



T.C.
NECMETTİN ERBAKAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



SÜREKLİ OPTİMİZASYON PROBLEMLERİ
İÇİN DEĞİŞKEN KOMŞULUK ARAMA
TEKNİKLERİNİN GELİŞTİRİLMESİ
Yusuf UZUN
DOKTORA TEZİ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

Mart - 2017
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Yusuf Uzun tarafından hazırlanan “SÜREKLİ OPTİMİZASYON PROBLEMLERİ İÇİN DEĞİŞKEN KOMŞULUK ARAMA TEKNİKLERİNİN GELİŞTİRİLMESİ” adlı tez çalışması 20/03/2017 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği / oy çokluğu ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof.Dr. Sabri KOÇER

.....

Danışman

Prof.Dr. Hüseyin ARIKAN

.....

Üye

Doç.Dr. Gülay TEZEL

.....

Üye

Doç.Dr. Ahmet SAMANCI

.....

Üye

Yrd.Doç.Dr. Murat DİLMEÇ

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Ahmet COŞKUN
Enstitü Müdürü

TEZ BİLDİRİMİ

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

DECLARATION PAGE

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

Yusuf UZUN

20.03.2017

ÖZET

DOKTORA TEZİ

SÜREKLİ OPTİMİZASYON PROBLEMLERİ İÇİN DEĞİŞKEN KOMŞULUK ARAMA TEKNİKLERİNİN GELİŞTİRİLMESİ

Yusuf UZUN

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Makine Mühendisliği Anabilim Dalı

Danışman: Prof.Dr. Hüseyin ARIKAN

2017, 115 Sayfa

Jüri

Prof.Dr. Hüseyin ARIKAN
Prof.Dr. Sabri KOÇER
Doç.Dr. Gülay TEZEL
Doç.Dr. Ahmet SAMANCI
Yrd.Doç.Dr. Murat DİLMEÇ

Metasezgisel yöntemlerden biri olan Değişken Komşuluk Arama (DKA) algoritması, ayrık optimizasyon problemlerine uygulanmak amacıyla geliştirilmiş bir metottur. Literatürde, sürekli optimizasyon problemleri için DKA'nın diğer metasezgisel algoritmalar ile birlikte kullanıldığı hibrit yapılar mevcuttur. Ayrıca, literatürde DKA'nın yapay zeka yöntemleri ile birlikte kullanıldığı çalışma sayısı azdır ve kural çıkarımına yönelik her hangi bir çalışmaya rastlanılmamıştır.

Bu tez çalışmasında sınıflandırma problemleri için, eğitilmiş yapay sinir ağlarından (EYSA) bilgi kazanımına yönelik (sınıflandırma kuralları şeklinde) bir algoritma geliştirilmiştir. Geliştirilen algoritma, YSA' daki bağlantı ağırlıklarında bulunan gizli bilgiyi keşfetmek için eğitilmiş yapay sinir ağları üzerinde çalıştırılmıştır. DKA metasezgisel algoritması iki adımlı hiyerarşik bir yapıya sahiptir. İlk adımda çok katmanlı bir sinir ağı eğitilmekte ve ağırlıkları çıkarılmaktadır. İkinci adımda, elde edilen ağırlıklar DKA algoritması ile sınıflandırma kurallarının üretimi için kullanılmaktadır.

Önerilen algoritma iki farklı veri kümesi (Echo ve WBC) üzerinde deneysel olarak değerlendirilmiş ve literatürde yer alan diğer yöntemler ile karşılaştırılmıştır. Veri kümeleri üzerinde yapılan analizler, geliştirilen algoritmanın doğru ve etkin sınıflandırma kuralları üretebildiğini göstermektedir.

Anahtar Kelimeler: Değişken komşuluk arama, Yapay sinir ağı, Kural çıkarımı, Optimizasyon

ABSTRACT

PhD THESIS

DEVELOPING VARIABLE NEIGHBORHOOD SEARCH TECHNIQUES FOR CONTINUOUS OPTIMIZATION PROBLEMS

Yusuf UZUN

The Graduate School of Natural and Applied Science
of Necmettin Erbakan University
Department of Mechanical Engineering

Advisor: Prof. Dr. Hüseyin ARIKAN

2017, 115 Pages

Jury

Prof.Dr. Hüseyin ARIKAN
Prof.Dr. Sabri KOÇER
Doç.Dr. Gülay TEZEL
Doç.Dr. Ahmet SAMANCI
Yrd.Doç.Dr. Murat DİLMEÇ

Variable Neighborhood Search (VNS) algorithm, one of metaheuristic methods, is a method developed to be applied to discrete optimization problems. In literature, are available hybrid structures using the VNS method along with different algorithm for continuous optimization problems. Additionally, have been observed that the small number of studies using the VNS method along with artificial intelligence methods in the literature and hasn't been found a study for rule extraction.

This thesis study presented a method of extraction correct and intelligible rules from trained artificial neural networks (TANN) using variable neighborhood search (VNS) metaheuristic method. The developed method, were run on trained artificial neural networks to discover of confidential information contained in ANN connection weight. VNS metaheuristic algorithm has a two-step hierarchical structure. In the first step, a multi-layer neural network has been trained and weights are removed. In the second step, the weights obtained are used for the production of classification rules.

The suggested algorithm was assessed experientially on two different data group (Echo and WBC) and was compared with other methods in the current literature. The analyses on the test data sets show the fact that the developed algorithm can produce accurate and effective classification rules.

Keywords: Variable neighborhood search, Artificial neural network, Rule extraction, Optimization

ÖNSÖZ

Çalışmama verdiği destek ve gösterdiği anlayıştan dolayı danışmanım sayın Prof. Dr. Hüseyin ARIKAN'a, tez süresince belirttikleri görüşler ve önerilerle tezin yönlendirilmesinde büyük ölçüde yardımcı olan sayın Doç. Dr. Gülay TEZEL'e, çalışmamda bilgi ve görüşlerini esirgemeyen sayın Prof. Dr. Mehmet CUNKAŞ'a, Doç. Dr. Ahmet SAMANCI'ya, sayın Yrd. Doç. Dr. Murat DİLMEÇ'e, sayın Doç. Dr. Halife KODAZ'a, sayın Doç. Dr. Harun UĞUZ'a ve sayın Yrd. Doç. Dr. Ömer Kaan BAYKAN'a, Değişken Komşuluk Arama yöntemleri alanında çalışmama vesile olan sayın Doç. Dr. Erkan ÜLKER'e, Selçuk Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümündeki değerli arkadaşlarıma en içten teşekkürlerimi sunarım.

Özellikle manevi desteğini hiç bir zaman esirgemeyen her zaman ve her konuda hep yanımda olan aileme, gösterdikleri sabır, anlayış ve destekle her zaman yanımda olan eşime ve çocuklarıma desteklerinden dolayı çok teşekkür ederim.

Yusuf UZUN
KONYA, 2017

İÇİNDEKİLER

TEZ KABUL VE ONAYI	ii
TEZ BİLDİRİMİ	iii
ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER LİSTESİ	xi
ÇİZELGELER LİSTESİ	xiii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
3. MATERYAL VE YÖNTEM.....	15
3.1. Değişken Komşuluk Arama Yöntemi	15
3.1.1. Değişken komşuluk iniş	18
3.1.2. İndirgenmiş değişken komşuluk arama	20
3.1.3. Genel değişken komşuluk arama.....	21
3.1.4. Temel değişken komşuluk arama	22
3.1.5. Esnek değişken komşuluk arama	23
3.1.6. Değişken komşuluk ayrıştırırmalı arama	24
3.2. Komşuluk Yapıları	24
3.2.1. λ - opt yöntemi	25
3.2.2. Düğüm ekleme	25
3.2.3. Düğüm yer değiştirme	26
3.2.4. Yol yer değişimi yöntemi	26
3.2.5. Yol ekleme yöntemi	27
3.3. Yapay Sınır Ağları	27
3.3.1. Tanım ve tarihsel gelişimi	28
3.3.2. Biyolojik hücre modeli	28
3.3.3. Yapay hücre modeli.....	30
3.3.4. YSA'nın temel özellikleri	35

3.3.5. Temel YSA mimarileri.....	36
3.3.5.1. İleri Beslemeli YSA	36
3.3.5.2. Geri beslemeli YSA.....	37
3.3.6. Çok katmanlı algılayıcı model	37
3.3.7. Aktivasyon fonksiyonları	41
3.3.8. Yapay sinir ağlarında öğrenme.....	43
3.3.8.1. İleri doğru hesaplama	46
3.3.8.2. Geriye doğru hesaplama.....	47
3.4. Veri Madenciliği	52
3.4.1. Veri madenciliği işlem süreci.....	54
3.4.2. Veri madenciliği modelleri.....	58
3.4.3. Veri madenciliği problemleri	60
3.5. Optimizasyon ve Temelleri.....	61
3.5.1. Optimizasyon yöntemleri	62
3.5.2. Doğrusal olmayan programlama	64
3.6. Sınıflandırma.....	67
3.6.1. Sınıflandırma kuralları	68
3.6.2. Kural çıkarımı.....	69
3.7. Veri Ayırıklaştırma	70
3.7.1. Eşit genişlikli ayırıklaştırma yöntemi.....	71
3.7.2. Eşit frekanslı ayırıklaştırma yöntemi.....	73
3.8. Verilerin İkili Formda Kodlanması.....	73
3.9. Kalite ve Uygunluk Fonksiyonları	76
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	78
4.1. Veri Kümeleri	78
4.2. Deneysel Çalışma.....	83
5. SONUÇLAR VE ÖNERİLER	101
5.1. Sonuçlar	101
5.2. Öneriler	104
KAYNAKLAR	105
ÖZGEÇMİŞ	114

SİMGELER VE KISALTMALAR

Simgeler

N	Komşuluk yapısı
k	Komşuluk sırası
k_{\max}	En son komşuluk sırası
S	Çözüm kümesi
A_n	Veritabanı özellikler kümesi
x_{m_n}	İkili alt dizi
GD	Giriş katmandaki düğüm sayısı
C_k	Çıktı sınıf vektörü
X_m	Giriş katmanındaki değişkenlerinin vektörü
K	İkili çıktı sınıf sayısı
KU	Kural uzunluğu
tp	Pozitif doğru
tn	Negatif doğru
fp	Pozitif yanlış
fn	Negatif yanlış
j	Gizli düğüm sayısı
$f(x)$	Uygunluk fonksiyonu
y	Yerel aramada bulunan çözüm
λ	Rota içerisindeki doğru
$\lambda\text{-opt}$	Rotadaki optimum doğru
w_k	k. hücrenin ağırlığı
v_k	k. hücrenin aralık aktivitesi
y_k	k. hücrenin nöronun çıkışı
θ_k	k. hücrenin eşik değeri
AG1	Giriş katmanı ve gizli katman arasındaki ağırlıklar
AG2	Gizli katman ve çıkış katmanı arasındaki ağırlıklar
GDG	Gizli düğümüne verilen toplam giriş
CDG	Çıkış düğümüne verilen toplam giriş
S_k	k. düğümün son çıkış değeri
η	Öğrenme katsayısı
α	Momentum katsayısı
Δw	Bağlantı ağırlığındaki değişim miktarı
δ_m	m . çıktı biriminin hatası

Kısaltmalar

YSA	Yapay sinir ağı
EYSA	Eğitilmiş yapay sinir ağı
DKA	Değişken komşuluk arama
DKİ	Değişken komşuluk iniş
İDKA	İndirgenmiş değişken komşuluk arama
GDKA	Genel değişken komşuluk arama
TDKA	Temel değişken komşuluk arama
EDKA	Esnek değişken komşuluk arama
DKAA	Değişken komşuluk ayrıştırılmalı arama

GA	Genetik algoritma
EA	Evrimsel algoritma
ACO	Ant colony optimization
TACO	Touring ant colony optimization
PSO	Particle swarm optimization
SVM	Support vector machines
GFS	Genetic fuzzy system
BPSO	Binary particle swarm optimization
ÇKA	Çok katmanlı algılayıcı
Echo	Echocardiogram
WBC	Wisconsin breast cancer
EGA	Eşit genişlikli ayırma
YBS	Yapay bağışıklık sistemi
GFS	Genetic fuzzy system



ŞEKİLLER LİSTESİ

Şekil 3.1. Metasezgisel algoritmanın çalışma prensibi.....	15
Şekil 3.2. DKA algoritmasının benzetimsel kodu	16
Şekil 3.3. Komşuluk yapıları kullanarak değişken komşuluk arama.....	17
Şekil 3.4. Değişken komşuluk iniş metodu.....	19
Şekil 3.5. Değişken komşuluk iniş algoritmasının çalışma prensibi	19
Şekil 3.6. İndirgenmiş değişken komşuluk arama metodu	20
Şekil 3.7. Genel değişken komşuluk arama metodu	21
Şekil 3.8. Temel değişken komşuluk arama metodu	22
Şekil 3.9. Esnek değişken komşuluk arama metodu.....	23
Şekil 3.10. Değişken komşuluk ayrıştırırmalı arama metodu	24
Şekil 3.11. λ -opt yönteminin örnek gösterimi	25
Şekil 3.12. Düğüm ekleme yönteminin örnek gösterimi	26
Şekil 3.13. Düğüm yer değiştirme yönteminin örnek gösterimi.....	26
Şekil 3.14. Düğüm yer değiştirme yönteminin örnek gösterimi.....	27
Şekil 3.15. Biyolojik sinir hücreleri.....	30
Şekil 3.16. Bir hücrenin içyapısı.....	31
Şekil 3.17. Basit bir yapay sinir ağı modeli	33
Şekil 3.18. YSA modeli	34
Şekil 3.19. YSA'nın yapısı, (Elalfi ve ark., 2004).....	38
Şekil 3.20. ÇKA YSA modelinin işlem diyagramı.....	40
Şekil 3.21. Aktivasyon fonksiyonları	41
Şekil 3.22. Danışmanlı öğrenme kuralı	44
Şekil 3.23. Danışmansız öğrenme kuralı	44
Şekil 3.24. İleri beslemeli çok katmanlı ağ modeli.....	46
Şekil 3.25. Veritabanında bilgi keşfi süreci	54
Şekil 3.26. Bir fonksiyonun minimum ve maksimum noktaları.....	62
Şekil 3.27. Yerel ve global optimumluk noktaları.....	66
Şekil 3.28. Eşit genişlikli ayrıştırma örneği.....	72
Şekil 3.29. Ekokardiyografi görüntüsü	79
Şekil 4.1. Önerilen algoritmanın temel adımları.....	85
Şekil 4.2. Önerilen metodun akış şeması	87

Şekil 4.3. Echo veri kümesi için eğitim hata grafiği.....	88
Şekil 4.4. Echo veri kümesi için performans değerlendirmeleri.....	93
Şekil 4.5. WBC veri kümesi için eğitim hata grafiği.....	94
Şekil 4.6. WBC veri kümesi için performans değerlendirmeleri.....	99



ÇİZELGELER LİSTESİ

Çizelge 2.1. Kaynak arařtırmalarının özeti.....	12
Çizelge 3.1. Biyolojik ve yapay hücre modellerinin karşılaştırılması.....	31
Çizelge 3.2. Sık kullanılan aktivasyon fonksiyonları (Öztemel, 2003).....	42
Çizelge 3.3. Tenis oynama veri kümesi.....	75
Çizelge 3.4. Tenis oynama veri kümesi için ikili kodlama vektör gösterimi.....	75
Çizelge 3.5. Tenis oynama veri kümesi deęişkenlerinin ikili kodlanması.....	75
Çizelge 3.6. Tenis oynama veri kümesi için ikili kodlama gösterimi.....	76
Çizelge 4.1. Veri Kümelerinin Temel Özellikleri.....	78
Çizelge 4.2. Echo veri kümesine ait deęişkenler.....	81
Çizelge 4.3. WBC veri kümesine ait deęişkenler.....	82
Çizelge 4.4. Echo için giriş ve gizli katman arasındaki aęırlıkları ($AG1_{i,j}$).....	89
Çizelge 4.5. Echo için gizli ve çıkış katmanı arasındaki aęırlıklar ($AG2_{j,k}$).....	89
Çizelge 4.6. Çıkış sınıf deęerinin 0 olma durumunda çıkarılan örnek kurallar..	90
Çizelge 4.7. Çıkış sınıf deęerinin 1 olma durumunda çıkarılan örnek kurallar..	91
Çizelge 4.8. Echo için metotlar arasındaki performans deęerlendirmeleri.....	93
Çizelge 4.9. WBC için giriş ve gizli katman arasındaki aęırlıkları ($AG1_{i,j}$).....	95
Çizelge 4.10. WBC için gizli ve çıkış katmanı arasındaki aęırlıklar ($AG2_{j,k}$) ...	95
Çizelge 4.11. Çıkış sınıf deęerinin 0 olma durumunda çıkarılan örnek kurallar	96
Çizelge 4.12. Çıkış sınıf deęerinin 1 olma durumunda çıkarılan örnek kurallar	96
Çizelge 4.13. WBC için metotlar arasındaki performans deęerlendirmeleri.....	99

1. GİRİŞ

Bilişim teknolojilerinde meydana gelen gelişmelerle doğru orantılı olarak veri miktarının boyutları da gün geçtikçe artmaktadır. Elde edilen büyük boyutlu verilerin işlenmesi ve işlenen verilerden anlamlı sonuçların elde edilmesi problemi çözülmesi gereken önemli konulardan biridir. Bu tür problemlerin üstesinden gelebilmek için geliştirilen birçok yöntem bulunmaktadır. Bunlardan bir tanesi de bilgi keşfine dayalı sınıflandırma kurallarının çıkarımıdır.

Veritabanı uygulamalarında geliştirilen yöntemlerden dolayı çok fazla miktarda bilgi depolanmasına olanak sağlanmıştır. Büyük miktarlardaki verilerin içerisinde saklı bulunan gizli bilgilerin geleneksel yöntemler kullanılarak tespit edilmesi gayet zordur. Veri miktarı büyüdükçe ve elde edilen verilerdeki karmaşıklık arttıkça, veri işleme ve çözümleme yöntemlerine olan ihtiyaçta gün geçtikçe artmaktadır (Adriaans ve Zantige, 1996).

Veri madenciliği büyük boyutlu veriler içerisinde gizli bilgi ve örüntülerin keşfi için geliştirilen yöntemlerden en yaygın ve en çok kullanılanlardan birisidir. Veri madenciliği, temel olarak verilerin içerisindeki örüntülerin, veri ilişkilerinin, tutarsızlıkların ve kuralların çıkarımı üzerinde çalışmaktadır. Amaç, daha önceden fark edilmemiş örüntüleri tespit edebilmek ve bu gizli örüntüleri keşfedebilmektir (Hand ve ark., 2001).

Karar değişkenleri ve bir takım parametreler kullanılarak tanımlanan bir problemde, değişkenlerin uygun değerler alarak ulaşabileceği en iyi çözümü elde etmeye çalışan problem tiplerine optimizasyon problemleri denir. Günümüzde bu tip problemleri çözmek için pek çok algoritma ve yöntemler geliştirilmiştir. Bu algoritmalar kesin çözümü veren ve yaklaşık çözümü veren algoritmalar olarak iki grupta sınıflandırılabilir. Kesin çözümü veren algoritmalar en iyi çözümü bulmayı garanti ederken yaklaşık çözümlü algoritmalar ise sadece bazı çözümleri değerlendirerek en iyiye yakın çözümleri bulurlar (Blum ve Roli, 2003).

Deneme yanılmaya dayalı bir arama stratejisi kullanarak çözüm kümesinin hangi çözümlere ulaşacağını belirleyen yöntemlere genel olarak sezgisel yöntemler denir. En basit sezgisel yöntemler yapıcı ve yerel arama metotlarıdır. Yapıcı metotlar çözümü,

boş bir çözüm kümesinden başlayarak ve belli bir amaç gözeterek çözümün bileşenlerini teker teker eklemekle oluşturur. Yerel arama ise başlangıç bir çözüm kümesinden başlayarak bir döngü içerisinde çözüme ait belirli bir komşuluk yapısında daha iyi çözümü arama metodudur (Şevkli, 2010).

Geleneksel YSA, desen tanıma ve sınıflandırma gibi geniş çeşitliliğe sahip uygulamalarda sıklıkla kullanılan metotlardan birisidir. YSA'nın, nöronlar arasındaki bağlantılarının şekli, bağlantılardaki ağırlıkların öğrenme algoritması ve aktivasyon fonksiyonu ile karakterize edildiği bilinmektedir (Fauset, 1994). Geleneksel YSA modelinde, bir katmanda bulunan ve önceden tanımlanan aktivasyon fonksiyonları sabittir ve herhangi bir düğüm için aktivasyon fonksiyonunun matematiksel ifadesi değişmemektedir (Segee, 1993).

Son zamanlarda yapılan çalışmalarda, gerçek dünya problemlerinin çözümünde basit sezgisel metotların bir araya getirilmesiyle daha soyut ve üst seviyeye sahip metasezgisel yöntemler kullanılmaktadır.

Metasezgisel yöntemler yüksek kalitede bir çözüm elde edebilmek için kullandığı sezgiselleri bir kural mekanizması ile değiştiren tekrarlı bir işlemdir. Her bir tekrarda tek bir çözümü veya çözüm kümesini değiştirerek yeni en iyi çözümlere ulaşır. Bünyesinde kullandığı sezgisel yöntemler basit düzeyde yerel arama yada yapıcı metotlar olabileceği gibi yüksek seviyeye sahip sezgisel metotlarda olabilir (Voß ve ark., 1999).

Metasezgisel yöntemlerin geleneksel metotlara göre dikkati çeken en büyük avantajı daha kısa sürede çözüm sunmalarıdır. Ayrıca yöntemin problemlerden bağımsız olması sezgisel metotlara nazaran daha yaygın bir kullanım alanına sahiptir. Yöntemin dezavantajları ise, problem üzerinde en iyi çözümü garanti etmemesi ve iyi çözümü elde edebilmek için birçok parametrenin kullanılarak bunların uygun değerler kullanılarak ayarlanmasıdır. Bu işlem de genellikle uygulamalarda deneysel bir takım çalışmalar yapılarak gerçekleştirilir (Şevkli, 2010).

Metasezgisel algoritmalar günümüzde optimizasyon problemlerinin çözümünde kullanılan en etkili yöntemlerden biridir. Bu yöntemler içerisinde, bir takım komşuluk yapıları kullanılmaktadır. Algoritma; mevcut çözüme ait komşu çözümleri dikkate alarak, çözüm uzayının içerisinde bir arama işlemi yürütür ve belli bir durdurma

kriterini sağladığında süreci sonlandırır, son olarak elde edilen çözümü kullanıcıya sunmaktadır (Şevkli, 2010).

Ayrıca metasezgisel yöntemler, arama uzayında çalışırken yerel iyilere yoğunlaşarak takılıp kalma riskini en aza indirgeyen bir takım metotları da içerisinde barındırmalıdır. Yöntem arama uzayı üzerinde iyi yayılabilmeli ve elde edilen yerel iyilere iyi yoğunlaşabilmelidir. Bu iki durumun sağlanması durumunda üzerinde çalışılan problemlerden daha kaliteli sonuçlar elde edilecektir. Bundan dolayı metasezgisel yöntemler kullanılırken rasgeleliğin yanı sıra sistematik bir işlevin de kullanılması gerekmektedir (Alba ve Marti, 2006).

Günümüze kadar farklı arama metotlarına sahip birçok metasezgisel yöntem geliştirilmiştir. Bunlardan bazıları yerel arama tekniğini daha etkin bir şekilde kullanan bir takım mekanizmalarla donatılmıştır. Tabu Arama, Tavlama Benzetimi, Değişken Komşuluk Arama ve Kılavuzlu Yerel Arama bu şekilde geliştirilmiş yöntemlere örnek olarak verilebilir. Tamamen farklı arama metotlarını kullanan metasezgisel yöntemlerde mevcuttur. Genetik Algoritma, Karınca Kolonileri Algoritması, Parçacık Sürü Optimizasyonu ve Yapay Arı Kolonisi algoritması bu tip metasezgisel yöntemlere örnek olarak verilebilir (Blum ve Roli, 2003).

Metasezgisel yöntemlerden biri olan Değişken Komşuluk Arama (DKA) algoritması, ayrık değerlere sahip olan optimizasyon problemlerine uygulanmak amacıyla geliştirilmiş bir metottur. Literatürde DKA'yı sürekli optimizasyon problemlerine uygulayabilmek için diğer metasezgisel algoritmalar ile beraber kullanıldığı bir takım hibrit yapılar da mevcuttur. Ayrıca, literatürde DKA'nın yapay zekâ yöntemleri ile birlikte kullanıldığı çalışma sayısı oldukça az olmakla birlikte kural çıkarımına yönelik her hangi bir çalışmaya rastlanılmamıştır.

Yerel arama tabanlı birçok metasezgisel algoritma, her bir iterasyonda tek bir komşuluk yapısı üzerinde çalışır. Yöntem bir iterasyondan diğerine geçerken güncelleme yapabilir veya yapamaz. Bunun yerine, algoritmanın her bir iterasyonunda birden fazla komşuluk yapısı kullanıldığı zaman çözüm süreci önemli derecede geliştirilmiş ve iyileştirilmiş olur. Böylece her bir iterasyon sonucunda birkaç yeni aday çözüm üretilir. Bu gerçekleştirilen işlem, DKA metasezgiselinin temel fikridir.

DKA yönteminin temel yapısının basit olması ve az sayıda parametre gereksinimi onu diğer metasezgisel yöntemlerden farklı kılmaktadır. Bundan dolayı, DKA ile daha basit yollarla çok iyi çözümler elde edilmektedir. Bu performansından dolayı DKA karmaşık problemler üzerinde daha verimli ve etkin bir şekilde uygulanabilmektedir.

Günümüze kadar DKA algoritmasının kullanıldığı birçok uygulama geliştirilmiş ve geliştirilmeye de devam edilmektedir. Bu çalışmalar daha çok ayırık (kombinatoryal) optimizasyon problemler üzerinde gerçekleştirilmiştir. Yapılan çalışmaların bazılarında farklı yöntemler ile hibrit bir sistem geliştirilirken, bazılarında da DKA üzerinde bir takım parametrik ve işlevsel değişiklikler yapılarak modifikasyonlar yapılmıştır. DKA algoritmasını sürekli optimizasyon problemleri üzerinde uygulayabilmek için ise hibrit yaklaşımlar ortaya atılmıştır.

Bunun yanında, DKA yöntemi pek çok optimizasyon problemleri üzerinde başarı ile uygulanmasına karşın, literatürde eğitilmiş YSA'dan kural çıkarımına yönelik herhangi bir çalışmaya rastlanılmamıştır.

Bu tez çalışmasında sınıflandırma problemleri için, eğitilmiş yapay sinir ağlarından (EYSA) bilgi kazanımına yönelik (sınıflandırma kuralları şeklinde) bir kural çıkarım algoritması geliştirilecektir. Geliştirilecek olan algoritma, YSA'daki nöronlar arasındaki bağlantı ağırlıklarında bulunan gizli bilgiyi ortaya çıkarmak için eğitilmiş yapay sinir ağları üzerinde çalışacaktır.

Önerilen algoritma, Değişken Komşuluk Arama (DKA) metasezgisel algoritmasına dayanmaktadır ve iki adımlı hiyerarşik bir yapıya sahiptir. İlk adımda, çok katmanlı bir sinir ağı eğitilmekte ve düğümler arası bağlantılarda bulunan ağırlık değerleri elde edilmektedir. İkinci adımda, elde edilen ağırlıklar DKA algoritması kullanılarak eğitilmiş yapay sinir ağından sınıflandırma kurallarının çıkarımı için kullanılmaktadır.

Bu tez çalışmasında sezgisel ve metasezgisel algoritmalar incelenecek ve metasezgisel yöntemlerden DKA algoritması sürekli optimizasyon problemlerine uygulanmak amacıyla geliştirilecektir. Gerçekleştirilecek olan DKA algoritmasında çözüm kalitesi, çözüm süreci ve algoritma sağlamlığı gibi bir takım kriterler ön planda tutulmaya çalışılacaktır.

2. KAYNAK ARAŞTIRMASI

Eğitilmiş Yapay Sinir Ağlarından (EYSA) kural çıkarımına yönelik çalışmalar ilk olarak teorik tartışmalar ve uygulamalar şeklinde gerçekleştirilmiştir. Araştırmacıların bazıları sezgisel yöntemleri kullanarak kural çıkarımı üzerinde çalışmışlar bazıları da hibrit modeller kullanarak bilgi işleme mantığıyla sınıflandırma kurallarının çıkarımı üzerinde çalışmışlardır. Aşağıda bu çalışmaların bazılarına kısaca yer verilmiştir.

Zhang ve ark. (1996) genetik algoritma (*GA*) kullanarak budamalı eğitilmiş sinir ağından kural çıkarımına yönelik bir metot sunmuşlardır. Her bir ağaç analiz edilerek her bir sınıf için kural kümeleri çıkarılmıştır. Çalışmada *RulExt* isiminde bir algoritma geliştirilmiştir. *RulExt* algoritmasının temelini standart çok katmanlı ileri beslemeli bir sinir ağı oluşturmaktadır. Çalışmada aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. İleri beslemeli sinir ağını standart geri yayılım algoritması ile eğitmiş ve eğitilmiş ağı *GA* kullanarak basitleştirmişlerdir. Algoritmalarında *GA*'daki her bir popülasyon, eğitilmiş ağın bağlantılarını ifade etmektedir.

Fukumi ve ark. (1998b) EYSA'dan kural çıkarımı için deterministik mutasyonlu bir genetik algoritmaya sahip evrimsel algoritma (*EA*) kullanan yöntem geliştirmişlerdir. Kullandıkları *EA*, rastgele optimizasyon metoduna dayanmaktadır. Deterministik mutasyonu sinir ağı öğrenme sonuçlarına uygulamışlardır. Basit kurallar çıkarabilmek için ağ boyutunun *EA* kullanılarak azaltılması gerekmektedir. Dolayısıyla *EA*'yı sinir ağ bağlantılarının sayısını azaltmak için de kullanmışlardır.

