteneğinden kaynaklanmaktadır (Arslan ve ark., 2007). Çok katmanlı sinir ağı yapısı Şekil 3.4' de gösterilmektedir.



Şekil 3.4. Çok Katmanlı Sinir Ağı Yapısı (Arslan ve ark., 2007)

Çizelge 3.2. Yapay sinir hücrelerinde kullanılan aktivasyon fonksiyonları (Öğücü,2006).

Fonksyonun Adı	Fonksyonun Şekli	Matematiksel İfadesi	Açıklama
Lineer Fonksyon		$F(NET) = NET$	Hesaplanan net giriş değeri, hücrenin çıkışı olarak kabul edilir
Sigmoid Fonksyonu		$F(NET) = \frac{1}{1+e^{-\beta NET}}$	β değeri keyfi şekilde değiştirilerek farklı eğimlere sahip eğriler elde edilebilir
Hiperbolik Tanjant Fonksyonu		$F(NET) = \frac{e^{NET} + e^{-NET}}{e^{NET} - e^{-NET}}$	Hesaplanan net giriş değeri, tanjant fonksyonuna uygulanır
Sinüs Fonksyonu		$F(NET) = Sin(NET)$	Öğrenilmesi istenen olayın, sinüs fonksyonuna uygun dağılım gösterdiği durumlarda kullanılır
Basamak Fonksyon		$F(NET) = \begin{cases} 1 & NET > k \\ 0 & NET \le k \end{cases}$	k, eşik değeri göstermektedir. Hesaplanan değerin, eşik değerden büyük yada küçük olmasına göre 1 yada 0 çıkışları üretilir

Sinir ağının çalıştırılması (öğrenmesi), ağın sonuç değeri ile hedeflenen sonuçlar arasındaki hata ölçüsünün (farkın) minimum yapılması esasına dayanır (Arslan ve ark., 2007). Bu değerler çalıştırma (eğitim) aşamasında 'geriye doğru' yayılarak ve yinelemeli biçimde gerçekleştirilir. Algoritma bu nedenle Geri Yayılma Algoritması olarak adlandırılmıştır (Arslan ve ark., 2007).

Geri yayılma kuralı, hata değerlerini ağ boyunca dağıtmakta ve dolayısı ile gizli nöronlara uyumunu sağlamaktadır. Bu uyum Gradyen İniş Öğrenme Kuralı teorisi kullanılarak, ağ içindeki her bir ağırlığa düzeltme getirilmesiyle adaptasyonu

sağlanmaktadır. Geri yayılma kuralı, aslında ağ içindeki her ağırlığın ilgili hata fonksiyonunun duyarlılığını hesaplamakta ve bu ağırlığı bu duyarlılığa göre güncelleştirmektedir (Arslan ve ark., 2007). Dolayısıyla hata yüzeyinin global minimum değeri arandığından, yalnızca yerel bilgiyi kullanmakta ve hesaplanması kolay bir yöntem olmaktadır. Aynı zamanda yerel minimuma yakalanabileceği için sakıncalıdır. Geri yayılma kuralının temel prensibi budur. Bu sakıncayı bertaraf etmek amacıyla farklı yöntemler geliştirilmiştir (Arslan ve ark., 2007). Çok katmanlı sinir ağları, genellikle geri yayılma algoritması ile çalıştırılmaktadır. Sinir ağının öğrenme işlemi hata fonksiyonunun minimum yapılması esasına dayanmaktadır. En küçük hata değerine sahip noktaya ilişkin ağırlıklar, optimum ağırlıklardır. Muhtemel olan en düşük hata değerine sahip teorik çözüm, *"*global minimum*"* değeridir (Arslan ve ark., 2007). Geri yayılma kuralı, çözüm vektör uzayında, hata yüzeyinin basamaklı kesitleri boyunca global minimum değerine ulaşmak için gradyen iniş tekniği uygular. Hata yüzeyinin çoğu kez düzensiz ve karmaşık bir yapıya sahip (birçok 'vadi ve tepe' içerir) olması nedeniyle, sinir ağı en uygun çözüm olmayan "yerel minimum" değerine yakınsayabilir. Ağ ağırlıkları istenen ve gerçekleşen ağ sonuçları arasındaki fark ölçüsüne dayanan bir hata değerini minimum yapacak şekilde dengelenmektedir (Arslan ve ark., 2007).

Hata fonksiyonu (3.18)' de ifade edildiği gibidir.

$$E^p = 0{,}5 \sum (d_j^p - o_j^p) \tag{3.18}$$

(3.19)' daki d_j elde edilmek istenen sonuç olup, o_j elde edilen sonuçtur ve p örüntü sayısı olarak tanımlanmaktadır. Ağırlıkları dengelemek (ağ düzeltmesi) için hata fonksiyonunun gradyeni kullanılır (Arslan ve ark., 2007).

Hata fonksiyonunun gradyeni (3.19) eşitliği ile ifade edilir.

.

$$\Delta^p W_{ji} = - \eta . (dE^p / dW_{ji}) \tag{3.19}$$

Burada ağırlık düzeltmesi için kullanılan η sabitine "öğrenme oranı" denilmekte olup; ağırlık değişimi için hesaplanacak hata duyarlık değerini belirlemektedir. En iyi öğrenme oranı (η) değeri hata yüzeyinin özelliklerine bağlıdır. Ağırlıklar (3.20)' deki gibi hesaplanmaktadır (Arslan ve ark., 2007).

$$W_{ij}(t+1) \cong W_{ij}(t) + \eta.\delta_j.i_i \quad (3.20)$$

Burada $W_{ij}(t)$, t zamanında i giriş (veya gizli) düğümü ile j düğümü arasındaki ağırlıktır. i_i ise i düğümünün sonuç değeri veya bir giriştir. δ_j, j düğümünün hata terimidir. Bir çıkış düğümü için hata terimi olan δ_j (3.21)' de ifade edildiği gibidir (Arslan ve ark., 2007).

$$\delta_j = o_j .(1 - o_j).(d_j - o_j) \quad (3.21)$$

Burada d_j, j düğümünün istenen sonuç değeri olup, o_j gerçekleşen sonuç değerdir. j düğümü bir gizli düğüm olmak üzere hata terimi olan δ_j (3.22) ile ifade edilir (Arslan ve ark., 2007).

$$\delta_j \cong o_j .(1 - o_j). \sum \delta_k . W_{jk} \quad (3.22)$$

Buradaki k, j düğümü üstündeki tabakalardaki düğümdür. Bazen bir "moment" teriminin (α) eklenmesi yakınsamayı hızlandırmakta ve ağırlık değişimleri yumuşatılmaktadır. Bu durum (3.23) eşitliği ile gösterilmiştir (Arslan ve ark., 2007).

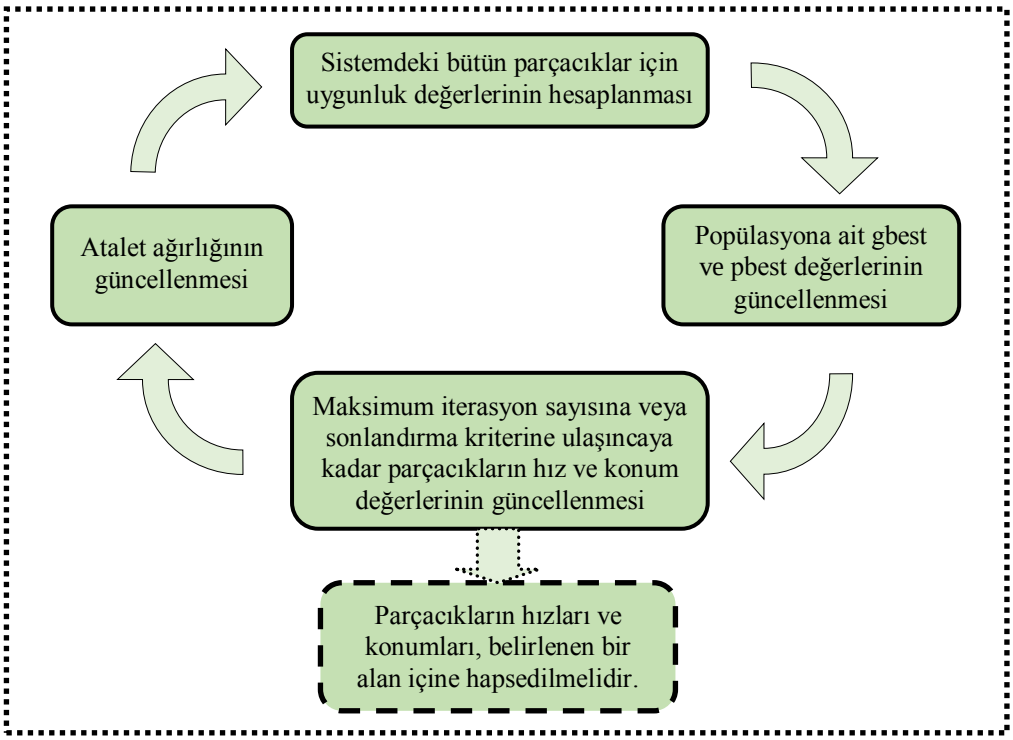
$$W_{ij}(t+1) \cong W_{ij}(t) + \eta.(d_j - o_j).i_i + \alpha.(W_{ij}(t) - W_{ij}(t-1)) \quad (3.23)$$

Öte yandan yaygın olarak kullanılmakta olan bu algoritma uzun zaman gerektirmektedir. Bu nedenle algoritmanın çözümünü hızlandırmak için çeşitli algoritmalar önerilmektedir. Ayrıca ağırlık güncellemelerini ardışık biçimde ve lokal olarak hesaplandığı için yerel minimuma yakalanma riski vardır (Arslan ve ark., 2007).

4. GERÇEKLEŞTİRİLEN HİBRİT SINIFLANDIRICI YAPILARI

4.1. Hibrit PSO-YSA Yapısı

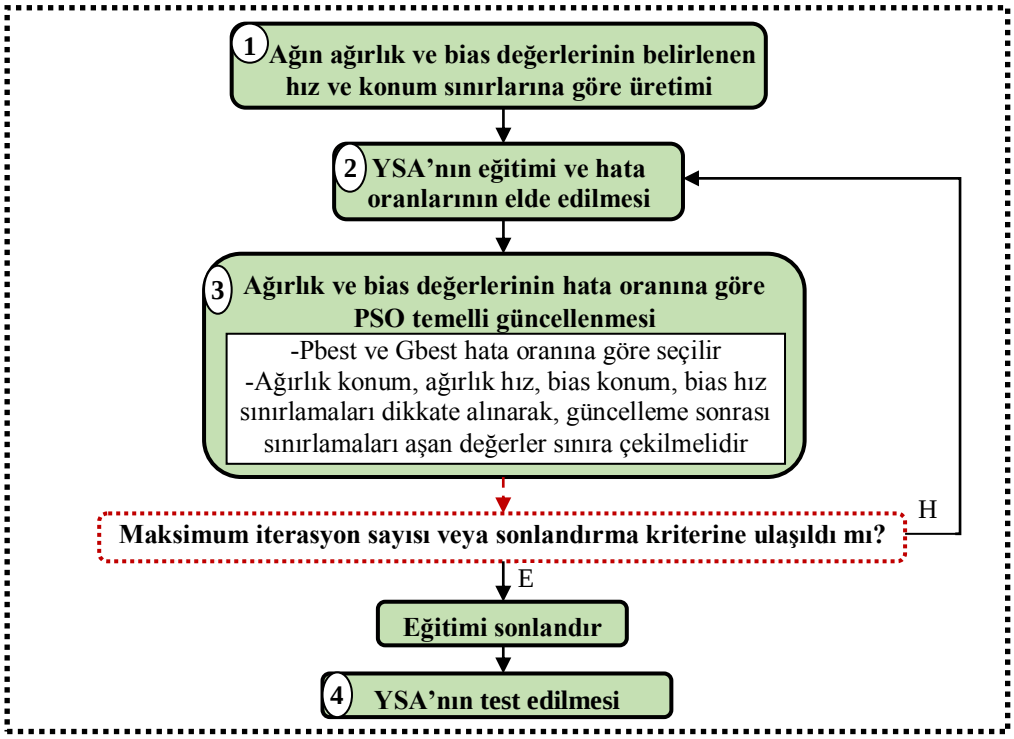
Tez çalışmasında ilk olarak hibrit PSO-YSA sınıflandırıcı yapısının tasarımı gerçeklenmiştir. Bu yapıda ağırlık ve bias değer atamaları doğrudan PSO algoritması ile gerçekleştirilmiştir. PSO algoritması için akış şeması Şekil 4.1' de görülebilir.



Şekil 4.1. Temel PSO algoritması

Şekil 4.1' de görüldüğü gibi; PSO tekniği ile bütün parçacıklar için değer ataması ve maliyet fonksiyonuna göre uygunluk değerlerinin elde edilmesi, elde edilen bu değerlere göre Pbest ve Gbest değerlerinin güncellenmesi, güncellenen bu değerlere göre hız ve konum güncellemeleri, atalet ağırlığının değiştirilmesi işlemleri ardışıl olarak gerçekleştirilir. Böylece parçacıkların optimum konumlara hareket etmesi sağlanır. Bu işlemler, sonlandırma kriterine veya maksimum iterasyon sayısına ulaşılıncaya kadar devam eder.

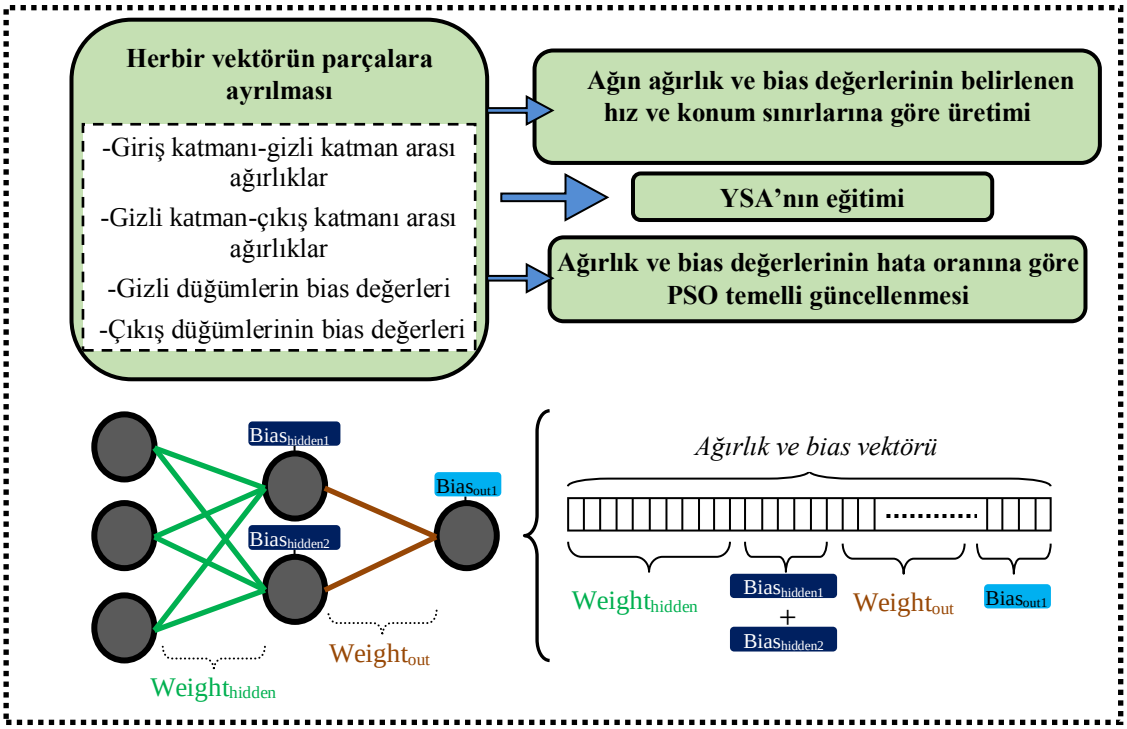
Tez çalışmasında gerçekleştirilen hibrit PSO-YSA algoritmasına ait akış şeması Şekil 4.2' de gösterilmektedir.



Şekil 4.2. Hibrit PSO-YSA akış şeması

Hibrit PSO-YSA yapısı içerisinde, ilk aşamada ağırlık ve bias vektörleri, kullanıcı tarafından belirlenen sınırlamalara göre üretilir. Literatür çalışmaları sonucunda, ağırlık ve bias değerlerinin sınırlanacağı alanların eşit seçildiği görülmüştür. Ancak, yapılan incelemeler sonunda, bias değerlerinin ağırlık değerleri ile aynı sınırlara sahip olması, bazı durumlarda sistem performansını olumsuz yönde etkilemektedir. Bu durumlarda bias sınırlarının, ağırlık sınırlarından (bu sınırlar hız ve konum sınırlarıdır) daha küçük aralıklarda tanımlanması gerektiği tespit edilmiştir. Bu nedenle, literatürde gerçekleştirildiği gibi ağırlık ve bias değerlerinin aynı sınırlar içerisinde olması yerine, farklı sınırlar seçilmesi daha uygun olduğundan Şekil 4.2' deki Hibrit PSO-YSA algoritması oluşturulmuştur. Bu amaçla ilk olarak ağırlık ve bias değerleri farklı matrisler içerisine yerleştirilmiş, bu matrislerin sınırları istenilen formatta (istenilen sınırlama aralıklarında) ayarlandıktan sonra bütün matrisler tek bir matriste birleştirilmiştir. Böylece temel popülasyon matrisi meydana getirilmiştir. Matris içerisindeki her bir vektörün (ağırlık ve bias vektörünün) sırasıyla YSA yapısına sunulması sağlanmıştır. Bu sayede, YSA yapısına sunulacak olan vektörün parçalarına ayrılması (giriş katmanı-gizli katman ile gizli katman-çıkış katmanı arası ağırlıkların ve gizli katman ile çıkış katmanında yer alan bias değerlerinin elde edilmesi) işlemi ile her

bir ağırlık ve bias matrisi için farklı sınırlamaların gerçekleştirilmesi kolaylaştırılmıştır. Bu durum Şekil 4.3' deki gibidir.



Şekil 4.3. Hibrit PSO-YSA yapısında vektörlerin parçalanması

Söz konusu YSA yapısı içerisinde hata fonksiyonu olarak, Karesel Ortalama Hata (Mean Squared Error) fonksiyonu kullanılmıştır. Elde edilen hata oranlarına göre her bir vektörün (ağırlık ve bias değerlerinin) YSA yapısında gösterdikleri performanslar karşılaştırılmıştır. İlk iterasyonda her bir vektörün Pbest değeri kendisine eşit kabul edilmiştir. Sistem içerisindeki vektörlerin adedi, popülasyon sayısı ile ayarlanmaktadır. Ayrıca vektörlerin her bir iterasyonda, bulundukları iterasyona kadar elde ettikleri en iyi konum değeri olan Pbest değerleri kaydedilmiştir. Söz konusu Pbest vektörlerinden en düşük hata oranını elde eden, popülasyonda bulunan ve o ana kadar en iyi performansı sergileyen vektör Gbest olarak atanmıştır. Daha sonra, sistemde popülasyon sayısı adedince bulunan ağırlık ve bias vektörlerinin (3.12) ve (3.13)' te belirtilen eşitliklere göre yeni değerler alması sağlanmıştır. Burada dikkat edilmesi gereken husus, vektörlerin yeniden parçalanarak üzerlerinde işlem yapılmasıdır. Buradaki temel amaç, ağırlık ve bias değerlerinin kullanıcı tarafından belirtilen sınırların dışına çıkmasının önlenmesidir.

Literatürde maskeleme olarak bilinen ve bir değişkenin belirli bir çerçeve içerisinde kalmasını öngören bu işlem ile söz konusu ağırlık-bias vektörlerinin kontrollü bir şekilde değer alması sağlanmıştır. Bu sayede ağırlık ve bias vektörlerinin sınırları aynı seçilebileceği gibi, bu sınırlar farklı da seçilebilecektir.

Kısaca, hibrit yapı içerisinde vektörlerin parçalanması işlemi, YSA yapısı girişinde ve ağırlık-bias vektörlerinin hız-konum güncellemesi esnasında olmak üzere iki kez gerçekleştirilir. Buradaki birinci amaç, vektörün YSA sistemine uygun hale getirilmesidir. İkinci amaç ise, güncelleme esnasında ağırlık ve bias katsayılarının farklı aralıklarda değişmesine (farklı sınırlamalara tabi tutulabilmesine) olanak sağlamaktır.

Gerçeklenen hibrit sınıflandırıcıya ait başlıca parametreler şu şekildedir: Ağırlık hız limitleri, ağırlık konum limitleri, bias hız limitleri, bias konum limitleri, popülasyon sayısı (pop-size), gizli düğüm sayısı (H), c_1 katsayısı, c_2 katsayısı, maksimum iterasyon sayısı (maxiter) ve atalet ağırlığı (inertia_w).

Ağırlık hız limiti, ağırlık vektörlerinin adım büyüklüğü sınırlarını ifade etmektedir. Diğer bir deyişle, ağırlık değerlerinin ileri veya geri yönde yapacakları hareketi sınırlayan çerçevedir.

Ağırlık konum limiti, ağırlık vektörlerinin sahip oldukları konum değerine adım büyüklüğü eklenmeden önce veya sonra sahip olacakları maksimum konum sınırlarını ifade etmektedir. Yani, ağırlık değerlerinin istenmeyen değerlere kaymasını önleyen bir mekanizmadır.

Bias hız limiti, bias vektörlerinin adım büyüklüğü sınırlarını ifade etmektedir. Diğer bir deyişle, bias değerlerinin ileri veya geri yönde yapacakları hareketi sınırlayan çerçevedir.

Bias konum limiti, bias vektörlerinin sahip oldukları konum değerine adım büyüklüğü eklenmeden önce veya sonra sahip olacakları maksimum konum sınırlarını ifade etmektedir. Yani bias değerlerinin istenmeyen değerlere kaymasını önleyen bir mekanizmadır.

Popülasyon sayısı, hibrit sistem içerisindeki ağırlık ve bias vektörlerinin sayısını ifade etmektedir. Kısaca, sistem içerisinde kaç adet PSO-YSA yapısının karşılaştırılacağını belirtir.

Gizli düğüm sayısı, YSA bölümünde bulunan gizli katman içerisinde kaç adet düğümün kullanılacağını belirten parametredir. Gizli düğüm sayısının artırılmasına ihtiyaç duyan sistemler komplike sistemler iken, bu sayının düşük kaldığı sistemler basit sistemlerdir.

c_1 ve c_2 katsayıları öğrenme faktörleridir. Bu katsayılar parçacıkları Pbest ve Gbest konumlarına doğru yönlendiren sabitlerdir.

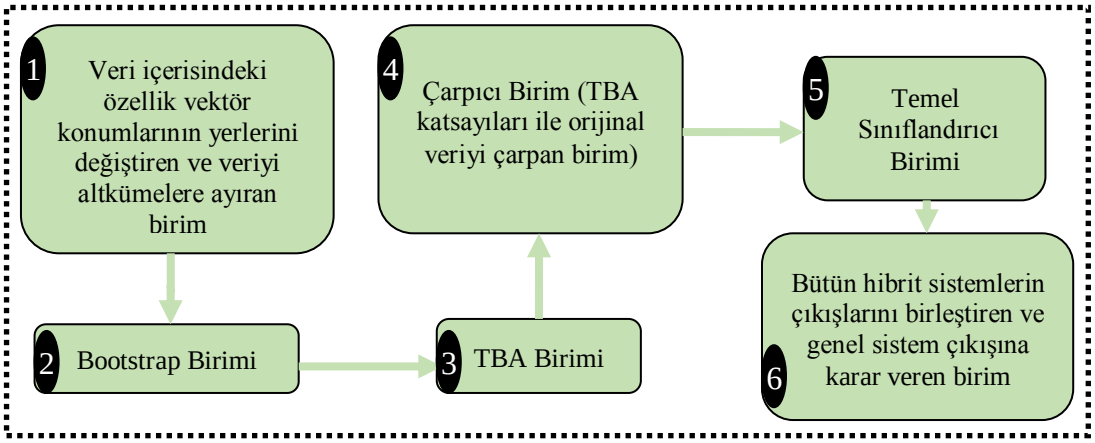
c_1 ve c_2 katsayıları için düşük değerler seçilmesi, parçacıkların hedef bölgeye doğru çekilmeden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin es geçilmesine sebep olabilir. Genel olarak c_1 ve c_2 katsayılarının toplamı 4 değerinden küçük veya bu değere eşit olmalıdır.

İterasyon sayısı, hibrit sistem içerisindeki optimum hata oranının elde edilme süresi için bir limit niteliğindedir.

Atalet ağırlığı, hibrit sistem içerisindeki ağırlık ve bias değerlerinin adım büyüklüklerini sınırlayan bir katsayıdır.

4.2. RO (Hibrit PSO-YSA) Yapısı

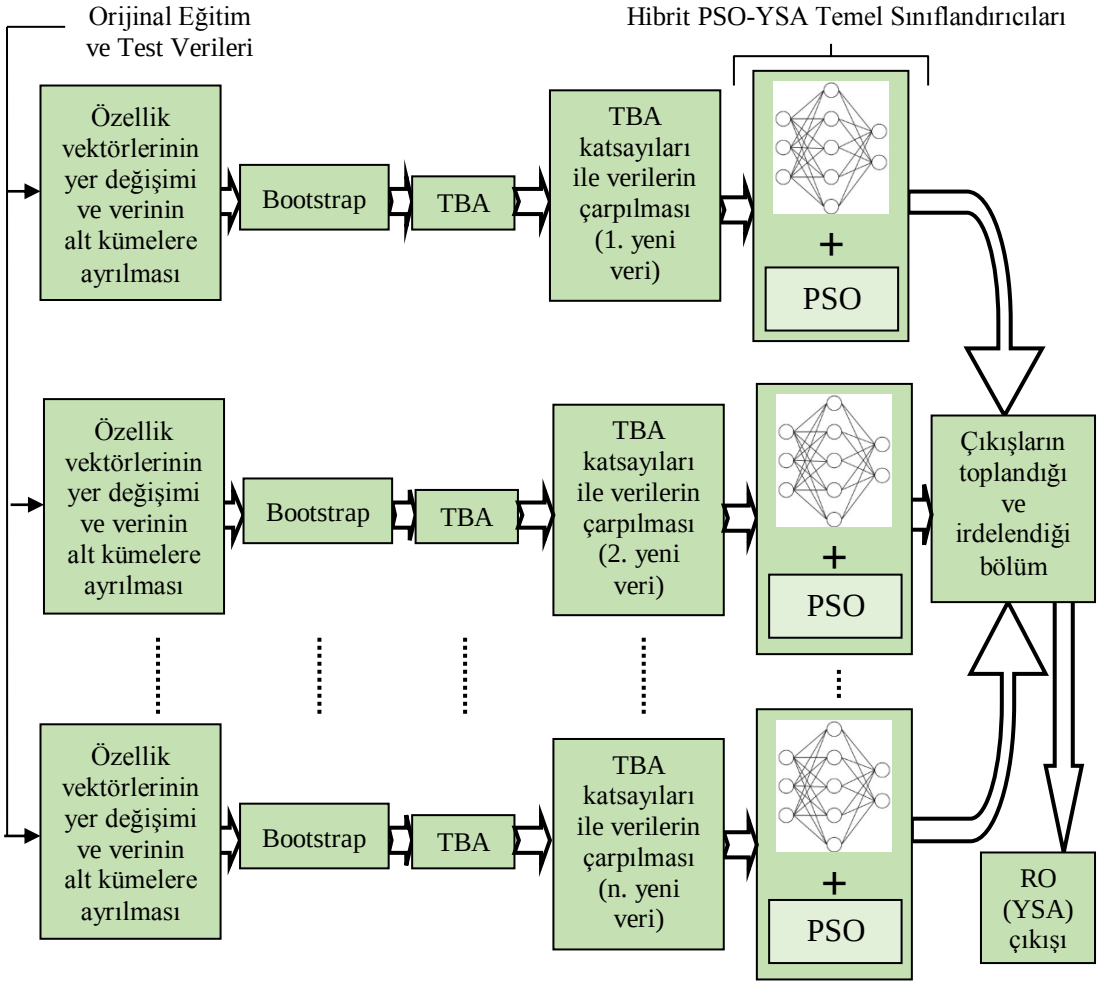
Tez çalışmasında önerilen bir diğer sınıflandırıcı ise RO (hibrit PSO-YSA) yapısıdır. Rotasyonel Orman tekniğinin algoritması ve RO (hibrit PSO-YSA) yapısı sırasıyla Şekil 4.4 ve Şekil 4.5' de görüldüğü gibidir.



Şekil 4.4. Rotasyonel Orman tekniğinin algoritması

Şekil 4.4' de görüldüğü gibi, RO tekniği 6 bölümden oluşmaktadır. İlk birimde, üzerinde işlem yapılacak olan veri seti içerisindeki özelliklerin sırası bir bütün olarak değiştirilir. Ardından verinin altkümelere ayrılması sağlanır. Bu sayede *bootstrap* işleminin uygulanacağı altkümeler elde edilir. İkinci birim, elde edilen altkümeler

içerisindeki örüntülerin sırasını değiştiren ve verinin belirli oranda azaltılmasını sağlayarak çeşitlilik faktörünü artıran bootstrap birimidir. Örüntü sırası değiştirilmiş ve örüntü sayısı azaltılmış altkümelerin özvektörlerini elde eden birim ise üçüncü birim olan TBA birimidir. Elde edilen katsayıların ilk baştaki özellik sırasına göre tekrar ayarlanmasını sağlayan ve bu katsayıları orijinal veri ile çarparak iyileştirilmiş veri elde eden birim, çarpıcı birimdir. İçerisine iyileştirilmiş veriyi alarak işleyen ve bir sınıflandırma işlemi yaparak çıkış üreten birim, temel sınıflandırıcı birimidir. Son birim olan yapı içerisinde, bütün temel sınıflandırıcılardan gelen sonuçlar irdelenerek ortak bir çıkış elde edilmesi sağlanır. Bu nedenle bu birim, karar veren birimdir.



Şekil 4.5. hibrit sınıflandırıcılı RO (PSO-YSA) yapısının algoritması

Şekil 4.5' de görüldüğü gibi, RO (hibrit PSO-YSA) yapısı, Şekil 4.4' de gösterilen her bir birimin temel sınıflandırıcı sayısı kadar tekrarlanması ve temel sınıflandırıcı bölümüne hibrit PSO-YSA yapısının eklenmesi ile meydana gelmektedir.

Her bir temel sınıflandırıcı, birbirinden farklı düzene sahip olan iyileştirilmiş veri setleri ile işleme tabi tutulmaktadır.

Söz konusu iyileştirme işlemi, özellik vektörlerinin sıralarının değiştirilmesi ve veri setinin altkümelerine ayrılması ile başlar. Bu aşamayı bootstrap işlemi takip eder. Bootstrap işlemi ile altkümelerin içerisindeki örüntüler rasgele olarak seçilir ve seçilen veri matrisi üzerinden belirli bir oran ile (RO tekniğinde bu oran %75' tir) veri sayısı azaltılır. Oluşturulan alt veri kümelerine TBA tekniğinin uygulanması ile elde edilecek katsayılar, orjinal veri matrisi ile çarpılarak iyileştirme işlemi sonlandırmaktadır. Burada dikkat edilmesi gereken en önemli aşamalardan bir tanesi de elde edilen katsayıların, orjinal veri seti içerisindeki özellik sırasına uygun bir şekilde, yeniden düzenlendikten sonra orjinal veri seti ile çarpılmasıdır.

Kısaca, orjinal veriden esinlenerek elde edilen ve birbirinden farklı olan (iyileştirilmiş) veriler, her bir hibrit temel sınıflandırıcı sisteme sunularak çeşitli sınıflandırma sonuçları elde edilir. Sınıflandırma sonuçlarının elde edilmesinin ardından son aşama olan ve karar veren bölüme geçilir. Bu aşamada bütün sınıflandırma sonuçları bir havuzda toplanarak elde edilen ortak karar üzere RO (hibrit PSO-YSA) çıkışı üretilmiş olur.

RO tekniğinde sistem çıkışını etkileyen en önemli parametre temel sınıflandırıcı sayısıdır. Bu parametre, sistem içerisinde kaç adet iyileştirme işlemi olacağını ve buna bağlı olarak kaç adet temel sınıflandırıcı kullanılacağını belirlemektedir. Bir diğer parametre ise, veri setinin kaç adet altkümeye bölüneceğidir. Ancak yapılan çalışmalarda bu sayının, sistem çıkışında dikkate değer bir değişim sağlamadığı tespit edilmiştir.

Tez çalışmasındaki RO temelli sınıflandırıcılar, Kuncheva ve Rodriguez (2006) tarafından önerilen algoritma üzerine oluşturulmuştur.

4.3. Elde Edilen Sonuçlar

Tez çalışmasında hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının performansının değerlendirilmesi amacı ile Wisconsin Breast Cancer Database verileri kullanılmıştır. Bu veriler *Physionet Database'* den alınmıştır (Erişim Tarihi: 20 Temmuz 2012). Wisconsin Breast Cancer (WBC) dataseti içerisindeki özellikler Çizelge 4.1' deki gibidir.

Hibrit PSO-YSA yapısı için sınıflandırma işlemi, 2 farklı istatistiksel yöntem ile gerçekleştirilmiş ve sonuçlar değerlendirilmiştir. Bu yöntemler, eğitim ve test verisi olmak üzere iki adet veri seti ile sistem performansının ölçülmesi temeline dayanmaktadır.

Çizelge 4.1. WBC dataseti içerisindeki özellik bilgisi

Numara	Özellik Bilgisi
1	Clump thickness
2	Uniformity of cell size
3	Uniformity of cell shape
4	Marginal adhesion
5	Single epithelial cell size
6	Bare nuclei
7	Bland chromatin
8	Normal nucleoli
9	Mitoses

Birinci istatistiksel yöntemde eğitim verisi 342 adet örüntüden oluşmaktadır. Eğitim verisi içerisindeki ilk 222 veri sağlıklı veri olup, kalan 120 veri hastalıklı veridir. Test verileri 341 örüntüden oluşturulmuştur. Bu verilerin ilk 222 tanesi sağlıklı veri olup, sonraki 119 veri hastalıklı verilerdir.