Zhenya ve ark. (1998) *PSO (Particle Swarm Optimization)* ismini verdikleri yöntemi kullanarak bulanık YSA'dan kurallar çıkarmakla ilgili, dört-katmanlı bir bulanık sinir ağını, girdi-çıkı örneklerinden bilgi kazancı elde etmek için önermişlerdir. Eğitim bilgisi ve evrimsel hesaplama temelli öğrenme algoritmasının eksikliklerini gidermek için, *PSO*'nun değiştirilmiş şeklini uygulamışlardır. Değiştirilmiş bu *PSO* algoritması, değişimin toplanmış oranı yerine, arama işlemini hızlandıran, önceki en iyi değeri güncellemek için komşulardaki her bir parçacığın güncel en iyi değerini kullanmaktadır.

Dorado ve ark. (2002) kural araştırma yöntemi olarak genetik programlamayı kullanan ve her türlü YSA'ya uygulanabilen bir kural çıkarım sistemi sunmuşlardır. Algoritmaları YSA'ları kara kutu olarak ele alan eğitimci bir yaklaşıma sahiptir.

Tsukimoto ve Hatano (2003), eğitilmiş üç katmanlı sinir ağının gizli katmanından ikili fonksiyonlar çıkarmak için bir metot geliştirmişlerdir. Elde edilen ikili fonksiyonlar gizli birimleri anlaşılır hale getirmektedir. Gizli katmandan çıkarılan ikili fonksiyon basit olduğunda, gizli birim fonksiyonel olarak belirlenmiş demektir. Eğer ikili fonksiyon karmaşıksa, bu durumda gizli birim fonksiyonel olarak belirgin değildir. Bundan dolayı çalışmalarında, gizli katmandan çıkarılan ikili fonksiyonların basitliği sayesinde, gizli birimlerin fonksiyonel belirginliğini ele almışlardır. Optimizasyon probleminin çözümünü genetik algoritma yöntemi kullanarak gerçekleştirmişlerdir. Uygunluk fonksiyonunu ise gizli birimler ve gizli birimlerden çıkarılan ikili fonksiyonlar arasındaki hataya göre tanımlamışlardır.

Fukumi ve Akamatsu (1998a), evrimsel algoritma (EA) kullanılarak EYSA'dan kurallar çıkarmak için bir yöntem önermişlerdir. Kullandıkları EA, deterministik mutasyonlu bir genetik algoritma ve deterministik mutasyonlu rastgele bir optimizasyon metoduna dayanmaktadır. Deterministik mutasyonu sinir ağı eğitime sonuçları üzerinde uygulamışlardır. Basit kurallar çıkarabilmek için ağ boyutunun EA kullanılarak azaltılması gerekmektedir. Bundan dolayı EA'yı sinir ağı bağlantılarının sayısını azaltmak için kullanmışlardır.

Elalfi ve ark. (2004) genetik algoritma kullanan EYSA'dan doğru ve anlaşılabilir kurallar çıkarabilmek için YSA'dan bağımsız olan bir metot sunmuşlardır. Çalışmalarında geliştirdikleri metot YSA eğitim algoritmalarından bağımsız ve eğitim sonuçları üzerinde herhangi bir değişiklik yapmamaktadır. Algoritmaları ayırık ve sürekli değişkenlere uygulanabilmekte, gizli birim aktivasyon fonksiyonu üzerine herhangi bir tahmin işlemi yapmamaktadır. Buna ek olarak EYSA'da herhangi bir sayıda gizli katmanı ele alabilmektedir. Veri kümesini ikili sayı formunda kodlamış ve EYSA'dan elde edilen ağırlıkları kullanarak YSA çıktı fonksiyonunu maksimize eden girdi vektörünü genetik algoritma kullanarak bulmuşlardır. Geliştirilen algoritma her bir sınıf için ayrı ayrı çalışmaktadır. Elde edilen kuralları YSA çıktı fonksiyon değerlerine

göre büyükten küçüğe doğru sıralamışlardır. Daha sonra her bir kuralı basitleştirerek anlaşılabilir dilsel kurallara dönüştürmüşlerdir.

Tokenaga ve ark. (2005) akıllı ve açıklayıcı değerlendirme sistemlerini oluşturmak için genetik programlama tekniğini kullanan çıkarım sistemini geliştirmişlerdir. Basit ve uygun sınıflandırma kuralları elde etmek için sinir ağlarını ve ikili sınıflandırmayı kullanmışlardır. Kuralları oluşturan ikili ifadeler nöronlar arasındaki ağırlıkların budanması ile elde edilmektedir. Yöntemi iflas tahmini için kural çıkarımı üzerine uygulamışlardır.

Santos ve ark. (2000), YSA'dan doğru ve anlaşılır kurallar çıkarmak için bir metot önermişlerdir. Geliştirmiş oldukları metot ağ topolojisi oluşturma ve sinir ağlarından kurallar çıkarma işlemlerini birleştirmektedir. Metot, tek gizli katmanlı ileri beslemeli sinir ağı topolojisi elde etmek için GA'yı kullanmaktadır. GA'da popülasyonda bulunan her bir birey *RPROP (Resilient Propagation)* algoritması ile eğitilen bir sinir ağı topolojisini ifade etmektedir. Bu topoloji, daha sonra çıktısı bir sınıflandırma kümesi olan kural çıkarım algoritmasına geçmektedir.

Hruschka ve ark. (2006) eğitilmiş çok katmanlı algılayıcılardan kurallar çıkarmak için bir kümeleme GA metodu geliştirmişlerdir. Kural çıkarım algoritması temel olarak iki adımdan oluşur. İlk olarak, bir kümeleme GA gizli katmandaki aktivasyon değerlerinin kümesini bulmak için uygulanır. Daha sonra bu kümeleri tanımlayan sınıflandırma kuralları giriş değerlerine bağlı olarak üretilir. Geliştirdikleri metot, EYSA'dan kurallar çıkarmaya yönelik ve gizli birim aktivasyon değerlerinin kümelenmesine dayanan *RX* algoritmasını temel almaktadır. EYSA'dan kural çıkarımıyla ilgili bir takım metotlar sadece bağlantı ağırlıkları bilgisine ihtiyaç duyarken, geliştirilen metot aynı zamanda eğitim kümesine de ihtiyaç duymaktadır. Yöntemlerinde başlangıçta tanımlanması gereken küme sayısı ile birlikte kümeleme sürecinin optimizasyonunu sağlamak için kümeleme problemlerine yönelik tasarlanan GA'dan yararlanmışlardır.

Kahramanlı ve Allahverdi (2009) eğitilmiş uyarlanabilir sinir ağlarından kural çıkarımı için yapay bağışıklık sistemi (YBS) algoritmasını sunmuşlardır. Ortaya konulan bu algoritma bütün giriş özelliklerini dikkate almakta ve uyarlanabilir transfer

fonksiyonuna sahip EYSA'dan etkin bir biçimde kurallar çıkarmaktadır. Algoritma veri kodlama, kodlanan verinin sınıflandırılması ve kural çıkarımı aşamalarını içermektedir.

Özbakır ve ark. (2009) EYSA'dan kural çıkarımı için *TACO (Touring Ant Colony Optimization)* adını verdikleri tur atan karınca koloni optimizasyon algoritmasını önermişlerdir. Önerilen metot YSA yapısındaki bağlantı ağırlıkları içinde bulunan gizli bilginin keşfi için kullanılmıştır. Bu çalışmada ilk olarak çok katmanlı ağ yapısındaki YSA eğitilerek uygun ağırlık değerleri elde edilmiştir. Daha sonra bu ağırlıklar kullanılarak, *TACO-Miner* algoritması aracılığı ile sınıflandırma kurallarını elde etmişlerdir.

Özbakır ve Delice (2011) EYSA'dan kural çıkarımı için *BPSO (Binary Particle Swarm Optimization)* ismini verdikleri ikili parçacık sürü optimizasyon metodu geliştirmişlerdir. Zaman değişimli *PSO* atalet ağırlığı ve hızlandırma katsayısı YSA çıkış fonksiyonunu optimize eden en iyi özellik değerini bulmak için kullanılmıştır. En uygun YSA ağırlık değerleri kullanılarak parçacıkların arama uzayını etkin bir biçimde taraması sağlanmıştır. Geliştirilen algoritma ilk olarak eğitim verisini ele almakta ve bu veri kümesini ifade eden kurallar çıkarılmaktadır. Daha sonra eğitim veri kümesinden bağımsız olan test veri kümesine elde edilen kuralların uygulanmasıyla kural kümesinin doğruluk değeri belirlenmektedir.

Kasiri ve ark. (2011) *GFS (Genetic Fuzzy System)* ismini verdikleri genetik bulanık sistem yöntemi ile YSA'dan kural çıkarım işlemini gerçekleştirmişlerdir. GA kullanan sinir ağından bulanık kural çıkarımı tribün girişindeki istenmeyen rüzgârı tanır. Böylece tribünde arzu edilen eğim açısı oluşturulur.

Marghny (2011) önerdiği metotta yapısal öğrenmeye sahip sinir ağı eğitmiş ve eşik değeri olmayan sinir ağının hatasının yakınsama oranının analizlerini sunmuştur. Bu problemi çözmek için GA kullanılmış.

EIAlami (2012) EYSA'dan kural çıkarımı için katmanlar arasındaki gizli düğüm ve ağırlıkların ortadan kaldırılmasıyla sinir ağının boyutunu indirgeyen tahribatlı bir teknik kullanmıştır.

Markowska-Kaczmar ve Wnuk-Lipinski (2004), *GenPar* adını verdikleri, sinir ağlarından kural çıkarımı algoritmasını önermişlerdir. Algoritmaları pareto optimizasyonlu genetik yaklaşıma dayanmaktadır. Çıkardıkları kurallar "Eğer... İse"

biçiminde önermeli kurallardır. Uygunluk fonksiyonunda doğruluk ve anlaşılabilirlik ölçütlerini ele almışlardır.

Markowska-Kaczmar (2005), bir önceki çalışmalarından farklı olarak sinir ağlarından kural çıkarmak için GA'ya dayanan *GEX* isimli algoritmasını geliştirmiştir. Yöntemin parametrelerinin son sonuçlardaki etkilerini deneysel çalışmalarla incelemiştir. Metodu, evrimsel yaklaşıma dayanmaktadır. İlk olarak genetik parametrelerin etkilerini incelemiş, daha sonra ise kural çıkarımında kullanacağı parametreleri test etmiştir. İkili, sürekli ve ayrık değişkenler için farklı kromozom yapıları kullanmıştır. Her popülasyonu, bir sınıfın kurallarını içerecek şekilde özelleştirmiştir. Bu da sınıflandırma probleminde mevcut olan sınıf sayısı kadar popülasyon olması anlamına gelmektedir.

Kamruzzaman (2012) EYSA'dan sembolik kural çıkarımı için *ERANN* ismini verdiği bir yöntem gerçekleştirmiştir. Ağ problemlerinin nasıl çözüleceğini daha iyi anlamak için YSA'dan bilgi çıkarımı üzerinde çalışılmıştır. Gizli katmanlardaki sürekli veriler kesikli hale getirmek için sezgisel kümeleme algoritması kullanılmış.

Craven ve Shavlik (1994) *TREPAN* metodunu önermişlerdir. Bu metod YSA'nın eğitilmesi için her hangi bir eğitim algoritmasına veya her hangi bir ağ yapısına ihtiyaç duymadan, sınıflandırma ağacı yapısından faydalanarak kural çıkarımı yapmaktadır.

Chen ve ark. (2006), *Ant-Classifier* isimli bir karınca koloni sınıflandırma kural çıkarımı yöntemi sunmuşlardır. Aynı zamanda sinir ağı grubunu ile Ant-Classifier sistemini birleştiren *NeAnt* algoritmasını önermişlerdir. Bu yöntemde sinir ağı yapısı eğitim kümesinin değerlendirilmesinde, Ant-Classifier algoritması ise yeni eğitim kümesinden kurallar çıkarma işleminde kullanılmıştır.

Kulluk'un (2009) yapmış olduğu tez çalışmasında, sınıflandırma problemleri için, EYSA'dan bilgi kazanımına yönelik bir algoritma geliştirmiştir. Önerilen algoritma, temelde Tur Atan Karınca Koloni Optimizasyon Algoritması (TAKKO) olarak bilinen bir metasezgiselle dayanmaktadır ve iki-adımlı hiyerarşik bir yapıya sahiptir. İlk adımda çok katmanlı algılayıcı tipi sinir ağı eğitilmekte ve ağırlıkları çıkarılmaktadır. Ağırlıklar elde edildikten sonra, ikinci adımda TAKKO algoritması sınıflandırma kurallarının üretimi için kullanılmaktadır.

Andrews ve ark. (1995) EYSA'dan kural çıkarımı için çalışma dönemine kadar geliştirilen önemli tekniklerin anket ve eleştirileri üzerinde çalışmalar gerçekleştirmişlerdir.

Arbatlı ve Akın (1997) EYSA'dan kural çıkarımı için GA kullanan bir yöntem önermişlerdir. Kural yapısının omurgasını oluşturmak için giriş özellikleri kullanılmıştır. Son olarak ilgili girdiler kullanılarak bağlaç kuralları çıkarılmıştır. Geliştirdikleri algoritmalarının temel farklılığı, YSA topolojisini optimize ederek kural çıkarmalarıdır. Kural çıkarımında bir metasezgisel yöntem yerine klasik bir kural çıkarım algoritması kullanmışlardır.

Mantas ve ark. (2006) sınıflandırma işlemi için EYSA'dan bulanık kural tabanlı bir sistem sunmuşlardır. Elde edilen bulanık sistem sinir ağına karşılık gelmektedir. Bulanık kurallarda girdi verileri ve ağırlık vektörleri arasındaki benzerlik kullanılmış. Böylece daha anlaşılır kuralları elde edilmiş.

D'Avila Garcez ve ark. (2001) EYSA'dan sembolik bilgilerin çıkarımı üzerine çalışmışlardır. Ağın giriş vektör kümesi üzerinde kısmi bir sıralama önermişler ve bir takım budama kuralları ile basit ve anlaşılabilir kuralları çıkarmışlardır.

Kulluk ve ark. (2013) bulanık kuralları daha anlaşılır ve kesin kurallara göre belirsizlik ve muğlaklıkla daha iyi başa çıktığı için YSA'dan bulanık sınıflandırma kuralları çıkarımı üzerine bir yöntem önermişlerdir. Veri madenciliği aracı olan *DIFACONN-miner* hesaplama yazılımını temel alınarak bulanık kural çıkarım yöntemi üzerinde çalışmışlardır.

Saito ve Nakano (1988) ikili verilere sahip problemlerde önce enine arama algoritmasını kullanarak bileşik kuralları çıkarmışlardır. Çalışmalarında kural kümelerinin sayısını azaltmak için iki kısıtlama kullanılmıştır. İlkinde derinlik adı verdikleri bir $t < n$ sayısı belirlemişler ve kural kümelerinin eleman sayısını en fazla t ile sınırlamışlar. İkinci kısıtlamada ise bazı kuralları kural kümesi içerisinde çıkararak elemişlerdir.

Ma ve ark. (2005) sundukları çalışmada PSO yöntemi ve ağırlıklandırılmış bulanık sinir ağına dayanan iki-amaçlı bir algoritma önermişlerdir. Bulanık sinir ağı topolojisi ve ağırlık parametresinin her ikisini de geliştirmiş ve çay tadı tanımlama uygulaması için en iyiye yakın bulanık sinir ağı yapısını elde etmişlerdir

Lin (2010) güç sistemlerinin kararlılığının ölçülmesi ve değerlendirilmesi için çok katmanlı algılayıcıya sahip YSA'dan kural çıkarımı üzerinde çalışmıştır.

Bhalla ve ark. (2012) trafo yağı içindeki çözünmüş gazların konsantrasyonlarını girdiler gibi kullanarak başlangıç arıza tespitlerine uygulanması ile YSA'dan kural çıkarımı üzerinde bir yöntem üzerinde sunum yapmışlardır.

Remm ve Alexandre (2002) çok katmanlı algılayıcı YSA'nın gizli katmanlarından bilgi ve kurallar çıkarma yöntemini geliştirmişlerdir. Çalışma radar hedef belirleme uygulaması üzerinde uygulanmıştır.

Krishnan ve ark. (1999) nöronları aktive eden girdilerin kombinasyonlarını bularak sinir ağlarından kural çıkarımı üzerine çalışmışlardır. Bir nöronun giriş ağırlıklarını sıralayarak arama uzayını budama işlemini gerçekleştirmişler.

Blum ve Socha (2005) örüntülerin sınıflandırılma problemleri için ileri beslemeli sinir ağlarının eğitiminde karınca koloni optimizasyon algoritmasını uygulamışlardır. Bu algoritmanın, GA'ya göre daha uygun olduğu sonucuna varmışlardır

Mohamed (2011) yapısal öğrenme yöntemi ile sinir ağını eğitme ve ağın basit yapısını elde etmek için yapısal bir metot yardımıyla öğrenilen eşikli veya eşiksiz bir sinir ağındaki hatanın yakınsama oranının analizini sunmuştur.

Hayashi ve ark. (2016) bir sinir ağının genişletilmiş ayrık girdiler ile nasıl eğitilebileceği yönünde bir çalışma yapmışlardır. İlave girişler, her bir sürekli özelliklerin orijinal aralığını eşit uzunluklara sahip alt aralıklara bölerek elde edilmiş. Önceden belirlenen minimum doğruluk gereksinim değeri elde edilene kadar orijinal sürekli özellikler gibi ayrık girdilerin çoğu budanarak ağdan kaldırılmış. Daha sonra ağın gizli katmanındaki aktivasyonların analiz edilmesiyle budanan ağdan daha anlaşılabilir sınıflandırma kurallarının nasıl çıkarılabileceğine değinilmiştir.

Özbakır ve ark. (2010) YSA'nın ileri beslemeli tipinden sınıflandırma kuralları elde etmek için bir yöntem önermişlerdir. Önerilen yaklaşım eğitim için diferansiyel gelişim algoritması ve kural çıkarımı için tur atan karınca koloni optimizasyon algoritmasını kullanmaktadır. Yönteme DIFACONN-miner ismini vermişler.

Tripathy ve ark. (2013) MTACO-Miner ismini verdikleri bir kural çıkarım algoritması geliştirmişlerdir. Önerilen yöntem etkili ve daha iyi bir sınıflandırma

yapmak için eşik değeri değiştirilebilen karınca kolonisi optimizasyonunda sürü zekası metodunu kullanmıştır.

Martens ve ark. (2007) çalışmalarında kural çıkarım işleminde destek vektör makineleri yöntemini kullanmışlardır.

Thabtah ve Cowling (2007) birleşik kurala dayalı bir açgözlü sınıflandırma algoritması önermişlerdir. RMR ismi verdikleri yöntem çok etiketli kurallar çıkarmaktadır.

Smaldon ve Freitas (2006) düzensiz kurallar kümesini keşfetmek için karınca kolonisi algoritmasının yeni bir yöntemini sunmuşlardır.

Çizelge 2.1’de, kaynak araştırması kısmında araştırması yapılan kaynaklara ait özet verilmiştir.

Çizelge 2.1. Kaynak araştırmalarının özeti

Yazar(lar)	Metot	Açıklama
Zhang ve ark. (1996)	GA	Budamalı EYSA’dan kural çıkarımı önermişler
Fukumi ve ark. (1998b)	GA+EA	EYSA’dan kural çıkarımı için deterministik mutasyonlu bir GA’ya sahip EA yöntemi geliştirmişlerdir.
Zhenya ve ark. (1998)	PSO	Bulanık YSA’dan kurallar çıkarmakla ilgilenmişler.
Dorado ve ark. (2002)	GA	EYSA’dan kural çıkarımı sunmuşlar
Tsukimoto ve Hatano (2003)	GA	Üç katmanlı EYSA’dan kurallar çıkarmak için bir metot geliştirmişlerdir.
Fukumi ve Akamatsu (1998a)	EA	EA kullanılarak EYSA’dan kurallar çıkarmak için bir yöntem önermişlerdir.
Elalfi ve ark. (2004)	GA	EYSA’dan kurallar çıkarabilmek için GA kullanan YSA’dan bağımsız bir metot sunmuşlardır.
Tokenaga ve ark. (2005)	GA	Akıllı ve açıklayıcı değerlendirme sistemlerini oluşturmak için GA kullanan kural çıkarım sistemini geliştirmişlerdir.
Santos ve ark. (2000)	GA	EYSA’dan kural çıkarımı sunmuşlar
Hruschka ve ark. (2006)	GA	EYSA’dan kural çıkarımı sunmuşlar
Kahramanlı ve Allahverdi (2009)	YBS	EYSA’dan kural çıkarımı için YBS algoritmasını sunmuşlardır.
Özbakır ve ark. (2009)	TACO	EYSA’dan kural çıkarımı için TACO adını verdikleri tur atan karınca koloni optimizasyon algoritmasını önermişlerdir.
Özbakır ve Delice (2011)	BPSO	EYSA’dan kural çıkarımı için BPSO ismini verdikleri ikili parçacık sürü optimizasyon metodu geliştirmişlerdir.
Kasiri ve ark. (2011)	GFS	GFS ismini verdikleri genetik bulanık sistem yöntemi ile EYSA’dan kural çıkarım işlemini gerçekleştirmişlerdir.
Marghny (2011)	GA	Yapısal öğrenmeye sahip sinir ağını eğitmiş ve

ElAlami (2012)	Tahribatlı bir teknik kullanmış	eşik değeri olmayan sinir ağının hatasının yakınsama oranının analizlerini sunmuştur. EYSA'dan kural çıkarımı için çalışmış
Markowska-Kaczmar ve Wnuk-Lipinski (2004)	GA	Pareto optimizasyonlu GA ile EYSA'dan kural çıkarımı için metot önermişler
Markowska-Kaczmar (2005)	GA	Bir önceki çalışmalarından farklı olarak EYSA'dan kural çıkarmak için yeni bir yöntem geliştirmiştir.
Kamruzzaman (2012)	Sezgisel Kümeleme	EYSA'dan sembolik kural çıkarımı için <i>ERANN</i> ismini verdiği bir yöntem gerçekleştirmiştir.
Craven ve Shavlik (1994)	Sınıflandırma Ağacı	<i>TREPAN</i> algoritmasını kullanarak kural çıkarma önermişlerdir.
Chen ve ark. (2006),	ACO	<i>Ant-Classifer</i> isimli bir karınca koloni sınıflandırma kural çıkarımı yöntemi sunmuşlardır.
Kulluk'un (2009)	TACO	EYSA'dan bilgi kazanımına yönelik bir algoritma geliştirmiştir. Önerilen algoritma, temelde Tur Atan Karınca Koloni Optimizasyon Algoritması olarak bilinen bir metasezgisel dayanmaktadır.
Andrews ve ark. (1995)	Anket	EYSA'dan kural çıkarımı için çalışma dönemine kadar geliştirilen önemli tekniklerin anket ve eleştirileri üzerinde çalışmalar gerçekleştirmişlerdir.
Arbatlı ve Akın (1997)	GA	EYSA'dan kural çıkarımı için GA kullanan bir yöntem önermişlerdir.
Mantas ve ark. (2006)	Bulanık Kural	Sınıflandırma işlemi için EYSA'dan bulanık kural tabanlı bir sistem sunmuşlardır.
D'Avila Garcez ve ark. (2001)	Budama Kuralı	EYSA'dan sembolik bilgilerin çıkarımı üzerine çalışmışlardır.
Kulluk ve ark. (2013)	<i>DIFACONN-miner</i>	Bulanık kural çıkarım yöntemi üzerinde çalışmışlardır.
Saito ve Nakano (1988)	Enine arama	İkili verilere sahip problemlerden kural çıkarımı üzerine çalışmışlardır.
Ma ve ark. (2005)	PSO	Bulanık sinir ağından kural çıkarımı yapmışlardır.
Lin (2010)	ÇKA	Güç sistemlerinin kararlılığının ölçülmesi ve değerlendirilmesi için çok katmanlı algılayıcıya sahip YSA'dan kural çıkarımı üzerinde çalışmıştır.
Bhalla ve ark. (2012)	Pedagojik yaklaşım	YSA'dan kural çıkarımı üzerinde bir yöntem üzerinde sunum yapmışlardır.
Remm ve Alexandre (2002)	Budama yöntemi	YSA'dan kural çıkarımı üzerinde bir yöntem üzerinde bir çalışma yapmışlardır.
Krishnan ve ark. (1999)	Budama yöntemi	YSA'dan kural çıkarımı üzerinde bir yöntem üzerinde sunum yapmışlardır.
Blum ve Socha (2005)	ACO	İleri beslemeli sinir ağlarının eğitiminde karınca koloni optimizasyon algoritmasını uygulamışlardır
Mohamed (2011)	Yapısal Öğrenme	Eşikli veya eşiksiz bir sinir ağındaki hatanın yakınsama oranının analizini sunmuştur.
Hayashi ve ark. (2016)	Budama Yöntemi	YSA'dan kural çıkarımı üzerinde bir yöntem üzerinde sunum yapmışlardır.
Özbakır ve ark. (2010)	<i>DIFACONN-miner</i>	Önerilen yaklaşım eğitim için diferansiyel gelişim algoritması ve kural çıkarımı için tur atan karınca koloni optimizasyon algoritması kullanmaktadır.

Tripathy ve ark. (2013)	MTACO-Miner	Eşik değeri değiştirilebilen karınca kolonisi optimizasyonunda sürü zekası metodunu kullanmıştır.
Martens ve ark. (2007)	SVM	kural çıkarım işleminde destek vektör makineleri yöntemini kullanmışlardır.
Thabtah ve Cowling (2007)	RMR	Birleşik kurala dayalı bir açgözlü sınıflandırma algoritması önermişlerdir.
Smaldon ve Freitas (2006)	ACO	Düzensiz kurallar kümesini keşfetmek için karınca kolonisi algoritmasının yeni bir yöntemini sunmuşlardır.

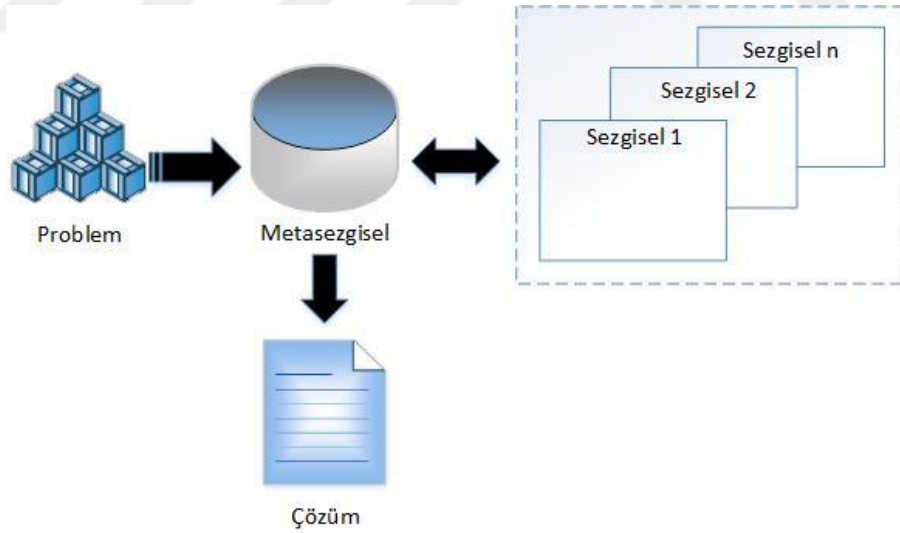


3. MATERYAL VE YÖNTEM

Eğitilmiş yapay sinir ağlarından (EYSA) kural çıkarım işlemi için gerçekleştirilecek olan çalışmada tek nokta ile arama yapan, çoklu komşuluk yöntemi kullanan ve hafızayı az kullanan Değişken Komşuluk Arama metasezgisel yöntemi kullanılacaktır.

3.1. Değişken Komşuluk Arama Yöntemi

Değişken Komşuluk Arama (DKA) yöntemi Mladenović ve Hansen (1997) tarafından geliştirilmiştir. Metasezgisel yöntemlerin sınıflandırılmasında DKA, tek nokta ile arama işlemi yapan, birden fazla komşuluk kullanan ve hafızayı az kullanan yöntemler arasında sayılır. Metasezgisel yöntemlerin çalışma prensibi Şekil 3.1’de gösterilmektedir.



Şekil 3.1. Metasezgisel algoritmanın çalışma prensibi

DKA, arama uzayını sistematik bir şekilde komşuluk yapılarının değiştirilmesi ilkesine göre tarayan bir metasezgisel yöntemdir. Algoritma oldukça genel bir çerçevede verilmiş olup, ele alınan problemin yapısına göre ayarlamalar yapılabilir. DKA yönteminin benzetimsel kodu Şekil 3.2’de verilmiştir.

```

Komşuluk yapılarını tanımla  $N_k$  ( $k=1, \dots, k_{max}$ )
Başlangıç çözümleri oluştur  $x \in X$ 
    Durdurma şartı gerçekleşinceye kadar devam et
         $k=1$  yap
        ( $k \leq k_{max}$ ) olduğu sürece tekrar et
             $x' = \text{Sarsma}(x), x' \in N_k(x)$ 
             $x'' = \text{Yerel Arama}(x'), x'' \in X$ 
            Eğer ( $\text{AmaçFonk}(x'') < \text{AmaçFonk}(x)$ )
                 $x = x''; k = 1; \text{ yap}$ 
            Değilse  $k = k + 1; \text{ yap}$ 

```

Şekil 3.2. DKA algoritmasının benzetimsel kodu

Bu algoritmaya göre ilk olarak komşuluk yapıları tanımlanmalıdır. N_k , ($k=1,2,\dots,k_{max}$) k . komşuluk yapısını gösterir. X , mümkün olan tüm çözüm kümelerini temsil ederken, $N_k(x)$ ise x çözümünün k . ($k=1,2,\dots,k_{max}$) komşuluk yapısındaki tüm çözümleri temsil eder (Şevkli, 2010).

Komşuluk yapılarını hem seçmek ve hem de sıralarını belirlemek algoritmanın performansı açısından oldukça önemli bir karardır. Genellikle komşuluk yapılarının sıralanması, yakın alandan uzak alana doğru arama yapabilecek bir şekilde belirlenir. Algoritma, X çözüm kümesi içerisindeki herhangi bir çözümü başlangıç çözüm kabul ederek çalışmaya başlar ve iç içe kullanılan döngüler sayesinde X çözüm kümesi içerisindeki çözümleri dolaşarak en iyi çözümü bulmaya çalışır.

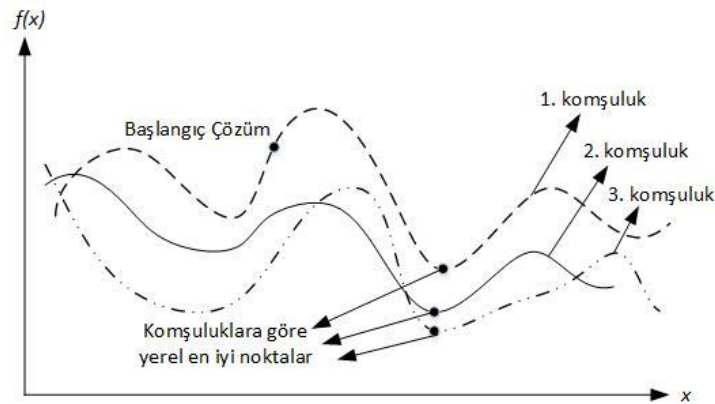
Algoritmada kullanılan *yerel arama (local search)* ve *sarsma (shaking)* fonksiyonları çözüm kümesi içerisinde sistematik olarak arama yapmayı sağlar. Sarsma, aramayı farklı yerlere konumlandırmayı sağlarken, yerel arama ise çözüm kümesi içerisindeki yerel en iyilere yoğunlaşmayı sağlar. Başka bir deyişle sarsma yerel en iyilerden kurtulmayı, yerel arama ise sarsma fonksiyonu sonucu oluşan x' çözümünün yakın komşularını arayarak o bölgedeki en iyi çözümü bulmaya çalışır. Yerel aramadan çıkan sonuç x'' , başlangıç çözüm x ile karşılaştırılır. Eğer x'' çözümü, x başlangıç

çözümünden daha iyi bir sonuç verirse algoritma birinci komşuluk yapısından ve x'' çözümü ile çalışmaya devam eder. Yerel aramadan çıkan sonuç kabul edilebilir bir değerde değilse algoritma x çözümünün bir sonraki komşuluk yapısından çalışmaya devam eder. İçteki döngü tanımlanan tüm komşuluk yapılarını dolaştıktan sonra sonlanır. Dıştaki döngü ise durdurma kriteri gerçekleşinceye kadar çalışmaya devam eder. Durdurma kriteri, belli bir CPU zamanı, belli bir tekrar miktarı veya iki iyileşme arasındaki en fazla tekrar miktarı olarak kabul edilebilir (Hansen ve Mladenovic', 2003b).

Hansen ve Mladenović (2003a) DKA yöntemini üç basit gerçek üzerinde temel almaktadır. Bunlar;

- Komşuluk yapısına göre yerel minimum değerleri için gerekli değildir;
- Global minimum muhtemel tüm komşuluk yapılarına göre yerel minimumdur;
- Birçok problem için yerel minimum bir veya birkaç komşuluğa göre bir birlerine göreceli olarak yakındır.

DKA, farklı yerel en iyiler elde etmek ve çalıştığı komşuluk yapısındaki yerel en iyilerden kurtulmak için kullandığı komşuluklar kümesini rastgele veya sistematik yöntemler kullanarak araştırır. Yerel arama işleminde çeşitli komşuluklar kullanılarak farklı yerel en iyiler oluşturulabilir ve ele alınan bir komşuluk yapısı için genel en iyi yerel en iyi olarak kabul edilir. Bu durumun temsili gösterimi Şekil 3.3'de belirtilmektedir.



Şekil 3.3. Komşuluk yapıları kullanarak değişken komşuluk arama

Gerçekleştirilen deneysel çalışmalarda, yerel en iyinin sıklıkla genel en iyi hakkında bir takım bilgileri sağladığı gözlenmiştir. Bu her ikisinin de aynı değere sahip birkaç değişken olabileceğini göstermektedir. Ancak, genellikle hangisinin yerel en iyi hangisinin genel en iyi olduğu bilinmemektedir. Yerel en iyinin komşulukları, en iyi değer bulunana kadar sırasıyla incelenecek ve yeniden organize edilecektir. Komşuluk yapıları kullanılarak ilgili problemleri çözmek için üç farklı yöntem kullanılabilir. Bu yöntemler sırasıyla deterministik, stokastik ve ikisinin bir arada kullanıldığı hibrit uygulamalar şeklindedir.

Günümüze kadar DKA yönteminin birçok farklı metodu geliştirilmiştir. Bunlar sırasıyla aşağıda belirtilmektedir.

3.1.1. Değişken komşuluk iniş

Değişken Komşuluk İniş (DKİ) metodu (Variable Neighborhood Descent) komşuluk değişimini deterministik bir yolla elde eder. Yöntemin işlem adımları Şekil 3.2’de gösterilmektedir (Talbi ve Dhaenens, 2009).

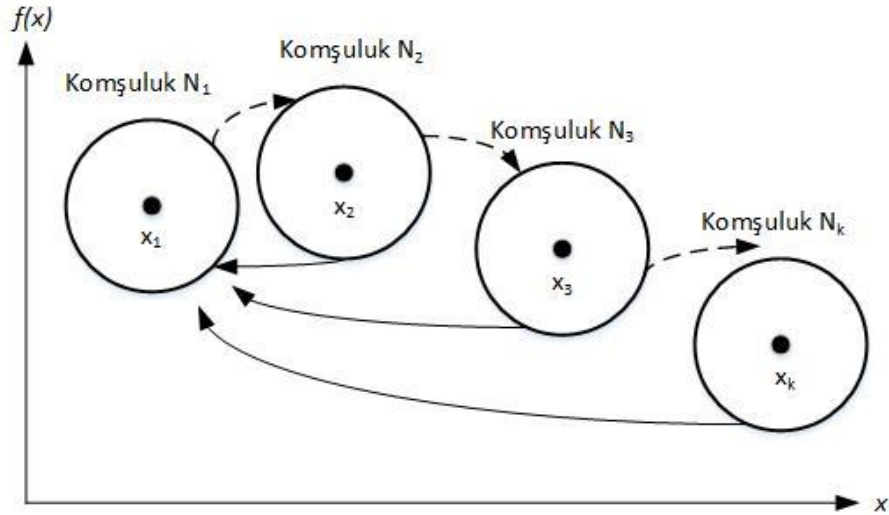
Birçok yerel arama sezgiseli tek veya bazen iki komşuluk inişi kullanır ($k_{max} \leq 2$). Son çözüm bütün komşuluklara (k_{max}) göre yerel en iyi olacaktır. Böylece genel en iyiye ulaşma şansı tek bir yapının kullanılması durumuna göre daha büyük olacaktır. DKİ’de komşuluk yapılarının sıralı olmasından dolayı yuvalanmış bir strateji geliştirilebilir. Varsayım olarak örneğin $k_{max} = 3$ olursa, o zaman mümkün olan yuvalama stratejisi, Şekil 3.4’de ilk iki komşuluk için DKİ’yi gerçekleştirir. Her bir x' noktası üçüncüye aittir ($x' \in N_3(x)$) (Hansen ve Mladenovic', 2003b).

DKİ yöntemi, DKA için belirtilen üç temel gerçekten birincisi dikkate alınarak geliştirilmiştir. Yani, $x = x' (N_1(x))$ komşuluğunda taşıma işleminin ilk tipi için yerel en iyi $x = x'' (N_2(x))$ komşuluğu içinde taşıma işleminin diğer tipine gerek duymamaktadır. Buda birleşik iniş yöntemi sezgisel yöntemle avantaj oluşturmaktadır.

1. Bir x başlangıç çözümünü bul.
2. Gelişme elde edilmediği sürece aşağıdakileri tekrar et:
 - (i) $k = 1$ yap;
 - (ii) $k = k_{max}$ oluncaya kadar aşağıdaki adımları tekrar et:
 - a. Komşuluğun Araştırılması. x çözümünün en iyi komşusu x' bul ($x' \in N_k(x)$);
 - b. Taşı veya Taşıma. Eğer x' çözümü x den daha iyi elde edilirse, $x = x'$ ve $k = 1$ yap; aksi takdirde $k = k+1$;

Şekil 3.4. Değişken komşuluk iniş metodu

Eğer x çözümü için her hangi bir gelişme elde edilirse komşuluk araması Şekil 3.5’de gösterildiği gibi tekrar $N_I(x)$ komşuluğundan başlatılır. Şekilde kesik çizgili oklar gelişmemiş çözümleri, düz çizgili oklar ise gelişmiş çözümleri göstermektedir.



Şekil 3.5. Değişken komşuluk iniş algoritmasının çalışma prensibi

3.1.2. İndirgenmiş değişken komşuluk arama

İndirgenmiş değişken komşuluk arama (İDKA) metodu (Reduced VNS) $N_k(x)$ komşuluğundan iniş takip edilmeksizin rastgele (stokastik olarak) seçilerek elde edilir. Yöntemin işlem adımları Şekil 3.6'de gösterilmektedir.

İDKA, yerel arama maliyeti çok büyük olan örnekler için avantajlıdır. Genellikle k_{max} parametresi için en iyi değer 2 olarak alınır. Durdurma koşulu olarak genelde iki gelişme arasında bulunan döngülerin maksimum sayısı kullanılır.

1. Bir x başlangıç çözümünü bul; bir durdurma koşulu seç;
2. Durdurma koşulu ile karşılaşıncaya kadar aşağıdaki işlemleri tekrar et:
 - (i) $k = 1$ yap;
 - (ii) $k = k_{max}$ oluncaya kadar aşağıdaki adımları tekrar et:
 - (a) Sarsma (Shaking). $N_k(x)$ 'den rastgele bir x' çözümü al;
 - (b) Taşı veya Taşıma. Eğer x' çözümü en iyi çözümden daha iyiyse, $(x = x')$ yap ve $N_1(k = 1)$ ile aramayı sürdür; değilse $k = k + 1$;

Şekil 3.6. İndirgenmiş değişken komşuluk arama metodu

İDKA yöntemi, DKA için belirtilen üç temel gerçekten ikincisi dikkate alınarak geliştirilmiştir. Şekil 3.6'da belirtilen İDKA işlem adımlarında, $N_1(x)$, $N_2(x)$, ..., $N_{kmax}(x)$ komşuluklarının kümesi geçerli bir x noktası (lokal optimum olabilir veya olmayabilir) etrafında dikkate alınmaktadır. Genellikle bu komşuluklar yuvalanmış durumdadır. İlk komşuluk içerisinden rastgele bir nokta seçilir. Seçilen bu noktanın değeri öncekinden daha iyi ise yani $f(x') < f(x)$ ise arama işlemi $x = x'$ yapılarak devam ettirilir. Aksi takdirde işlem bir sonraki komşuluk üzerinden devam ettirilir. Tüm komşuluklar üzerinde arama işlemi gerçekleştirildikten sonra, uygun bir durdurma koşulu sağlanıncaya kadar işlemler baştan itibaren tekrar ettirilir (Hansen ve Mladenovic', 2003b).

Bu tez çalışmasında EYSA'dan kural çıkarımı yapmak için iki komşuluk yapısına sahip İDKA algoritması kullanılmıştır.

3.1.3. Genel değişken komşuluk arama

Eğer yerel arama işleminde basit yerel arama metodu yerine DKİ algoritması kullanılırsa ve İDKA yöntemi kullanılarak daha gelişmiş bir başlangıç çözüm bulunursa Genel Değişken Komşuluk Arama (GDKA) metodu elde edilir. GDKA metodu Şekil 3.7'de gösterilmektedir (Hansen ve Mladenovic', 2003a).

Genel değişken komşuluk arama yöntemi komşulukların değişimini hem deterministik hem de stokastik yolla elde eder. Bu yöntemde durdurma koşulu olarak maksimum CPU süresi, maksimum yineleme sayısı veya iki iyileşme arasındaki maksimum yineleme sayısı ele alınabilir.

Başlangıç: İDKA kullanarak bir x başlangıç çözümü ve onun gelişmesini bul

Döngüler:

Durdurma koşulu ile karşılaşınca kadar aşağıdaki sıralamayı tekrar et:

- (i) $k=1$ yap;
- (ii) $k=k_{max}$ oluncaya kadar aşağıdaki adımları tekrar et;
 - a. Sarsma (Shaking), x çözümünün k . komşuluğunda rastgele bir x' noktası oluştur ($x' \in N_k(x)$);
 - b. DKİ ile yerel arama, $z=1$ yap; ve $z=z_{max}$ oluncaya kadar aşağıdaki işlem adımlarını tekrar et;
 1. $N_z(x')$ de x çözümünün x'' en iyi komşusunu bul
 2. Eğer $f(x'') < f(x')$ ise $x' = x''$ ve $z=1$ yap; aksi durumda $z= z+1$ yap;
 3. Taşı veya taşıma, Eğer bu yerel en iyi öncekinden daha iyi ise oraya ($x= x''$) taşı ve $N_I(k=1)$ ile aramaya devam et; aksi takdirde $k=k+1$ yap;

Şekil 3.7. Genel değişken komşuluk arama metodu

3.1.4. Temel deęişken komşuluk arama

Temel Deęişken Komşuluk Arama (TDKA) algoritması deterministik ve stokastik komşuluk deęişim işlemlerinin birleştirilmiş halidir. Yöntemin işlem adımları Şekil 3.8’de gösterilmektedir (Hansen ve Mladenovic’, 2003a).

Başlangıç. Aramada kullanılacak olan $k = 1, \dots, k_{max}$ için N_k komşuluk yapılarının kümesini seç. x başlangıç çözümünü bul; Durdurma koşulunu seç;

Durdurma koşulu ile karşılaşınca kadar aşağıdaki işlem adımlarını tekrar et:

1. $k = 1$ yap;
2. $k = k_{max}$ oluncaya kadar aşağıdaki işlem adımlarını tekrar et;
 - a. *Sarsma (Shaking)*. x çözümünün k . komşuluğunda rastgele bir x' noktası oluştur ($x' \in N_k(x)$);
 - b. *Yerel Arama (Local search)*. Başlangıç çözüm gibi x' ile yerel arama uygula; elde edilen yerel en iyiyi x'' ile göster;
 - c. *Taşı veya taşıma*. Eğer bu yerel en iyi öncekinden daha iyi ise oraya ($x = x''$) taşı ve $N_1 (k=1)$ ile aramaya devam et; aksi takdirde $k=k+1$ yap;

Şekil 3.8. Temel deęişken komşuluk arama metodu

Durdurma koşulu olarak maksimum CPU zamanı, maksimum döngü sayısı veya iki gelişme arasındaki döngülerin maksimum sayısı alınabilir. Genellikle ardışık komşuluklar yuvalanmış durumdadır. Adım 2(a)’da döngülerden kaçınmak ve aramayı farklı yerlere konumlandırmak için noktaların rastgele olarak üretildiği görülmektedir. Bu döngü ise eğer herhangi bir deterministik kural kullanılırsa ortaya çıkmaktadır. Ayrıca, yerel arama adımı (2b) DKİ yöntemi ile deęiştirilebilir. TDKA yaklaşımının birçok problemin çözümünde başarılı bir şekilde kullanıldığı görülmektedir.

3.1.5. Esnek deęişken komşuluk arama

Esnek Deęişken Komşuluk Arama (EDKA) metodunda (Skewed VNS) sarsma ve yerel arama işlemleri aynen kullanılır. Burada yerel arama işlemi sonucunda elde edilen x'' çözümünü kabul etme şartı DKA metoduna kıyasla esnetilmiştir. Yöntemde x'' çözümünün x çözümünden uzaklığı $\alpha\rho(x, x'')$ fonksiyonundan çıkan deęerden küçük ise çözüm kabul edilir. Burada, $\rho(x, x'')$ fonksiyonu x ile x'' arasındaki uzaklığı geri döndürürken, α iki çözüm arasındaki uzaklığın önem katsayısını belirler (Hansen ve Mladenovic', 2003a). EDKA yönteminin işlem adımları Şekil 3.9'da gösterilmiştir.

Başlangıç. Arama işleminde kullanılacak olan $k = 1, \dots, k_{max}$ için N_k komşuluk yapılarının kümesini seç. x başlangıç çözümünü bul; başlangıç $f(x)$ amaç fonksiyonunu oluştur. Durdurma koşulunu seç;

Durdurma koşulu ile karşılaşınca kadar aşağıdaki işlem adımlarını tekrar et:

1. $k = 1$ yap;
2. $k = k_{max}$ oluncaya kadar aşağıdaki işlem adımlarını tekrar et;
 - a. *Sarsma (Shaking)*. x çözümünün k . komşuluęunda rastgele bir x' noktası oluştur ($x' \in N_k(x)$);
 - b. *Yerel Arama (Local search)*. Başlangıç çözüm gibi x' ile yerel arama uygula; elde edilen yerel en iyiyi x'' ile göster;
 - c. *Taşı veya taşıma*. Eğer $f(x'') - \alpha\rho(x, x'') < f(x)$ ise $x = x''$ taşı ve N_1 ($k=1$) ile aramaya devam et; aksi takdirde $k=k+1$ yap;

Şekil 3.9. Esnek deęişken komşuluk arama metodu

3.1.6. Değişken komşuluk ayrıştırılmalı arama

Değişken komşuluk ayrıştırılmalı arama (DKAA) metodu (Variable Neighborhood Decomposition Search) temel DKA'nın problemin ayrıştırılmasına dayalı olan iki seviyeli DKA şeması içine genişletilmesidir (Hansen ve Mladenovic', 2001). DKAA yönteminin işlem adımları Şekil 3.10'da belirtilmektedir.

Başlangıç. Aramada kullanılacak olan $k = 1, \dots, k_{max}$ için N_k komşuluk yapılarının kümesini seç. x başlangıç çözümünü bul; Durdurma koşulunu seç;

Durdurma koşulu ile karşılaşınca kadar aşağıdaki işlem adımlarını tekrar et:

1. $k = 1$ yap;
2. $k = k_{max}$ oluncaya kadar aşağıdaki işlem adımlarını tekrar et;
 - a. *Sarsma (Shaking)*. x çözümünün k . komşuluğunda rastgele bir x' noktası oluştur ($x' \in N_k(x)$); y , x çözümünde bulunan mevcut k komşuluğunun bir çözüm kümesi olsun. Ancak x ($y = x' \setminus x$) de değil.
 - b. *Yerel Arama (Local search)*. y çözüm kümesinde yerel en iyiyi bul ve bulunan çözümü y' ile belirt ve x'' ile karşılaştır ($x'' = (x' \setminus y) \cup y'$);
 - c. *Taşı veya taşıma*. Eğer $f(x'') - \alpha \rho(x, x'') < f(x)$ ise $x = x''$ taşı ve N_1 ($k=1$) ile aramaya devam et; aksi takdirde $k=k+1$ yap;

Şekil 3.10. Değişken komşuluk ayrıştırılmalı arama metodu

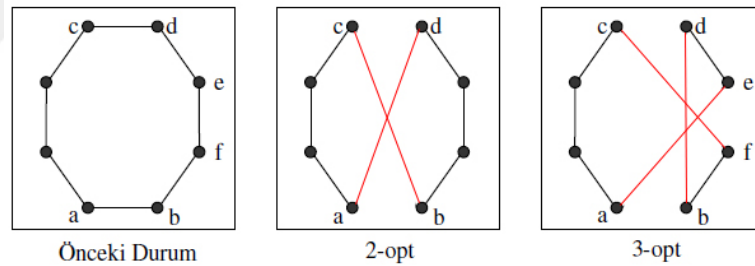
3.2. Komşuluk Yapıları

Komşuluk tabanlı algoritmalar, çözüm kümesi üzerindeki arama işlemini bir başlangıç çözüm kullanarak yerine getirmekte ve ele alınan bir komşuluk yapısını

kullanarak mevcut çözümleri optimize etmeye çalışmaktadırlar. DKA metodunda komşuluk yapılarını doğru tespit etmek ve yöntem içinde doğru bir şekilde sıralama yapmak metodun performansını doğrudan etkilediğinden dolayı en önemli işlemlerden biridir. Aşağıda çok sık kullanılan komşuluk yapıları hakkında bilgiler verilmektedir.

3.2.1. λ -opt yöntemi

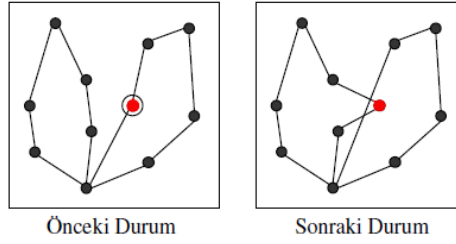
Gezgin satıcı problemleri için geliştirilen bu metotta λ adet doğru parçası mevcut rota içerisinde çıkarılarak olası tüm çözümlerde rotanın belirli noktalarına eklenmektedir. Mevcut çözümden daha iyi bir çözümün elde edilmesi durumunda metod, elde edilen bu yeni rotayı sonuç olarak değerlendirmektedir. λ -opt metodu, bir biri ile kesişmeyen doğrulardan oluşan bir rota elde etmeye çalışır (Laporte ve ark., 2000). Şekil 3.11’de λ -opt yönteminin örnek bir uygulaması gösterilmektedir.



Şekil 3.11. λ -opt yönteminin örnek gösterimi

3.2.2. Düğüm ekleme

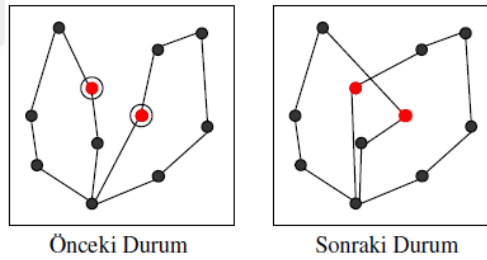
Düğüm ekleme yöntemi mevcut çözümün yakınında bir komşu çözüm elde etmek için çok kullanılan bir yöntemdir. Bu yapıda, çözüm içerisinde rasgele olarak seçilen bir kontrol noktası rasgele seçilen diğer bir kontrol noktasının önüne eklenir. Şekil 3.12’de düğüm ekleme yönteminin örnek bir uygulaması gösterilmektedir (Erol, 2006).



Şekil 3.12. Düğüm ekleme yönteminin örnek gösterimi

3.2.3. Düğüm yer değiştirme

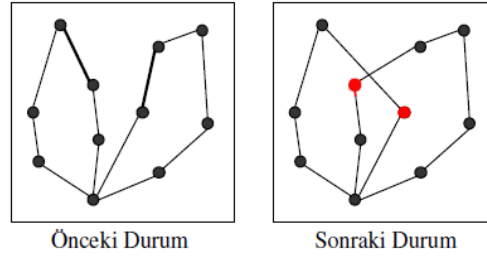
Düğüm yer değiştirme komşuluk yapısında permütasyon içinde rastgele iki kontrol noktası belirlenir ve daha sonra bu noktalar birbirleri ile yer değiştirir. Şekil 3.13’de düğüm yer değiştirme yönteminin örnek bir uygulaması gösterilmektedir (Erol, 2006).



Şekil 3.13. Düğüm yer değiştirme yönteminin örnek gösterimi

3.2.4. Yol yer değişimi yöntemi

Bu yöntemde farklı iki rota arasında, rasgele seçilen iki noktayı bir birine bağlayan bir yol (kenar) üzerinde yer değiştirme işlemi yapılarak yeni yollar elde edilir. Şekil 3.14’de yol yer değiştirme yönteminin örnek bir uygulaması gösterilmektedir (Erol, 2006).



Şekil 3.14. Düğüm yer değiştirme yönteminin örnek gösterimi

Yol yer değiştirme işlemi ile elde edilen yeni çözüm, en iyi çözümü çok daha uzak alanlarda arar.

3.2.5. Yol ekleme yöntemi

Yol ekleme yöntemi düğüm ekleme yöntemine benzemektedir. Aralarındaki tek fark çözüm üzerinde rasgele iki nokta seçmek yerine üç tane nokta seçmektir. Bu noktaların ikisi yolun başlangıç ve bitiş noktasını gösterir üçüncüsü ise yolun nereye ekleneceğini belirtir (Şevkli, 2010).

3.3. Yapay Sinir Ağları

Bu bölüm, çalışmanın temel konusu olan eğitilmiş yapay sinir ağlarından (EYSA) kural çıkarımına temel oluşturması amacıyla yapay sinir ağlarının (YSA) tanıtımına ayrılmıştır. Burada sırasıyla YSA'ların tanımı ve tarihçesi, avantaj ve dezavantajları ile günümüz teknolojisindeki uygulama alanlarına değinilmiştir. İnsan beynindeki biyolojik sinir sistemi ve biyolojik sinir sisteminden esinlenerek geliştirilen yapay sinir hücresi açıklanmıştır. Ayrıca bu bölümde yapay sinir ağlarının yapısı, aktivasyon fonksiyonları ve öğrenme algoritmaları gibi temel bir takım özelliklere de değinilmiştir.

3.3.1. Tanım ve tarihsel gelişimi

YSA, insan beyninin öğrenme yeteneğinden esinlenerek beynin yeni bilgiler keşfedebilme yeteneğini taklit ederek sistemlere herhangi bir yardım almadan otomatik öğrenme, genelleme yapma, hatırlama gibi kabiliyetleri kazandırmayı amaçlayan, birbirleriyle bağlantılı çok sayıda nöron birleşiminin matematiksel modeli ile oluşturulan bilgi işleme yöntemidir (Hush ve Horne, 1993).

YSA yapısı, birden fazla giriş değişkenini birkaç çıkış değişkenine doğrusal olmayan bir şekilde temsili olarak yapay ağ mantığı ile eşleştiren genel bir çerçeve sunar ve birçok geleneksel eşleştirme tekniklerinin uzantısı olarak kabul edilebilir. YSA yapısının en önemli görevlerinden biri ağırlıklar olarak bilinen parametre kümesinin ayarlanabilmesidir. Eğitilen YSA (EYSA) daha sonra çıktıları kategorize edilmiş sınıflandırma veya çıktıları sürekli değişkenler ile belirtilen tahmin problemlerinde kullanılabilir (Haykin, 1999).

YSA'nın eğitim sürecinde karşılaşılan önemli problemlerden biri, eğitim seti olarak isimlendirilen giriş / çıkış çiftlerinin kümesine karşılık gelen hatayı en aza indirgeyen ağırlık değerlerinin bulunması işlemidir. İlk aşamada, problem kısıtsız doğrusal olmayan bir optimizasyon problemidir. Burada karar değişkenleri ağırlıklardır ve amaç eğitim hatasını en aza indirmektir. Ancak YSA'nın tasarım ve eğitimindeki temel amaç, yeni girdiler için daha iyi bir tahmin yürüten bir model elde etmektir. Bu nedenle, EYSA özel ayrıntılardan ziyade eğitim verisinin sistematik yönlerini dikkate alır (Alba ve Marti, 2006).

3.3.2. Biyolojik hücre modeli

YSA'nın tanımlanmasında iki temel yaklaşım mevcuttur. Bunlar sırasıyla mühendislik problemlerine uygulanabilen biyolojik model ve doğrusal veya doğrusal olmayan istatistiksel modellerle ilgili olan matematiksel modeldir.

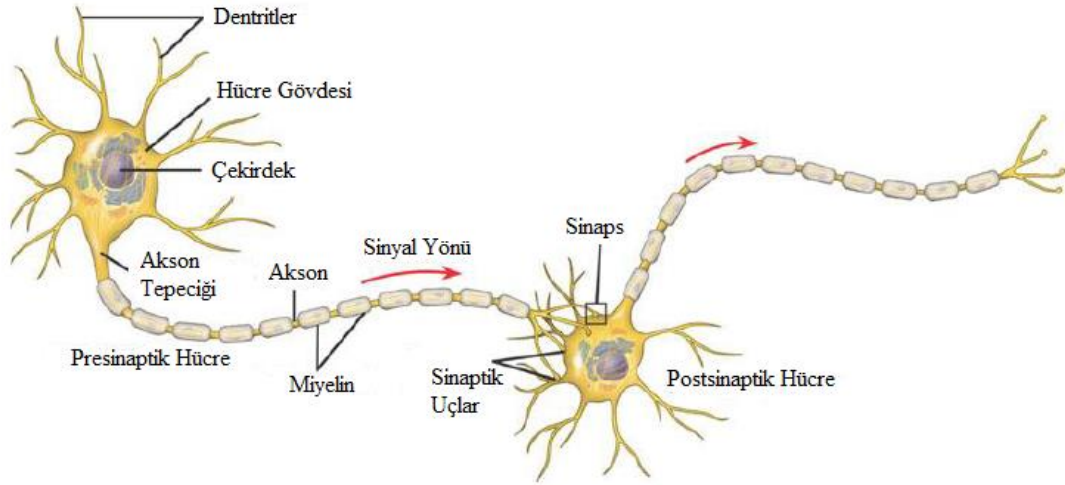
Biyolojik model matematiksel ve istatistiksel alanlarda pratik olarak uygulanabilen yeni YSA modellerinin geliştirilmesine öncülük sağlamaktadır (Soria ve ark., 2006). YSA modelinde üç temel yapı bulunmaktadır. Bunlar, geçerli bir problemi çözmek

için kullanılan sinir modeli, sinir modeli vasıtasıyla elde edilen çözümün kalitesi hakkında değerlendirme sağlayan maliyet fonksiyonu ve ele alınan problemin çözümünü sağlayan sinir modellerinin parametrelerini elde etmek için kullanılan öğrenme algoritmasıdır (Öztemel, 2003).

İnsan beyninde bulunan sinir sistemi, iç veya dış ortamlardan algıladıkları bir takım uyarıları elektriksel sinyallere dönüştürüp ve beyne ileten alıcı sinirlerden oluşmaktadır. Ayrıca iletilen elektriksel sinyalleri yorumlayan ve uygun bir karar üreten beyindeki elektriksel darbeleri uygun tepkilere dönüştüren tepki sinirlerinden de oluşmaktadır (Öztemel, 2003).