Diğer sınıflandırma yöntemi ise literatürde k kat çapraz geçerlilik testi (*k-fold cross validation*) olarak bilinen yöntemdir. Burada k değeri, metodun çevrim sayısını belirtmektedir. k değerinin literatürde sıklıkla 10 olarak alınması ve tasarlanan sistemin diğer tekniklerle karşılaştırılabilmesi amacına uygun olarak, tez çalışmasında bu değer 10 olarak alınmıştır. 10-kat çapraz geçerlilik yönteminin temel prensipleri; sahip olunan veri setini 10 (yaklaşık) eşit parçaya bölmesi (10-kat olduğu için), bölünen her bir alt veri kümesini sırasıyla test işlemi için kullanması ve bu işlem esnasında test verisi dışındaki alt veri setlerini (geriye kalan 9 parçalık veri setini) eğitim işlemi için ayırmasıdır. Gerçeklenen sınıflandırıcı, veri setinin 10 (yaklaşık) eşit parçaya bölünmesi sayesinde 10 kere eğitim ve test işlemine tabi tutulacaktır. Böylece elde edilecek 10 adet sınıflandırma doğruluğunun ortalaması alınarak, gerçeklenen sınıflandırıcının performansı daha hassas ve daha doğru bir şekilde ölçülecektir.

RO (Hibrit PSO-YSA) yapısı için sınıflandırma işlemi, literatüre göre bir performans değerlendirmesi sunabilmek adına, yalnızca 10-kat çapraz geçerlilik yöntemi ile gerçeklenmiştir.

4.3.1. Hibrit PSO-YSA yapısı ile elde edilen sonuçlar

Gerçeklenen hibrit PSO-YSA sınıflandırıcısı, ilk aşamada birinci istatistiksel yöntem ile test edilmiştir. Yani eğitim aşamasında 342 adet örüntü kullanılırken, test aşamasında 341 örüntü kullanılmıştır.

Önceden belirtildiği gibi, ağırlık ve bias vektörlerinin limit değerlerinin bazı durumlarda farklı olması gerekmektedir. 1. test yöntemi ile gerçekleştirilen denemelerde bu sınırlamalar önemli olmamakta, sınırların farklı seçilmesi sistemin çıkışını olumlu etkilememektedir. Ancak 1. test yöntemi içerisinde eğitim ve test işlemi sürekli aynı örüntülerle yapılmaktadır. Diğer bir deyişle, sistem sürekli olarak aynı şartlarda kendini zorlamaktadır. Bu yüzden ağırlık ve bias sınırlarındaki farklılıkların çıkışta oluşturacağı etki açık bir şekilde görülememektedir. Ağırlık ve bias vektörlerindeki değişimin çıkışa etkisini daha açık bir şekilde görmek amacıyla, ikinci aşamada 10-kat çapraz geçerlilik test yöntemi üzerine denemeler gerçekleştirilmiştir.

İlk aşamada 1. test yöntemi için, ağırlık konum limiti (*x*), ağırlık hız limiti (*v*), bias konum limiti (*bx*), bias hız limiti (*bv*) aralıkları değiştirilerek, hibrit sistemin elde ettiği sınıflandırma doğrulukları tespit edilmiştir.

Söz konusu denemeler gerçekleştirilirken sabit bir şablona bağlı kalınmamaktadır. Yani, sınırların büyümesi ve küçülmesinin çıkışta oluşturduğu etki temel alınarak, ağırlık ve bias limit değerleri değiştirilecektir. Bu nedenle, farklı ağırlık konum temelli denemelerde (ağırlık konum limitine bağlı olarak değiştirilen denemelerde) sabit bir taslak bulunmamakta, denemeler *değişimlerin çıkışa etkisi* ve *deneme-yanılma* ile tespit edilmektedir. Diğer bir deyişle, ağırlık konum limit değeri dışındaki limit değerleri çıkış odaklı değiştirilmektedir.

Bu aşamada popülasyon sayısı (*pop-size*) 5 olarak belirlenmiştir. Çünkü popülasyon sayısının bu değerden büyük seçilmesi sistemin sonuca ulaşmasında aşırı zaman kaybına neden olmakta, sayının bu değerden küçük seçilmesi ise sistem çıkışının olumsuz etkilenmesine (daha düşük sınıflandırma doğruluğuna) sebep olmaktadır.

Maksimum iterasyon sayısı (*maxiter*) bütün denemelerde 500 olarak sabitlenmiştir. Çünkü bu değerden sonraki iterasyon değerlerinde (1000, 1500, 2000) sistem çıkışında kayda değer bir değişim gözlenmemekte, çıkış sürekli aynı sonucu vermektedir.

c_1 ve c_2 katsayıları, literatürde önerildiği gibi birbirine eşit ve 2 değerine sahiptir.

Atalet ağırlığı (*inertia_w*) ise, yine literatürde önerildiği gibi 0.9 değerinden 0.4 değerine doğru iniş hareketi yapacaktır.

Gizli düğüm sayısı (H) 10 olarak seçilmiştir.

Ağırlık konum limiti (*x*), belirli aralıklarda sabit tutularak, ağırlık hız limiti, bias konum limiti ve bias hız limiti değiştirilmiştir. Bu durum Çizelge 4.2' de gösterilmektedir.

Çizelge 4.2' e göre, bütün sınırlamalar içinde elde edilen en optimum sınıflandırma doğruluğu %98.53 olarak elde edilmiştir. Bu sonucun elde edildiği parametreler:

- H=10
- Pop-size=5
- İnertia_w=0.9→0.4
- c_1=c_2=2
- maxiter=500
- x(-100,100)
- v(-80,80)
- bx(-100,100)
- bv(-80,80)

şeklindedir.

Önerilen sınıflandırıcı ile pekçok deneme yapılmış, ancak tez çalışmasında en önemlileri sunulmuştur.

Bir sonraki adımda optimum gizli düğüm sayının tespiti yapılmıştır. Yukarıda belirtilen optimum parametre değerleri sabit tutulmuş, gizli düğüm sayısı değiştirilerek sistemin sınıflandırma performansı ölçülmüştür.

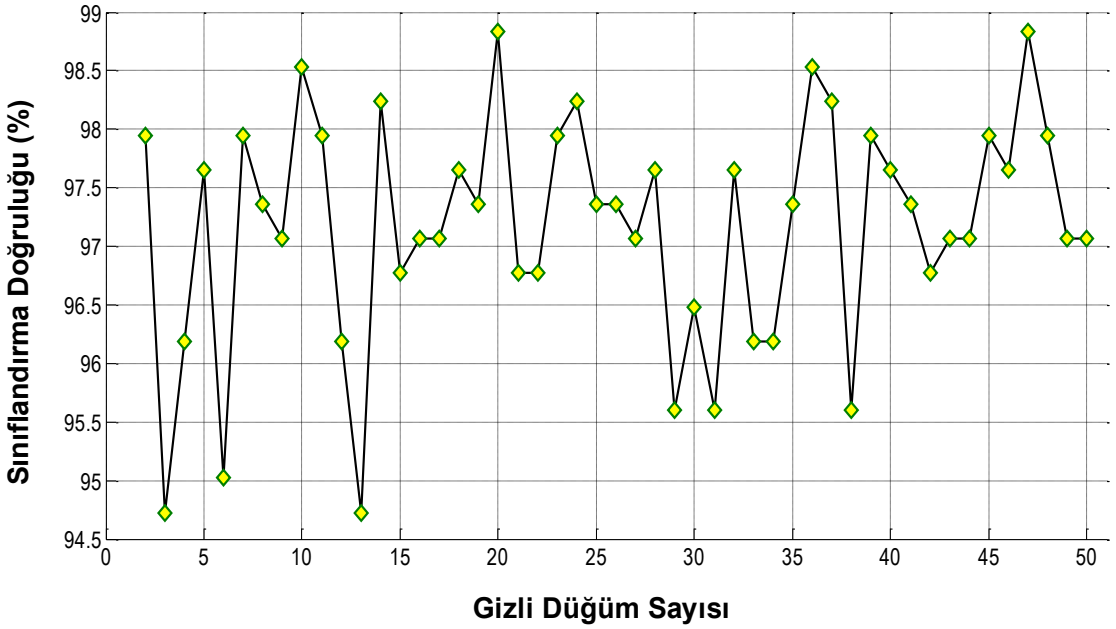
Gizli düğüm sayısı 2' den 50' ye kadar lineer bir şekilde artırılmış olup, hibrit sistemin ürettiği sınıflandırma doğrulukları tespit edilmiştir. Elde edilen en yüksek doğruluk oranı %98.83 olup, bu değer gizli düğüm sayıları sırasıyla 20 ve 47 olduğunda elde edilmiştir. Kısaca tasarlanan sistem, göğüs kanseri verileri üzerine teşhis işlemi

esnasında yüksek gizli düğüm sayılarına ihtiyaç duymakta, karmaşık bir ağ yapısı ile istenilen performansı vermektedir.

Çizelge 4.2. Parametre sınırlarındaki değişimin sınıflandırma doğruluğuna etkisi (maxiter=500, H=10, pop-size=5, c_1=2, c_2=2, inertia_w=0.9→0.4)

Ağırlık Konum Limitleri (*x*)	Ağırlık Hız Limitleri (*v*)	Bias Konum Limitleri (*bx*)	Bias Hız Limitleri (*bv*)	Sınıflandırma Doğruluğu (%)
x(-5,5)	v(-5,5)	bx(-5,5)	bv(-5,5)	97.65
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-4,4)	97.65
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-2.5,2.5)	97.36
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-1,1)	97.65
x(-5,5)	v(-4,4)	bx(-4,4)	bv(-3.2,3.2)	97.65
x(-5,5)	v(-4,4)	bx(-3,3)	bv(-3,3)	97.65
x(-5,5)	v(-4,4)	bx(-3,3)	bv(-2.4,2.4)	97.65
x(-5,5)	v(-4,4)	bx(-1,1)	bv(-0.8,0.8)	97.07
x(-5,5)	v(-4,4)	bx(-1,1)	bv(-0.5,0.5)	97.65
x(-5,5)	v(-4,4)	bx(-1,1)	bv(-0.1,0.1)	97.36
x(-5,5)	**v(-1,1)**	**bx(-5,5)**	**bv(-4,4)**	**98.24**
x(-5,5)	**v(-1,1)**	**bx(-5,5)**	**bv(-2.5,2.5)**	**98.24**
x(-5,5)	v(-1,1)	bx(-5,5)	bv(-1,1)	97.95
x(-10,10)	v(-1,1)	bx(-10,10)	bv(-1,1)	95.60
x(-10,10)	v(-1,1)	bx(-10,10)	bv(-5,5)	96.48
x(-10,10)	v(-1,1)	bx(-10,10)	bv(-8,8)	96.19
x(-10,10)	v(-1,1)	bx(-5,5)	bv(-0.5,0.5)	93.55
x(-10,10)	v(-1,1)	bx(-1,1)	bv(-0.1,0.1)	91.20
x(-10,10)	**v(-2,2)**	**bx(-10,10)**	**bv(-5,5)**	**97.65**
x(-10,10)	v(-5,5)	bx(-10,10)	bv(-5,5)	96.77
x(-10,10)	v(-5,5)	bx(-5,5)	bv(-4,4)	97.07
x(-10,10)	v(-10,10)	bx(-10,10)	bv(-10,10)	95.31
x(-50,50)	v(-50,50)	bx(-50,50)	bv(-50,50)	96.48
x(-50,50)	**v(-40,40)**	**bx(-50,50)**	**bv(-40,40)**	**96.77**
x(-50,50)	**v(-40,40)**	**bx(-50,50)**	**bv(-10,10)**	**96.77**
x(-50,50)	v(-40,40)	bx(-50,50)	bv(-5,5)	96.19
x(-50,50)	v(-40,40)	bx(-10,10)	bv(-1,1)	94.13
x(-50,50)	v(-40,40)	bx(-5,5)	bv(-1,1)	93.26
x(-100,100)	v(-100,100)	bx(-100,100)	bv(-100,100)	97.65
x(-100,100)	v(-90,90)	bx(-100,100)	bv(-90,90)	97.65
x(-100,100)	**v(-80,80)**	**bx(-100,100)**	**bv(-80,80)**	**98.53**
x(-100,100)	v(-70,70)	bx(-100,100)	bv(-70,70)	97.07
x(-100,100)	v(-60,60)	bx(-100,100)	bv(-60,60)	95.89
x(-100,100)	v(-50,50)	bx(-100,100)	bv(-50,50)	97.36
x(-100,100)	v(-40,40)	bx(-100,100)	bv(-40,40)	97.07
x(-100,100)	v(-30,30)	bx(-100,100)	bv(-30,30)	97.65
x(-100,100)	v(-10,10)	bx(-100,100)	bv(-10,10)	96.48
x(-100,100)	v(-80,80)	bx(-90,90)	bv(-72,72)	97.95
x(-100,100)	v(-80,80)	bx(-80,80)	bv(-80,80)	97.95
x(-100,100)	v(-80,80)	bv(-80,80)	bv(-64,64)	98.24
x(-100,100)	v(-80,80)	bx(-70,70)	bv(-56,56)	97.65
x(-100,100)	v(-80,80)	bx(-50,50)	bv(-40,40)	97.65

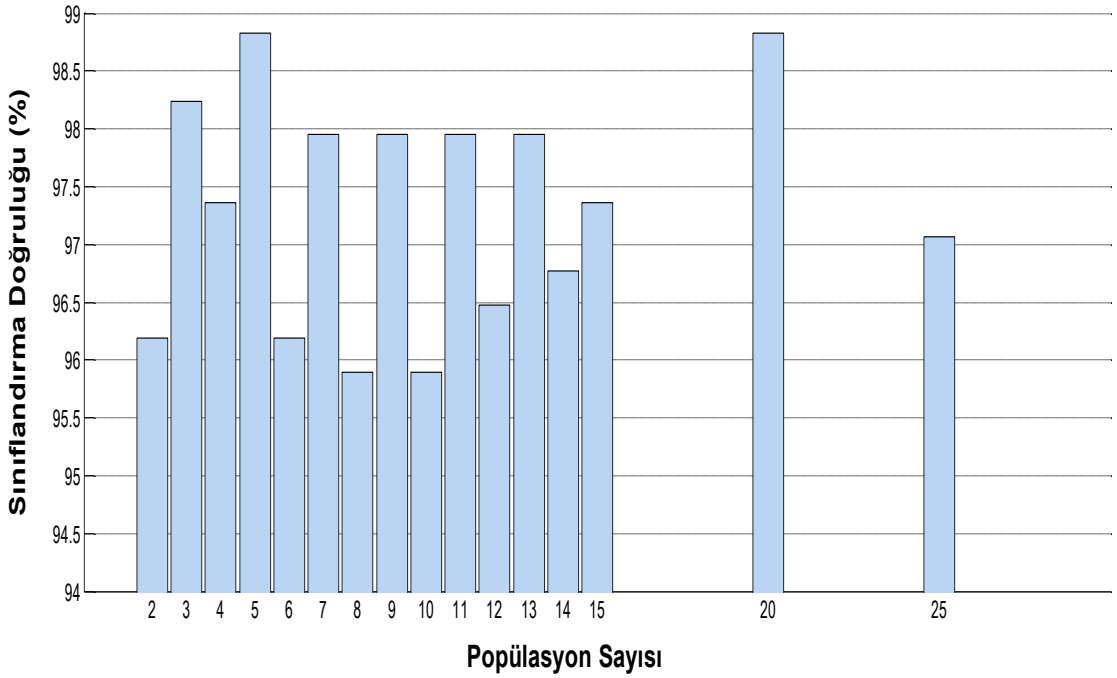
Gizli düğüm sayısındaki değişimin sınıflandırma doğruluğuna etkisi Şekil 4.6' da görüldüğü gibidir.



Şekil 4.6. Gizli düğüm sayısındaki değişimin sınıflandırma doğruluğuna etkisi (x(-100,100), v(-80,80), bx(-100,100), bv(-80,80), maxiter=500, pop-size=5, c_1=2, c_2=2, inertia_w=0.9→0.4)

Bir sonraki adımda optimum popülasyon sayısının tespiti yapılmıştır. Bu amaçla popülasyon sayısı 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20 ve 25 olarak değiştirilmiş, Şekil 4.7' deki değişim elde edilmiştir.

Yapılan incelemeler sonunda, popülasyon sayısının sürekli artırılması, sistem çıkışının sürekli iyileşmesi anlamına gelmediği kanıtlanmıştır. Popülasyon sayısının (hibrit yapıdaki ağırlık ve bias vektör sayısının) belirli değerlerde sistem çıkışını olumlu etkilediği, diğer durumlarda iyi sonuçlar sağlamadığı görülmüştür. Elde edilen en yüksek sınıflandırma doğruluğu %98.83' tür. Bu değer popülasyon sayısı sırasıyla 5 ve 20 olduğu durumlarda elde edilmiştir. Daha önceden (optimum gizli düğüm sayısı 20 olarak belirlendiğinde) elde edilen bu oran, popülasyon sayısının değiştirilmesi ile artırılmak istenmiş, ancak %98.83 değerinin ötesine geçilememiştir. Popülasyon sayısının 20 olması işlem maliyetlerini artırmaktadır. Yani sistemin çıkış verme süresi uzamaktadır. Bu nedenle, aynı sınıflandırma doğruluğunun elde edildiği ve sonuç verme süresi daha kısa olan durum optimum seçenek olacaktır. Bu seçenek, popülasyon sayısının 5 olarak seçilmesidir.



Şekil 4.7. Popülasyon sayısındaki değişimin sınıflandırma doğruluğuna etkisi (x(-100,100), v(-80,80), bx(-100,100), bv(-80,80), maxiter=500, c_1=2, c_2=2, inertia_w=0.9→0.4, H=20)

Bir sonraki adımda c_1 ve c_2 katsayıları değiştirilerek sistem çıkışı iyileştirilmeye çalışılmıştır. Diğer parametrelerin sabit tutulması kaydıyla (x(-100,100), v(-80,80), bx(-100,100), bv(-80,80), pop_size=5, maxiter=500, inertia_w=0.9→0.4 iken) gizli düğüm sayısı (H) 20 iken katsayılar değiştirilmiş, Çizelge 4.3' de verilmiştir.

Çizelge 4.3. c1 ve c2 katsayılarındaki değişimin sınıflandırma doğruluğuna etkisi (x(-100,100), v(-80,80), bx(-100,100), bv(-80,80), maxiter=500, inertia_w=0.9→0.4, pop_size=5, H=20)

c_1	c_2	Sınıflandırma Doğruluğu (%)
1,9	2,1	97.65
1,92	2,08	97.36
1,94	2,06	97.95
1,96	2,04	96.77
1,98	2,02	97.65
2	2	98.83
2,02	1,98	98.83
2,04	1,96	95.60
2,06	1,94	96.48
2,08	**1,92**	**99.12**
2,1	1,9	97.65

Elde edilen en yüksek sınıflandırma doğruluğu %99.12 olup, bu değer c_1 katsayısı 2.08 ve c_2 katsayısı 1.92 değerine sahip olduğu şartlarda kaydedilmiştir.

c_1 ve c_2 katsayıları değiştirilerek, önceki şart üzerine (c_1=2 ve c_2=2 iken) elde edilen sınıflandırma doğruluğu (%98.83) artırılarak %99.12 oranı elde edilmiştir.

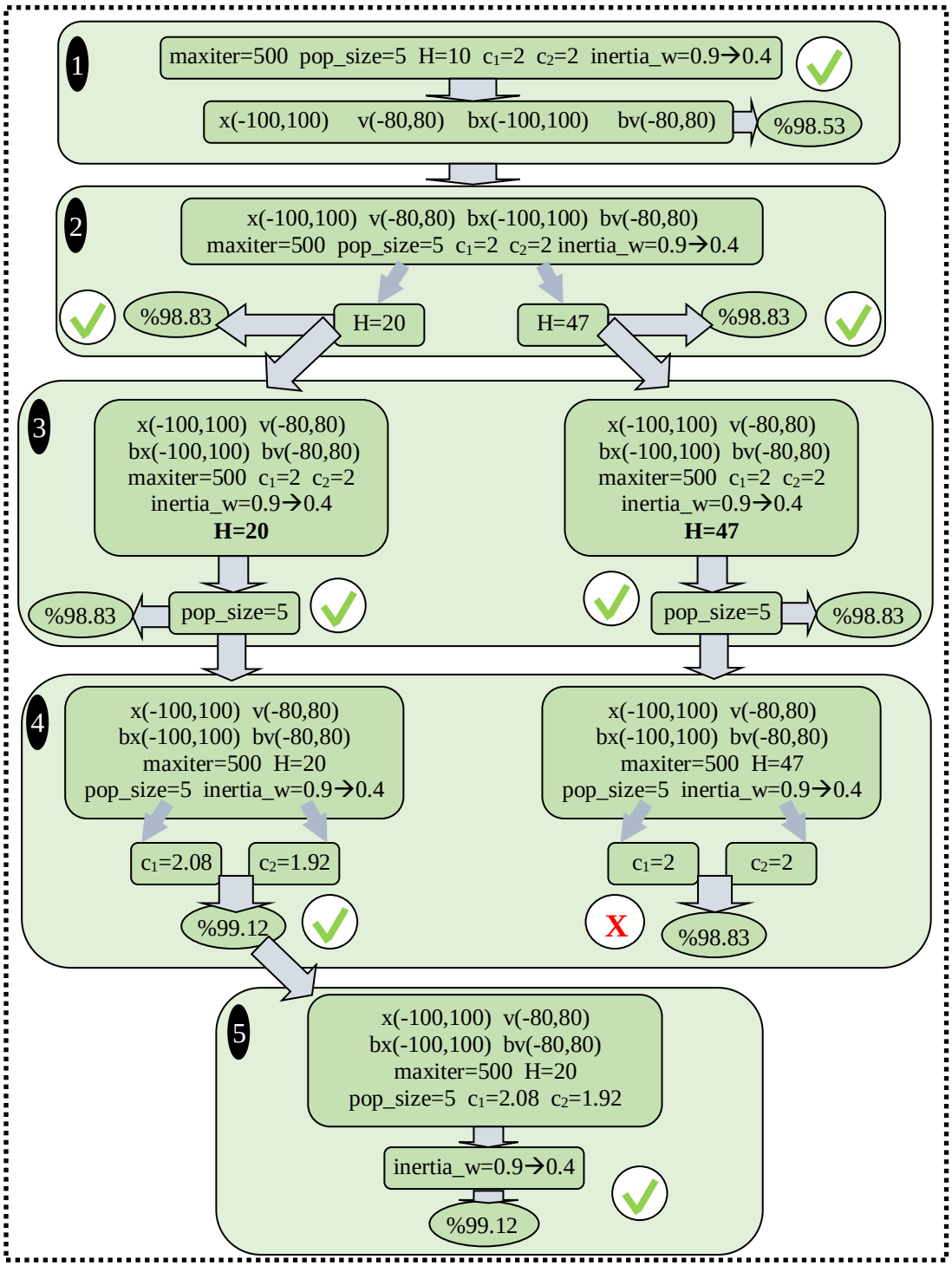
c_1 ve c_2 toplamının farklı sayılara eşit olduğu veya herhangi bir eşitliğe bağlı kalmadan rasgele değerler verilmesi gibi durumlar da gözlemlenmiş, ancak bu şartlarda istenilen performans artışı elde edilememiştir. Bunun yanı sıra, söz konusu şartlarda sistem çıkışında sürekli bir gerileme meydana geldiği için, c_1 ve c_2 katsayılarının değişimi Çizelge 4.3' deki gibi ayarlanmıştır.

Son adımda atalet ağırlığı değiştirilerek sistem çıkışı iyileştirilmek istenmiştir. Bu amaçla, atalet ağırlığı 0.1-0.9 aralığında 0.1 artımlarla değiştirilmiş, sistem çıkışı gözlemlenmiştir. Ancak sınıflandırma doğruluğu %99.12 değerinde sabit kalmıştır. Herhangi bir gelişim kaydedilemediği için, literatürde önerilen ve atalet ağırlığının 0.9 değerinden başlayarak 0.4 değerine düşürülmesini öngören parametre değişimi optimum değer olarak seçilmiştir.

Sonuç olarak, ilk test yöntemi üzerine yapılan denemelerde elde edilen en yüksek sınıflandırma doğruluğu %99.12 olup, hibrit sistem parametreleri için tespit edilen optimum değerler şu şekildedir:

- H=20
- Pop-size=5
- İnertia_w=0.9→0.4
- c_1=2.08
- c_2=1.92
- maxiter=500
- x(-100,100)
- v(-80,80)
- bx(-100,100)
- bv(-80,80)

Söz konusu optimum parametre değerlerinin elde edilişi Şekil 4.8' de bir akış diyagramı olarak verilmiştir.



Şekil 4.8. Optimum parametre değerlerinin ve sınıflandırma doğruluğunun elde edilişi

Önceden de belirtildiği gibi, ağırlık ve bias vektörlerinin limit değerlerinin bazı durumlarda farklı olması gerekmektedir. 1. test yöntemi ile gerçekleştirilen denemelerde bu sınırlamalar önemli olmamakta, sınırların farklı seçilmesi sistemin çıkışını olumlu etkilememektedir. Ancak 1. test yöntemi içerisinde, eğitim ve test işlemi sürekli aynı

örüntülerle yapılmaktadır. Diğer bir deyişle, ağırlık ve bias sınırlarındaki farklılıkların çıkışta oluşturacağı etki açık bir şekilde görülememektedir. Ağırlık ve bias vektörlerindeki değişimin çıkışa etkisini daha açık bir şekilde görmek amacıyla, ikinci aşamada 10-kat çapraz geçerlilik test yöntemi üzerine denemeler gerçekleştirilmiştir.

Bu metodun seçilmesinin önemli sebeplerinden biri, sistemin sürekli bir şekilde, farklı eğitim ve test veri parçaları ile kendini test etmesi, elde edilen sonuçların ortalaması alınarak sonuçlar genellenebilmektedir. 10-kat çapraz geçerlilik yönteminin bir diğer önemi de literatürde sıklıkla kullanılan test yöntemi olmasıdır. Yani elde edilecek sonuçların literatürdeki diğer çalışmalarla karşılaştırılması sağlanacaktır.

Söz konusu denemeler gerçekleştirilirken sabit bir şablon bulunmamaktadır. Yani 1. test yönteminde olduğu gibi, sınırların büyümesi ve küçülmesinin çıkışta oluşturduğu etki baz alınarak, limit değerleri değiştirilecektir. Bu nedenle, farklı ağırlık konum (x) temelli denemelerde (ağırlık konum limitine bağlı olarak değiştirilen denemelerde) sabit bir taslak bulunmamakta, denemeler *değişimlerin çıkışa etkisi* ve *deneme-yanılma* ile tespit edilmektedir.

Bu amaçla ağırlık konum limiti (*x*), çeşitli aralıklarda sınırlamalara tabi tutularak, ağırlık hız limiti, bias konum limiti ve bias hız limiti değiştirilmiştir. Bu aşamada ilk olarak Çizelge 4.4 elde edilmiştir.

Çizelge 4.4' de görüldüğü gibi, *x* sabit tutularak, *v, bx* ve *bv* limitleri değişken olduğunda elde edilen en iyi sınıflandırma doğruluğu %97.07 olarak elde edilmiştir. Ağırlık ve bias vektörlerinin farklı değerler almasının sistem çıkışına olan etkisi açık bir şekilde görülmektedir. Bu nedenle, hibrit bir yapı tasarlanacağında söz konusu sınırların hangi aralıklarda tanımlanacağı, dikkat edilmesi gereken en önemli unsurlardan bir tanesidir. Aynı zamanda ağırlık ve bias sınırlarının (konum ve hız değerlerinin) birbirine eşit olmasının optimum sonucu vermediği, bu sınırların her birisi için optimal aralığın bulunması gerektiği görülmüştür.

Sınıflandırma doğruluğunun artırılması amacı ile gizli düğüm sayısı 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30, 35, 40, 45, 50 ve 100 şekilde değiştirilmiş olup, hibrit sistemin ürettiği sınıflandırma doğrulukları tespit edilmiştir. Elde edilen en yüksek doğruluk oranı %97.22 olup, bu değer gizli düğüm sayısı 15 olduğunda elde edilmiştir. Kısaca tasarlanan sistem, göğüs kanseri verileri üzerine teşhis işlemi esnasında yüksek gizli düğüm sayılarına ihtiyaç duymakta, karmaşık bir ağ yapısı ile istenilen performansı vermektedir.

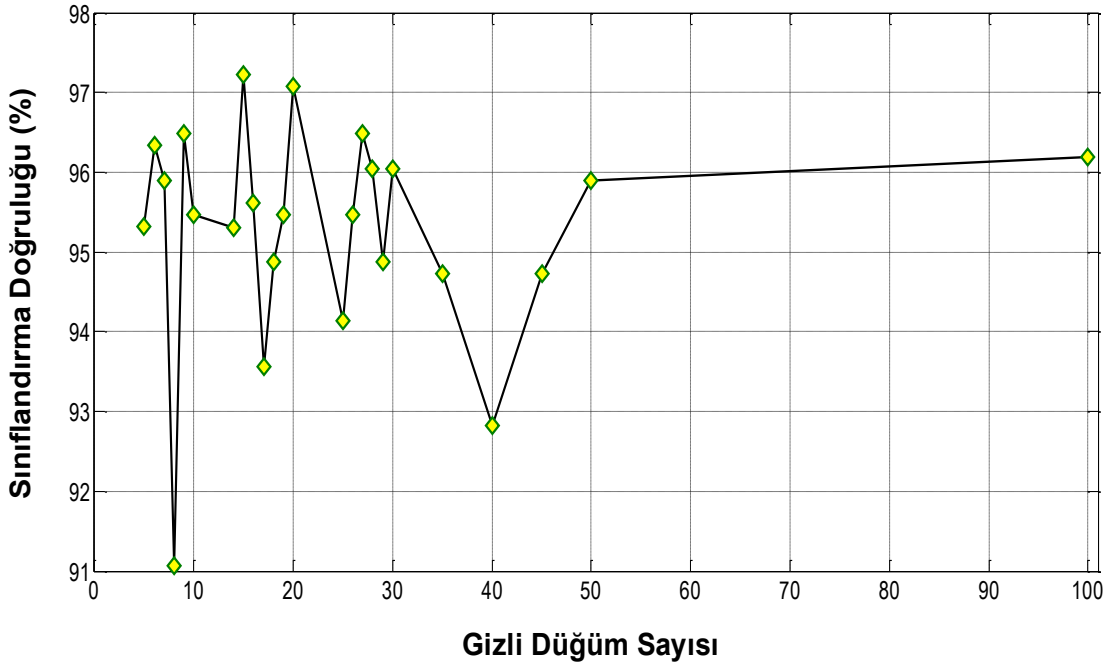
Çizelge 4.4. Parametre sınırlarındaki değişimin 10-kat çapraz doğrulamada sınıflandırma doğruluğuna etkisi (maxiter=500, H=20, pop-size=5, c_1=2.08, c_2=1.92, inertia_w=0.9→0.4)

Ağırlık Konum Limitleri (*x*)	Ağırlık Hız Limitleri (*v*)	Bias Konum Limitleri (*bx*)	Bias Hız Limitleri (*bv*)	Sınıflandırma Doğruluğu (%)
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-4,4)	96.19
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-2.5,2.5)	96.05
x(-5,5)	v(-4,4)	bx(-5,5)	bv(-0.5,0.5)	95.90
x(-5,5)	**v(-4,4)**	**bx(-4,4)**	**bv(-3.2,3.2)**	**96.34**
x(-5,5)	v(-4,4)	bx(-2.5,2.5)	bv(-0.25,0.25)	95.90
x(-5,5)	v(-4,4)	bx(-1,1)	bv(-0.1,0.1)	95.90
x(-10,10)	v(-8,8)	bx(-10,10)	bv(-8,8)	95.90
x(-10,10)	v(-8,8)	bx(-10,10)	bv(-5,5)	95.31
x(-10,10)	**v(-8,8)**	**bx(-10,10)**	**bv(-1,1)**	**96.63**
x(-10,10)	v(-8,8)	bx(-8,8)	bv(-0.8,0.8)	96.19
x(-10,10)	v(-8,8)	bx(-5,5)	bv(-2.5,2.5)	95.75
x(-10,10)	v(-8,8)	bx(-5,5)	bv(-0.5,0.5)	94.29
x(-10,10)	v(-8,8)	bx(-2,2)	bv(-0.2,0.2)	92.68
x(-10,10)	v(-8,8)	bx(-1,1)	bv(-0.1,0.1)	92.09
x(-100,100)	v(-80,80)	bx(-100,100)	bv(-80,80)	95.31
x(-100,100)	v(-80,80)	bx(-100,100)	bv(-50,50)	96.49
x(-100,100)	v(-80,80)	bx(-100,100)	bv(-10,10)	95.31
x(-100,100)	**v(-80,80)**	**bx(-80,80)**	**bv(-64,64)**	**96.93**
x(-100,100)	v(-80,80)	bx(-50,50)	bv(-5,5)	93.56
x(-100,100)	v(-80,80)	bx(-90,90)	bv(-72,72)	96.19
x(-100,100)	v(-80,80)	bx(-80,80)	bv(-64,64)	96.93
x(-100,100)	**v(-80,80)**	**bx(-70,70)**	**bv(-56,56)**	**97.07**
x(-100,100)	v(-80,80)	bx(-60,60)	bv(-48,48)	95.90
x(-100,100)	v(-80,80)	bx(-50,50)	bv(-40,40)	95.90
x(-100,100)	v(-80,80)	bx(-40,40)	bv(-32,32)	95.75
x(-100,100)	v(-80,80)	bx(-30,30)	bv(-24,24)	95.75
x(-100,100)	v(-80,80)	bx(-20,20)	bv(-16,16)	91.65
x(-100,100)	v(-90,90)	bx(-70,70)	bv(-56,56)	95.9
x(-100,100)	**v(-80,80)**	**bx(-70,70)**	**bv(-56,56)**	**97.07**
x(-100,100)	v(-70,70)	bx(-70,70)	bv(-56,56)	95.75
x(-100,100)	v(-60,60)	bx(-70,70)	bv(-56,56)	96.63
x(-100,100)	v(-50,50)	bx(-70,70)	bv(-56,56)	96.34
x(-100,100)	v(-40,40)	bx(-70,70)	bv(-56,56)	96.19
x(-100,100)	v(-30,30)	bx(-70,70)	bv(-56,56)	96.49
x(-100,100)	v(-20,20)	bx(-70,70)	bv(-56,56)	96.05
x(-100,100)	v(-10,10)	bx(-70,70)	bv(-56,56)	95.02

Gizli düğüm sayısındaki değişimin sınıflandırma doğruluğuna etkisi Şekil 4.9' da görüldüğü gibidir.

k-kat çapraz geçerlilik ile gerçekleştirilen denemelerde, popülasyon miktarının artırılması, sistem maliyetlerini çok fazla artırdığından dolayı sistemin popülasyon sayısının 5 olarak sabit tutulması uygun görülmüştür. Aksi takdirde sistem çıkışının elde

edilme süresi aşırı derecede uzamaktadır. Yani, optimum popülasyon sayısı 5 olarak atanmıştır.