Yapay sinirin temel modeli biyolojik sinirin işlevselliğinden elde edilmiştir. Şekil 3.15’de insan beyninde bulunan biyolojik sinir hücreleri gösterilmektedir. Tanım olarak nöronlar, hücre gövdesinden yayılan N adet işlevi ayrık bir hücre şeklinde temsil eden her bir nöronun sinir sisteminin temel sinyal birimidir (Haykin, 1999).

Biyolojik nöronlar Şekil 3.15’de görüldüğü gibi 4 temel yapıya sahiptir. Bunlar sırasıyla 2 kapalı filiz, dentritler ve akson’a sahip hücre gövdesi şeklindedir. Hücre gövdesi temel olarak hücrenin kalbidir. Çekirdek içerir ve protein sentezini korur. Nöron, diğer nöronlardan sinyaller alan ağaç yapısına benzeyen birçok dentritten oluşmaktadır. Tek bir nöron genel olarak hücre gövdesinden genişleyen bir aksona (akson tepesi olarak etiketlenmiş) sahiptir. Aksonun temel amacı akson tepesinde üretilen elektrik sinyallerini aşağıya doğru dağıtmaktır. Bu sinyaller aksiyon potansiyelleri olarak etiketlenmektedirler. Aksonun diğer ucu ise presinaptik terminalde N adet dala bölünmüştür. Elektrik sinyalleri beynin verilerini nöronlara iletmektedir (Haykin, 1994).



Şekil 3.15. Biyolojik sinir hücreleri

Son derece karmaşık ve etkin bir yapıya sahip olan insan beyni yaklaşık 10^{11} sinir hücresinden ve $6 \cdot 10^{13}$ sinaps bağlantısından oluşmaktadır (Haykin, 1994). Bundan dolayı insan beyni; öğrenme, kavrama, genelleme ve uyarılma yetenek ve becerilerinden lineer olmayan bir bilgi işleme sistemi gibi yorumlanabilir (Karna ve Bren, 1989).

3.3.3. Yapay hücre modeli

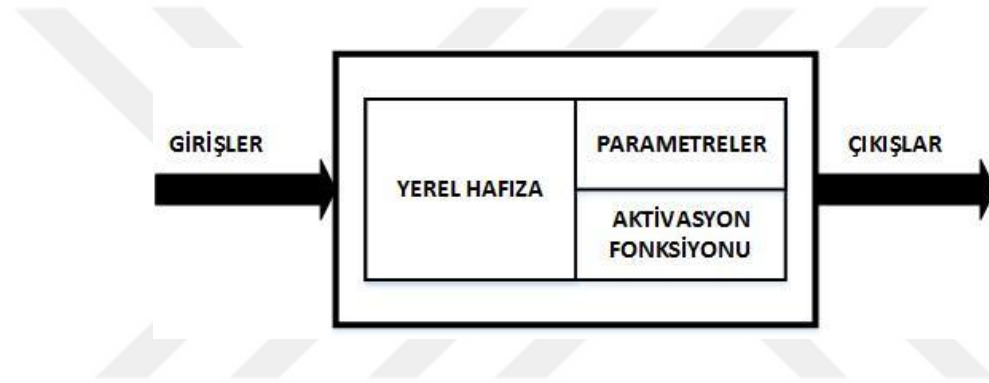
İnsan beyninin bütün davranış ve yeteneklerini modelleyebilmek için beynin biyolojik bileşenlerine karşılık gelen bir takım yapay hücre ve ağ topolojileri geliştirilmiştir. Bu sebeple, günümüzde kullanılan algoritmik yöntemeye dayalı hesaplama tekniklerinden farklı bir bilim dalı olarak “yapay sinir ağları” ortaya çıkmıştır (Pankaj ve Benjamin, 1992).

Yapay hücre modelinde, biyolojik hücre modelinde bulunan elemanlara karşılık gelen kavramlar aşağıda Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Biyolojik ve yapay hücre modellerinin karşılaştırılması

Biyolojik Hücre Modeli	Yapay Hücre Modeli
Nöron	İşlemci eleman
Dentrit	Toplama fonksiyonu
Hücre gövdesi	Transfer fonksiyonu
Aksonlar	Yapay hücre çıkışı
Sinapslar	Ağırlıklar

Bir hücrenin içyapısı Şekil 3.16’da gösterilmektedir. Bu yapı yerel hafıza birimi, parametreler ve aktivasyon fonksiyonundan oluşmaktadır (Weigend ve Gershenfeld, 1993).



Şekil 3.16. Bir hücrenin içyapısı

Yerel hafıza: Sinir ağlarında zaman serisi modeli tercih edileceği zaman bu bileşen kullanılır. Yerel hafızalar önceki girişleri veya önceki çıkışları depolayabilir. Böylece hücreler önceki davranışlarını hatırlayabilir. Hafızaların bu yöntemini kullanmak için birçok yöntem mevcuttur.

Parametreler: Bu parametreler genellikle sinaptik ağırlıklar olarak ta bilinir. Belli bir fonksiyona göre girişler ile birlikte kullanılır. Giriş ve katsayı vektörleri göz önüne alındığında en çok kullanılan fonksiyon skaler çarpım ve Öklid mesafe fonksiyonlarıdır. Sinaptik ağırlıklar ve giriş vektörü arasında yukarıda belirtilen birliktelik dolayısıyla doğrusal olmayan bir fonksiyondur.

Aktivasyon fonksiyonu: Bu fonksiyon karmaşık problemleri çözmek için sinir ağının yeteneklerini kullanabilmesinden dolayı çok önemlidir. En çok kullanılan aktivasyon fonksiyonları işaret fonksiyonu, hiperbolik tanjant, sigmoid ve Gauss fonksiyonudur.

YSA, insan beynindeki sinir sisteminin çalışmandan esinlenerek ele alınan sistemlere öğrenme, hatırlama ve genelleme yapma gibi bir takım yetenekleri kazandıran bir bilgi işleme yöntemidir (Öztemel, 2003). YSA, yapay sinir hücrelerinin farklı şekillerde birbirleri ile bağlanmasından oluşur. YSA'nın en önemli özelliklerinden birisi de onun öğrenme becerisinin olmasıdır. YSA'lar bir takım öğrenme algoritmalarını kullanarak eğitildikten sonra bilgi toplama, hücreler arası bağlantı ağırlıkları ile bilgi saklama ve genelleme yapma yeteneklerine sahip olurlar (Zurada, 1992).

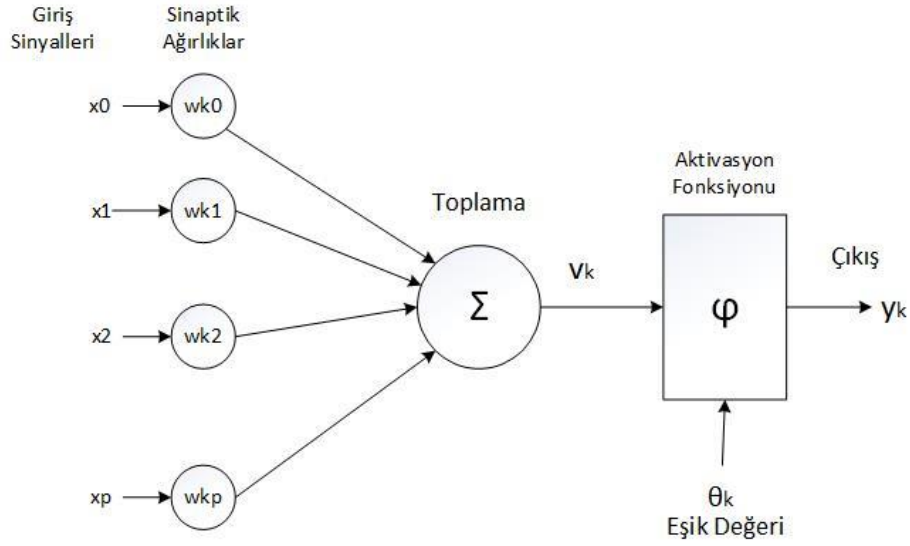
Biyolojik nöron modeli YSA'ya birkaç anahtar kavram ile uyarlanabilir. Bunlar sırasıyla, biyolojik nöronun sinapsları YSA'da ağırlıklar şeklinde modellenir. Biyolojik nöronun sinapsı sinir ağları arasında bağlantı yapar ve bağlantının gücünü sağlar. YSA'da ağırlık, sinapsı temsil eden bir değeri göstermektedir. Pozitif ağırlık değeri uyarıcı bir bağlantıyı temsil ederken negatif ağırlık değeri ise baskılayıcı bir bağlantıyı temsil eder. Şekil 3.17'de görüldüğü gibi tüm girişler bir araya toplanır ve ağırlıklar tarafından düzenleme yapılır. Aktivasyon fonksiyonu, çıkışın genliğini kontrol etmek için tanımlanmak zorundadır. Örnek olarak, pek çok durumda çıkış değerlerinin kabul edilebilir aralığı 0 ve 1 arasında yada -1 ve 1 arasında olmalıdır (Öztemel, 2003).

Şekil 3.17'de nöronun aktivitesi v_k ile belirtilmiştir. Ağırlık değerleri w_k ile belirtilmektedir. Burada y_k nöronun çıkışıdır ve v_k nin değeri üzerindeki aktivasyon fonksiyonudur.

$$v_k = \sum_{j=1}^p w_{kj} x_j \quad (3.1)$$

$$y_k = \sum_{j=1}^p w_{kj} x_j + \theta_k \quad (3.2)$$

Her bir giriş değerlerinde meydana gelen değişim, hücrenin çıkışında bir takım değişimlere yol açmakta ve bu değişimin değeri, girdinin etkisini belirleyen bağlantıların kazançlarına, toplayıcıda kullanılan eşik değere ve kullanılacak olan aktivasyon fonksiyonununun bağlı olmaktadır (Haykin, 1999).



Şekil 3.17. Basit bir yapay sinir ağı modeli

Girdiler (x_1, x_2, \dots, x_p), dış ortamdan alınan probleme ait bilgilerdir ve ağın öğrenmesi istenen örneklerin özellikleri tarafından belirlenirler.

Ağırlıklar (w_1, w_2, \dots, w_p), dış ortamdan alınan veriyi nörona bağlar ve ilgili girdilerin etkisini belirlerler. Ağırlıkların büyük yada küçük olması onların değerli veya değersiz olduğu manasına gelmez. Ağırlığın pozitif veya negatif olması sisteme olan etkisinin pozitif veya negatif olduğunu belirtir. Sıfır olması sistem üzerinde herhangi bir etkinin olmadığını belirtir. Ağırlıklar değişken veya basit değerler şeklinde olabilirler (Öztemel, 2003).

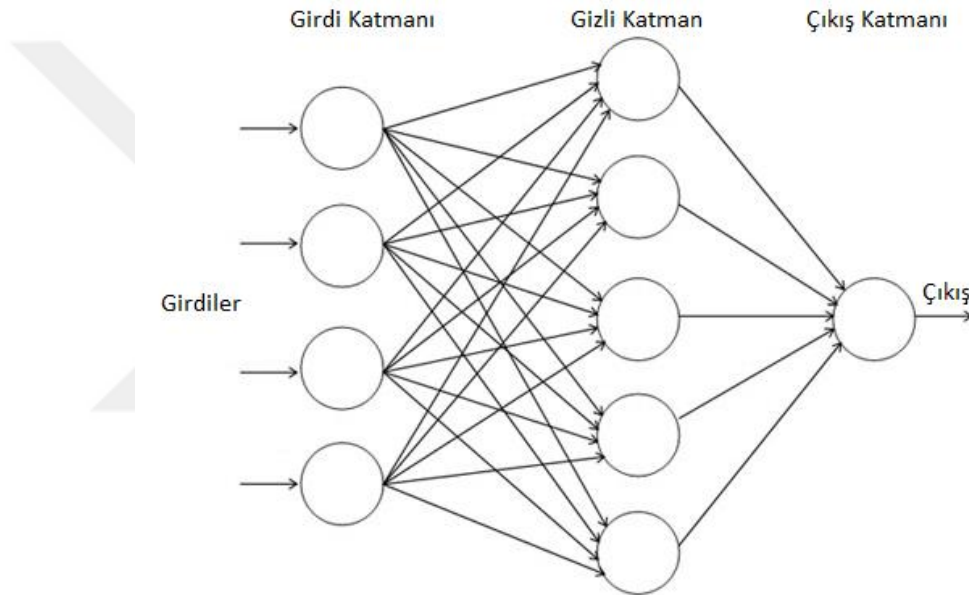
Toplama fonksiyonu, YSA'da bir hücreye gelen net girdiyi hesaplar. Toplama fonksiyonu için bir takım fonksiyonlar kullanılmakla birlikte en yaygın kullanılan fonksiyon ağırlıklı toplam fonksiyonudur. Burada sisteme gelen her girdi değeri kendi ağırlık değeri ile çarpılarak toplanır, daha sonra ise aktivasyon fonksiyonun eşik değeri bu toplama dâhil edilir. Böylece ağa gelen net girdi değeri hesaplanır (Şevkli, 2010).

Aktivasyon fonksiyonu, hücreye gelen net girdi değeri üzerinde işlem yaparak hücrenin bu girdiye karşılık üreteceği net çıktı değerini hesaplar. Toplama fonksiyonundaki gibi aktivasyon fonksiyonunda da bir takım formüller

kullanılmaktadır. Bu fonksiyon $f(x)=(x \geq 0 \text{ ise } 1; x < 0 \text{ ise } -1)$ şeklinde tanımlanabilir (Arbib, 2003).

Bir YSA aşağıdaki çerçevede işlemlerini yürütmektedir (Öztemel, 2003);

1. Giriş, nörona bir sinyal gibi gelir;
2. Sinyal hücre içerisine girer;
3. Sonuç olarak, hücre sinyali eşik değerine bağlı olarak çıkışa iletir;
4. Hücre tekrar sinyal almaya devam eder;



Şekil 3.18. YSA modeli

Şekil 3.18’de gösterildiği gibi YSA üç katmanlı bir mimariye sahiptir. Bunlar sırasıyla girdi katmanı, gizli katman ve çıktı katmanı şeklinde tanımlanmaktadır.

Girdi katmanı, dış ortamdan aldıkları bilgiyi gizli katmana taşırlar. Girdi katmanında bulunan veri üzerinde herhangi bir değişiklik yapılmaz.

Gizli katman, girdi katmanından aldığı verileri toplama fonksiyonu ve ağırlıkları kullanarak işler ve sonucu çıktı katmanına iletir.

Çıktı katmanı, gizli katmandan gelen bilgiyi aktivasyon fonksiyonu ve eşik değerini kullanarak işler ve girdi katmanındaki bilgilere karşılık gelen çıktıyı üretir.

3.3.4. YSA'nın temel özellikleri

YSA'nın temel özelliklerini aşağıdaki gibi maddeler halinde özetleyebiliriz (Gurney, 1997).

Paralellik: Genellikle işlemlerin seri bir şekilde çözümlenmesinden dolayı problemlerin hesaplanmasında hız problemi ile karşı karşıya kalınmaktadır. YSA'da gerçekleştirilen işlemler doğrusal değildir, işlemler bütün ağ üzerine yayılmış durumdadır ve katmanların arasında her hangi bir zaman bağımlılığı bulunmamaktadır. Bundan dolayı, tüm sistem eş zamanlı olarak çalışmakta ve hesaplama hızı da bariz bir şekilde artırmaktadır. Buda bize doğrusal olmayan karmaşık problemlerin çözümünde kolaylık sağlamaktadır.

Uyarlanabilirlik: Ele alınan bir problemi çözmek için YSA'da hücreler arası bağlantı ağırlıklarının güncellenebilir olması, eğitilen YSA'nın farklı durum ve koşullara göre yeniden adapte edilebilirliğini sağlamaktadır. Bu özellik, YSA'nın örnek tanıma ve sinyal işleme gibi çalışma alanlarında etkin bir şekilde kullanılabilmesine imkân sağlamıştır.

Genelleme: YSA, eğitim işlemi tamamlandıktan sonra eğitim aşamasında karşılaşılmayan test verilerini de değerlendirip istenilen tepkiler üretebilir. Örneğin, karakter tanıma probleminde bozuk karakterli verilerin girişinde doğru karakterlerin tespit edilmesi mümkündür.

Öğrenebilirlik: Geleneksel yöntemler belirli bir algoritma kullanmakta ve bağlantılarda bulunan ağırlıkları güncelleyememektedir. Bundan dolayı ele alınan bir takım problemler de çözülememektedir. YSA ile eğitme işleminde kullanılan örnekler kullanılarak ağırlıklar elde edilmekte ve her bir iterasyonda öğrenme işlemi yenilenmektedir.

Hata Toleransı: Geleneksel yöntemlere kıyasla YSA mevcut bilginin tüm bağlantılar üzerine yayılmasını sağlamaktadır. Bundan dolayı katmanlardaki bazı bağlantıların veya yapı içerisindeki hücrelerin etkisiz olması, ağın doğru sonucu üretmesini çok fazla etkilemez. Ağların geleneksel metotlara göre hatayı tolere edebilme yetenekleri oldukça yüksektir.

Yerel Bilgi İşleme: YSA'da problemin tümü üzerinde uğraşmak yerine sadece belli parçaları üzerinde çalışılması çok karmaşık ve zor problemlerin çözülebilmeye imkân sağlamaktadır.

Gerçekleme Kolaylığı: Karışık ve anlaşılması zor fonksiyonların kullanılması yerine basit yöntemlerin kullanılması problemin çözümü yönünde gerçekleştirme kolaylığı sağlamaktadır.

Donanım ve Hız: YSA, işlemleri paralel olarak işleyebilme yeteneği sayesinde bilgi işleme sürecinin hızını önemli derecede artırmaktadır.

3.3.5. Temel YSA mimarileri

YSA mimarisinin, sınırlar arası bağlantıların yönlerine göre ileri beslemeli ve geri beslemeli olmak üzere iki temel mimari yapısı mevcuttur.

3.3.5.1. İleri Beslemeli YSA

İleri beslemeli bir ağda düğümler katmanlar üzerinde bulunur. Giriş katmanı, dış ortamdan aldığı bilgileri üzerinde herhangi bir değişiklik yapmadan gizli katmandaki düğümlere iletir. Bilgi, gizli ve çıkış katmanları üzerinde işlenerek ağ çıkışına iletilir. Girişler, giriş katmandan başlayıp çıkış katmana doğru tek yönlü bağlantılarla iletilirken bir katmandaki çıkışlar bir sonraki katmana girdi olarak verilir (Rumelhart ve ark., 1986). Bir düğüm genellikle kendinden sonraki herhangi bir düğüme bağlanabilirken bazen kendisine de bağlanmaktadır. Son katmanda üretilen değerler ağın çıkışıdır. Herhangi bir anda elde edilen çıkış değeri, o anki girişin bir fonksiyonudur. Ağ statik yapılı bir fonksiyondur (Elmas, 2003).

İleri beslemeye sahip 3 katmanlı bir YSA'nın, gizli katmanında yeterli sayıda hücre olması şartıyla, herhangi bir sürekli fonksiyonu arzu edilen doğruluk değerine yaklaştırabileceği gösterilmiştir. En yaygın geriye yayılım öğrenme algoritması, YSA'ların eğitim işleminde çok kullanılmaktadır ve bu tip ağlara geriye yayılım ağları da denmektedir.

YSA'da ele alınan herhangi bir problemi çözümlmek için, katman sayısı ve gizli katmandaki hücre sayısı gibi belirsiz bilgilere rağmen ileri beslemeli YSA örnek tanıma, sinyal işleme, sınıflandırma ve tahmin gibi çalışma alanlarda çok yaygın olarak kullanılmaktadır.

3.3.5.2. Geri beslemeli YSA

Geri beslemeli bir sinir ağı, en az bir düğüm çıkışının kendisine veya diğer düğümlere giriş olarak verildiği ve geri beslemenin bir gecikme elemanı üzerinden yapıldığı ağıdır. Geri besleme, bir katmandaki düğümler arasında olduğu gibi katmanlar arasındaki düğümler arasında da olabilir (Haykin, 1994). Bu yapısından dolayı geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir ve herhangi bir andaki çıkış, hem o andaki hem de önceki giriş değerlerine bağlıdır.

Dolayısıyla, geri beslemenin yapılış biçimine göre farklı yapı ve davranışa sahip geri beslemeli YSA yapıları oluşturulabilir.

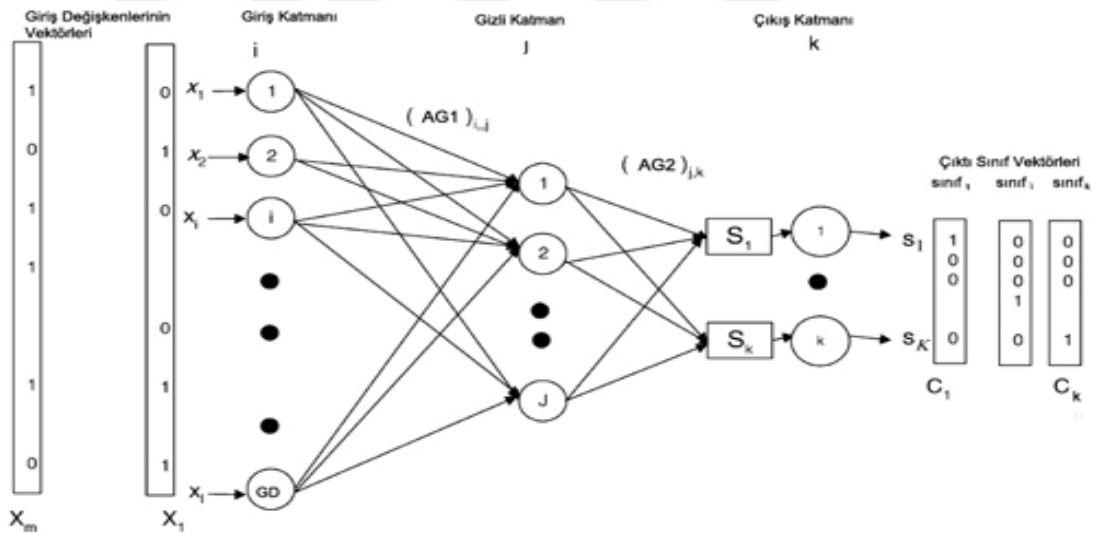
3.3.6. Çok katmanlı algılayıcı model

En çok kullanılan sinirsel model danışmanlı öğrenme algoritmasına sahip Çok Katmanlı Algılayıcı (ÇKA) modelidir. ÇKA modeli, özellikle sınıflandırma, tanıma ve genelleme gibi bir takım problemler üzerinde başarılı sonuçlar veren bir çözüm aracıdır.

ÇKA modeli, bir giriş katmanı, en az bir gizli katman ve bir çıkış katmanından oluşmaktadır. Her bir katmanda ele alınan probleme bağlı olarak bir veya daha fazla nöron bulunabilir. Rumelhart ve ark. (1986) tarafından geliştirilen bu modele hata yayma modeli veya geri yayılım modeli de denmektedir. Bu model günümüzde mühendislik problemlerinin hemen hemen hepsine çözüm getirmektedir. Özellikle problemler üzerinde sınıflandırma, tanıma ve genelleme gibi işlemler için çok önemli bir çözümlene aracıdır. Bu model delta öğrenme kuralı denilen bir öğrenme yöntemini kullanmaktadır. Sistemin temel çalışma amacı ağın beklenen çıktı değeri ile ürettiği çıktı değeri arasındaki hatayı en aza indirmektir (Öztemel, 2003).

Ele alınan tez çalışmasında, YSA modeli olarak günümüzde en yaygın kullanıma sahip danışmanlı öğrenme algoritmalarından ÇKA modeli kullanılacaktır. ÇKA'nın bir başka dikkat çeken yönü ise katmanlar arası tam bağlantı içermesi ve sınıflandırmada yaygın olarak kullanılmasıdır. Eğitilmiş ÇKA'dan elde edilen ağırlık değerlerine göre ÇKA'nın çıktı fonksiyonunu eniyileyen girdi vektörü, ağırlıkların nöronları aktif hale getirmelerine göre belirlenmiştir.

Şekil 3.19, YSA'nın giriş ve çıkış düğümlerine göre yapısını göstermektedir. Burada YSA ikili girdi ve çıktı vektör değerleri kullanılarak eğitilmektedir. Bu eğitim sonucunda, katmanlar arası uygun ağırlık değerleri elde edilmekte ve bu eğitime işlemi arzu edilen çıktı değerler elde edilinceye kadar devam ettirilmektedir.



Şekil 3.19. YSA'nın yapısı, (Elalfi ve ark., 2004)

Modelde yakınsama değeri, tekrar sayısı, gizli düğümlerin sayısı (J), öğrenme değeri ve momentum değeri değiştirilerek geliştirilebilir. YSA eğitildikten sonra iki grup ağırlık değeri elde edilebilir. İlk grup giriş katmanı ve gizli katman arasında bulunan ağırlık değerlerini $(AG1)_{i,j}$ içerir. İkinci grup ise gizli katman ve çıkış katmanı arasında bulunan ağırlık değerlerini $(AG2)_{j,k}$ içerir. YSA'nın gizli ve çıkış düğümlerinde kullanılan aktivasyon fonksiyonu sigmoid fonksiyonudur. YSA'da, j. gizli düğüme verilen toplam giriş $(GDG)_j$ aşağıdaki Eşitlik 3.3 kullanılarak hesaplanır (Elalfi ve ark., 2004).

$$GDG_j = \sum_{i=1}^I x_i (AG1)_{i,j} \quad (3.3)$$

YSA'da, j. gizli düğümün çıkışı (GDC_j) ise aşağıdaki Eşitlik 3.4 kullanılarak hesaplanır.

$$GDC_j = \frac{1}{1 + e^{-[\sum_{i=1}^I x_i (AG1)_{i,j}]} \quad (3.4)$$

YSA'da, k. çıkış düğümüne verilen toplam giriş (CDG_k) aşağıdaki Eşitlik 3.5 kullanılarak hesaplanır.

$$CDG_k = \sum_{i=1}^I (AG2)_{j,k} \frac{1}{1 + e^{-[\sum_{i=1}^I x_i (AG1)_{i,j}]} \quad (3.5)$$

Burada, k. çıkış düğümünün son değeri s_k ise aşağıdaki Eşitlik 3.6 kullanılarak hesaplanır.

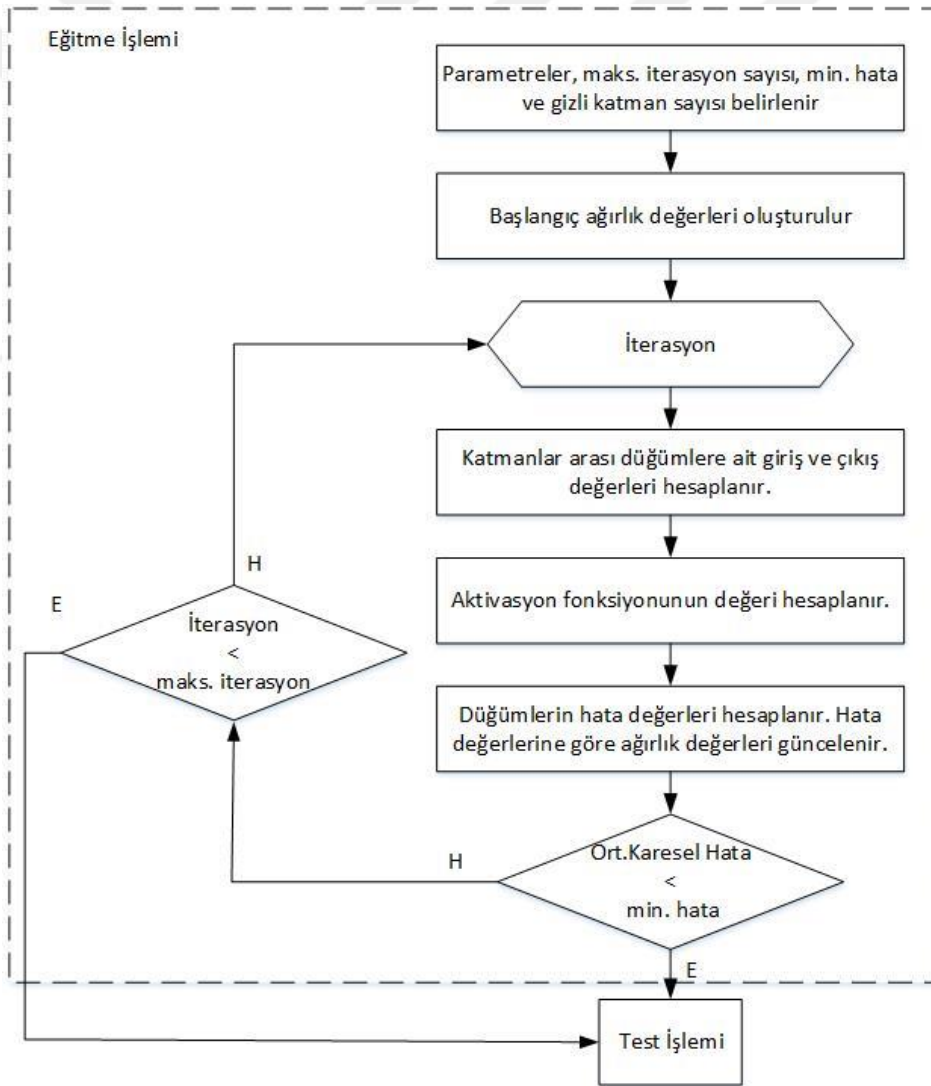
$$s_k(x_i) = \left\{ \frac{1}{1 + e^{-\left[\sum_{i=1}^I (AG2)_{j,k} \frac{1}{1 + e^{-[\sum_{i=1}^I x_i (AG1)_{i,j}]} \right]} \right\} \quad (3.6)$$

ÇKA ağlarında, ağa bir örnek ve bu örnek için üretilmesi gereken sonuç değeri verilir. Giriş katmanına girdi olarak verilen örnek veriler gizli katmanda işlenir ve çıkış katmanda çıktı değerleri elde edilir. Belirli bir eğitim algoritması kullanılarak ağ çıktısı ve arzu edilen çıktı arasındaki hata geriye doğru yayılır. Bu ağırlık güncelleme işlemi hata belirli bir değerin altına düşene kadar devam ettirilir. Burada, s_k üstel bir fonksiyondur ve alacağı en büyük çıktı değeri "1" dir.