Şekil 4.9. Gizli düğüm sayısındaki değişimin 10-kat çapraz doğrulamada sınıflandırma doğruluğuna etkisi (x(-100,100), v(-80,80), bx(-70,70), bv(-56,56), maxiter=500, pop-size=5, c_1=2.08, c_2=1.92, inertia_w=0.9→0.4)

Maksimum iterasyon sayısı ise bütün denemelerde 500 olarak alınmıştır. Çünkü bu parametrenin artırılması, sistem çıkışında gözle görülür bir değişime neden olmamaktadır.

Bir sonraki adımda c_1 ve c_2 katsayılarının sınıflandırma doğruluna olan etkileri gözlemlenmiştir. Bu durum Çizelge 4.5' deki gibidir.

Elde edilen en yüksek sınıflandırma doğruluğu %97.22 olup, bu değer c_1 katsayısı 2.08 ve c_2 katsayısı 1.92 değerine sahip olduğu şartlarda kaydedilmiştir. Kısaca, c_1 ve c_2 katsayılarındaki değişim, sistem çıkışında büyük oranda bir iyileşmeye neden olmamıştır.

Çizelgelerden de görüldüğü gibi c_1 ve c_2 katsayıları, toplamları 4' e eşit olacak şekilde değiştirilmiştir. Bu katsayıların toplamlarının farklı sayılara eşit olduğu ve herhangi bir eşitliğe bağlı kalmadan rasgele değerler verilmesi gibi durumlar da gözlemlenmiş, ancak söz konusu şartlarda istenilen performans artışı elde edilememiştir. Ayrıca, bu şartlarda çıkışta sürekli bir gerileme meydana geldiği için, c_1 ve c_2 katsayılarının değişimi Çizelge 4.5' deki gibi ayarlanmıştır.

Çizelge 4.5. c_1 ve c_2 katsayılarındaki değişimin 10-kat çapraz doğrulamada sınıflandırma doğruluğuna etkisi (x(-100,100), v(-80,80), bx(-70,70), bv(-56,56), maxiter=500, inertia_w=0.9→0.4, pop_size=5, H=15)

c_1	c_2	Sınıflandırma Doğruluğu (%)
1,9	2,1	96.63
1,92	2,08	96.49
1,94	2,06	96.63
1,96	2,04	96.93
1,98	2,02	96.63
2	2	96.93
2,02	1,98	95.90
2,04	1,96	96.19
2,06	1,94	96.93
2,08	**1,92**	**97.22**
2,1	1,9	95.31

Son adımda atalet ağırlığı değiştirilerek sistem çıkışı iyileştirilmek istenmiştir. Bu amaçla, atalet ağırlığı 0.1-0.9 aralığında 0.1 artımlarla değiştirilmiş, ancak sistem çıkışında bir iyileşme sağlanamamıştır. Sınıflandırma doğruluklarından en yükseği %95.9 olup, bu değer atalet ağırlığı 0.9 iken elde edilmiştir. Herhangi bir gelişim kaydedilemediği için, literatürde önerilen ve atalet ağırlığının 0.9 değerinden başlayarak 0.4 değerine düşürülmesini öngören parametre değişimi seçilecektir. Bu parametre değişimi daha önceki denemelerde gerçekleştirilmiş olup, sağlamış olduğu sınıflandırma doğruluğu %97.22' dir.

Sonuç olarak, 10-kat çapraz geçerlilik test yöntemi ile yapılan denemelerde elde edilen en yüksek sınıflandırma doğruluğu %97.22 olup, hibrit sistem parametreleri için tespit edilen en optimum değerler şu şekildedir:

- H=15
- Pop-size=5
- İnertia_w=0.9→0.4
- c_1=2.08
- c_2=1.92
- maxiter=500
- x(-100,100)
- v(-80,80)
- bx(-70,70)
- bv(-56,56)

Araştırmalardan da görüldüğü gibi, hibrit sistemin ağırlık konum, ağırlık hız, bias konum ve bias hız limit aralıkları birbirine eşit olmadığı gibi, ağırlık konum-hız ve bias konum-hız ikilileri de birbirlerine eşit değillerdir. Yani, optimum şartların sağlanması amacıyla söz konusu limit aralıklarının çok iyi ayarlanması gerektiği, limit sınırlarının sürekli eşit seçilmemesi, bu aralıkların *değişimlerin çıkışa etkisi* ve *deneme-yanılma* yöntemlerine göre oluşturulmasının uygun olacağı görülmüştür. Hibrit PSO-YSA yapısının optimum çalışma koşullarının sağlanmasında, bu sınırlamaların büyük bir etkisinin olduğu tespit edilmiştir.

Önceden de belirtildiği gibi, hibrit PSO-YSA yapısının elde ettiği en yüksek sınıflandırma doğruluğu %97.22' dir. Ayrıca bu sınıflandırma doğruluğunun elde edildiği şartlarda hibrit yapının duyarlılık (*sensitivity*) ve belirlilik (*specificity*) yüzdeleri ile hata matris (*confusion matrix)* çıkışı belirlenmiştir. Duyarlılık ve belirlilik oranı ile hata matrisini belirtmeden önce, bu kavramları kısaca açıklamakta yarar vardır.

Duyarlılık ve belirlilik, test edilmek istenen yapının istatistiksel performans ölçümüdür. Duyarlılık, doğru olarak sınıflandırılan kesin pozitif değerlerin, pozitif değerlere oranıdır. Belirlilik ise doğru olarak sınıflandırılan kesin negatif değerlerin, negatif değerlere oranıdır. Hata matrisi Çizelge 4.6' daki gibidir (Sharma ve ark., 2009).

Çizelge 4.6. Hata matrisi

		Asıl Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	Doğru Pozitif	Yanlış Pozitif
	Negatif	Yanlış Negatif	Doğru Negatif

- Doğru Pozitif (DP), hasta olup hasta olarak sınıflandırılan kişi sayısıdır.
- Yanlış Pozitif (YP), sağlıklı olup hasta olarak sınıflandırılan kişi sayısıdır.
- Yanlış Negatif (YN), hasta olup sağlıklı olarak sınıflandırılan kişi sayısıdır.
- Doğru Negatif (DN), sağlıklı olup sağlıklı olarak sınıflandırılan kişi sayısıdır.

Duyarlılık ve belirlilik, sırasıyla (4.1) ve (4.2)' de belirtildiği gibi elde edilir (Sharma ve ark., 2009).

$$Duyarlılık\ (Sensitivity) = DP/\ (DP+YN) \tag{4.1}$$

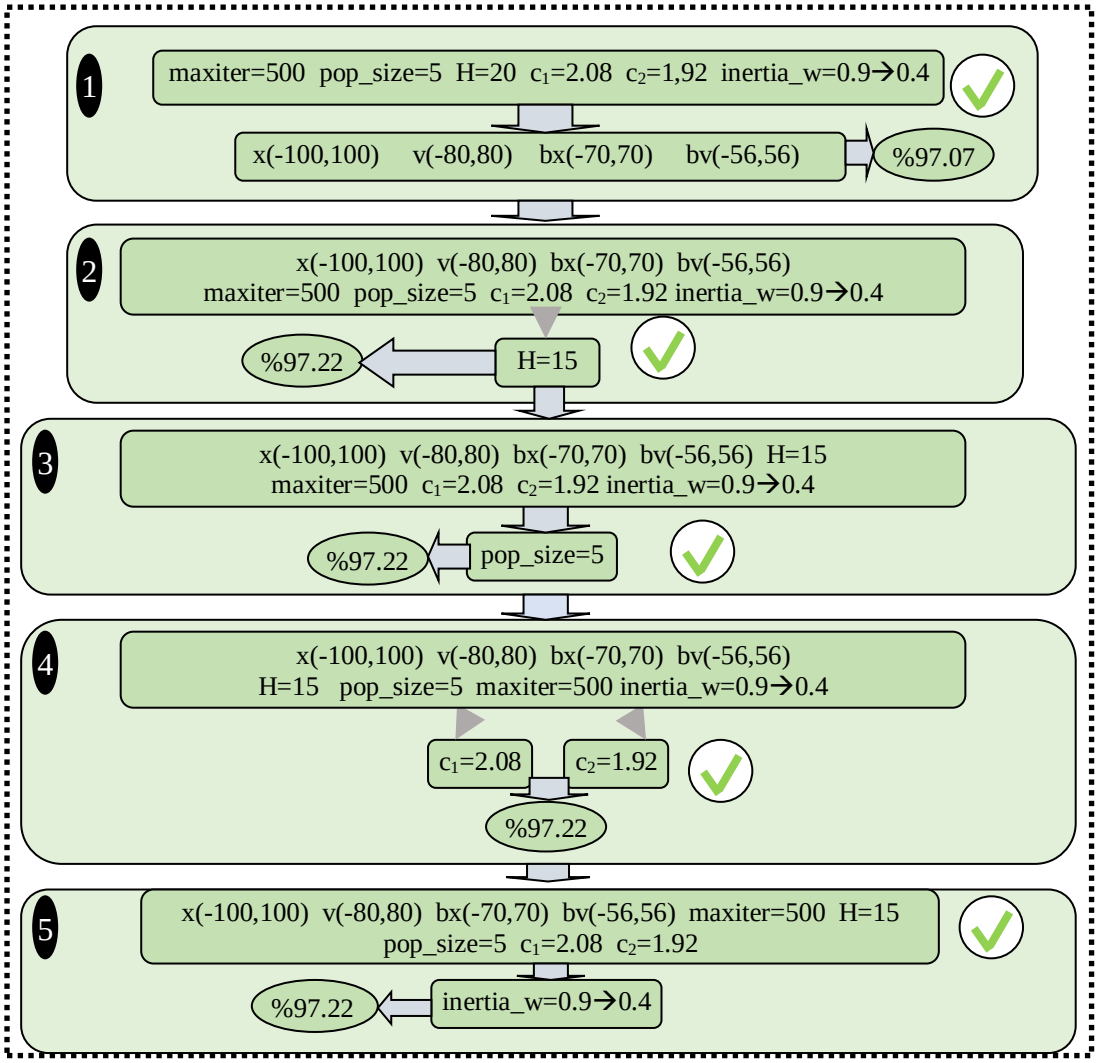
$$Belirlilik\ (Specificity) = DN/\ (DN+YP) \tag{4.2}$$

Hibrit sistemin duyarlılık oranı %99.16 olup, belirlilik oranı %96.17' dir. Diğer bir deyişle, söz konusu hibrit yapı hastalıklı örüntülerin %99.16' sını doğru sınıflandırırken, sağlıklı örüntülerin %96.17' sini doğru sınıflandırmıştır. Oluşturulan hata matrisi Çizelge 4.7' deki gibidir.

Çizelge 4.7. Elde edilen hata matrisi

		Asıl Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	**237**	**17**
	Negatif	**2**	**427**

Hibrit yapı için optimum parametre değerlerinin elde ediliş adımlarını gösteren akış şeması Şekil 4.10' da verilmiştir.



Şekil 4.10. Optimum parametre değerlerinin ve sınıflandırma doğruluğunun elde edilişi

4.3.2. RO (Hibrit PSO-YSA) yapısı ile elde edilen sonuçlar

Yüksek lisans seminer çalışmasında gerçekleştirilen kaskat PSO-YSA yapısı, sınıflandırma doğruluğu olarak %96.63 değerini (10-kat çapraz geçerlilik test yöntemi ile) elde etmişti. Bu değer, gizli düğüm sayısı 10, öğrenme oranı 0.85 ve PSO yapısı içerisinde Sphere uygunluk fonksiyonu mevcut iken gerçeklenmişti. Bunun yanı sıra, RO (kaskat PSO-YSA) yapısı (10-kat çapraz geçerlilik test yöntemi ile) %96.78 sınıflandırma doğruluğu elde etmişti. Bu aşamada gizli düğüm sayısı 10, öğrenme oranı 0.85, temel sınıflandırıcı sayısı 9 ve PSO içerisinde kullanılan uygunluk fonksiyonu Sphere idi. Kısaca, kaskat PSO-YSA ve RO (kaskat PSO-YSA) yapıları sınıflandırma doğruluğu bakımından %97 değerine dahi ulaşamamışlardır.

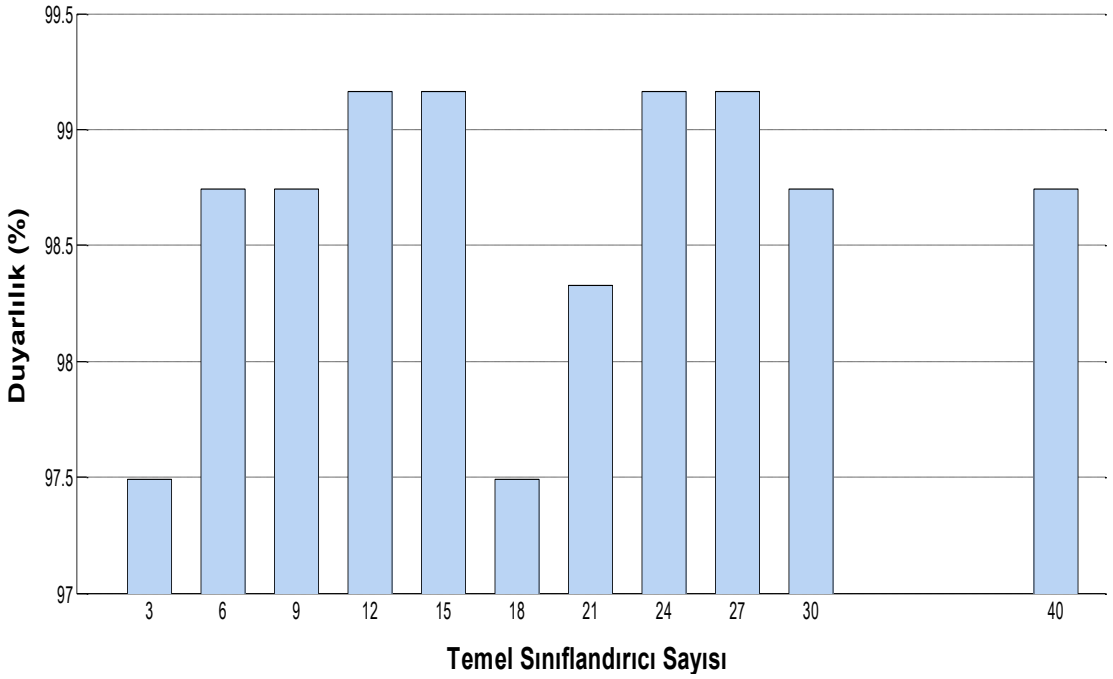
Hibrit PSO-YSA yapısı, tasarlanmasının ve parametrelerinin optimum değerlerinin tespitinin ardından, Rotasyonel Orman yapısı içerisinde temel sınıflandırıcı olarak kullanılmıştır.

Tez çalışmasında önerilen ikinci yapı olan RO (hibrit PSO-YSA) sınıflandırıcısının performansı, göğüs kanseri verisi ile değerlendirilmiştir. Bu amaçla, hibrit PSO-YSA sistemi, optimum parametreleri (x(-100,100), v(-80,80), bx(-70,70), bv(-56,56), pop_size=5, maxiter=500, c1=2.08, c2=1.92, H=15, inertia_w=0.9→0.4) sabit tutularak, RO topluluk sınıflandırıcısında temel sınıflandırıcı olarak kullanılmıştır. Diğer bir deyişle, hibrit sistemin optimum şekilde çalıştığı şartlar sağlanarak, RO (hibrit PSO-YSA) çıkışı gözlemlenmiştir. Sınıflandırma aşamasında kullanılacak olan test yöntemi, 10-kat çapraz geçerlilik testidir.

RO (hibrit PSO-YSA) yapısındaki temel sınıflandırıcı sayısı değiştirilerek, sınıflandırma doğruluğu, duyarlılık, belirlilik ve hata matrisi parametrelerine göre sistem çıkışı test edilmiştir.