Sonuç olarak, girdi nitelikleri ile ilgili sınıflar arasında kurallar çıkarabilmek için s_k fonksiyonunu maksimize eden girdi vektörünün bulunması gerekir. Bu problem bir optimizasyon problemidir (Elalfi ve ark., 2004);

Burada x_i ikili değerlere (0 veya 1) sahip girdi vektörüdür. Kısıtlar ikili ve amaç fonksiyonu ise doğrusal olmadığı için bu bir doğrusal olmayan tamsayı optimizasyon problemidir. DKA algoritması bu optimizasyon problemini çözmek için kullanılacaktır.

Şekil 3.20’de, ÇKA modele sahip YSA’nın işlem diyagramı gösterilmektedir. Bu tez çalışmasında ÇKA modelinin sadece eğitme kısmı kullanılacaktır.



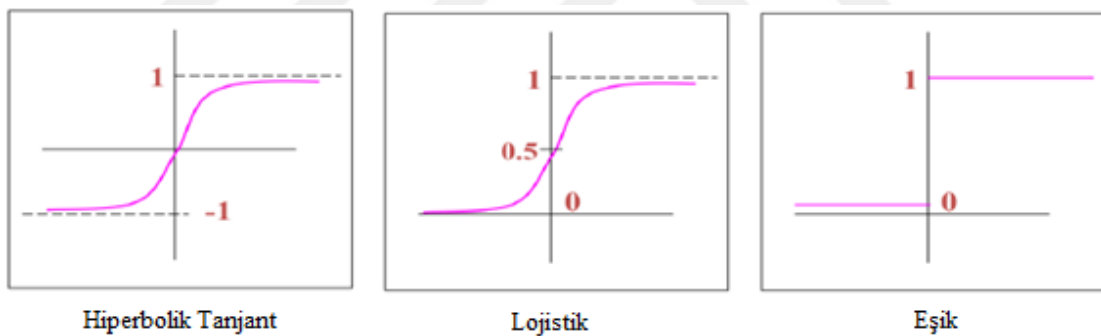
Şekil 3.20. ÇKA YSA modelinin işlem diyagramı

3.3.7. Aktivasyon fonksiyonları

Nöron davranışını belirleyen önemli etkenlerden biri nöronun aktivasyon fonksiyonudur. Biyolojik nöronlarda, toplam belli bir değeri aştığında nöronun kısa süreli bir darbe gönderdiği bilinmektedir. Bu davranışa benzer bir davranışı yapay nöronlarla da elde etmek için aktivasyon fonksiyonları kullanılır (Fauset, 1994).

Aktivasyon veya transfer fonksiyonları, YSA'da nöronun çıkış genliğini istenilen değerler arasında sınırlamak için kullanılmaktadır. Bu değerler genellikle $[0, 1]$ veya $[-1, 1]$ arasındadır.

YSA'da kullanılacak olan fonksiyonların türevi alınabilir ve süreklilik arz eden fonksiyonlar olması gerekir. Yapılan çalışmalarda daha çok sigmoid veya hiperbolik tanjant aktivasyon fonksiyonları kullanılmaktadır (Sağıroğlu ve ark., 2003). Aktivasyon fonksiyonları ve formülleri Şekil 3.21'de gösterilmektedir.



Şekil 3.21. Aktivasyon fonksiyonları

YSA tarafından öğrenilen kavramların anlaşılması zordur. Çünkü YSA ağırlık olarak adlandırılan büyük miktarda reel değerlere sahip parametre kümesinden oluşmaktadır. EYSA'yı anlamak için yapılan işlemlerden birisi, ağırlık sınıflandırma karakterini gösteren sembolik kuralları çıkarmaktır (Craven ve Shavlik, 1994).

Çalışmalarda hangi aktivasyon fonksiyonunun kullanılacağı problemin türüne ve özelliğine göre değişmektedir. Genellikle bu belirleme işlemi problemler üzerinde deneme yanılma yoluyla yapılan çalışmalar ile tespit edilmektedir.

YSA'larda en çok kullanılan aktivasyon fonksiyonları aşağıda Çizelge 3.2'de verilmiştir.

YSA sınıflandırma ve tahmin işlemlerinde yüksek doğruluk değerine sahiptir. Ancak YSA'nın sahip olduğu kara-kutu özelliği nedeniyle, ağın öğrenme işini nasıl yaptığı tam olarak anlaşılamamaktadır. Ağın yapısı incelendiğinde sadece ağırlık olarak adlandırılan, büyük miktardaki gerçek değerli parametre görülmektedir.

EYSA'nın üstünlüklerinden faydalanabilmek için ele alınan problemin daha anlaşılır bir yapıya dönüştürülmesi gerekmektedir. EYSA'dan sınıflandırma kuralları çıkarım işlemi, YSA'nın kara-kutu problemini gidererek, onu daha anlaşılır bir yapıya dönüştürmektedir (Delice, 2008).

Çizelge 3.2. Sık kullanılan aktivasyon fonksiyonları (Öztemel, 2003)

Aktivasyon Fonksiyonu Adı	Formül
Hiperbolik Tanjant	$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$
Lojistik	$f(x) = \frac{e^x}{1 + e^x}$
Eşik	$f(x) = \begin{cases} 0 & , x < 0 \\ 1 & , x > 0 \end{cases}$
Doğrusal Fonksiyon	$f(x) = x$
Parçalı Doğrusal fonksiyon	$f(x) = \begin{cases} 0 & , x \leq -0.5\alpha \\ \alpha x + 0.5 & , x < 0.5 \\ 1 & , x \geq 0.5\alpha \end{cases}$
Sigmoid	$f(x) = \frac{1}{1 + e^x}$

Gizli katmanları olmayan, sadece giriş ve çıkış katmanından oluşan ağlar, karmaşık problemleri hesaplama yeteneğinden yoksundur. Bundan dolayı karmaşık problemlerin hesaplanması için oluşturulan ağlarda en az bir gizli katman bulunmalıdır. Gizli düğümlerin sayısı belirlenirken ağın en iyi bir şekilde çalışmasını sağlayacak sayıda olması göz önünde bulundurulmalıdır.

Ağdaki düğüm sayılarının belirlenmesinde bir takım yöntemler kullanılmaktadır. Eğer gizli düğüm sayısı çok artırılırsa, istenen sonuç değerinin üzerinde bir değer elde edilir. Bu durum, ağ üzerinde bir genelleştirme problemini ortaya çıkaracaktır. Bir

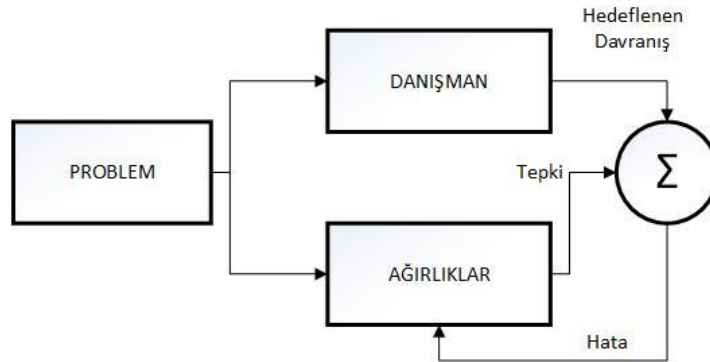
düğümün çıkışı, diğer bir düğüme giriş olarak bağlanır. Düğümler birçok giriş alır, fakat tek çıkış üretir (Elmas, 2003).

3.3.8. Yapay sinir ağlarında öğrenme

Bir öğrenme algoritması, belirli bir problemi çözmek için parametrelere veya sinir modelinin mimarisine uyarlanabilen bir yordamdır. Algoritmaları, budama metotları veya gürlleştirme metotları olarak bilinen çok katmanlı algılayıcı modelin mimarisine uyarlanmıştır. Budama metotları, gürlleştirme metotlarından daha yaygın bir kullanım alanına sahiptir (Haykin, 1999). YSA'nın eğitim işlemi için kullanılan öğrenme kuralları genellikle danışmanlı öğrenme ve danışmansız öğrenme olmak üzere iki başlık altında incelenmektedir.

Danışmanlı öğrenme kuralı, YSA'dan istenilen çıktının elde edilebilmesi için, çıktı hatasının düşürülmesinde ağırlıkların uyarlanabilir olmasını gerektirir. Danışmanlı öğrenmede her bir giriş değerine karşılık gelen çıkış değeri bellidir ve bunlar eğitime yöntemi ile sisteme tanıtılır. Daha sonra YSA'nın giriş-çıkış ilişkisi ayarlanana kadar ağırlıklar uyarlanır.

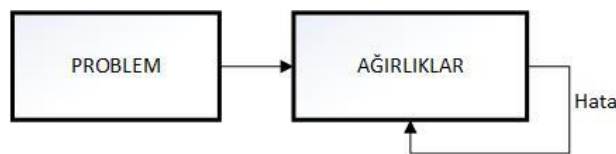
Burada, gerçek çıkış değeri istenen çıkış değeri ile karşılaştırılır. Başlangıçta rasgele olarak atanan ağırlık değerleri ağ tarafından güncellenerek, bir iterasyonda gerçek çıkış istenen çıkışa biraz daha yaklaşmış olur. Hata değeri kabul edilebilir eşik değerine yaklaşıncaya kadar ağırlık değerleri sürekli güncellenir. Öğrenme aşaması tamamlandıktan sonra elde edilen ağırlık değerleri sabit olarak kalır ve bir daha üzerinde her hangi bir değişiklik yapılmaz (Elmas, 2003). Danışmanlı öğrenme kuralının çalışma prensibi Şekil 3.22'de gösterilmektedir.



Şekil 3.22. Danışmanlı öğrenme kuralı

Danışmansız öğrenmede, YSA'da bulunan girişin hangi sınıfa ait olduğu veya ağın nerede daha iyi sonuç vereceği bilgisi bilinmez. Ağ veriyi üyeleri birbirine benzer olan öbeklere yol açmadan ayırır. Danışmansız öğrenme yöntemi danışmanlı öğrenme yöntemine göre çok daha hızlıdır. Ayrıca matematiksel algoritmaları da daha basittir (Elmas, 2003).

Danışmansız öğrenme kuralının çalışma prensibi Şekil 3.23'de gösterilmektedir.



Şekil 3.23. Danışmansız öğrenme kuralı

En çok kullanılan öğrenme algoritması, Geri Yayılım Algoritmasıdır. Algoritma anlaşılabilir ve kolay uygulanabilir olmasından dolayı birçok problemin çözümünde en çok tercih edilen öğrenme algoritmasıdır. Bu algoritma, hata değerlerini çıkış katmanından giriş katmanına doğru azaltmaya çalışması sebebiyle geri yayılım algoritması ismini almıştır. Geri Yayılım Algoritması, katmanlar arasında bulunan bağlantılardaki ağırlıkların, çıkış hatasına bağlı olarak optimize edilmesini amaçlar. Giriş ve giriş değerleri ile ilgili çıkış vektörü ağın eğitilmesi için kullanılmaktadır (Fauset, 1994).

Bir öğrenme algoritmasının en önemli karakteristiklerini aşağıdaki gibi belirtebiliriz (Haykin, 1999);

- *Verimlilik:* Bir problemi en az maliyet ile çözebilme yeteneği.
- *Sağlamlık:* Algoritmanın arzu edilmeyen gürültüye karşı bağımsızlığı bulunmalıdır.
- *Başlangıç koşullarından bağımsız:* Algoritma başlangıçta kullanılan verilerden bağımsız benzer çözümler göstermelidir.
- *Genelleme yetenekleri:* Algoritma, eğitim veri kümesinde farklı girişler kullanıldığı zaman uygun çıkışları sağlamalıdır.
- *Verilerin büyüklüğü ve karmaşıklığı ile ölçeklenebilirliği:* Algoritma bir problemin boyutunu ve büyüklüğünü dikkate almadan çözümü hesaplayabilmelidir.

Mevcut öğrenme algoritmalarının birçoğu Delta, Hebb, Kohonen ve Hopfield öğrenme kurallarından esinlenilerek geliştirilmiştir.

Hebb kuralı: Hebb (1949) tarafından geliştirilen bu yöntem diğer öğrenme kurallarının temelini oluşturmaktadır. Bu kuralın temelinde, eğer bir YSA hücresi diğer bir hücreden bilgi alırsa ve her iki yapay hücrede aktif ise iki YSA hücresi arasındaki bağlantının kuvvetlendirilmesi fikri vardır. Kural şu şekilde özetlenebilir; bir yapay hücre kendisi aktif ise bağlı olduğu YSA hücresini aktif yapmaya, pasif ise pasif yapmaya çalışmaktadır.

Hopfield kuralı: Bu kural Hebb kuralına benzemektedir. Hebb kuralından farkı kuralda YSA bağlantılarının ne derece kuvvetlendirilmesi veya zayıflatılması gerektiğinin belirlenmesidir. Eğer beklenen çıktı ve girdi değerlerin her ikisi de aktif veya pasif ise öğrenme katsayısı kadar ağırlık değerleri kuvvetlendirilir veya zayıflatılır. Ağırlık değerlerinin kuvvetlendirilmesi veya zayıflatılması işlemi öğrenme katsayısı kullanılarak gerçekleştirilmektedir. Öğrenme katsayısı 0-1 arasında ağ eğitimi öncesinde belirlenen sabit pozitif bir değerdir (Hopfield, 1982).

Delta kuralı: En yaygın kullanılan öğrenme algoritmalarından birisidir. Beklenen çıktı ile gerçekleşen çıktı arasındaki farklılığı azaltmak için YSA'nın elemanlar arası bağlantılarına ait ağırlık değerlerinin sürekli değiştirilmesi fikrine dayanmaktadır. YSA'nın ürettiği çıktı değeri ile beklenen çıktı değeri arasındaki hatanın

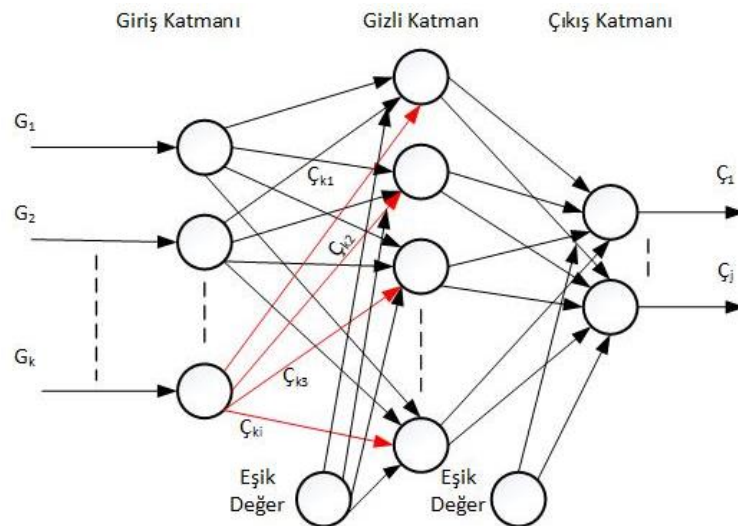
karelerinin ortalamasının en küçük değer olması amaçlanmaktadır. Hata, katmanlar arasında geri doğru yayılarak azaltılır. Bu kural, geri yayılım öğrenme kuralı olarak da adlandırılır (Widrow ve Hoff, 1960).

Kohonen kuralı: Kohonen (1990) tarafından geliştirilen bu kuralda hücreler öğrenmek için yarışır ve kazanan hücrenin ağırlıkları güncelleme işlemine tabi tutulur. Yarışta en büyük çıkış değerine sahip hücre kazanır. Kohonen kuralı, hedef çıkışa ihtiyaç duymadığından danışmansız öğrenme metodu olarak değerlendirilir.

ÇKA ağının öğrenme kuralı, en küçük kareler yöntemini temel alan delta öğrenme kuralının genelleştirilmiş halidir. Genelleştirilmiş delta kuralı iki kısımdan oluşur. Bunlar sırasıyla; ağın çıktılarının hesaplanma safhası olan *ileri doğru hesaplama* kısmı ve ağırlıkları değiştirme aşaması olan *geriye doğru* hesaplama kısmıdır.

3.3.8.1. İleri doğru hesaplama

Bu işlem eğitim setindeki bir örneğin girdi katmanından ağa iletilmesi ile başlar. Gelen girdiler hiç bir değişiklik yapılmadan gizli katmana gönderilir. İleri beslemeli çok katmanlı ağ modelinin çalışma prensibi Şekil 3.24’de gösterilmektedir (Zurada, 1992).



Şekil 3.24. İleri beslemeli çok katmanlı ağ modeli

Girdi katmanındaki k . işlem elemanının çıktısı ζ_{ki} , $\zeta_{ki} = G_k$ şeklinde belirlenir. Gizli katmandaki her bir işlem elemanı girdi katmanında bulunan bütün işlem elemanlarından gelen bilgileri hücreler arası bağlantı ağırlıklarının (w_1, w_2, \dots, w_j) etkisi ile ele alır. İlk önce gizli katmandaki işlem elemanlarına gelen net girdi aşağıdaki Eşitlik 3.8 kullanılarak hesaplanır.

$$NET_j^a = \sum_{k=1}^n w_{kj} \zeta_{ki} \quad (3.8)$$

Burada w_{kj} , k . girdi elemanını j . gizli katman elemanına bağlayan bağlantıya ait ağırlık değerini göstermektedir ve j . gizli katman elemanının çıktısı ise bu net girdinin aktivasyon fonksiyonu ile kullanılmasıyla hesaplanır.

Aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılması halinde çıktı değer Eşitlik 3.9 ile hesaplanır.

$$\zeta_j^a = \frac{1}{1 + e^{-NET_j^a \beta_j^a}} \quad (3.9)$$

Burada β_j , gizli katmanda bulunan j . elemana bağlanan eşik değer elemanına ait ağırlık değerini göstermektedir. Bu eşik değeri çıktısı sabit olup 1'e eşittir. YSA, eğitim işlemi sırasında bu değeri kendisi belirlemektedir.

Gizli katmanın bütün işlem elemanları ve çıktı katmanının işlem elemanlarının çıktı değerleri aynı şekilde kendilerine gelen NET girdi değerinin hesaplanması ve sigmoid fonksiyonunun kullanılması sonucu belirlenirler. Çıktı katmanından çıkan değerler bulunca ağırlık ileri doğru hesaplama işlemi tamamlanmış olur.

3.3.8.2. Geriye doğru hesaplama

Geriye doğru hesaplama algoritması, ÇKA modeli için ilk kullanılan algoritmadır ve geçerliliği günümüzde de halen devam etmektedir. Bu algoritmaya

bazen geri yayılım algoritması da denilmektedir. Amaç, maliyet fonksiyonun en küçük değerini elde etmektir.

Birçok uygulamada amaç fonksiyonun gerçek çözüm değerinin hesaplanabilmesi olanaksızdır. Geri yayılım algoritması ileri beslemeli ve çok katmanlı bir ağ mimarisine gerek duymaktadır. Anlaşılması ve yorumlanmasının kolay ve yöntemin matematiksel olarak kolayca ispatlanabilir olması sebebiyle en çok tercih edilen öğrenme algoritmasıdır. Hataları çıkış katmanından giriş katmana doğru azaltmaya çalıştığından dolayı geri yayılım algoritması olarak isimlendirilmektedir (Öztemel, 2003).

Yöntem istenilen çıkış değeri ile ağın çıkış değerinin arasındaki hatanın ağırlıklara bağlı olarak düşürülmesi prensibine dayanmaktadır. Geri yayılım algoritmasında eğitim veri kümesi kullanılarak ağ öğretilir ve ağa hedef değeri verilerek eğitime işlemi gerçekleştirilir. Öğrenme aşamasında, her bir örnek için ağdan elde edilen çıktı değeri ile hedef değeri karşılaştırılır (Witten ve Frank, 2005). Hata değeri, tekrardan ağa geri besleme biçiminde verilir. Eğitim kümesindeki hata kareleri toplamını en küçük hale getirmek için hücreler arasındaki bağlantı ağırlıkları güncellenir.

Problemlerde öğrenme algoritması olarak geri yayılım algoritması tercih edildiğinde öğrenme katsayısı (η) ve momentum katsayısı (α) parametreleri önem kazanmaktadır (Curram ve Minger, 1994).

Öğrenme katsayısı, YSA'da hücreler arasındaki ağırlıkların bir sonraki tekrarda hangi oranda değiştirileceğini göstermektedir. Öğrenme katsayılarındaki küçük değerler ağın sonuca ulaşmasını yavaşlatırken, büyük değerler ağın sonuca daha kısa sürede erişmesini sağlar. Ağın hesaplamalarında büyük salınımlara neden olduğundan ve ağın en iyi noktayı bulmasını engellediğinden dolayı yüksek değerlere sahip kat sayı değerlerinden kaçınmak gerekir (Fauset, 1994).

Momentum katsayısı ise, ağdaki salınımları engelleyerek ağın hata yüzeyindeki bölgesel minimum noktalarına takılmadan daha en iyi noktalara ulaşmasına yardımcı olur. En iyi öğrenme oranı ve momentum katsayısının belirlenmesi deneysel çalışmalar ile belirlenebilir (Haykin, 1994).

Ağa verilen girdi değeri için ağın ürettiği çıktı değeri beklenen çıktı değerleri ile kıyaslanır. Elde edilen fark değeri hata olarak kabul edilir. Amaç elde edilen hata değerinin düşürülmesidir. Bu hata değeri, bağlantılarda bulunan ağırlık değerlerine

dağıtılarak bir sonraki iterasyonda hata değerinin azaltılması sağlanır. Çıktı katmanındaki m . işlem elemanı için oluşan hata değeri $E_m = B_m - \zeta_m$ şeklinde hesaplanır. Burada B_m , beklenen çıktıyı belirtir ve ζ_m , ağın ürettiği çıktıyı belirtir.

Yukarıdaki E_m hatası, bir işlem elemanı için oluşan hatadır. Ağın çıktı katmanı için elde edilen toplam hatayı (TH) hesaplamak için bütün hata değerlerinin toplanması gerekir. Bu işlem için Eşitlik 3.10 kullanılır.

$$TH = \frac{1}{2} \sum_{i=1}^m E_i^2 \quad (3.10)$$

Toplam hatayı en aza indirmek için bu hataya neden olan işlem elemanlarına dağıtılması gerekmektedir.

Ağın ağırlıklarını değiştirmek için iki durum söz konusudur. Bunları aşağıdaki gibi maddeleyebiliriz;

- Gizli katman ile çıktı katmanı arasındaki bağlantı ağırlıklarının değiştirilmesi
- Gizli katmanlar arası veya gizli katman ile girdi katmanı arasındaki bağlantı ağırlıklarının değiştirilmesi

Gizli katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesinde aşağıda belirtilen işlem adımları dikkate alınmaktadır.

Gizli katmandaki j . işlem elemanını çıktı katmanındaki m . işlem elemanına bağlayan bağlantının ağırlık değerlerindeki değişim miktarına Δw denirse; herhangi bir t zamanındaki ağırlık değerinin değişim miktarı Eşitlik 3.11 ile hesaplanır (Öztemel, 2003).

$$\Delta w_{jm}(t) = \lambda \delta_m \zeta_j^a + \alpha \Delta w_{jm}(t-1) \quad (3.11)$$

Burada, λ öğrenme katsayısını, α momentum katsayısını göstermektedir. Momentum katsayısı, ağın öğrenmesi esnasında yerel bir en iyiye takılıp kalmaması için

ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar. Yukarıdaki formül dikkate alındığında δ_m , m . çıktı biriminin hatasını göstermektedir ve aşağıdaki Eşitlik 3.12 ile hesaplanır.

$$\delta_m = f'(NET)E_m \quad (3.12)$$

Burada $f'(NET)$ aktivasyon fonksiyonunun türevidir. Aktivasyon fonksiyonu olarak sigmoid fonksiyonun kullanılması durumunda Eşitlik 3.13 elde edilir.

$$\delta_m = \zeta_m(1 - \zeta_m)E_m \quad (3.13)$$

Değişim miktarı hesaplandıktan sonra ağırlık değerlerinin t . iterasyondaki yeni değerleri Eşitlik 3.14 ile hesaplanır.

$$w_{jm}(t) = w_{jm}(t - 1) + \Delta w_{jm}(t) \quad (3.14)$$

Benzer şekilde eşik değer biriminin de ağırlıklarını değiştirmek gerekmektedir. Çıktı katmanında bulunan işlem elemanlarının eşik değer ağırlıkları β_c ile gösterilirse; bu birimin çıktısının sabit ve 1 değeri olmasından dolayı değişim miktarı Eşitlik 3.15 ile hesaplanır.

$$\Delta\beta_m^c(t) = \lambda\delta_m + \alpha\Delta\beta_m^c(t - 1) \quad (3.15)$$

Eşik değer t . iterasyonundaki ağırlığının yeni değeri ise aşağıdaki Eşitlik 3.16 ile hesaplanır.

$$\beta_m^c(t) = \beta_m^c(t - 1) + \Delta\beta_m^c(t) \quad (3.16)$$

YSA'nın ara katmanı ile çıktı katmanı arasında bulunan ağırlık değerlerinin değişiminde her bir ağırlık değeri için sadece çıktı katmanında bulunan bir işlem

elemanının hatası göz önüne alınmıştır. Ancak bu hata değerlerinin oluşmasında girdi katmanı ve gizli katmanı arasındaki ağırlık değerlerinin önemi göz ardı edilemez.

Girdi katmanı ile gizli katman arasındaki ağırlık değerlerinin değişimi Δw_i ile gösterilirse değişim miktarı Eşitlik 3.17 ile hesaplanır.

$$\Delta w_{kj}^i(t) = \lambda \delta_j \zeta_k^i + \alpha \Delta w_{kj}^i(t-1) \quad (3.17)$$

Buradaki hata terimi δ^a aşağıdaki Eşitlik 3.18 ile hesaplanır.

$$\delta_j^a = f'(NET) \sum_m \delta_m w_{jm}^a \quad (3.18)$$

Burada, $f'(NET)$ aktivasyon fonksiyonunun türevidir. Aktivasyon fonksiyonu olarak sigmoid fonksiyonun kullanılması durumunda Eşitlik 3.19 elde edilir.

$$\delta_j^a = \zeta_j^a (1 - \zeta_j^a) \sum_m \delta_m w_{jm}^a \quad (3.19)$$

Değişim miktarı hesaplandıktan sonra ağırlık değerlerinin t . iterasyondaki yeni değerleri Eşitlik 3.20 ile hesaplanır.

$$\Delta w_{kj}^i(t) = w_{kj}^i(t-1) + \Delta w_{kj}^i(t) \quad (3.20)$$

Benzer şekilde eşik değer biriminin de ağırlık değerlerinin güncellenmesi gerekmektedir. Ara katmanda bulunan eşik değer ağırlıkları β^a ile gösterilirse değişim miktarı aşağıdaki Eşitlik 3.21 ile hesaplanır.

$$\Delta \beta_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1) \quad (3.21)$$

Eşik değer t . iterasyonundaki ağırlığa ait yeni değer ise aşağıdaki Eşitlik 3.22 ile hesaplanır.

$$\beta_j^a(t) = \beta_j^a(t-1) + \Delta\beta_j^a(t) \quad (3.22)$$

Böylece ağırlıklarının hepsi güncelleştirilmiş olacaktır. Bir iterasyonda hem ileri hem de geriye doğru hesaplama işlemleri yapılarak tamamlanmış olacaktır.

Yukarıda belirtilen adımlar, ÇKA ağının eğitilmesi tamamlanıncaya kadar devam ettirilir. Başka bir deyişle elde edilen ağ çıkışları ile hedeflenen ağ çıkışları arasındaki hata kabul edilebilir düzeye ininceye kadar tekrar ettirilir. ÇKA modellerinde, en büyük iterasyon sayısı ve kabul edilebilir hata değeri (minimum hata kriteri) sistemde durdurma kriteri olarak kabul edilmiştir.

Eğer ağ maksimum iterasyon sayısına erişmemişse, her iterasyonun sonunda hesaplanan karesel hata değeri minimum hata kriteri ile karşılaştırma yapılarak öğrenmenin gerçekleşip gerçekleşmediği incelenir. Ağı eğitme işleminin tamamlanması, ağ girişine sunulan verilere karşılık gelecek uygun çıkış değerinin üretilmesini sağlayan ağırlık değerlerinin elde edildiği anlamına gelmektedir. Eğitim tamamlandıktan sonra ağın başarısını ölçmek ve değerlendirmek için test veri kümesi kullanılmaktadır (Tezel, 2007).

3.4. Veri Madenciliği

Eğitilmiş YSA'dan DKA yöntemi kullanılarak elde edilen kurallar sınıflandırma kuralları şeklinde ve ağırlıklarda bulunan gizli bilgilerin keşfine yönelik bir çalışma olduğundan dolayı bu bölümünde veri madenciliği yöntemi ve sınıflandırma detaylı olarak incelenmiştir.