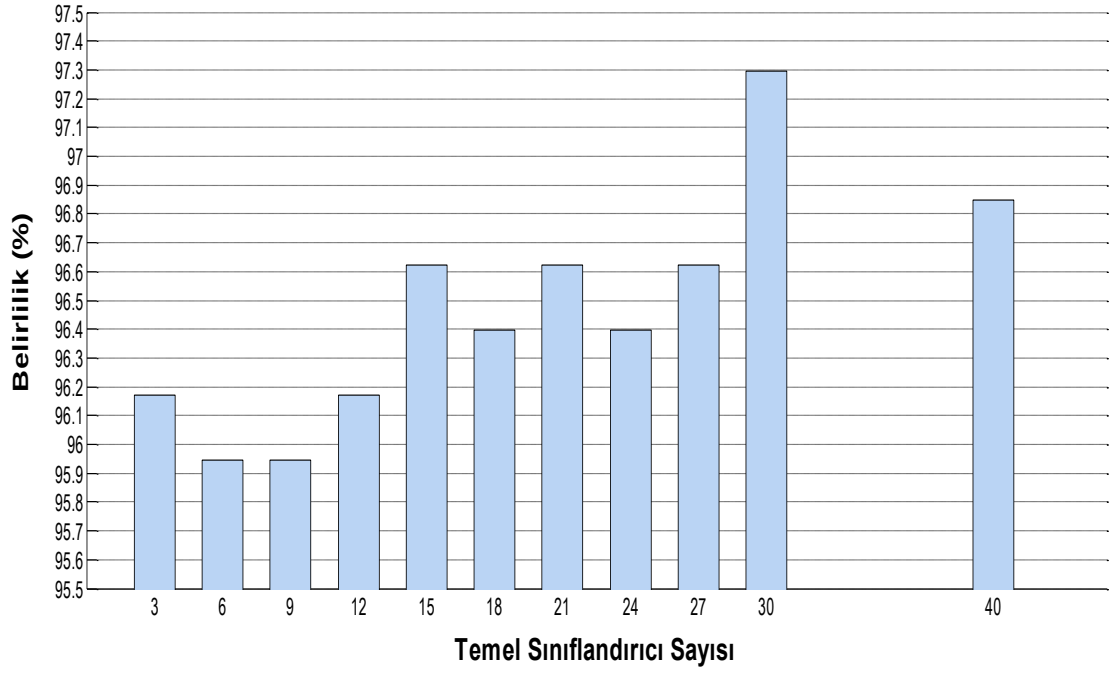
RO (hibrit PSO-YSA) yapısının temel sınıflandırıcı sayısı sırasıyla 3, 6, 9, 12, 15, 18, 21, 24, 27, 30 ve 40 olarak değiştirilmiş, Şekil 4.11' deki duyarlılık grafiği elde edilmiştir.

Şekil 4.11' e göre, elde edilen en yüksek duyarlılık oranı %99.16 olup, bu değer temel sınıflandırıcı sayısı sırasıyla 12, 15, 24 ve 27 olduğunda elde edilmiştir. Diğer bir deyişle, temel sınıflandırıcı sayısı sırasıyla 12, 15, 24 ve 27 olduğunda, hastalıklı örüntülerin %99.16' sı doğru teşhis edilmiştir.



Şekil 4.11. Temel sınıflandırıcı sayısının duyarlılık faktörüne etkisi

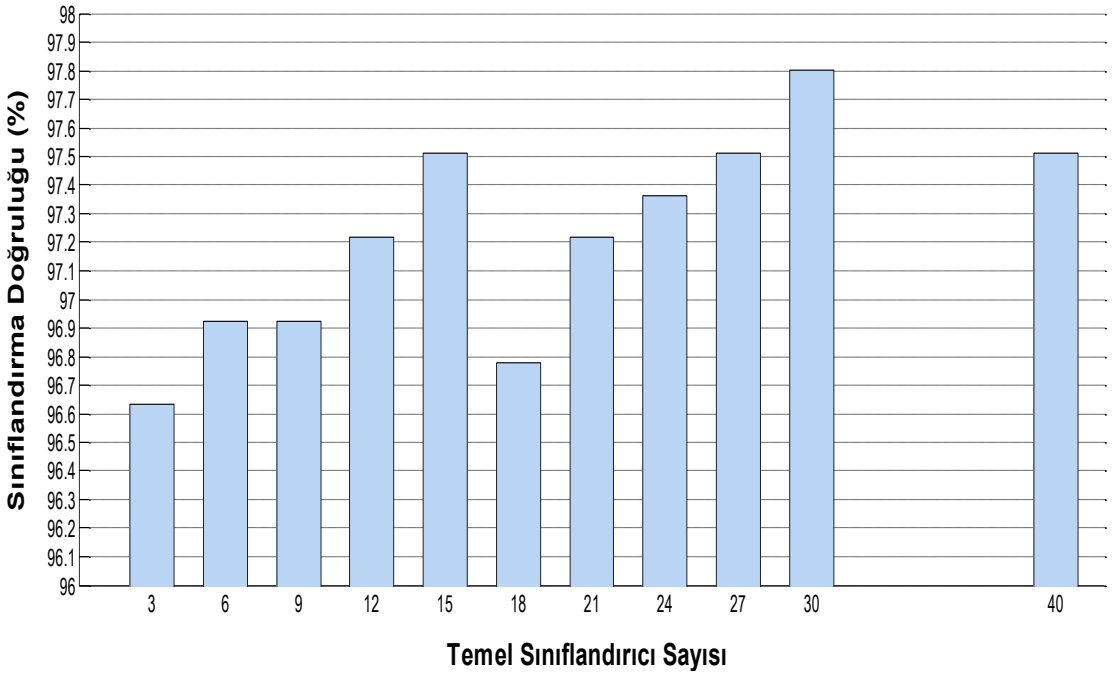
Aynı temel sınıflandırıcı sayıları için RO (hibrit PSO-YSA) yapısında elde edilen belirlilik ve sınıflandırma doğruluğu grafikleri sırasıyla Şekil 4.12 ve Şekil 4.13' da sunulmuştur.



Şekil 4.12. Temel sınıflandırıcı sayısının belirlilik faktörüne etkisi

Şekil 4.12 incelenirse, en yüksek belirlilik oranına %97.30 olarak, temel sınıflandırıcı sayısı 30 iken ulaşıldığı görülebilir. Temel sınıflandırıcı sayısı 30 olduğunda, sağlıklı örüntülerin %97.30' u doğru teşhis edilmiştir.

Şekil 4.13' de görüldüğü gibi, en yüksek sınıflandırma doğruluğuna temel sınıflandırıcı sayısı 30 iken erişilmiştir ve bu doğruluk oranı %97.80' dir. RO (hibrit PSO-YSA) yapısı, temel sınıflandırıcı sayısındaki artışa paralel bir cevap vermemektedir. Yani, temel sınıflandırıcı sayısı da optimum değeri tespit edilmesi gereken bir parametredir. Temel sınıflandırıcı sayısı 30 iken elde edilen %97.80 sınıflandırma doğruluğu, 10-kat çapraz geçerlilik yöntemi ile yapılan denemeler arasında elde edilen en yüksek sınıflandırma doğruluğudur.



Şekil 4.13. Temel sınıflandırıcı sayısının sınıflandırma doğruluğuna etkisi

Maksimum sınıflandırma doğruluğu %97.80, temel sınıflandırıcı sayısı 30 iken elde edilen hata matrisi Çizelge 4.8' de görüldüğü gibidir.

Çizelge 4.8. Temel sınıflandırıcı sayısı 30 iken elde edilen hata matrisi

		Asıl Sınıf	
		Pozitif	Negatif
Tahmin Edilen Sınıf	Pozitif	**236**	**12**
	Negatif	**3**	**432**

4.3.3. Sınıflandırıcı yapıların karşılaştırılması

Yüksek lisans seminer çalışmasında kaskat PSO-YSA ve RO (kaskat PSO-YSA) tekniklerinin göğüs kanseri verisi kullanılarak performans tespiti yapılmış, bu amaçla 10-kat çapraz geçerlilik test yöntemi kullanılmıştı. Tez çalışmasında tasarlanan hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının her biri, kaskat PSO-YSA ve RO (kaskat PSO-YSA) tekniklerinin göğüs kanseri verilerinde elde ettikleri sınıflandırma doğruluklarından daha iyi oranlar elde etmişlerdir. Bu oranlar Çizelge 4.9' da görüldüğü gibidir.

Çizelge 4.9. Dört adet sınıflandırıcının (göğüs kanseri verisi üzerine) sınıflandırma doğruluğuna göre karşılaştırılması

Sınıflandırıcılar	Sınıflandırma Doğruluğu (%)	Sonuçların Elde Edilme Süreleri
Kaskat PSO-YSA	**96.63**	**Eğitim süresi = 1.5 saat Test süresi = 32 milisaniye**
RO (kaskat PSO-YSA)	**96.78**	**Eğitim süresi = 13.5 saat Test süresi = 0.288 saniye**
Hibrit PSO-YSA	**97.22**	**Eğitim süresi = 2.5 saat Test süresi = 32 milisaniye**
RO (hibrit PSO-YSA)	**97.80**	**Eğitim süresi = 75 saat Test süresi = 0.96 saniye**

En yüksek sınıflandırma doğruluğu RO (hibrit PSO-YSA) yapısı tarafından elde edilmiş olup, bu oran %97.80' dir.

Hibrit PSO-YSA yapısı hem RO (kaskat PSO-YSA) yapısına hem de kaskat PSO-YSA yapısına üstünlük sağlamıştır.

Sonuç olarak, hibrit sınıflandırıcılar (hibrit PSO-YSA) ve bu sınıflandırıcıları içerisinde bulunduran sınıflandırıcı topluluklar (RO (hibrit PSO-YSA)), hibrit olmayan sistemlere kolaylıkla üstünlük sağlamaktadırlar. Hibrit yapıların üstünlüklerinin sebebi ise, bu sistemlerin yapay sinir ağları, destek vektör makinesi gibi yüksek performansa sahip temel sınıflandırıcıların eğitim aşamasının geliştirilmesidir. Bu nedenle hibrit sistemler, kaskat sınıflandırıcılardan daha iyi bir performans sergilemektedirler.

Tasarlamış olduğumuz hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapıları, literatürdeki simülasyon programlarından biri olan ORANGE CANVAS kullanılarak çeşitli sınıflandırıcı sistemlerle de karşılaştırılmıştır. Bu sınıflandırıcı sistemler; Sınıflandırma Ağacı (*Classification Tree*), k-En Yakın Komşu Tekniği (*k-Nearest*

Neighbours), Naive Bayes sınıflandırıcısı, CN2 sınıflandırıcısı, Yapay Sinir Ağları (gizli düğüm sayısı 15 iken), Rasgele Orman (*Random Forest*) ve Destek Vektör Makinesi (*Support Vector Machine*) şeklindedir. Tasarlanan yapıların, literatürdeki diğer çalışmalarla ve ORANGE CANVAS simülasyon programındaki sınıflandırıcı yapılarla elde edilen sonuçlar ile karşılaştırılması Çizelge 4.10' da ifade edildiği gibidir.

Çizelge 4.10. Tasarlamış olduğumuz yapıların çeşitli sınıflandırıcı sistemlerle karşılaştırılması

Çalışmalar	Sınıflandırıcılar	Sınıflandırma Doğruluğu (%)
(Abonyi ve Szeifert, 2003)	Denetimli Belirsiz Kümeleme (Supervised Fuzzy Clustering)	95.57
(Goodman ve ark., 2002)	Optimize Edilmiş Vektör Niceleme Öğrenimi (Optimized LVQ)	96.70
(Goodman ve ark., 2002)	Büyük Vektör Niceleme Öğrenimi (Big LVQ)	96.80
(Goodman ve ark., 2002)	Yapay Bağışıklık Tanıma Sistemi (AIRS)	97.20
(Salama ve ark., 2012)	SMO+J48+NB+IBk hibrit yapısı	97.28
Orange Canvas simülasyon programı ile	Sınıflandırma Ağacı (*Classification Tree*)	93.40
Orange Canvas simülasyon programı ile	Rasgele Orman (*Random Forest*) Tekniği	94.13
Orange Canvas simülasyon programı ile	CN2 Kuralları Tekniği (*CN2 Rules*)	94.87
Orange Canvas simülasyon programı ile	Yapay Sinir Ağları (*Neural Network*)	95.01
Orange Canvas simülasyon programı ile	k-en yakın komşu tekniği (*k-Nearest Neighbours*)	96.04
Orange Canvas simülasyon programı ile	Naive Bayes Ağları	96.77
Orange Canvas simülasyon programı ile	Destek Vektör Makinesi (*Support Vector Machine*)	97.21
Tez çalışması	**Hibrit PSO-YSA** (10 kat çapraz geçerlilik kullanılmamıştır)	**99.12**
Tez çalışması	**Hibrit PSO-YSA**	**97.22**
Tez çalışması	**RO (hibrit PSO-YSA)**	**97.80**

Yapay sinir ağları yapısının gizli düğüm sayısı, tasarlanan hibrit yapıdaki değerle (H=15) aynı seçilmiştir. Rasgele Orman tekniği içerisindeki ağaç sayısı, RO tekniği içerisindeki temel sınıflandırıcı sayısına (30) eşit olarak seçilmiştir. K-en yakın

komşu tekniği içerisindeki komşuluk sayısı 5 olarak seçilmiştir. Tasarlanan yapıların farklı tekniklerle karşılaştırılabilmesi için, 2. test yöntemi olarak belirtilen 10-kat çapraz geçerlilik yöntemine göre deneme işlemleri gerçeklenmiştir.

Çizelge 4.10' dan da görüldüğü gibi, hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapıları, genel olarak diğer sınıflandırıcı yapıların her birine sınıflandırma üstünlüğü sağlamışlardır. 14 adet sınıflandırıcı yapı arasında elde edilen en yüksek sınıflandırma doğruluğu, RO (hibrit PSO-YSA) yapısı tarafından elde edilmiş olup, bu değer %97.80' dir.

Hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının yazılımları, sırasıyla EK-1 ve EK-2 kısımlarında sunulmaktadır.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Tez çalışmasında hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının tasarımı gerçekleştirilmiştir.

Hibrit PSO-YSA tekniği, iki adet istatistik test yöntemi kullanılarak test edilmiştir. Hibrit PSO-YSA yapısı, göğüs kanseri verisi üzerine 1. test yöntemi sonuçlarına göre %99.12 sınıflandırma doğruluğu elde etmiştir. Hibrit yapının 10-kat çapraz geçerlilik yöntemi ile elde ettiği en yüksek sınıflandırma doğruluğu %97.22' dir.

RO (hibrit PSO-YSA) yapısı, yalnızca 10-kat çapraz geçerlilik yöntemi ile test işlemine tabi tutulmuştur. Çünkü 10-kat çapraz geçerlilik test yöntemi, literatürde çok sık kullanılan, parametre sınırlamalarının sistem çıkışına etkisinin daha rahat görülebileceği, aynı zamanda yapılan çalışmayı literatürdeki diğer çalışmalarla kolaylıkla karşılaştırabileceğimiz bir tekniktir. Bunun yanı sıra, ağırlık ve bias sınırlarındaki değişimin çıkışa etkisinin gözlenebilmesi için iyi bir seçenektir. 1. test yöntemi ile yapılan denemelerde, ağırlık ve bias limitlerindeki değişimin çıkışa olan etkisi iyi bir şekilde gözlemlenememektedir. Çünkü 1. test yöntemi, sınanması istenen yapıyı sabit eğitim ve test verileri ile denemeye tabi tutmaktadır. 10-kat çapraz geçerlilik yöntemi ise, sahip olunan veriyi 10 parçaya ayırarak, her bir parçayı test işleminde kullanmaktadır. Test dışında kalan 9 parça veri kümesi ise eğitim işleminde kullanılmaktadır.

RO (hibrit PSO-YSA) yapısı, %97.22 sınıflandırma doğruluğu elde eden hibrit PSO-YSA yapısının, RO yapısının temel sınıflandırıcı birimine entegre edilmesi ile gerçeklenmiştir. RO (hibrit PSO-YSA) yapısı, göğüs kanseri verisi üzerinde test edilmiş, elde edilen en yüksek sınıflandırma doğruluğu %97.80 olarak bulunmuştur.

YSA' da ağırlıkların güncellenmesi işlemindeki temel mantık, yapay sinir ağları yapısının, girişine sunulan veri ile istenen çıkış değerine göre bulunan optimum ağırlık katsayılarının tespitidir. Bu sayede YSA yapısı, test aşamasında girişine sunulan test verisini, eğitim işleminde tespit ettiği ağırlık-bias katsayılarını kullanarak test eder. Ancak bu noktada dikkat edilmesi gereken en önemli hususlardan bir tanesi de bias değerlerinin ağırlık değerlerini baskı altına alarak, sistem çıkışını olumsuz yönde etkileyebileceğidir. Bu amaçla, ağırlık ve bias değişkenlerinin konum-hız sınırlamaları üzerinde incelemeler yapılmıştır. 10-kat çapraz geçerlilik test yöntemi ile göğüs kanseri

verileri üzerine gerçekleştirilen denemelerde, ağırlık ve bias değişkenlerinin konum-hız sınırlamalarının sistem çıkışını önemli bir şekilde etkilediği tespit edilmiştir. Kısaca, optimum sınırların belirlenmesi işleminin sınıflandırma doğruluğuna olan etkisi incelenmiş, herbir sınırlamanın bireysel olarak incelenmesi gerektiği görülmüştür.