Veri madenciliğine olan ilginin en önemli sebeplerinden birisi bilgi teknolojisinde büyük miktardaki verilerin kullanılabilirliği ve bu büyük boyutlu verilerin içinden anlamlı bilginin elde edilmesine yönelik taleplerdir. Veri madenciliği yöntemi kısa ve öz bir tanımla, büyük ölçekli veriler arasından önceden tahmin edilemeyen bilgiye ulaşma işidir. Bir başka deyişle büyük veri yığınları içerisinde çalışma ile ilgili tahminlerde bulunabilmemizi sağlayabilecek kuralların bilgisayar programı kullanılarak elde edilmedir (Han ve Kamber, 2001). Bu teknoloji veriyi

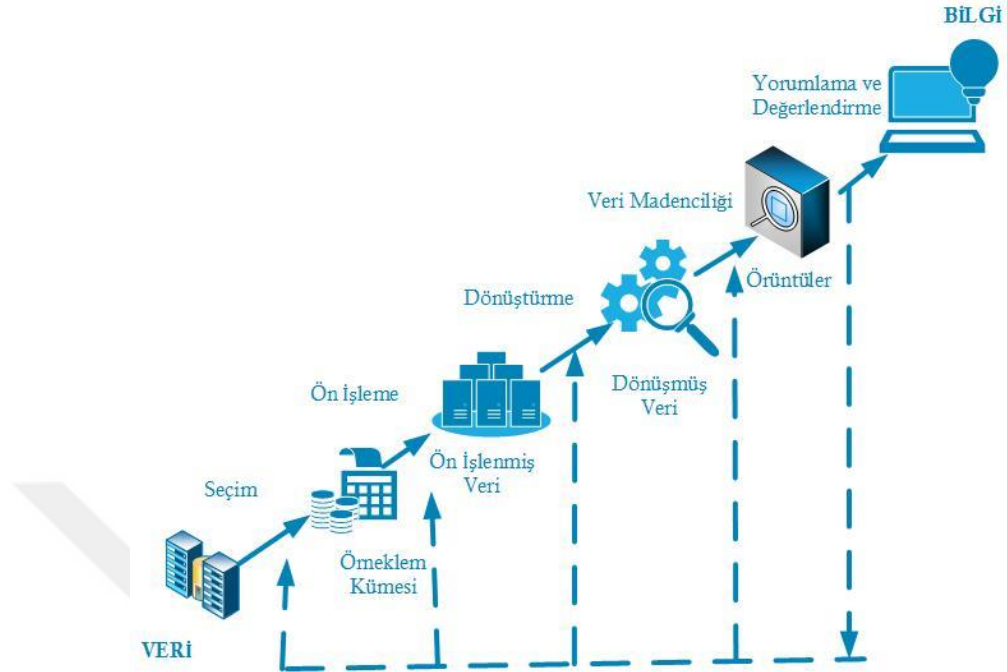
faydalı ve anlaşılabilir yeni biçimlere dönüştürmek ve aralarındaki beklenmedik ilişkileri bulmak için veri kümelerinin analiz edilmesi üzerinde çalışır (Argüden ve Erşahin, 2008).

Veri madenciliği, veritabanı yönetim sistemleri, veri ambarları, istatistik, makine öğrenme, veri görselleştirme, bilgi kurtarma ve yüksek performanslı hesaplama gibi çeşitli alanlar ile disiplinler arası çalışan yeni bir alandır. Diğer çalışılan alanlar ise YSA, örüntü tanıma, veri analizi, sinyal işleme ve ekonomi gibi birçok uygulama alanlarıdır (Hand ve ark., 2001).

Ele alınan problemleri çözmek, çözüm üzerinde kritik kararları almak veya geleceğe yönelik tahminlerde bulunmak için gereken bilgileri elde etmek temel çalışma prensiplerindedir. Elde edilmesi gereken bilgiler; karmaşık, belirsiz, önceden bilinmeyen, daha önce ortaya çıkarılmamış kullanışlı ve kritik bilgilerdir (Han ve Kamber, 2001).

Fayyad ve ark. (1996) veritabanlarından bilgi keşfine yönelik bir süreç öne sürmüşlerdir. Veri tabanından bilgiyi keşfetme işlemi, veri temizleme, veri bütünleştirme, veri seçme, veri dönüştürme, veri madenciliği, örüntü değerlendirme ve bilgi sunumu adımlarını içermektedir. Veri madenciliği bu sürecin içerisinde temel bir adımdır ve verilerden modeller veya örnekler elde etmek için veri analizleri ve bilgi keşif algoritmalarını kullanır. Bilgi keşfi süreci birkaç adımdan oluşan interaktif ve döngüsel bir süreçtir. Bu sürecin çalışma prensibi Şekil 3.25’de gösterilmektedir.

Süreç problemin ele alınması ile başlar ve problemin amaçları doğrultusunda hedef veri kümesi seçilerek devam eder. Bir sonraki aşamada, gürültülü ve tutarsız verilerin çıkarıldığı ön işleme süreci işletilir. Burada bir başka deyişle veri temizleme işlemi gerçekleştirilir. Veri madenciliği aşamasında, zeki yöntemler aracılığıyla büyük miktarda veriden anlamlı bilgileri çıkarılır. Çıkarılan örüntüler değerlendirilerek içlerinden yararlı olanlar değerlendirilir. Bilgi keşfi sürecinin son aşaması ise, elde edilen bilginin belirlenerek kullanıcıya sunulmasıdır (Tan ve ark., 2006b).



Şekil 3.25. Veritabanında bilgi keşfi süreci

Veri madenciliği herhangi bir veri deposuna uygulanabilen bir yöntemdir. Bu veri depoları, ilişkisel veritabanları, veri ambarları, işlevsel veritabanları, ileri veritabanı yönetim sistemleri, düz dosyalar ve web olarak sıralanabilir (Han ve Kamber, 2001).

3.4.1. Veri madenciliği işlem süreci

Veri madenciliği projelerinin daha verimli, hızlı ve maliyetsiz bir şekilde gerçekleştirilmesi için altı adımdan oluşan bir süreç kullanılmaktadır (Frawley ve ark., 1991).

Problemin Tanımlanması: Veri madenciliğinin ilk aşaması çalışmasının hangi amaç için yapılacağını net olarak tanımlanmasıdır. Veri madenciliği çalışmalarında başarılı bir sonuç elde etmek için uygulamanın hangi amaçla yapılacağını açık ve net bir şekilde ortaya konulması gerekmektedir. Amaç, açık ve anlaşılır bir dille ifade edilmeli, elde edilecek olan sonuçların başarılarının nasıl ölçüleceği önceden belirgin bir şekilde tanımlanmalıdır.

Veri Anlama: Bu aşama verilerin toplanması, mevcut verilerin uygunluğunun değerlendirilmesi, modeli oluşturmada kullanılacak veri ihtiyaçlarının tespiti, mevcut kayıt sayısının yeterliliği gibi işlemlerin gerçekleştirildiği bir aşamadır. Bu döngü kendi içinde devam ettikçe çalışmada kullanılacak olan veriler netlik kazanır.

Veri anlama aşaması başlangıç verilerini toplama, veri tanımlama, veri keşfi ve son olarak veri kalitesinin belirlenmesi adımlarından oluşmaktadır (Witten ve Frank, 2005).

Başlangıç Verilerini Toplama: Proje içerisinde kullanılacak olan başlangıç verilerinin ilgili veri kaynaklarından toplanması aşamasıdır.

Veri Tanımlama: Veri kaynaklarından toplanan verilerin tanımlanması ve yeterliliklerinin değerlendirildiği aşamadır.

Veri Keşfi: Başlangıçta toplanan veriler üzerinde bir takım hipotezlerinin kurulduğu ve verilerden çıkarımların yapıldığı aşamadır. Bu aşamada sonuca ulaşmada kullanılacak bilgilerin elde edilmesinden daha çok çalışmanın gerçekleştirilebilmesi için veri anlamında bulunan eksikliklerin tespit edilmesi amaçlanır.

Veri Kalitesinin Belirlenmesi: Verilerde eksiklik var mı, doğru mu, içerisinde bir takım hatalar içeriyor mu, eğer hatalar mevcut ise bu hataların türü ne şeklinde sorular ile veri kalitesinin belirlendiği aşamadır.

Veri Hazırlama: Veri madenciliğinin en önemli aşamalarından birisidir ve uygulamanın büyük bir bölümünü kapsamaktadır. Bu aşama başlangıçta bulunan verilerin, çalışmalara temel oluşturacak sonuç verilere dönüştürülmesidir. Bu çalışmanın adımlarına ait her hangi bir sıra veya tekrar sayısı yoktur. Modelin kurulması aşamasında ortaya çıkacak problemler, sürecin bu aşamaya geri dönülmesine ve verilerin yeniden düzenlenmesine sebep olabilmektedir.

Verilerin hazırlanması aşaması veri setini tanımlama, veri seçimi, veri temizleme, veri kurma, veri birleştirme ve veri biçimleme adımlarından oluşmaktadır (Ye, 2003).

Veri Setini Tanımlamak: Ele alınan problem için gerekli olan veri setinin belirlendiği aşamadır. Bazen problemler için büyük boyutlarda veri toplanıyor olmasına rağmen, toplanan veriler mevcut problem için yeterli olmayıp, başka veri gruplarından eklemelerin yapılması gerekebilir.

Veri Seçimi: Yapılacak olan analiz işleminde kullanılacak verilerin belirlendiği aşamadır. Değerlendirme sırasında verinin kalitesine de dikkat etmek gerekir. Problemden kullanılacak olan kayıt sayısı da önemlidir. Veri kümesinde gereğinden az veri bulunması durumunda çalışmanın değerlendirilmesi eksik olabilir. Gereğinden fazla veri ise veri kirliliğine ve işlem sürecinin uzamasına neden olabilir. Bundan dolayı veri seçimi dikkat edilmesi gereken önemli bir işlem adımıdır.

Veri Temizleme: Gürültülü ve tutarsız verilerin veri kümesinden çıkartılarak verinin kalitesinin artırıldığı aşamadır. Yanlış girilen verilerin, değerlendirmeye almadan veri kümesinden çıkartılması tercih edilir. Çok büyük boyutlu veri tabanı ile çalışmak yerine örnekleme yapılması uygun olabilir. Verileri temizlemek sadece veri kümesinden eksik verileri çıkarmak gibi düşünülmemelidir, bazen eksik verilerin tamamlanması için modellemeler yapmak da söz konusu olabilir.

Veriyi Kurmak: Verileri tanımlayan mevcut değişkenler üzerinde düzeltmeler yaparak model için daha kullanılabilir değişken setleri oluşturma aşamasıdır.

Veri Birleştirmek: Veri madenciliğinde problemler için kullanılacak verilerin farklı ortamlardan toplanması veri uyumsuzluklarına sebep olmaktadır. Bu uyumsuzlukların başlıca sebepleri verilerin farklı zamanlara ait olmaları, güncelleme sırasında meydana gelen hatalar, kullanılan veri formatlarının farklı olması, gerçekleştirilen kodlama farklılıkları, farklı ölçü birimleri ve varsayım farklılıklarıdır. Bu adımda farklı ortamlardan toplanan verilerdeki uyumsuzluklar mümkün olduğu oranda giderilerek, verilerin tek bir veri tabanında toplanması amaçlanır. Bu aşamada çok dikkatli ve titiz davranmak gerekmektedir. Yapılacak bir hata, ileriki aşamalarda daha büyük sorunların ortaya çıkmasına neden olacaktır.

Veri Biçimleme: Veri seti oluşturulduktan sonra kullanılacak modele göre anlam üzerinde her hangi bir değişikliğe meydan vermeyecek biçimlemelerin yapılması aşamasıdır.

Modelleme: Veri madenciliği problemleri için birden çok çözüm tekniği bulunmaktadır. Bazı teknikler verilerde bir takım özel durumlara ihtiyaçlar duyarlar. Bundan dolayı veri hazırlama ve model kurma aşamaları, en iyi modele ulaşıncaya kadar tekrar ettirilen bir süreçtir (Adriaans ve Zantige, 1996).

Verilerin modellenmesi aşaması aşağıda belirtilen adımlarından oluşmaktadır.

Model Tekniğini Seçmek: Kullanılacak veri madenciliği fonksiyonunun ve algoritmasının belirlendiği aşadır.

Model Test Tasarımı Yapmak: Modelin çalıştırılıp sonuçlarının elde edilmesine başlanmadan önce, modelin kalitesini ve geçerliliğini test etmek gerekmektedir. Verilerin hazırlanmasından sonra, ilk aşamada veri kümesinde bulunan verilerin bir kısmı modelin eğitilmesi, kalan kısmı ise modelin geçerliliğinin test edilmesi için ayrılır. Modelin eğitilmesi işlemi, eğitim kümesi kullanılarak gerçekleştirildikten sonra, test kümesi ile modelin doğruluk derecesi hesaplanır. Modelde yanlış olarak sınıflanan örnek sayısının, tüm örnek sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan örnek sayısının tüm örnek sayısına bölünmesi ile ise doğruluk oranı hesaplanır (Akpınar, 2000).

Sınırlı oranda veriye sahip olunması durumunda, çapraz geçerlilik test yöntemi kullanılır. Bu yöntemde ele alınan veri kümesi rasgele iki eşit parçaya bölünür. İlk aşamada, ele alınan bir veri parçası üzerinde model eğitimi ve diğer veri parçası üzerinde test işlemi; ikinci aşamada ise ikinci veri parçası üzerinde model eğitimi ve birinci veri parçası üzerinde test işlemi gerçekleştirilerek elde edilen hata oranlarının ortalaması kullanılır.

Aynı teknikle farklı parametre değerlerinin kullanıldığı veya başka algoritma ve yöntemlerin kullanıldığı farklı modeller oluşturulabilir. Modellemeye başlamadan önce, hangi tekniğin daha uygun olduğuna karar verebilmek oldukça güçtür. Modellerin doğruluk derecelerine göre en uygun modeli bulmak için bir takım denemelerin yapılmasında fayda görülmektedir. Ancak oluşturulan modelin doğruluk derecesi ne kadar yüksek olursa olsun, gerçek dünya problemlerini tam olarak modellediğini garanti etmek mümkün değildir.

Modeli Kurmak: Modelde kullanılacak metodun ele alınan veri kümesi üzerinde çalıştırılma aşamasıdır. Oluşturulan model doğrudan bir uygulama olabileceği gibi, bir başka uygulamaya ait bir alt parça olarak da kullanılabilir.

Modeli Değerlendirmek: Elde edilen tecrübeler ve test sonuçlarına göre modelin değerlendirildiği aşamadır. Burada modelin teknik olarak değerlendirilmesi amaçlanır.

Değerlendirme: Bu aşama, modelin tüm detaylarıyla değerlendirilip ve ana hedef ile uyumlu olup olmadığının kontrolünün yapılmasını amaçlar. Sonuçlar

değerlendirilerek modelin geçerliliği ve uygunluğu konusunda kararlar alınır. Bu aşamada modelin hedefleri ne ölçüde karşıladığı değerlendirilir. Modelleme esnasında meydana gelen fakat ana hedefler ile ilişkili olmayan bir takım ek faydaların da bu aşamada tanımlanması uygun olur (Argüden ve Erşahin, 2008).

Yayma: Modelin tamamlanmış olması projenin bitmiş olduğu anlamına gelmez. Modelin amacı veriler hakkında bir takım bilinen bilgileri artırmak dahi olsa, elde edilen veri kullanılacak biçimde düzenlenmeli ve sunulmalıdır. Bu işlem genellikle gerçek veriler içerisinde örneklerin sunulması şeklinde olur.

3.4.2. Veri madenciliği modelleri

Veri madenciliği yöntemi, veriyi giriş olarak alır ve örüntüler şeklinde çıktı üretir. Burada ele alınan işlem kural kümesi olarak da kodlanabilmektedir (Hand ve ark., 2001).

Veri madenciliği modellerini işlevlerine göre iki ana başlık altında incelemek mümkündür (Klösger ve Zytkow, 2002).

1. Tahmin Öngörü Fonksiyonları
 - Sınıflandırma
 - Regresyon Eğri Uydurma
2. Tanımlama Fonksiyonları
 - Kümeleme
 - Birliktelik Kuralları
 - Sıralı Dizi Analizi

Tahmin Öngörü Fonksiyonları: Geçmiş verilerden faydalanarak gelecekle ilgili bir sonucun tahmin edilmesi için kullanılan fonksiyonlardır. Model oluşturulurken geçmişteki deneyimlerde, faktörlerin aldığı bir takım değerlere göre elde edilen sonuçlar girdi değeri olarak kullanılır. Beklenen sonuç değeri kategorik değer veya rakamsal değerdir (Argüden ve Erşahin, 2008).

Sınıflandırma: Mevcut verilerden hareket ederek sonucun tahmin edilmesinde faydalanılan ve veri madenciliği teknikleri içerisinde en yaygın kullanım alanına sahip olan temel veri madenciliği fonksiyonlarından biridir. Modeli oluşturabilmek için, sonuçları önceden bilinen durumlar ve bu durumlarda ilgili faktörlerin yer aldığı eğitim verisi gereklidir. Modelin doğruluğunu tespit etmek için test verileri kullanılır (Akpınar, 2000).

Regresyon Eğri Uydurma: Sürekli değerlerin tahmin edilmesi için kullanılan fonksiyonlardır. Regresyon ile amaç, girdi değerler ile çıktıyı ilişkilendirecek modeli elde edip, en iyi tahmine ulaşmaktır. Sonuç “bağımlı değişken”, girdi değerleri ise “bağımsız değişken” olarak tanımlanır. Sonucun alacağı değer genellikle bir güvenlik aralığı içinde tanımlanır (Argüden ve Erşahin, 2008).

Tanımlama Fonksiyonları: Bu fonksiyonların temel amacı belirli bir hedefi tahmin etme değildir. Amaç veri kümesindeki veriler arasında bulunan ilişkileri, bağlantıları ve davranışları ortaya çıkarmaktır. Mevcut verileri yorumlayarak verilerin davranış biçimleri hakkında bir takım tespitler yapmak ve bu davranış biçimini belirten alt veri kümelerinin özelliklerini tanımlamayı hedefler (Argüden ve Erşahin, 2008).

Kümeleme: Kümeleme modellerinde amaç, veriyi birbirlerine benzeyen elemanlardan oluşan sınıflara bölerek, heterojen bir veri grubundan, homojen alt veri grupları elde edilmesi aşamasıdır. Başlangıç aşamasında veritabanında bulunan kayıtların hangi kümelere ayrılacağı veya kümelemenin hangi değişken özelliklere göre yapılacağı bilinmemektedir. Uzman kişiler tarafından kümelerin neler olacağı tahmin edilmektedir (Hand ve ark., 2001). Kümeleme algoritmaları; küme içi uzaklıkların minimize edilmesiyle küme içinde benzerliğin en üst seviyeye çıkarılması, kümeler arası uzaklıkların maksimize edilmesi ile kümeler arası benzerliğin en aza indirgenmesi kavramına dayanır. Sonuçta elde edilen farklı kümelere ait elemanlar arasında benzerlik azdır.

Birliktelik Kuralları: Eş zamanlı olarak gerçekleştirilen bir takım ilişkilerin tanımlanmasında kullanılır. Büyük veri kümeleri içerisinde bulunan farklı veriler arasındaki birliktelik ilişkilerinin bulunması işlemidir. Birliktelik analizi, belirli bir veri kümesi içerisinde yüksek sıklıkta birlikte görülen bir takım özellik değerlerine ait ilişki kurallarının keşfidir (Argüden ve Erşahin, 2008).

Sıralı Dizi Analizi: Gözlem sonuçlarını zaman ve mekân özelliklerine göre sıralanmış bir şekilde gösteren sayı dizileri olarak tanımlanır. Sayısal olarak sıralı durumda bulunan verilerdeki eğilimleri ve döngüleri anlamak ve tanımlamak için kullanılır. Bu fonksiyonda birbiriyle ilişkili olan kayıtlar incelenir ve zaman içinde sıkça karşılaşılan eğilimler ve benzer eğilimler bulunur. Bu eğilimler daha sonra veri içindeki bir takım ilişkileri tanımlamak için kullanılır (Han ve Kamber, 2001).

3.4.3. Veri madenciliği problemleri

Veri madenciliği büyük boyutlu veri kümeleri ile uğraştığından dolayı işlev içerisinde bir takım problemler ile karşılaşmaktadır. Bundan dolayı küçük veri kümeleriyle doğru çalışan sistemler büyük hacimli, eksik, gürültülü, boş değerli, artık veya dinamik verilerle yanlış çalışabilir. Veri madenciliğinde karşılaşılan bir takım problemleri aşağıdaki gibi belirtebiliriz (Sever ve Oğuz, 2002).

Veri Boyutu: Veri madenciliği uygulamalarında ele alınan veri kümesinin büyük boyutlu olması bulunacak örüntüleri ne kadar iyi tanımlıyorsa, bu büyük küme ile uğraşma zorluğu da o kadar artmaktadır. Veri madenciliğinde veri hacminin büyüklüğünden kaynaklanan problemlerle başa çıkmak için veri kümesinin yatay ve dikey boyutta indirgenmesi veya veri madenciliği metotlarının sezgisel bir yaklaşımla arama uzayını taraması gerekmektedir (Kulluk, 2009).

Gürültülü Veri: Verilerin toplanması veya sisteme girilmesi esnasında meydana gelen hatalar gürültü olarak ele alınır. Bu hatalardan dolayı veri tabanlarına girilen birçok niteliğin değeri yanlış olabilir. Veri madenciliğinde bu sorun üzerinde durulması gereken ciddi bir problemdir. Bu nedenle veri madenciliği yöntemlerinin gürültülü verilere karşı daha fazla duyarlı olması ve gürültülü verilerin sistem tarafından tanınarak dikkate alınmaması istenir (Hand ve ark., 2001).

Boş Değerler: Veri kümesinde bulunan boş değerler, bilinmeyen ve üzerinde işlem yapılamayan bir değer olarak dikkate alınır. Boş değerler ile çalışabilmek için bir takım yöntemler mevcuttur. Bunlar sırasıyla; boş değerleri dikkate almama, boş değerlerin belli bir değerle doldurulması işlemleridir (Quinlan, 1986).

Eksik Veri: Veri kümesindeki eksiklik nitelik veya niteliğe ait kayıtların eksik olması şeklindedir.

Artık Veri: Veri kümesinde gereksiz istenmeyen nitelikler artık veri olarak kabul edilir. Eksik verileri veri kümesinden çıkarmak için geliştirilen algoritmalar özellik seçimi algoritmaları olarak bilinir. Özellik seçimi, arama uzayı içerisinde daraltma yaparak sınıflandırma işleminin kalitesini artırır (Adriaans ve Zantige, 1996).

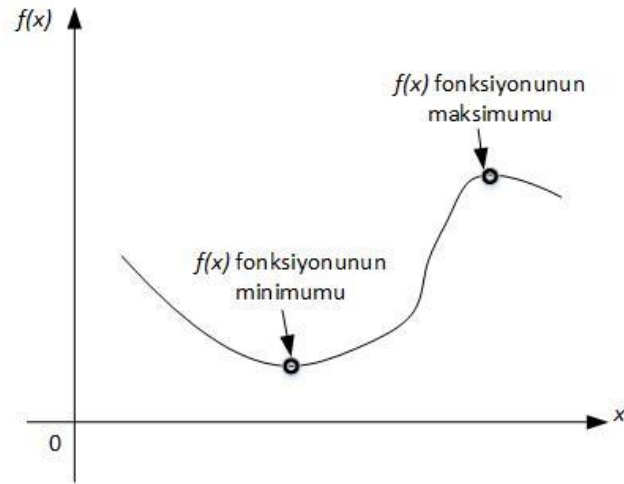
Dinamik Veri: İçeriği sürekli değişen veri tabanları dinamik veriler içerirler. Veri kümesinde bulunan içeriğin sürekli değişmesi veri madenciliği uygulamalarında bir takım sorunların meydana çıkmasını sağlar (Han ve Kamber, 2001).

3.5. Optimizasyon ve Temelleri

Optimizasyon ele alınan bir problemin üzerinde belli koşullar altında en iyi sonucu elde etme eylemidir. Mühendisler tasarım, inşaat ve her türlü mühendislik sisteminin kurulumunun çeşitli aşamalarında birçok teknolojik ve yönetsel kararlar almak zorundadır. Tüm bu kararların nihai hedefi ya gereken çabayı en aza indirmek veya istenilen faydayı maksimize etmektir. Gerekli uğraş veya herhangi bir pratik durumda arzu edilen fayda belli bir karar değişkenlerin bir fonksiyonu gibi ifade edilebilir olduğundan dolayı, optimizasyon bir fonksiyonun maksimum veya minimum değer alan koşulları bulma süreci olarak tanımlanabilir (Rao, 2009).

Bir optimizasyon problemi, bir veri kümesi içerisinde giriş değerlerini seçme ve fonksiyonunun değerini hesaplayarak gerçek işlevi maksimize veya minimize etmek olarak tanımlanabilir. Daha genel olarak, optimizasyon amaç fonksiyonları ve bir etki alanı (veya kısıtlamalar kümesi) verilen bazı amaç fonksiyonlar için "mevcut en iyi" değerleri bulmayı içerir (Joshi ve Moudgalya, 2004).

Şekil 3.26'da bir $f(x)$ fonksiyonun minimum ve maksimum değerler aldığı noktalar gösterilmektedir.



Şekil 3.26. Bir fonksiyonun minimum ve maksimum noktaları

Optimizasyon modelleri bir takım matematiksel terimleri kullanarak bir problemin en iyi şekilde çözülmesini amaçlar (Rao, 2009). Bu amaçları örneklendirmek gerekirse, en fazla kâr, en az zarar hesaplamaları ve en düşük risk gibi hedefleri belirtebiliriz. Bir inşaatın kolon veya kiriş dayanıklılığın en yüksek olması istenebilir. Bir araç rotasının belirlenmesinde en kısa yol, en az süre veya en düşük maliyet olması istenebilir. Bu tip hesaplama istemleri hemen hemen tüm uygulama alanlarında karşımıza çıkmaktadır. Artık birçok problemde, bu modeller kullanılmadan kararların alınması mümkün olmamaktadır.

3.5.1. Optimizasyon yöntemleri

Optimizasyon problemlerinin çözümünde kullanılan yöntemlerin bazılarını aşağıdaki gibi gruplandırabiliriz (Rao, 2009);

1. Matematiksel programlama veya optimizasyon teknikleri

- Hesaplama metotları
- Değişkenler hesaplaması
- Doğrusal programlama
- Doğrusal olmayan programlama
- Geometrik programlama

- Dinamik programlama
 - Tamsayı programlama
 - Çok amaçlı programlama
 - Rasgele olasılıklı programlama
2. Rasgele olasılıklı işlem teknikleri
- İstatiksel karar teorisi
 - Markov işlemi
 - Kuyruk teorisi
 - Yenilenme teorisi
 - Benzetim metotları
 - Güvenilirlik teorisi
3. İstatiksel yöntemler
- Regresyon analizi
 - Kümeleme analizi
 - Örüntü tanıma
 - Deneysel tasarımlar
 - Ayırt etme analizleri (faktör analizi)
4. Modern optimizasyon teknikleri
- Genetik algoritmalar
 - Parçacık sürü optimizasyonu
 - Karınca koloni optimizasyonu
 - Tavlama benzetimi
 - Değişken komşuluk arama
 - Bulanık optimizasyon
 - Sinir ağları

Matematiksel programlama teknikleri, kısıtlamaları önceden belirlenmiş bir kümesi altında bir takım değişkenlerin fonksiyonunun minimum değerini bulma işleminde yararlıdır.

Rasgele olasılıklı işlem teknikleri, bilinen olasılık dağılımlara sahip rastgele değişkenlerin bir kümesi tarafından açıklanan problemleri analiz etmek için kullanılabilir.

İstatistiksel yöntemler, deneysel verileri analiz etmek ve fiziksel durumun en doğru temsilini elde etmek için deneysel modeller oluşturmak için kullanılır.

Modern optimizasyon yöntemleri, bazen geleneksel olmayan optimizasyon yöntemleri olarak da adlandırılır. Son yıllardaki karmaşık mühendislik optimizasyon problemlerini çözmek için güçlü ve popüler yöntemler olarak ortaya çıkmıştır. Günümüzde de en çok tercih edilen yöntemler arasındadır.

Ele alınan problem kısıtsız doğrusal olmayan bir optimizasyon problemidir. Bundan dolayı aşağıda doğrusal olmayan optimizasyon modeli için açıklayıcı bilgiler verilecektir.

3.5.2. Doğrusal olmayan programlama

Doğrusal olmayan programlama bir fonksiyonun değerinin bir takım kısıtlama fonksiyonları ele alınarak minimum veya maksimum yapılmasıdır. Buradaki fonksiyonlardan herhangi biri doğrusal olmayan bir fonksiyon olabilir (Joshi ve Moudgalya, 2004).

Doğrusal olmayan programlama modelleri çoğunlukla bilim ve mühendislik alanlarında karşımıza çıkar. Bir elektrik devresinde harcanan enerji, direncin doğrusal olmayan bir fonksiyonudur. YSA'nın eğitilme problemi kısıtlı doğrusal olmayan bir problem olarak görülebilir (Bazaraa ve ark., 1993).

Gerçek hayatta karşımıza çıkan birçok problem için geliştirilen karar modellere ait kısıtlarda ve amaç fonksiyonda doğrusal ilişkileri gözlemlemek oldukça zordur.

Doğrusal olmayan karar modelinin genel yapısı; X : karar değişkenleri vektörü, $X = (x_1, x_2, x_3, \dots, x_n)$, $g_i(x)$: i . kısıtın ifadesi ($i = 1, 2, \dots, m$), b_i : i . kısıta ait sağ taraf sabiti ($i = 1, 2, \dots, m$), $f(X)$: Amaç fonksiyonu ve en az bir $g_i(X)$ ve $f(X)$ doğrusal olmayan vektör fonksiyonları olmak üzere; $f(X)$ fonksiyonunu eniyileyen X vektörünün bulunması biçimindedir. Bu problem, eğer amaç fonksiyon doğrusal değilse veya

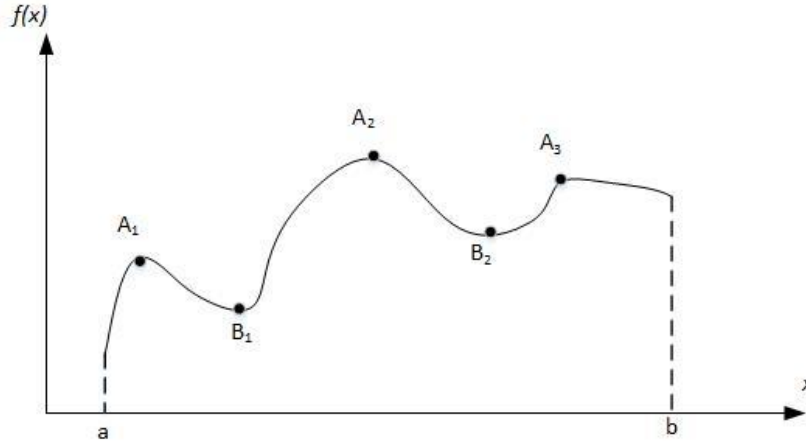
çözüm kümesinin yer aldığı uygulama doğrusal olmayan bir takım sınırlarla belirlenmiş ise doğrusal olmayan programlama problemi olarak isimlendirilir (Avriel, 2012).

Doğrusal olmayan karar modellerinin çözümü için genel bir yöntem ve etkin kullanılan bir algoritma geliştirilmemiştir. Amaç fonksiyonu ve kısıt değerlerinin yapılarına göre, bir takım özel modellerin çözüm teknikleri söz konusudur.

Optimizasyon yöntemlerinin uygulanmasında elde edilen sonucun gerçek optimum değer olup olmadığını tespit etmede bir takım gerek ve yeter şartlar dikkate alınmaktadır. Gerek şart, elde edilecek optimum noktada şartları sağlayan durumlardır. Eğer çözüm uzayındaki herhangi bir nokta gerek şartları sağlayamıyorsa optimum nokta olarak kabul edilemez. Gerek şartları sağlayan nokta optimum değerde olmayabilir veya tek bir nokta şeklinde olmayabilir. Gerek şartları sağlayan noktalar aday nokta olarak isimlendirilir.

Optimum nokta ile optimum olmayan noktaları ayırt etmek için bir takım farklı şartlara gereksinim duyulur. Bu gereksinimlere yeter şart adı verilir. Eğer aday optimum noktalar belirlenen yeter şartları sağlıyorsa bu nokta optimum nokta olarak kabul edilir ve daha ileri düzeyde testleri yapmaya gerek yoktur. Ancak bu şartların sağlanmadığı durumlarda aday noktalardan herhangi birisinin optimum olmadığı sonucu çıkartılamayabilir (Rao, 2009).

Tek değişkenli fonksiyonlarda dikkat edilmesi gereken husus problemden elde edilen minimum değer yerel minimum mu yoksa global minimum mu olduğunun tespit edilmesi işlemidir. Yerel minimum, tek değişkenli bir $f(x)$ fonksiyonunun h değerinin küçük, pozitif ve negatif değerinde $f(x^*) \leq f(x^* + h)$ ifadesini sağlıyorsa bu fonksiyonun değeri $x = x^*$ durumunda göreceli veya yerel minimumdur. Benzer bir şekilde x noktasında eğer $f(x^*) \geq f(x^* + h)$ ifadesi sağlanıyorsa bu değer $f(x)$ fonksiyonu maksimumdur. Global minimum veya maksimum değer kabulü için optimumluk kriterlerinin sağlanması gerekmektedir (Rao, 2009). Bu durum tek değişkenli bir fonksiyon için aşağıda Şekil 3.27'de verilmiştir.



Şekil 3.27. Yerel ve global optimumluk noktaları

Şekil 3.27’de; A_1 , A_2 ve A_3 noktaları göreceli maksimum noktalarıdır. Burada A_2 noktası global maksimumdur. B_1 ve B_2 noktaları göreceli minimum noktalarıdır. Burada B_1 noktası global minimum noktasıdır.

Doğrusal olmayan programlama problemlerinin hep birini çözen kabul edilmiş genel bir yöntem bulunmamaktadır. Farklı tipteki doğrusal olmayan programlama problemlerinin çözümü için farklı bir takım yöntemler bulunmaktadır. Optimizasyon problemlerinin analitik yöntemler kullanılarak çözülemediği durumlarda yaklaşık çözüm tekniklerine başvurulur. Doğrusal olmayan programlama problemleri için geliştirilen algoritmaların temeli, tek değişkenli fonksiyonların çözümündeki yöntemlere dayanır (Alba ve Marti, 2006).

Günümüz optimizasyon problemlerinin çözümünde klasik optimizasyon yöntemlerinin kullanımı zorlaşmıştır. Genellikle çözüm kümesinde amaç fonksiyonunun birkaç optimum noktası bulunmaktadır. Bundan dolayı optimizasyon problemlerinin çözümünde sezgisel veya evrimsel algoritmalarından birinin kullanımı zorunlu hale gelmiştir (Joshi ve Moudgalya, 2004).

Sezgisel veya metasezgisel yöntemler, herhangi bir amacı gerçekleştirmek veya hedefe ulaşmak için çeşitli bir takım işlevlerden etkili olanlara karar vermek amacıyla tanımlanan bir takım kriterler veya bilgisayar yazılım metotlarıdır. Bu yöntemler, çözüm uzayı içerisinde en iyi çözüme ulaşılması tespit edilemeyen algoritmalar olarak da adlandırılır. Bu tür metotlar yakınsama özelliğine sahiptir, fakat kesin çözümün

bulunmasını garanti edemezler ve sadece kesin çözüme yakın bir çözümü garanti ederler (Karaboga, 2004).

Evrimsel algoritmalar optimizasyon yöntemleri olup, doğal seçim ilkeleri üzerine kurulmaktadır. Yeni çözümler üretmek için seçime dayalı ve rasgele yöntemler kullanan popülasyon temelli döngüsel metotlar evrimsel algoritma olarak değerlendirilmektedir. Bir problemin çözümünde, çözüm uzayında araştırma yaparak en iyi çözümü aradığımızdan dolayı bu işlem optimizasyon olarak değerlendirilir (Baykal ve Beyan, 2004).

3.6. Sınıflandırma

Sınıflandırma, sınıf etiketi bilinmeyen girdi verilerinin sınıflarını tahmin etmek için verilere ait sınıf ve kavramlarını tanımlayan modeller veya fonksiyonlar kümesinin bulunması sürecidir (Han ve Kamber, 2001). Sınıflandırma işlemi genellikle model olarak danışmanlı öğrenme metotlarını kullanır. Burada, çıktı sınıfı bilinen bir eğitim veri kümesi verildiğinde, değişkenler ve sınıflar arasındaki gizli ilişkilerin bulunması amaçlanmaktadır (Hand ve ark., 2001).

Veri madenciliği uygulamalarında sınıflandırma kuralları, kullanıcıların kuralları anlamasının ve yorumlamasının daha kolay olmasından dolayı çıktı gösterimi için en çok tercih edilen yöntemlerden birisidir. Genellikle kurallar bilginin net bir şekilde ifade edilmesi için kullanılır ve kullanıcılar tarafından anlaşılması daha kolaydır (Adriaans ve Zantige, 1996).

Sınıflandırma işlemi, sınıf değeri belli olmayan verilerin hangi sınıfa dâhil edileceklerini tahmin etmek için bünyesinde bir takım fonksiyonlar içerir. İstatistik, veritabanı, sinir ağları, makine öğrenimi, evrimsel algoritmalar ve bunlar gibi birçok alanda geliştirilmiş pek çok sınıflandırma yöntemleri mevcuttur.

Geliştirilen yöntemler, veri kümesinde sınıflandırma için kullanılacak olan bir eğitim kümesinin analizine dayanmaktadır ve bir kaydın önceden belirlenen bir sınıf içerisine dâhil edilmesi amaçlanmaktadır (Ye, 2003). Bu işlemi gerçekleştirmek için kullanılacak olan bir sınıflandırma algoritması ile eğitim verileri kullanılarak hangi

sınıfların var olduğu tespit edilmelidir. Test verileri kullanılarak öğrenme yöntemi ile sınıflandırma kuralları elde edilir.

Veri sınıflandırma işlemi iki aşamadan oluşmaktadır. İlk aşama, veri kümesi sınıfını tanımlayan modelin oluşturulmasıdır. Model, özellikler kullanılarak veri kümelerinin analizi ile oluşturulur. İkinci aşama, modelin sınıflandırılması için kullanılmaktadır. İlk olarak, modelin sahip olduğu tahmin veya sınıflandırmadaki doğruluk değeri tespit edilir. Verilen test kümesindeki modelin doğruluğu, model tarafından doğru bir şekilde sınıflandırılan test kümesi örneklerinin oranı olarak ortaya çıkar (Han ve Kamber, 2001).

Test veri kümesinin her bir kaydı, eğitilmiş modelin sınıf tahmini ile karşılaştırılır. Eğitilmiş modelin doğruluğunun belirlenebilmesi için eğitim verisinden bağımsız olan test verisi kullanılmaktadır. Eğer modelin doğruluk derecesi önceden belirlenmiş bir eşik seviyesinin üzerinde ise, model sınıf değerleri bilinmeyen veri kümelerini sınıflandırmak için kullanılabilir.

Sınıflandırma ve tahminde doğruluğu geliştirmek için bir takım ön işlem adımları veri kümesinde yer alan veriler üzerinde uygulanabilir. Bunlar sırasıyla; veri içerisindeki gürültüyü yok etmek ve eksik değerleri gidermek için *veri temizleme* ile ön işlemden geçirilir. *Uygunluk analizi* kullanılarak ilgisiz veya gereksiz özelliklerin öğrenme işleminden çıkarılması ile sınıflandırma etkinliğinin ve ölçeklenebilirliğinin gelişimine yardımcı olunur. *Veri dönüşümü* yapılarak veri kümesinde bulunan veriler daha üst kavramlara genelleştirilebilmektedir. Bu işlem genel olarak sürekli değerlere sahip özellikler için faydalı olmaktadır (Han ve Kamber, 2001).

3.6.1. Sınıflandırma kuralları

Sınıflandırma kuralları, genellikle veritabanı bilgi keşfi süreci sonunda elde edilen bilginin ifade edilmesinde kullanılır ve kullanıcılar tarafından kolayca anlaşılabilir yapıdadır. Sınıflandırma işlemi geleneksel analiz yöntemleri ile karşılaştırıldığı zaman bir takım avantajlara sahip olduğu görülmektedir. Bu avantajları aşağıdaki gibi özetleyebiliriz (Edelstein, 1999);

- İşlemi çözümlene zamanı geleneksel analiz yöntemine göre daha kısadır
- Parametrik değişkenler kullanılabilir
- Elde edilen doğruluk değerleri daha yüksektir
- Düşük maliyetli sonuçlar elde edilir
- Elde edilen bilgiler tutarlıdır
- Analiz işlemleri için uzman gerektirmez
- Anlaşılır ve kolay yorumlanabilir
- Kullanım alanı geniştir.

Sınıflandırma kuralının genel yapısı “EĞER *koşul* O HALDE *değer*” biçimindedir. Bu kural yapısında, koşullar bölümü kuralın mantıksal operatörler ile kontrol edilen nitelik değerlerinin kontrol edilmesidir. Kuralın en sonunda bulunan değer kısmı ise koşul kısmını sağlayan durumlar için tahmin edilen sınıf değerini belirtir. Bu gösterim biçimi, keşfedilen bilginin anlaşılabilirliğine ve yorumlanabilirliğine katkı sağlamaktadır (Hand ve ark., 2001).

Sınıflandırma kuralları tanımlama ve ayırt etme kuralları olarak iki gruba ayrılır. Tanımlama kurallarında amaç bir sınıfın özelliklerini tanımlayan kuralları bulmaktır ve oluşturulan kurallar “EĞER *sınıf* O HALDE *nitelik*” biçimindedir. Ayırt etme kurallarında ise amaç bir sınıfa ait verilerin, veri kümesi içerisinde kalan veriler içinden ayırt edilmesine olanak sağlayan kuralların bulunmasıdır. Ayırt etme kuralları “EĞER *özellik* O HALDE *sınıf*” biçiminde oluşturulurlar. Burada tanımlama kurallarının tersi ayırt etme kuralları şeklindeki bir algı yanlış olur (Breiman ve ark., 1984).

3.6.2. Kural çıkarımı

Kural çıkarımı, üzerinde çalışılan veri kümesinde gizli olan bir bilgiyi yeterli doğrulukta tanımlamak ve mantıksal kavramlar ile sembolik olarak belirtmektir (Tan ve ark., 2006a). Kural çıkarımında temel amaç, verideki gizli bilgiyi keşfetmek ve bilgiyi anlaşılabilir bir şekilde ifade ederek tanımlama kabiliyetini artırmaktır.

Kural çıkarımı ile elde edilen kurallarda bir takım özellikler aranmaktadır. Bu özellikleri şu şekilde sıralayabiliriz (Tan ve ark., 2006a);

- Kolay anlaşılabilir olmalı
- Yapısı basit ve karmaşık olmamalı
- Yorumlanabilir olmalı
- Tutarlı olmalı
- Kural tekrarı olmamalı
- Eksiklikleri tanımlama kabiliyeti olmalı
- Muhakeme ve açıklama kapasitesine sahip olmalı
- Verideki bilgiyi tek başına özetleyebilmeli.

Sınıflandırma kural çıkarım teknikleri kural tabanlı metotlar ve kural tabanlı olmayan metotlar olarak ikiye ayrılır (Kuttiyil, 2004).

Kural tabanlı metotlar: Kural tabanlı sınıflandırma metotları veri kümesinden gizli bilgiyi doğrudan çıkartırlar ve anlaşılması kolaydır. Bu metotlara örnek olarak C4.5, karar tabloları vb. metotları verilebilir.

Kural tabanlı olmayan metotlar: Kural tabanlı olmayan sınıflandırma metotlarından, kural tabanlı sınıflandırma metotlarına nazaran daha doğru sonuçlar elde edilir. Ancak bilgiler kullanıcıların anlayacağı bir biçimde sunulamaz. Genel olarak yapay sinir ağları gibi kural tabanlı olmayan sınıflandırıcılar çok iyi sınıflandırma doğrulukları elde edebilmelerine rağmen anlaşılabilirlik yönünden rekabetçi değildirler. Destek vektör makineleri, yapay sinir ağları, doğrusal genetik programlama kural tabanlı olmayan metotlara örnek olarak verilebilir.

3.7. Veri Ayrıklaştırma

Veri ayrıklaştırma, veri ön işleme adımlarından birisidir. Jin ve ark. (2007) sürekli verinin ayırık veriye dönüştürülmesini, veri ayrıklaştırma işlemi en az veri kaybı ile sürekli verilerin sonlu komşu aralıklar şekline dönüştürülmesi süreci olarak tanımlamışlardır. Das ve Vyas (2010) ise bu işlemi, aynı veya yakın özellikler taşıyan sürekli verinin grup veya aralıklara dönüştürülmesi süreci olarak ifade etmektedir.

Sürekli verilerin ayırık verilere dönüştürülmesine veri madenciliğinde veri indirgeme adı verilir. Veri indirgeme metotları, veri kümesinin daha küçük hacimli olacak şekilde indirgenmiş bir örneğinin elde edilmesi amacıyla uygulanır. Bundan dolayı, elde edilen indirgenmiş veri kümesine veri madenciliği yöntemleri uygulanarak daha etkin ve verimli sonuçlar elde edilebilir. Şekil 3.25’de gösterildiği gibi veri indirgeme yöntemi veritabanından bilginin keşfedilmesi sürecinde kullanılmaktadır.

Bazı veri madenciliği algoritmaları sadece kategorik özellikli değerleri ele aldığından, sürekli özellikli verilerin ayırık değerlere dönüştürülmesi gerekmektedir. Sürekli verilerin ayırık değer aralıklarına dönüştürülmesiyle elde edilen kategorik özellikli değerler, orijinal veri değerlerinin yerine kullanılırlar. Örneğin yaş değişkeni içerisindeki sürekli veriler 1-18, 19-40, 40+ olacak biçimde ayırık halde ifade edilebilir. Bu şekilde veri indirgemedede detay bilgiler kaybolursa da, genelleştirilmiş veriler daha anlamlı olacak ve daha kolay yorumlanabilecektir (Oğuzlar, 2003).

Ayrıklaştırmada en temel sorun, aralıkların sınırlarını belirlemede kullanılacak olan metottur. Bu sınırlar değişkenleri eşit aralıklara bölümlenmek için tanımlanır. Böylece eşit genişlikli aralıklar oluşturulmuş olur. Bu metot, eşit genişlikli aralıkta ayrıklaştırma olarak isimlendirilir.

Veri ayrıklaştırma işlemi için birçok yöntem kullanılmakla beraber burada en çok kullanılan eşit genişlikli ayrıklaştırma ve eşit frekanslı ayrıklaştırma yöntemi hakkında bilgiler verilecektir. Tez çalışmasında ise eşit genişlikli ayrıklaştırma yöntemi kullanılacaktır.

3.7.1. Eşit genişlikli ayrıklaştırma yöntemi

Veri ayrıklaştırma yöntemlerinden biri olan Eşit Genişlikli Ayrıklaştırma (EGA) yönteminde bir değişkenin içerisindeki değerler için maksimum ve minimum değerler tanımlanır. Bu değerler, eşit genişlikte olan ve kullanıcı tarafından belirlenmiş t tane aralığa yerleştirilir (Kurgan ve Cios, 2004).

Yang ve Webb (2002), çalışmalarında bu yöntemi ve aralıkların kesim noktasını ifade eden sınır noktalarının hesaplanması için Eşitlik 3.23 ve 3.24’ü kullanmışlardır.

$$w = \frac{v_{maks} - v_{min}}{t} \quad (3.23)$$

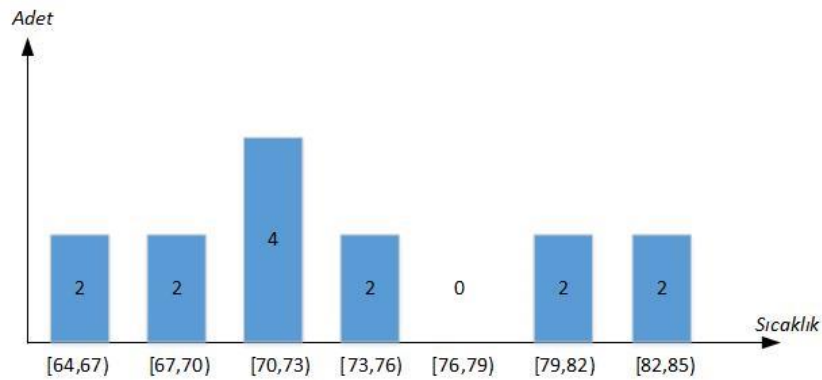
$$v_{min} + w, v_{min} + 2w, \dots, v_{min} + (t - 1)w \quad (3.24)$$

Burada, maksimum (v_{maks}) ve minimum (v_{min}) değerler arasındaki farkın istenilen aralık sayısına (t) bölünmesi ile elde edilen w değişkeni aralıkların genişliğini belirtmektedir.

EGA yöntemi aşağıdaki adımlar kullanılarak uygulanır;

- Ayrıklaştırılacak değişkenlerde bulunan verilerin en büyük ve en küçük değerleri hesaplanır.
- Belirli bir aralık (kategori) sayısı kullanılarak her bir aralığın alt ve üst sınır değerleri hesaplanır.
- Belirlenen aralıklara ait alt ve üst sınırlar kullanılarak her bir veri için bir aralık (kategori) numarası atanır.

Bu adımlar kullanılarak sürekli değerlere sahip olan değişkenler EGA yöntemi ile ayrık değerli hale dönüştürülmüş olur.



Şekil 3.28. Eşit genişlikli ayrıştırma örneği

Şekil 3.28’de, EGA yönteminin çalışmasını belirtmek için örnek bir uygulama gösterilmektedir. Şekilde, eşit aralıklı sıcaklık değerleri aralığına düşen veri kümesi içerisindeki sıcaklık değerlerinin adetleri gösterilmektedir.

3.7.2. Eşit frekanslı ayırıklaştırma yöntemi

Eşit frekanslı ayırıklaştırma yöntemi de eşit genişlikli ayırıklaştırma yöntemi gibi herhangi bir sınıf bilgisi kullanılmadan uygulanan denetimsiz ayırıklaştırma yöntemlerinden birisidir (Biba ve ark., 2007).

Bu yöntemde, öncelikle veri kümesinde yer alan her bir hedef niteliğin bağlantılı olduğu diğer nitelikler tanımlanarak elde edilen bağlantılı nitelikler artan bir sıraya göre dizilmektedir. Bu işlemin ardından bu nitelik değerleri her bir aralığa eşit sayıda düşecek şekilde aralıklara yerleştirilmektedir (Mehta ve ark., 2005).

3.8. Verilerin İkili Formda Kodlanması

İkili veri gösterim şekli ikili veya ayırık değişken içeren veri kümelerine doğrudan uygulanabilir. Geliştirilecek olan algoritma EYSA'dan sınıflandırma kuralları çıkarmak için YSA'ların ağırlık değerlerinden yararlanmaktadır. Ağırlık değerlerinin aktif yada pasif durumda olduğunu göstermenin en etkin yolu ise ikili gösterim biçimidir. Bu gösterimde kullanılan değişkenin "1" değerini alması ilgili nöronun aktif, "0" değerini alması ise pasif olduğunu belirtmektedir. Sürekli değişken içeren veri kümelerinde, verilerin ikili forma dönüştürülmeden önce ayırıklaştırılması gerekmektedir. Böylece her ayırık değer bir ikili değişken ile ifade edilir. İkili kodlama en çok kullanılan ve en basit kodlama biçimidir. Bu işleme yönelik yaklaşımlardan biri ayırık değerler içeren değişkenlerdeki her bir değer için bir bit kullanmaktır.

Verinin bu şekilde ikili gösterim biçimi ilk olarak, Elalfi ve ark. (2004) tarafından EYSA'dan genetik algoritma ile kural çıkarımı çalışmasında kullanılmıştır. Çalışmada veri kümesinde N adet değişkenin olduğu varsayılmıştır. Bu durumda her bir A_n ($n=1, 2, 3, \dots, N$) değişkeninin değerleri sabit uzunlukta bir ikili alt-diziye $(x_1, x_2, \dots, x_{m_n})$ kodlanır. Burada m_n , A_n değişkenleri için olası değerlerin sayısıdır. İkili kodlama gösterimine göre, m_n elemanlarından sadece aktif hale gelen bir elemanı "1" değerini alabilir ve diğer bütün elemanlar "0" değerini alacaktır. Bundan dolayı, YSA'nın giriş katmanındaki düğüm sayısı GD aşağıdaki Eşitlik 3.25 ile hesaplanır;

$$GD = \sum_{n=1}^N m_n \quad (3.25)$$

X_m giriş katmanındaki değişkenlerinin vektörü $X_m=(x_1, x_2, \dots, x_{GD})_m$ şeklinde gösterilir. Burada, $m=(1, 2, \dots, M)$, M girişte eğitilen örneklerin toplam sayısını belirtir. Çıktı sınıfının vektörü $C_k=(s_1, s_2, \dots, s_K)$ ile gösterilir. Burada ($k=1, 2, \dots, K$), K ikili çıktı sınıf sayısıdır. Eğer çıkış vektörü k sınıfına ait ise o zaman s_K “1” değerine eşittir. Diğer tüm elemanlar ise “0” değerine eşittir. Bundan dolayı YSA'nın çıkış katmanındaki çıktı düğümlerinin önerilen sayısı K 'dir. KU kural uzunluğu da Eşitlik 3.26 ile hesaplanır.

$$KU = GD + K \quad (3.26)$$

Ayrıklaştırılmış verilerin ikili forma dönüştürülmesinin daha kolay bir şekilde anlaşılması için bu işlem literatürde yer alan “Tenis Oynama” veri kümesi üzerinde anlatılacaktır. Bu problem dört girdi özellik ve bir çıktıdan meydana gelmektedir. Veri kümesi, (*hava durumu, sıcaklık, nem, rüzgâr*) olmak üzere dört kategorik (ayrık) girdi değişkeni ve tenis oynama kararının yer aldığı, (*oyna, oynama*) şeklinde iki değer alabilen bir çıktı değişkeni içermektedir. Orijinal veri kümesi Çizelge 3.3'de gösterilmektedir.

Çizelge 3.3'de görüldüğü gibi hava durumu değişkeni (*güneşli, bulutlu, yağmurlu*) olmak üzere üç; sıcaklık değişkeni (*sıcak, ılık, soğuk*) olmak üzere üç; nem değişkeni (*yüksek, normal*) olmak üzere iki ve rüzgar değişkeni (*hafif, sert*) olmak üzere yine iki farklı değer içermektedir. Dolayısıyla geliştirilen algoritma için toplam girdi vektörünün (GD) değeri 10 olacaktır. Çıktı değişkeni ise (*oyna, oynama*) olmak üzere iki sınıf ($K=2$) içermektedir. Buradan KU kural uzunluğunun değeri $GD+K$ yani 12 olacaktır. Çizelge 3.4 tenis oynama veri kümesi için elde edilen ikili girdi ve çıktı sınıf vektörlerinin ikili kodlama yöntemini göstermektedir.

Çizelge 3.3. Tenis oynama veri kümesi

Hava Durumu	Sıcaklık	Nem	Rüzgâr	Karar
Güneşli	Sıcak	Yüksek	Hafif	Oynama
Güneşli	Sıcak	Yüksek	Sert	Oynama
Bulutlu	Sıcak	Yüksek	Hafif	Oyna
Yağmurlu	Ilık	Yüksek	Hafif	Oyna
Yağmurlu	Soğuk	Normal	Hafif	Oyna
Yağmurlu	Soğuk	Normal	Sert	Oynama
Bulutlu	Soğuk	Normal	Sert	Oyna
Güneşli	Ilık	Yüksek	Hafif	Oynama
Güneşli	Soğuk	Normal	Hafif	Oyna
Yağmurlu	Ilık	Normal	Hafif	Oyna
Güneşli	Ilık	Normal	Sert	Oyna
Bulutlu	Ilık	Yüksek	Sert	Oyna
Bulutlu	Sıcak	Normal	Hafif	Oyna
Yağmurlu	Ilık	Yüksek	Sert	Oynama

Çizelge 3.4. Tenis oynama veri kümesi için ikili kodlama vektör gösterimi

Hava Durumu			Sıcaklık			Nem		Rüzgar		Karar	
Güneşli	bulutlu	yağmurlu	sıcak	ılık	soğuk	yüksek	normal	hafif	sert	oyna	oynama
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	s_1	s_2

Çizelge 3.5. Tenis oynama veri kümesi değişkenlerinin ikili kodlanması

Hava Durumu				Sıcaklık				Nem		
Değer	x_1	x_2	x_3	Değer	x_4	x_5	x_6	Değer	x_7	x_8
güneşli	1	0	0	sıcak	1	0	0	yüksek	1	0
bulutlu	0	1	0	ılık	0	1	0	normal	0	1
yağmurlu	0	0	1	soğuk	0	0	1			
Rüzgâr				Karar						
Değer	x_9	x_{10}		Değer	s_1	s_2				
hafif	1	0		oyna	1	0				
sert	0	1		oynama	0	1				

Girdi ve çıktı değişkenlerinin hepsi ikili forma dönüştürülerek çok katmanlı algılayıcılara girdi olarak sunulacak şekilde getirilmektedir. Çizelge 3.6'da tenis oynama veri kümesinin ikili kodlanmış biçimi gösterilmiştir.

Çizelge 3.6. Tenis oynama veri kümesi için ikili kodlama gösterimi

X _m	Hava Durumu (m ₁ =3)			Sıcaklık (m ₂ =3)			Nem (m ₃ =2)		Rüzgâr (m ₄ =2)		C _m	Karar	
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀		s ₁	s ₂
X ₁	1	0	0	1	0	0	1	0	1	0	C ₁	1	0
X ₂	1	0	0	1	0	0	1	0	0	1	C ₂	1	0
X ₃	0	1	0	1	0	0	1	0	1	0	C ₃	0	1
X ₄	0	0	1	0	1	0	1	0	1	0	C ₄	0	1
X ₅	0	0	1	0	0	1	0	1	1	0	C ₅	0	1
X ₆	0	0	1	0	0	1	0	1	0	1	C ₆	1	0
X ₇	0	1	0	0	0	1	0	1	0	1	C ₇	0	1
X ₈	1	0	0	0	1	0	1	0	1	0	C ₈	1	0
X ₉	1	0	0	0	0	1	0	1	1	0	C ₉	0	1
X ₁₀	0	0	1	0	1	0	0	1	1	0	C ₁₀	0	1
X ₁₁	1	0	0	0	1	0	0	1	0	1	C ₁₁	0	1
X ₁₂	0	1	0	0	1	0	1	0	0	1	C ₁₂	0	1
X ₁₃	0	1	0	1	0	0	0	1	1	0	C ₁₃	0	1
X ₁₄	0	0	1	0	1	0	1	0	0	1	C ₁₄	1	0

3.9. Kalite ve Uygunluk Fonksiyonları

Çalışmada kalite fonksiyonu olarak yukarıda Eşitlik 3.6'da verilen $s_k(x_i)$ fonksiyonu kullanılacaktır.

Algoritmanın çalışması sonucu elde edilen kuralların birbirleri ile kıyaslanabilmesi için bir uygunluk değeri belirlenmiştir. Her bir aday kural için belirlenmesi gereken bu değer Eşitlik 3.27 ile elde edilmektedir. Herhangi bir kural, verilen eğitim kümesindeki herhangi bir örnek durumu sınıflandırdığında karşımıza dört farklı durum çıkmaktadır. Bu durumlar, Eşitlik 3.27'de yer alan pozitif doğru (tp), negatif doğru (tn), pozitif yanlış (fp) ve negatif yanlış (fn)'den oluşmaktadır (Tan ve ark., 2006b). Burada yer alan tp ve tn doğru sınıflandırmalar iken fp ve fn yanlış sınıflandırmaları belirtmektedir.

$$Uygunluk = \frac{tp}{tp + fn} * \frac{tn}{tn + fp} \quad (3.27)$$

Eşitlik 3.27'de yer alan dört durum hakkında kısa bilgiler aşağıda belirtilmiştir (Baykasoğlu ve Özbakir, 2007).