Tasarlanan hibrit PSO-YSA yapısındaki en önemli parametrelerin; ağırlık konum sınırı, ağırlık hız sınırı, bias konum sınırı, bias hız sınırı, gizli düğüm sayısı ile c_1 ve c_2 katsayıları olduğu tespit edilmiştir. Atalet ağırlığının 0.9 değerinden başlayarak, iterasyon sonunda 0.4 değerine düşürülerek değiştirilmesinin literatürde önerildiği (Shi ve Eberhart, 1999) ve tez çalışmasında en iyi sonucu verdiği görülmüştür. Bunun yanı sıra, optimum popülasyon sayısı 5 olarak belirlenmiştir. Bu değerden daha büyük ve daha küçük değerler seçilmesi, sistem maliyetlerini artırmaktadır. Maksimum iterasyon sayısı ise 500 değerinde sabit tutulmuştur. Çünkü hibrit yapının hata oranı, bu değerden sonra gözle görülür bir iyileşme gösterememiştir. Kısaca hibrit yapının çıkışını (maxiter, pop_size, inertia_w değişkenleri optimum değerlerinde sabit iken), ağırlık-bias sınırlarının, gizli düğüm sayısının ve c_1-c_2 katsayılarının daha etkin bir şekilde değiştirdiği tespit edilmiştir.

Yüksek lisans seminer çalışmasının ilk aşamasında gerçeklenen kaskat PSO-YSA ve RO (kaskat PSO-YSA) yapıları ile, göğüs kanseri verileri üzerine sırasıyla %96.63 ve %96.78 sınıflandırma doğruluğu elde edilmişti. Tez çalışmasında gerçekleştirilen hibrit PSO-YSA ve RO (hibrit PSO-YSA) sınıflandırıcıları ise sırasıyla %97.22 ve %97.80 sınıflandırma doğruluğu elde etmiştir.

Ayrıca RO (hibrit PSO-YSA) yapısının, hibrit PSO-YSA yapısına %0.58 oranında sınıflandırma üstünlüğü sağladığı tespit edilmiştir.

Tez çalışmasında da seminer çalışmasında olduğu gibi, birden fazla temel sınıflandırıcıyı içinde barındıran sınıflandırıcı topluluğu (RO (hibrit PSO-YSA)) ile, bir adet temel sınıflandırıcı (hibrit PSO-YSA) kullanımından daha iyi bir performans elde edildiği görülmüştür.

Bunun yanı sıra, kaskat sınıflandırıcılar yerine hibrit sınıflandırıcı tasarımlarının YSA ve YSA tabanlı RO yapıları için daha uygun olduğu, bu yapıların hibrit temelli olmasının performans bakımından büyük bir önem arz ettiği tespit edilmiştir.

5.2 Öneriler

Elde edilen sonuçlar incelendiğinde, temel sınıflandırıcı sistemleri içerisinde barındıran yapıların (RO (hibrit PSO-YSA)) veya temel sınıflandırıcının ek yapılarla güçlendirilmiş hallerinin (hibrit PSO-YSA), bir adet temel sınıflandırıcı yapısından daha yüksek bir performansa sahip olduğu görülmüştür. Bu nedenle tek bir YSA yapısı yerine, YSA yapısının temel alındığı hibrit PSO-YSA veya içerisinde birçok hibrit PSO-YSA yapısı bulunan RO (hibrit PSO-YSA) yapılarının kullanılması daha uygun olacaktır.

Ayrıca kaskat PSO-YSA ve RO (kaskat PSO-YSA) yapıları yerine, hibrit yapılara sahip olan hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının tercih edilmesinin, performans ve sınıflandırma doğruluğu bakımından daha uygun olduğu belirlenmiştir.

Ayrıca gerçekleştirilen uygulamalar sonucunda, ağırlık ve bias konum-hız limitlerinin eşit olarak seçilmemesi ve bu limitlerin her birisi için ayrı ayrı optimum aralıkların belirlenmesi gerektiği açık bir şekilde görülmektedir.

RO temelli sınıflandırıcılarda, optimum temel sınıflandırıcı sayısı tespit edilmelidir.

Tez çalışmasında önerilen hibrit PSO-YSA ve RO (hibrit PSO-YSA) yapılarının sonuçlarının genelleştirilebilmesi için, farklı biyomedikal verilerle de uygulamalar yapılması uygun olacaktır. Böylece önerilen yapıların performansları daha net bir şekilde değerlendirilmiş olacaktır.

KAYNAKLAR

Abonyi, J. and Szeifert, F., 2003, Supervised fuzzy clustering for the identification of fuzzy classifiers, Pattern Recognition Letters, 24 (14), 2195-2207.

Amasyalı, M. F. ve Ersoy, O., 2011, Performance based pruning and weighted voting with classification ensembles, *IEEE,* 194-197.

Arslan, O., Kurt, O. ve Konak, H., 2007, Yapay sinir ağlarının jeodezide uygulamaları üzerine öneriler, *TMMOB Harita ve Kadastro Mühendisleri Odası 11. Türkiye Harita Bilimsel ve Teknik Kurultayı,* Ankara, 1-10.

Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S. and Abraham, A., 2011, Inertia weight strategies in particle swarm optimization, *IEEE,* 640-647.

Barton, M. B., 2001, Screening mammography for women aged 40-49: Are we off the fence yet?, *Canadian Medical Association*, 164(4), 498-499.

Baş, N., 2006, Yapay sinir ağları yaklaşımı ve bir uygulama, Yüksek Lisans Tezi, *Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 42-45.

Bock, K. W-D. and Poel, D. V-D., 2011, An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction, *Expert systems with applications,* Lille-France, 12293-12301

Bottorff, J. L., Johnson, J. L., Bhagat, R., Grewal, S., Balneaves, L. G., Clarke, H. and Hilton, B. A., 1998, Beliefs related to breast health practices: The perceptions of south asian women living in Canada, *Social Science and Medicine,* 47(12), 145-151.

Dou, Z. and Gao, L., 2012, A particle swarm optimizer based extension classifier for biomedical document, *Journal of Computational Information Systems,* 8(1), 249-258.

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

Fei, S-W., 2010, Diagnostic study on arrhythmia cordis based on particle swarm optimization-based support vector machine, *Expert systems with applications,* 6748-6752.

Goodman, D.E., Boggess, L. and Watkins. A., 2002, Artificial immune system classification of multiple-class problems, *Prof. Of the Artificial Neural Networks in Engineering ANNIE*, 179-183.

Günal, S. Y. and Günal, A. İ., 2000, 22-55 yaş çalışan kadınların meme kanserine yönelik davranışları, *Sağlık ve Toplum*, 10(1), 21-24.

Han, M., Zhu, M. and Yao, W., 2012, Remote sensing image classification based on neural network ensemble algorithm, *Neurocomputing*, 78, 133-138.

Heppner, F. and Grenander, U., 1990, A stochastic nonlinear model for coordinated bird flocks, *The Ubiquity of Chaos. AAAS Publications*, 233-238.

Kader, H. A. and Salam, M. A., 2012, Evaluation of differential evolution and particle swarm optimization algorithms at training of neural network for stock prediction, *International Arab Journal of e-Technology*, 2(3), 145-151.

Kara, İ., 1986, Yöneylem araştırması ve doğrusal olmayan modeller, *Anadolu Üniversitesi Basımevi*, Eskişehir, 8-15.

Kennedy, J. and Eberhart, R., 1995, Particle swarm optimization, *In Proceedings of IEEE International Conference on Neural Networks*, Piscataway-New Jersey, 1942-1948.

Kok, B. C., Goh, H. H. and Chua, H. G., 2012, Optimal Power Tracker for Stand-Alone Photovoltaic system using Artificial Neural Network (ANN) and Particle Swarm Optimisation (PSO), *International Conference on Renewable Energies and Power Quality (ICREPQ'12)*, Santiago de Compostela-Spain, 1-6.

Korürek, M. ve Doğan, B., 2010, ECG beat classification using particle swarm optimization and radial basis function neural network, *Expert systems with applications*, 7563-7569.

Kotsiantis, S., 2010, Combining bagging, boosting, rotation forest and random subspace methods, *Springer Science*, 1-18.

Kruk, J. and Aboul-Enein, H. Y., 2003, Occupational physical activity and the risk of breast cancer, *Cancer Detection and Preventation*, 27, 187-192.

Kuncheva, L. I. and Rodriguez, J. J., 2006, Rotation Forest: a new classifier ensemble method, *IEEE Computer Society*, 28, 1619-1630.

Kuncheva, L. I. and Rodriguez, J. J., 2007, An experimental study on rotation forest ensembles, *Springer-Verlag*, 459-468.

Liu, K-H. and Huang, D-S., 2008, Cancer classification using rotation forest, *Computers in Biology and Medicine*, 38, 601-611.

Mandal, I. And Sairam, N., 2012, SVM-PSO based feature selection for improving medical diagnosis reliability using machine learning ensembles, *Computer Science & Information Technology*, 267-276.

Marques, A. I., Garcia, V. and Sanchez, J. S., 2012, Two-level classifier ensembles for credit risk assessment, *Expert systems with applications*, 39, 10916-10922.

Melgani, F. and Bazi, B., 2008, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, *IEEE*, 12(5), 667-677.

Metehan M., Cihan G., 2007, Göğüs kanseri verilerinin sınıflandırılması, *Elektrik Elektronik-Bilgisayar Müshendisliği 12. Ulusal Kongresi ve Fuarı.*

Nayak, S. R., Pradhan, C. R., Ali, S. M. and Sabat, R. R., 2012, Application of solar energy using artificial neural network and particle swarm optimization, *International Journal of Advances in Engineering & Technology*, 4(1), 550-560.

Neto, A. B., Nievola, J. C., Figueredo, M. V. M. and Rogal, S. R., 2010, Automatic detection of cardiac arrhythmias using wavelets, neural Networks and particle swarm optimization, *IEEE*, 194-198.

Novakovic, J. and Veljovic, A., 2011, Interpretation of mammograms with rotation forest and pca, *6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, Timişoara-Romania, 571-575.

Nowak, A., Szamrej, J. and Latane, B., 1990, From private attitude to public opinion: A dynamic theory of social impact, *Psychological Review*, 97(3), 362-376.

Ondrusek, N., Warner, E. and Goel, V., 1999, Development of a knowledge scale about breast cancer and heridity (BCHK), *Breast Cancer Research and Treatment*, 53, 69-75.

Öğücü, M. O., 2006, Yapay sinir ağları ile sistem tanıma, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü,* İstanbul, 4-5.

Özbay, Y., 1999, EKG aritmilerini hızlı tanıma, Doktora Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya, 26-38.

Özçift, A. ve Gülten, A., 2011, Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms, *Computer Methods and Programs in Biomedicine*, 104, 443-451.

Reeves, W. T., 1983, Particle systems--A technique for modeling a class of fuzzy objects, *ACM Transactions on Graphics*, 2(2), 91-108,.

Reynolds, C. W., 1987, Flocks, herds, and schools: A distributed behavioral model, *ACM Computer Graphics*, 21(4), 25-34.

Rokach, L., 2010, Pattern classification using ensemble methods, *World Scientific*, Beer Sheva-Israel, 111-112.

Salama, G. I., Abdelhalim, M. B. and Zeid, M. A-E., 2012, Breast Cancer Diagnosis on Three Different Datasets Using Multi-Classifiers, *International Journal of Computer and Information Technology*, 1 (1), 36-43.

Segura, J. M., Castells, X., Casamitjana, M., Macia, F. and Ferrer, F., 2000, Utilization of screening mammography as a preventive practice prior to initiating a population based breast cancer screening program, *Journal of Clinical Epidemiology*, 53, 595-603.

Sharma, D., Yadav, U. B. and Sharma, P., 2009, The concept of sensitivity and specificity in relation to two types of errors and its application in medical research, *Journal of Reliability and Statistical Studies*, 2, 53-58.

Shi, Y. H. and Eberhart, R. C., 1999, Empirical study of particle swarm optimization, *Proceedings of the IEEE International Conference on Evolutionary Computation*, 3, 1945–1950.

Shi, Y., Liu, H., Gao, L. and Zhang, G., 2011, Cellular particle swarm optimization, *Information Sciences*, 4460-4493.

Sinem A. ve Bilal A., 2012, Güncel Sürü Zekası Optimizasyon Algoritmaları, *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi 1*, 36-50.

Tamer, S. ve Karakuzu, C., 2006, Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri, *ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu*, Bursa, 302-306.

Varmazabadi, M. A. and Aiouri, A. S., 2012, Training RBF neural network by hybrid adaptive modified particle swarm optimization-Tabu search algorithm for function approximation, *International Journal of Scientific & Engineering Research (IJSER)*, 3(8), 1-8.

Worden, J. K., Mickey, R. M., Vacek, P. M., Flynn, B. S., Solomon, L. J., Secker-Walker, R. H., Danigelis, N. L., Geller, B. M., Warner, S. L., Clark, R. A., Foster, R. S., Vezina, J. L. and Hooper, G., 2002, Evaluation of a community breast screening promotion program, *Preventive Medicine*, 35, 349-361.

Xu, R., Venayagamoorthy, G. K. and Wunsch, D. C., 2007, Modeling of gene regulatory Networks with hybrid differential evolution and particle swarm optimization, *Neural Networks*, 917-927.

Yadav, R. and Mandal, D., 2011, Optimization of artificial neural network for speaker recognition using particle swarm optimization, *International Journal of Soft Computing and Engineering (IJSCE)*, 1(3), 2231-2307.

Yazıcı, İ., 2008, EMG işaretlerinin işlenmesi ve sınıflandırılması, Yüksek Lisans Tezi, *Sakarya üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, 67-68.

Zhang, C-X. and Zhang, J-S., 2008, Rotboost: a technique for combining rotation forest and adaboost, *Pattern Recognition Letters*, 29, 1524-1536.

Zincir, H., 1999, Malatya il merkezinde 40 yaş üzeri kadınların meme kanseri ve korunma konusunda bilgi tutum ve davranışları, Yüksek Lisans Tezi, *İnönü Üniversitesi Sağlık Bilimleri Enstitüsü*, Malatya, 2-6.

http://www.alife.org [Erişim Tarihi:8 Ekim 2012]

http://www.cancer.org/ American Cancer Societ [Erişim Tarihi:8 Ekim 2012]

http://www.engr.iupui.edu/~eberhart/ [Erişim Tarihi:8 Ekim 2012]

http://www.haberler.com/7-8-kadindan-biri-meme-kanseri-oluyor-3267668-haberi/ [Erişim Tarihi:8 Ekim 2012]

http://www.nenedir.net/nedir/bilim/3326-optimizasyon-nedir.html [Erişim Tarihi:8 Ekim 2012]

http://www.physionet.org/ [Erişim Tarihi:27 Temmuz 2012]

http://www.red3d.com/cwr/boids/ [Erişim Tarihi:8 Ekim 2012]

http://www.scholarpedia.org/article/Particle_swarm_optimization [Erişim Tarihi: 27 Ekim 2012]

http://scipy-lectures.github.com/intro/scipy.html [Erişim Tarihi:8 Ekim 2012]

http://www.swarmintelligence.org/tutorials.php [Erişim Tarihi:8 Ekim 2012]

http://tr.wikipedia.org/wiki/Meme_kanseri#cite_note-0 [Erişim Tarihi:8 Ekim 2012]

http://www.yapay-zeka.org/modules/wiwimod/index.php?page=ANN [Erişim Tarihi:8 Ekim 2012]

Başaran,E.,2012,RadikalGazetesi, http://www.radikal.com.tr/Radikal.aspx?aType=RadikalYazar&ArticleID=1078899&Yazar=EZGI-BASARAN&CategoryID=97 [Erişim Tarihi:8 Ekim 2012]

Kakıcı,A.,2009,Yapay sinir ağlarına giriş [online], http://www.ahmetkakici.com/yazilim/yapay-sinir-aglarina-giris/ [Erişim Tarihi:8 Ekim 2012]

Karayurt,Ö., Meme Kanseri, http://www.saglik.gov.tr [Erişim Tarihi: 27 Ekim 2012].

Matsui,W., Breast Cancer, http://www.nlm.nih.gov.tr [Erişim Tarihi: 27 Ekim 2012].

Sağ,T., 2009, Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) [online], http://yazilimmutfagi.com/MakaleOku.aspx?MakaleKey=1a4ceabd-5ad7-4d8f-88e6-6feb15dd8f2d [Erişim Tarihi : 8 Ekim 2012]

Tükel,S., 2008, Dünyada meme kanseri görülme sıklığı nedir?, http://www.metam.org.sss.php?id=86 [Erişim tarihi: 20 Ekim 2012]

Yılmaz,S.,2012, https://docs.google.com/viewer?a=v&q=cache:mCkiK5qhbi8J:serhatyilmaz.org/dersler/sc/SayCozdosyalar/10.Optimizasyon.ppt+10.optimizasyon&hl=tr&gl=tr&pid=bl&srcid=ADGEEShiNhyiWetYgN9mD81Op41SppYvNQ5rspQDiXYCVa7O6wxsyRFnTpTc6nS5AnXtEtmaODR71pO32SQ66QLk2BLdDVYTyHHCWwTyT7UYT4ymbyzt9xqt8tsDFRcchyKfSgKx&sig=AHIEtbReQLSvw7jNw87sjLsahoAXN5B9ZQ [Erişim Tarihi: 9 Ekim 2012]

EKLER

EK-1 Hibrit PSO-YSA yazılımı

```
dat=xlsread('breastcancer.xls');
out_come=dat(:,10);NN=0;NH=0;HN=0;HH=0;confus_mat=zeros(2,2);
for hh=1:9
    for jj=1:size(dat,1)
       x(jj,hh)=(dat(jj,hh)-min(dat(:,hh)))/(max(dat(:,hh))-
min(dat(:,hh)));
    end
end
dat(:,1:9)=x;  k_num=2;
      indices = crossvalind('Kfold',out_come,k_num);
      cp = classperf(out_come); % initializes the CP object
      for i = 1:k_num
          test = (indices == i); train = ~test;
          class =
psonn_rand(dat(train,1:9),out_come(train),dat(test,1:9));
          oun=out_come(test);
          for ii=1:size(class,1)
               if((class(ii)==oun(ii)) && (oun(ii)==0))
                   NN=NN+1;
               elseif((class(ii)==oun(ii)) && (oun(ii)==1))
                   HH=HH+1;
               elseif((class(ii)~=oun(ii)) && (oun(ii)==1))
                   HN=HN+1;
               elseif((class(ii)~=oun(ii)) && (oun(ii)==0))
                   NH=NH+1;
               end
          end
          classperf(cp,class,test);
      end
      cp.CorrectRate % queries for the correct classification rate
      confus_mat(1,1)=NN;confus_mat(2,1)=HN;confus_mat(1,2)=NH;
confus_mat(2,2)=HH;
accuracy_rate=(sum(diag(confus_mat))/size(dat,1))*100;
```

```
function [test_targets] = psonn_rand(train_da,train_ou,test_da)
data=train_da';desired_out=train_ou';test=test_da';
rand('state',0);
%=================================================================
G=size(data,1); % number of input nodes
C=size(desired_out,1); % number of hidden nodes
num_pattern=size(desired_out,2); % number of pattern
%=================================================================
wh=zeros(H,size(data,1));
wc=zeros(size(desired_out,1),H);
xmax=100;xmin=-100;vmax=80;vmin=-80;pop_size=5;bxmin=-
70;bxmax=70;bvmin=-56;bvmax=56;
ww=zeros(pop_size,((H*size(data,1))+H+H+size(desired_out,1)));
ww1=(xmin*ones(pop_size,(H*size(data,1))))+((xmax-
xmin)*rand(pop_size,(H*size(data,1))));
ww2=(bxmin*ones(pop_size,H))+((bxmax-bxmin)*rand(pop_size,H));
ww3=(xmin*ones(pop_size,H))+((xmax-xmin)*rand(pop_size,H));
ww4=(bxmin*ones(pop_size,size(desired_out,1)))+((bxmax-
bxmin)*rand(pop_size,size(desired_out,1)));
```

```
ww=[ww1,ww2,ww3,ww4];func_err=ones(1,pop_size,maxiter+1);dfunc_err=one
s(1,pop_size,maxiter+1);
nn=size(ww,2);gbest=zeros(1,nn);c1=2.08;c2=1.92;v=zeros(pop_size,nn,ma
xiter+1);www=ones(pop_size,((H*size(data,1))+H+H+size(desired_out,1)),
maxiter+1);www(:,:,1)=ww;inertia_weight=0.9;

%=====================================================================
n=1;g=0;
inp_hidden=zeros(H,num_pattern); % input of hidden layer
out_hidden=zeros(H,num_pattern); % output of hidden layer
inp_out=zeros(C,num_pattern); % input of ouput layer
out_out=zeros(C,num_pattern); % output of output layer

%=====================================================================
clear train_error;
while n<=maxiter
    for dg=1:pop_size
        ww1=www(dg,:,n);
        wha=ww1(1,1:(H*size(data,1)));
        for i=1:H
          wh(i,:)=wha((((i-1)*size(data,1))+1):(size(data,1)*i));
        end
        bh=ww1(((H*size(data,1))+1):((H*size(data,1))+H));
wc=ww1(((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+H)
);
        bc=ww1(((H*size(data,1))+size(bh,2)+H+1):size(ww,2));
        for g=1:num_pattern
            inp_hidden(:,g)=(wh*data(:,g))+bh';
            out_hidden(:,g)=logsig(inp_hidden(:,g));
            inp_out(:,g)=(wc*out_hidden(:,g))+bc';
            out_out(:,g)=logsig(inp_out(:,g));
        end
           train_error(n)=((sum(sum(err.^2)))/size(desired_out,2));
        if n>1
             if train_error(n)<(1e-3),break,end;
        end
        func_err(1,dg,n)=train_error(n);
    end
%======================pbest güncellemesi=======================%
 if (n==1)
     pbest(:,:,n)=www(:,:,n);
     dfunc_err(1,:,n)=func_err(1,:,n);
 elseif (n~=1)
     for i=1:pop_size
      if(func_err(1,i,n)<dfunc_err(1,i,n-1))
             pbest(i,:,n)=www(i,:,n);
             dfunc_err(1,i,n)=func_err(1,i,n);
      else
             pbest(i,:,n)=pbest(i,:,n-1);
             dfunc_err(1,i,n)=dfunc_err(1,i,n-1);
      end
     end
 end
%====================gbest güncellemesi=========================%
[aa,ind]=min(dfunc_err(1,:,n));

fprintf('iterasyon sayısı=%d, hata oranı=%f\n',n,aa);
 if n>1
      if aa<(1e-3),break,end;
 end
```

```
%================ağırlık ve bias güncellemesi==================%
 for i=1:pop_size
    for j=1:(H*size(data,1))
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for
j=((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+H)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for j=((H*size(data,1))+1):((H*size(data,1))+H)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for j=((H*size(data,1))+size(bh,2)+H+1):size(ww,2)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
 end
    inertia_weight=0.9-((0.5*n)/maxiter);
    n=n+1;
end

%====TEST AŞAMASI ve gbest vektörünün test için kullanılması====%

s=size(test,2);
wha=gbest(1,1:(H*size(data,1)));
    for i=1:H
       wh(i,:)=wha((((i-1)*size(data,1))+1):(size(data,1)*i));
    end
bh=gbest(((H*size(data,1))+1):((H*size(data,1))+H));
wc=gbest(((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+
H));
bc=gbest(((H*size(data,1))+size(bh,2)+H+1):size(ww,2));

    for m=1:s
        thlg(:,m)=wh*test(:,m)+bh';
        thlc(:,m)=logsig(thlg(:,m));
        tckg(:,m)=wc*thlc(:,m)+bc';
        tckc(:,m)=logsig(tckg(:,m));
    end

test_targets=(round(tckc))';
end
```

EK-2 RO (hibrit PSO-YSA) yazılımı

```
dat=xlsread('breastcancer.xls');
out_come=dat(:,10);NN=0;NH=0;HN=0;HH=0;confus_mat=zeros(2,2);
for hh=1:9
    for jj=1:size(dat,1)
        x(jj,hh)=(dat(jj,hh)-min(dat(:,hh)))/(max(dat(:,hh))-
min(dat(:,hh)));
    end
end
dat(:,1:9)=x;k_num=10;
      indices = crossvalind('Kfold',out_come,k_num);
      cp = classperf(out_come); % initializes the CP object
      for i = 1:k_num
          test = (indices == i); train = ~test;
          class =
rfpsoysa(dat(test,1:9),dat(train,1:9),out_come(train));
          oun=out_come(test);
          for ii=1:size(class,1)
               if((class(ii)==oun(ii)) && (oun(ii)==0))
                   NN=NN+1;
               elseif((class(ii)==oun(ii)) && (oun(ii)==1))
                   HH=HH+1;
               elseif((class(ii)~=oun(ii)) && (oun(ii)==1))
                   HN=HN+1;
               elseif((class(ii)~=oun(ii)) && (oun(ii)==0))
                   NH=NH+1;
               end
          end
          classperf(cp,class,test);
      end
      cp.CorrectRate % queries for the correct classification rate
      confus_mat(1,1)=NN;confus_mat(2,1)=HN;confus_mat(1,2)=NH;
confus_mat(2,2)=HH;
accuracy_rate=(sum(diag(confus_mat))/size(dat,1))*100;
```

```
function [ trainDATA_subnew ] =
bootstrap_process(traindat_subset,train_DATALABEL,aa,Ratio)
        rand('state',sum(100*clock));
        Xpartini=traindat_subset(sel_column,:);
        Ypartini=train_DATALABEL(sel_column,:);
        trainDATA_subnew=[Xpartini];
        trainDATALABEL_new=[Ypartini];
        setrow=[sel_column];
end
```

```
function [ pca_eig_vector] = pca_process(data)
%%%young-eckart teoremine göre PCA yazılımı%%%
    [row_num, column_num]=size(data);
    fark=zeros(row_num,column_num);
    mean_data_column=mean(data,1);
    for i=1:row_num
        fark(i,:)=mean_data_column;
    end
    data=data-fark;
    cov_matrix=(data'*data);
```

```
    [vect,val]=eig(cov_matrix);
    pca_eig_vector=vect;
end
```

```
function
[newsetrow,Rot_matrix,eig_all,newtrainDATA,newtestDATA]=rotation_proce
ss(train_DATA,train_DATALABEL,test_DATA,K,Ratio)

    data_set=train_DATA;
    [data_num feature_num]=size(data_set); % datanın satır ve sütun
sayıları belirlendi
    each_samp_num=floor(feature_num/K); % herbir altküme içindeki veri
sayısı belirlendi


%=========================================================================%
    %söz konusu train_DATA veri matrisinde sütunların rasgele
dağıtılması ve matrisin yeniden oluşturulması
    rand('state',sum(100*clock));
    newsetrow=randperm(feature_num);
    new_data=data_set(:,newsetrow);
    eig_all=zeros(feature_num,feature_num);%özvektör matrisi
tanımlandı


%=========================================================================%
    for i=1:K
        trans1=(i-1)*each_samp_num;
        trans2=i*each_samp_num;
        traindat_subset=new_data(:,trans1+1:trans2);%data alt kümeye
(342x3 lük) ayrıldı

    %%%% alt kümelelerin(azaltılmış datalı) elde edilmesi için
bootstrap tekniğinin uygulanması %%%%
        aa=size(traindat_subset,1);
        [ trainDATA_subnew ] =
bootstrap_process(traindat_subset,train_DATALABEL,aa,Ratio);

    %%%% PCA prosesinin uygulanması ve özvektör matrisinin elde
edilmesi %%%%
        eig_vector=pca_process(trainDATA_subnew);
        eig_all(trans1+1:trans2,trans1+1:trans2)=eig_vector;
    end


%=========================================================================%
    %%%% özvektörlerin orjinal veriye göre düzenlenmesi %%%%

    eig_new=zeros(feature_num,feature_num);
    for i=1:feature_num
         eig_new(:,newsetrow(i))=eig_all(:,i);
    end
    Rot_matrix=eig_new;

%=========================================================================%
    %%%% yeni örneklerin elde edilmesi%%%%
    newtrainDATA=train_DATA*Rot_matrix;
```

```
    newtestDATA=test_DATA*Rot_matrix;
end
```

```
function [test_targets] = psonn_rand(train_da,train_ou,test_da)
data=train_da';desired_out=train_ou';test=test_da';
rand('state',0);
%=================================================================
G=size(data,1); % number of input nodes
C=size(desired_out,1); % number of hidden nodes
num_pattern=size(desired_out,2); % number of pattern
%=================================================================
wh=zeros(H,size(data,1));
wc=zeros(size(desired_out,1),H);
xmax=100;xmin=-100;vmax=80;vmin=-80;pop_size=5;bxmin=-
70;bxmax=70;bvmin=-56;bvmax=56;
ww=zeros(pop_size,((H*size(data,1))+H+H+size(desired_out,1)));
ww1=(xmin*ones(pop_size,(H*size(data,1))))+((xmax-
xmin)*rand(pop_size,(H*size(data,1))));
ww2=(bxmin*ones(pop_size,H))+((bxmax-bxmin)*rand(pop_size,H));
ww3=(xmin*ones(pop_size,H))+((xmax-xmin)*rand(pop_size,H));
ww4=(bxmin*ones(pop_size,size(desired_out,1)))+((bxmax-
bxmin)*rand(pop_size,size(desired_out,1)));
ww=[ww1,ww2,ww3,ww4];func_err=ones(1,pop_size,maxiter+1);dfunc_err=one
s(1,pop_size,maxiter+1);
nn=size(ww,2);gbest=zeros(1,nn);c1=2.08;c2=1.92;v=zeros(pop_size,nn,ma
xiter+1);www=ones(pop_size,((H*size(data,1))+H+H+size(desired_out,1)),
maxiter+1);www(:,:,1)=ww;inertia_weight=0.9;

%=================================================================
n=1;g=0;
inp_hidden=zeros(H,num_pattern); % input of hidden layer
out_hidden=zeros(H,num_pattern); % output of hidden layer
inp_out=zeros(C,num_pattern); % input of ouput layer
out_out=zeros(C,num_pattern); % output of output layer

%=================================================================
clear train_error;
while n<=maxiter
    for dg=1:pop_size
        ww1=www(dg,:,n);
        wha=ww1(1,1:(H*size(data,1)));
        for i=1:H
          wh(i,:)=wha((((i-1)*size(data,1))+1):(size(data,1)*i));
        end
        bh=ww1(((H*size(data,1))+1):((H*size(data,1))+H));
wc=ww1(((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+H)
);
        bc=ww1(((H*size(data,1))+size(bh,2)+H+1):size(ww,2));
        for g=1:num_pattern
            inp_hidden(:,g)=(wh*data(:,g))+bh';
            out_hidden(:,g)=logsig(inp_hidden(:,g));
            inp_out(:,g)=(wc*out_hidden(:,g))+bc';
            out_out(:,g)=logsig(inp_out(:,g));
        end
           train_error(n)=((sum(sum(err.^2)))/size(desired_out,2));
        if n>1
             if train_error(n)<(1e-3),break,end;
        end
```

```
        func_err(1,dg,n)=train_error(n);
    end
%======================pbest güncellemesi=======================%
 if (n==1)
     pbest(:,:,n)=www(:,:,n);
     dfunc_err(1,:,n)=func_err(1,:,n);
 elseif (n~=1)
     for i=1:pop_size
      if(func_err(1,i,n)<dfunc_err(1,i,n-1))
             pbest(i,:,n)=www(i,:,n);
             dfunc_err(1,i,n)=func_err(1,i,n);
      else
             pbest(i,:,n)=pbest(i,:,n-1);
             dfunc_err(1,i,n)=dfunc_err(1,i,n-1);
      end
     end
 end
%====================gbest güncellemesi=========================%
[aa,ind]=min(dfunc_err(1,:,n));

fprintf('iterasyon sayısı=%d, hata oranı=%f\n',n,aa);
 if n>1
      if aa<(1e-3),break,end;
 end

%================ağırlık ve bias güncellemesi==================%
 for i=1:pop_size
    for j=1:(H*size(data,1))
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for
j=((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+H)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for j=((H*size(data,1))+1):((H*size(data,1))+H)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
    for j=((H*size(data,1))+size(bh,2)+H+1):size(ww,2)
        v(i,j,n)= v(i,j,n)*inertia_weight + c1*rand*(pbest(i,j,n) -
www(i,j,n)) + c2*rand*(gbest(j)-www(i,j,n));
        www(i,j,n+1)=www(i,j,n)+v(i,j,n);
    end
 end
    inertia_weight=0.9-((0.5*n)/maxiter);
    n=n+1;
end

%====TEST AŞAMASI ve gbest vektörünün test için kullanılması====%

s=size(test,2);
wha=gbest(1,1:(H*size(data,1)));
    for i=1:H
       wh(i,:)=wha((((i-1)*size(data,1))+1):(size(data,1)*i));
    end
bh=gbest(((H*size(data,1))+1):((H*size(data,1))+H));
```

```
wc=gbest(((H*size(data,1))+size(bh,2)+1):((H*size(data,1))+size(bh,2)+
H));
bc=gbest(((H*size(data,1))+size(bh,2)+H+1):size(ww,2));

    for m=1:s
        thlg(:,m)=wh*test(:,m)+bh';
        thlc(:,m)=logsig(thlg(:,m));
        tckg(:,m)=wc*thlc(:,m)+bc';
        tckc(:,m)=logsig(tckg(:,m));
    end

test_targets=(round(tckc))';
end
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Hasan KOYUNCU
Uyruğu	**:**	Türkiye Cumhuriyeti
Doğum Yeri ve Tarihi	**:**	Meram/KONYA-26.07.1989
Telefon	**:**	0536 649 64 64
Faks	**:**	---
e-mail	**:**	hasanelektro89@gmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Karatay S.D.M.P. Anadolu Lisesi	2007
Üniversite	:	Selçuk Üniversitesi	2011
Yüksek Lisans	:	Selçuk Üniversitesi	Devam Ediyor
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2013	Selçuk Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI

Biyomedikal Yazılım

YABANCI DİLLER

İngilizce-İleri Seviye

YAYINLAR

Koyuncu H. and Ceylan R., 2013, Artificial neural network based on rotation forest for biomedical pattern classification, *IEEE 2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, 581-585.