- Pozitif Doğru (tp): Verilen bir örnek durumun gerçekte pozitif sınıfta olması ve kuralın da bu örnek durumun sınıfını pozitif olarak tahmin etmesi durumudur.
- Negatif Doğru (tn): Verilen bir örnek durumun gerçekte negatif sınıfta olması ve kuralın da bu örnek durumun sınıfını negatif olarak tahmin etmesi durumudur.
- Pozitif Yanlış (fp): Verilen bir örnek durumun gerçekte negatif sınıfta olması ancak kuralın bu örnek durumun sınıfını pozitif olarak tahmin etmesi durumudur.
- Negatif Yanlış (fn): Verilen bir örnek durumun gerçekte pozitif sınıfta olması ancak kuralın bu örnek durumun sınıfını negatif olarak tahmin etmesi durumudur.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Veri Kümeleri

Gerçekleştirilecek EYSA'dan kural çıkarımı çalışması için “The UCI KDD Archive Information and Computer Science University of California” makine öğrenme çalışma bölümünden elde edilen 2 farklı veri kümesi kullanılmıştır (<http://archive.ics.uci.edu/ml/datasets.html>) (son erişim 15 Eylül 2016) . Bu veri kümeleri ikili sınıfa sahip Echocardiogram (Echo) ve Wisconsin Breast Cancer (WBC) veri kümeleridir.

Algoritmanın çalışmaya başlaması için yapılması gereken en önemli işlem, ele alınan veri kümelerinin YSA'nın anlayacağı yapıya dönüştürülmesidir. Bu nedenle ilgili veri kümesine bağlı olarak birtakım dönüşüm işlemlerinin yapılması gerekmektedir. Veri kümesinde bulunan her bir özellik ikili forma dönüştürülmelidir.

Veri kümeleri eksik değerler içermektedir. Bu örnekler veri kümelerinden çıkartılmıştır. Çizelge 4.1'de belirtilen veri kümelerindeki sürekli değere sahip özellikler için ayrıklaştırma işlemi uygulanmıştır. Sürekli değişkenlerin ikili forma dönüştürülmeden önce ayrıklaştırması gerekmektedir.

Çizelge 4.1. Veri Kümelerinin Temel Özellikleri

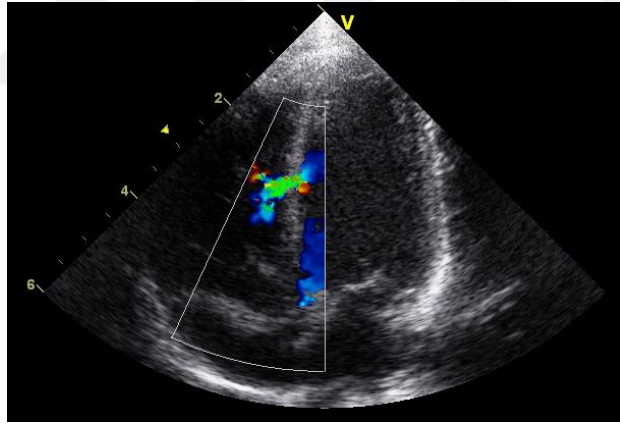
Veri Kümesi	Örnek Sayısı	Kategorik Özellikler	Sürekli Özellikler	Eksik Değerli Özellik	Sınıf Sayısı
Echo	132	1	6	Evet	2
WBC	699	-	9	Evet	2

Öncelikle problemimizin YSA'dan kural çıkarımı kısmına çözüm üretebilmek için kullanılacak olan veri kümelerine ve kural gösterim biçimlerine karar vermek gerekmektedir. Bu yöntemde problemde kullanılacak olan veri kümesinin ikili veya ayrık değişkenler içermesi gerekmektedir. Sürekli değişkenler içeren verilere ön işlem olarak ayrıklaştırma işlemi uygulanmalıdır. Böylece her bir ayrıklaştırma noktası bir ikili değişken ile ifade edilebilir.

Ayrıklaştırma işlemi, Weka 3.7 veri madenciliği yazılımı kullanılarak Basit Bölme (Simple Binning) ile gerçekleştirilmiştir. Tez çalışmasında, basit bölme algoritmasının “eşit genişlik” ölçütü kullanılmıştır. Daha sonra bu kesikli veriler ikili yapıya dönüştürülmüştür.

UCI makine öğrenme çalışma bölümünden elde edilen Echo ve WBC veri kümeleri hakkında detaylı bilgiler aşağıdaki gibidir.

Echo Veri Kümesi: Ekokardiyografi ultrasound, ve ultrason gibi ses dalgalarını kullanarak kalbin ve kalp içerisinde akış halinde bulunan kanın incelenmesi ve görüntülenmesi metodudur. Ekokardiyografiye tıp literatüründe kısaca eko veya kalp ultrasonu da denilmektedir. Ekokardiyografi, kalbin görüntülerini oluşturmak için standart iki boyutlu, üç boyutlu ve Doppler ultrasonografi yöntemlerini kullanmaktadır (Houghton ve Gray, 2012). Şekil 3.29’da örnek bir ekokardiyografi görüntüsü gösterilmektedir.



Şekil 3.29. Ekokardiyografi görüntüsü

Ekokardiyografi görüntüsü elde etmek için, prob denilen bir cihaz göğüs ile temas ettirilerek kalp üzerinde yavaş bir şekilde gezdirilir ve bu cihazın yaydığı ses dalgaları kalbe ve ilgili dokulara çarparak geriye doğru yansır. Geriye yansıyan bu dalgalar yine prob tarafından algılanır ve özel olarak hazırlanan bir takım bilgisayar yazılım işlemlerinden sonra ekrana kalbin görüntüsü olarak yansıtılır.

Elde edilen görüntü sayesinde kalp kasılması, kapakçıkların açılıp kapanması gibi hareketler gerçek zamanlı olarak izlenebilir. Ayrıca kalp içerisinde dolaşan kandan

yansıyan dalgalar da kalp kapaklarının fonksiyonları (kapaktaki darlık, yetmezlik vb.) hakkında ve kalbin kanı pompalama gücü hakkında çok değerli bir takım bilgiler verir. Eko, bizlere kalp hakkında önemli bilgiler veren, hızlı ve zararı olmayan bir test işlemidir (Houghton ve Gray, 2012).

Geçmiş bir zamanda tüm hastalar kalp krizi geçirirler. Bazı hastalar halen hayatta iken bazıları yaşamlarını yitirmektedirler. Araştırmacılar kalp krizini takip eden en az 1 yıl içerisinde hastanın hayatta olup olmayacağını belirlemek için bir takım nitelikler kullanmaktadır. Bu problem üzerinde uygulama geliştirmek için Echo veri kümesi oluşturulmuştur. Echo veri kümesinde 132 örnek, 7 sürekli ve 2 kategorik değişken bulunmaktadır. Bu veri kümesindeki eksik sınıfa sahip 70 adet örnek veri kümesinden çıkarılmış ve geriye kalan örnekler kullanılmıştır. Çizelge 4.2, Echo veri kümesi değişkenleri hakkında bilgi vermektedir. Ayrıklaştırma ve ikili yapıya dönüştürme işlemlerinden sonra toplam 26 adet ikili girdi ve 2 adet ikili çıktı değeri elde edilmiştir.

Veri kümesinde kullanılan özelliklere ait kısa açıklamalar aşağıdaki gibidir;

- Kalp krizi yaşı: Hastada kalp krizinin gerçekleştiği yaş
- Perikardiyal efüzyon: Kalp etrafındaki sıvıdır. İkili değere sahiptir. Değer 0 ise akışkan değil, 1 ise akışkandır.
- Fraksiyonel kısalma: Kalbin etrafındaki kontraktilitenin bir ölçüsüdür. Düşük değerli sayılar durumun anormal olduğunu bildirir.
- Eps: Açılımı E noktalı septum ayrılması şeklindedir. Kontraktilitenin bir başka ölçüsüdür. Büyük sayılar durumun giderek anormalleştiğini gösterir.
- Lvdd: Açılımı sol ventrikül diyastol sonu boyutu şeklindedir. Bu son diyastol kalbin boyutunun bir ölçüsüdür. Büyük kalpler, hastanın kalp krizi olması eğilimindedir.
- Duvar hareket skoru: Sol ventrikül kısımlarının nasıl hareket ettiğinin bir ölçüsüdür.
- Duvar hareket indeksi: Duvar hareket skorunun, görülen kısımların sayısına bölünmesi ile elde edilir. Genellikle ekokardiyografinin 12. ve

13. kısımlarında görülür. Bu değişken duvar hareket skorunun yerine kullanılır.

- Dikkate alma derecesi: Belli bir aralık değerlerine göre hastalığı göz ardı etme durumunu belirtir.
- Durum: Hastanın yaşam durumunu belirtir. İkili değere sahiptir. İlk iki nitelikten türetilmiştir. Değerin 0 olma durumunda hastanın ya 1 yıl sonra ölü ya da 1 yıl boyunca takip edildiğini ifade eder. 1 olma durumunda ise hastanın 1 yıl hayatta olduğu anlamına gelir.

Çizelge 4.2. Echo veri kümesine ait değişkenler

Değişken	Değişken Bilgisi	İkili Değişken Sayısı	Alt-Aralık
Kalp krizi yaşı	Sürekli	4	(-; 56], (56; 66], (66; 76], (76; -)
Perikardiyal efüzyon	0, 1	2	0, 1
Fraksiyonel kısalma	Sürekli	4	(-; 0.16], (0.16; 0.31], (0.31; 0.46], (0.46; -)
Epsc	Sürekli	4	(-; 10], (10; 20], (20; 30], (30; -)
Lvdd	Sürekli	3	(-; 4.52], (4.52; 5.62], (5.62; -)
Duvar hareket skoru	Sürekli	3	(-; 16.66], (16.66; 27.83], (27.83; -)
Duvar hareket indeksi	Sürekli	3	(-; 1.66], (1.66; 2.33], (2.33; -)
Dikkate alma derecesi	Sürekli	3	(-; 0.57], (0.57; 0.78], (0.78; -)
Durum (Sınıf)	0, 1	2	0, 1

WBC veri kümesi: Meme kanseri türü kadınlarda görülen kanser tipleri arasında birinci sırada yer almaktadır. Son zamanlarda erkeklerde de görülmektedir. Türkiye Halk Sağlığı Kurumu Kanser Daire Başkanlığı'nın araştırmalarına göre (<http://kanser.gov.tr/kanser/kanser-turleri/49-meme-kanseri.html>) her 8 kadından birinin meme kanserine yakalanma riski vardır. Meme kanseri, meme dokusunda bulunan hücrelerden ortaya çıkan bir kanser türüdür (son erişim 20 Eylül 2016).

Meme kanseri, meme dokusunun herhangi bir bölgesinden ortaya çıkabilir. Meme kanserinin en sık görülen tipi; meme kanallarından kaynaklanan “duktal” olarak isimlendirilen kanserlerdir. Meme içerisinde süt bezlerinden köken alan “lobüler” kanserler de çok sık görülmektedir. Ayrıca diğer meme dokularından kaynaklanan daha az görünen medüller, tübüler ve müsinöz gibi tipleri de vardır. Meme kanserinin meydana gelmesinde genetik durumlarda çok önemlidir. Genetik yapıda bir takım faktörlerin ve yaşlanmanın etkisiyle meydana gelen bozukluklar kansere neden olur.

Ancak meme kanserlerinin yaklaşık %8 lik bölümü ailesel genetik geçişlidir. Özellikle bireyin anne tarafında genç yaşta meme kanseri ve erkek meme kanseri görülmesi ailesel genetik geçişe örnek verilebilir (Topuz ve ark., 2007).

Dr. Wolberg'in, Wisconsin Üniversite Hastanesinde periyodik olarak meme kanseri tedavisi gören meme kanseri hastalarının tedavisi sürecinde rapor ettiği klinik verilerinden oluşmaktadır (Wolberg ve Mangasarian, 1990). Bu veri kümesi 699 örnekten oluşmakta ve her bir örnek 9 sürekli değişken içermektedir. WBC veri kümesinde, 16 eksik değerli örnek mevcuttur. Eksik değerli örnekler veri kümesinden çıkarılmış ve geriye kalan örnekler kullanılmıştır. Ayırıklaştırma ve ikili forma dönüştürme işleminden sonra 27 ikili girdi ve 2 adet ikili çıkıtu değeri elde edilmiştir. İlgili girdilere ait bilgiler Çizelge 4.3'de gösterilmektedir.

Çizelge 4.3. WBC veri kümesine ait değişkenler

Değişken	Değişken Bilgisi	İkili Değişken Sayısı	Alt-Aralık
Kitle kalınlığı	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Hücre boyutu homojenliği	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Hücre şekli homojenliği	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Marjinal yapışma	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Tek epitel hücre boyutu	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Yalın çekirdek	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Yumuşak kromatin	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Normal çekirdekçik	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Mitozlar	Tamsayı (1-10)	3	(-; 4], (4; 7], (7; -)
Sınıf	İyi huylu, Kötü huylu	2	0,1

Veri kümesinde kullanılan özelliklere ait kısa açıklamalar aşağıdaki gibidir (Salama ve ark., 2012);

- Kitle kalınlığı: Kitle kalınlığında iyi huylu hücreler tek katmanlı halde gruplanırken, kötü huylu (kanseri) hücreler çok katmanlı olarak gruplanır.
- Hücre boyutu/şekli: Hücre boyutu/şekli tekdüzeyken kanser hücreleri boyut ve şekil bakımından farklılıklar gösterir. Bu nedenle bu parametreler, hücrelerin kanserli olup olmadığını belirlemede değerlidir.
- Marjinal yapışma: Marjinal yapışma (tutunum) durumunda, normal hücreler, kanser hücrelerinin bu kabiliyeti kaybetme eğiliminde oldukları

yerde birbirine yapışırlar. Yapışma kaybı, hücrenin kötü huylu olması bulgusudur.

- Tek epitel hücre boyutu: Tek epitel hücre boyutunda boyut yukarıda belirtilen tekdüzeliğe bağlıdır. Önemli derecede büyümüş epitel hücreler kötü huylu bir hücre olabilir.
- Yalın çekirdek: Yalın çekirdekler, sitoplazma (hücrenin geri kalan kısmı) tarafından çevrelenmeyen çekirdekler için kullanılan bir terimdir. Bunlar genellikle iyi huylu tümörlerde görülür.
- Yumuşak kromatin: İyi huylu hücrelerde görülür ve çekirdeğin tek biçimli dokusunu açıklamaktadır. Kanser hücrelerinde kromatin daha iri taneli olma eğilimindedir.
- Normal çekirdekçik: Çekirdekte görünen küçük yapılardır. Normal hücrelerde çekirdekçik genelde görülür derece küçüktür. Kanser hücrelerinde çekirdekçik daha belirgin hale gelir ve bazen bunlardan daha fazlası vardır.
- Mitozlar: Mitoz ya da mitoz bölünme, ana hücre'nin bölünerek iki yeni hücre oluşturmasıdır. Hücrenin bölündüğü ve tekrarlandığı süreçtir. Patologlar mitoz sayısını sayarak kanser derecesini belirleyebilir.
- Sınıf: Sınıf değerinin 0 olma durumunda hücrenin iyi huylu, 1 olması durumunda ise hücrenin kötü huylu olduğuna karar verilir.

4.2. Deneysel Çalışma

Bu tez çalışmasında, ele alınan veri kümeleri üzerinde EYSA'dan DKA yöntemlerinden iki komşuluğa sahip İndirgenmiş Değişken Komşuluk Arama (İDKA) metodu kullanılarak kural çıkarım işlemi hedeflenmiştir. İDKA'nın bünyesinde bulunan sarsma işlemi genel olarak çözümü bozmak amaçlı kullanılacaktır. Sarsma fonksiyonu, çözüm uzayı içerisinde arama işlemini farklı yerlere konumlandırarak yerel en iyilerden kurtulmayı amaçlar. Burada İDKA, X çözüm kümesi üzerinde rasgele bir yön seçerek uygun bir çözüm bulmaya çalışır.

Çalışmada kullanılan İDKA yöntemindeki sarsma işleminde iki komşuluk yapısı kullanılmıştır. Bunlar sırasıyla rasgele nokta değişim ve rasgele çapraz yol değişim komşuluk yapılarıdır. Bu komşuluk yapılarıyla çözüm, belli bir başlangıç uzunluğundan başlayan noktaların değiştirilmesiyle veya yolların yer değiştirilmesiyle bulunduğu yerden uzak yerlere gönderilir. Kararlı yapıların kullanılmasıyla çözümü bozmak garanti altına alınır.

İDKA metodu uygunluk fonksiyonunun optimizasyonu için kullanılacaktır. S_k uygunluk fonksiyonunu maksimize eden en iyi çözüm ifadesini bulan algoritma Şekil 4.1’de ve akış şeması ise Şekil 4.2’de gösterilmiştir.

Algoritmada adım 1’de, veri ön işleme işlemleri gerçekleştirilmektedir. Ele alınan veri kümesinde sürekli veriler eşit genişlikli ayırıştırma yöntemi kullanılarak ayırık hale getirilir. Daha sonra veri kümesinin değişkenlerine ait ayırık veriler ikili forma dönüştürülür. İkili forma dönüştürme işleminde ele alınan değişkenin aldığı farklı değer sayısı kadar ikili formda hane değeri oluşturulur. YSA’nın eğitilmesi için ikili değerlerden oluşan giriş ve çıkış vektörleri oluşturulur.

Adım 2’de önceden tanımlanan parametreler, ikili giriş ve çıkış vektörleri kullanılarak YSA eğitilir.

Adım 3’de EYSA’dan giriş katmanı ve gizli katman arasındaki bağlantılarda bulunan $AG1_{i,j}$ ağırlık değerleri ve gizli katman ve çıkış katmanı arasındaki bağlantılarda bulunan $AG2_{j,k}$ ağırlık değerleri elde edilir. Bu ağırlıklar, İDKA yöntemi ile Eşitlik 3.6’da belirtilen fonksiyonun optimize edilmesi için saklanır.

İDKA metodu, adım 4 ve adım 9 aralığında uygulanmıştır. Burada, başlangıç parametreleri, X çözüm kümesi içerisinde rasgele belirlenen başlangıç çözüm kümesi ve $AG1_{i,j}$ ve $AG2_{j,k}$ ağırlık vektörleri İDKA yöntemi içerisinde kullanılır. Önceden belirlenmiş olan eşik değerinden daha büyük olan S_k uygunluk fonksiyonuna ait çözümler muhafaza edilir.

Üretilen çözümlerden uygunluk değeri en büyük olan çözümden başlanarak, bu çözümlere karşılık gelen giriş ve çıkış vektörleri bir kural gibi dikkate alınıp kural listesine eklenir. Adım 9’da kural çıkarım işlemi gerçekleştirilmiştir ve elde edilen çözümler “Eğer O Halde” yapısında olan dilsel kurallara dönüştürülmüştür.

Adım 1: Veri kümesini giriş ve çıkış vektörleri olarak ayır. Verileri ikili formatta kodla.

Adım 2: YSA'yı ikili giriş ve çıkış vektör değerlerini kullanarak eğit.

Adım 3: YSA'dan $AG1_{i,j}$ ve $AG2_{j,k}$ ağırlık vektör değerlerini elde et.

Adım 4: Başlangıç parametrelerini ayarla. Eşitlik 3.6'da belirtilen S_k fonksiyonunu uygunluk fonksiyonu olarak ayarla.

Adım 5: Başlangıç popülasyonunu (x) oluştur. Aramada kullanılan $(k=1, \dots, k_{max})$ için N_k komşuluk yapılarının kümesini ve durdurma koşulunu seç.

Adım 6: Durdurma koşuluna ulaşıncaya kadar aşağıdaki adımları tekrar et.

Adım 6.1: $k=1$ yap

Adım 6.2: $S_k(x_i)$ uygunluk fonksiyonunu maksimize etmek için $k = k_{max}$ oluncaya kadar aşağıdaki adımları tekrarla

a) Sarsma. x ($x' \in N_k(x)$) noktasının k . komşuluğundan rasgele bir x' noktası üret.

b) Taşı veya taşıma. Eğer bu x' noktası x noktasından daha iyi ise ($x = x'$) yap ve arama işlemini N_1 ($k = 1$) ile sürdür.

Aksi durumda $k = k + 1$ yap

Adım 6.3: Başlangıç popülasyonuna göre uygunluk fonksiyonunu değerlendir.

Adım 7: Eşik değerinden daha büyük olan S_k çözümlerini muhafaza et.

Adım 8: Kural çıkarımı: için aşağıdaki adımları gerçekleştir.

Adım 8.1: Adım 7'de üretilen çözümlerden uygunluğu en büyük olan çözümü bul.

Adım 8.2: Çözüme karşılık gelen giriş ve çıkış vektörlerini bul

Adım 8.3: Çözümü bir kural gibi kabul et ve kural listesine ekle.

Adım 9: Çözümleri dilsel kurallara (Eğer O Halde) dönüştür.

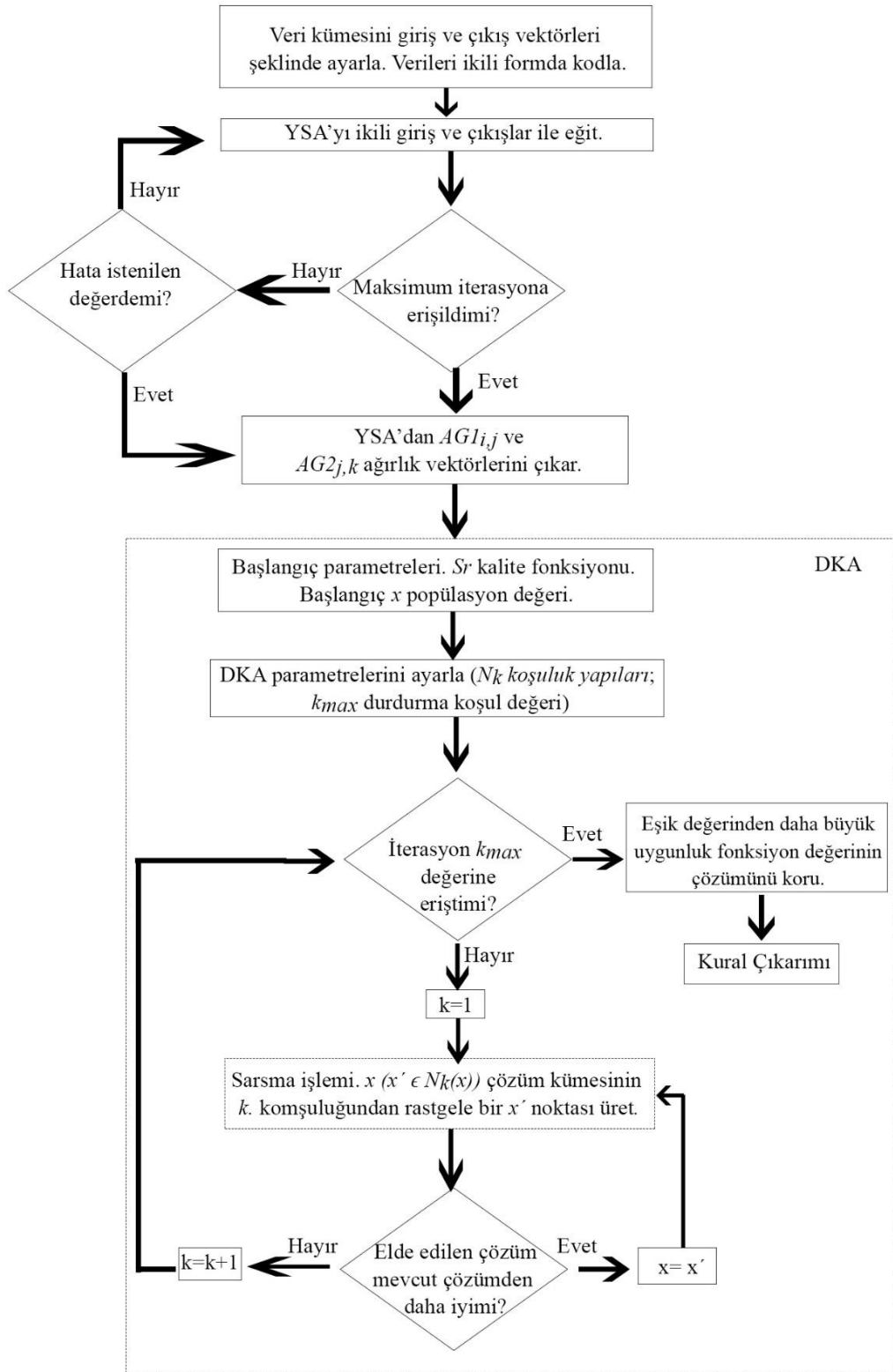
Şekil 4.1. Önerilen algoritmanın temel adımları

Kuralların sadeleştirilmesi işleminde, eğer kural içerisinde bir özelliğe ait birden fazla değer “veya” bağlacı ile bağlanmış ise kurala belirsizlik ve anlam karmaşıklığı

kattığı için bu özellik kural yapısından çıkartılır. Sistemden net ve kesin sonuçlar elde edebilmek için kural içerisinde bir özelliğe ait sadece tek bir değerin olmasına dikkat edilmelidir. Kural içerisinde bulunan farklı özellikler “ve” bağlaçları ile birbirlerine bağlanarak belirtilmelidir.

İDKA yöntemi kullanılarak EYSA’dan kural çıkarım işleminin kodlaması Matlab R2013a programı kullanılarak gerçekleştirilmiştir.





Şekil 4.2. Önerilen metodun akış şeması

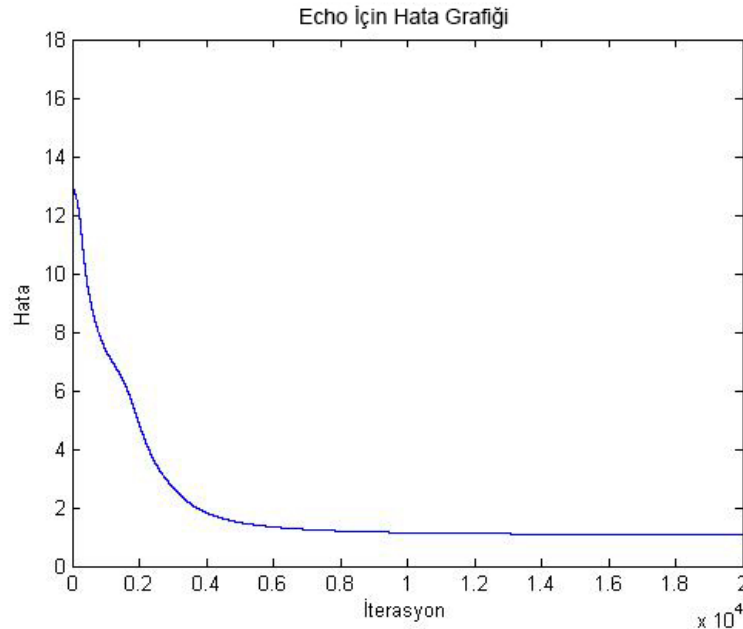
Şekil 4.2'deki kesik çizgi ile belirtilen kısım İDKA algoritmasının kullanıldığı ve kural çıkarım işleminin gerçekleştirildiği kısımdır.

Önerilen yöntemin performans değerlendirmesi için aşağıdaki Eşitlik 4.1 kullanılmıştır. Doğruluk değeri, çıkış sınıfının sınıflandırma yeteneğini ölçmek için kullanılmaktadır.

$$\text{Doğruluk} = \left\{ \frac{\text{Doğru sınıflandırılan çıkışların toplam sayısı}}{\text{Çıkışların toplam sayısı}} \right\} \quad (4.1)$$

Eğitilmiş YSA'dan İDKA kullanılarak kural çıkarım işlemi için Echo veri kümesi 62 adet giriş vektörü (X_m) üzerinde eğitildi. Giriş düğümlerinin sayısı (I) 26 ve çıkış düğümlerinin sayısı (K) 2 dir. Gerçek ve arzu edilen çıkış arasındaki yakınsama oranı, gizli düğüm sayısı= 7, öğrenme katsayısı= 0.001, momentum katsayısı= 1 ve 20,000 tekrar ile elde edildi. İzin verilen minimum hata değeri 0.01 dir.

Echo veri kümesi için eğitime sonucu elde edilen hata grafiği Şekil 4.3'de görülmektedir.



Şekil 4.3. Echo veri kümesi için eğitim hata grafiği

Çizelge 4.4, giriş katmanı ve gizli katman arasındaki S_k uygunluk fonksiyonunu maksimize eden ağırlıkları ($AG1_{i,j}$), Çizelge 4.5 ise gizli katman ve çıkış katmanı arasındaki S_k uygunluk fonksiyonunu maksimize eden ağırlıkları ($AG2_{j,k}$) göstermektedir.

Çizelge 4.4. Echo için giriş ve gizli katman arasındaki ağırlıkları ($AG1_{i,j}$)

Giriş Dügümler	Gizli Dügümler						
	G_1	G_2	G_3	G_4	G_5	G_6	G_7
x_1	2.33848	-0.79051	-1.00519	1.95106	0.89506	1.96284	-0.88337
x_2	0.93614	0.56277	0.10489	-0.48603	-0.41354	-1.23649	1.20337
x_3	-3.59942	-0.60049	-0.01386	-0.09467	0.76079	1.15984	-1.21075
x_4	-1.20374	1.29889	1.13657	-2.28957	-1.33118	-1.94209	1.49101
x_5	-3.38503	-0.86848	-0.64484	-1.30316	0.30117	-1.82582	-2.01506
x_6	-0.03678	0.17875	0.01396	-0.21016	0.08987	0.17369	-0.28402
x_7	-0.59818	-0.52559	-0.20474	-1.70804	-0.37512	-1.74295	-0.68887
x_8	-2.88949	1.22920	1.89629	-0.79010	-0.55352	-0.49537	1.98923
x_9	1.00668	-0.17421	-0.43336	0.75417	0.50437	1.13739	-0.29194
x_{10}	1.13169	-0.06933	-0.46341	0.530659	0.20024	0.55416	-0.11142
x_{11}	-1.82261	-1.32677	-0.42765	0.36975	1.15175	0.20981	-1.43073
x_{12}	2.39971	1.71163	0.87575	2.88070	-2.18438	0.01697	2.62866
x_{13}	-0.40794	-0.11802	0.15323	-2.06542	0.08548	-1.91729	-0.21670
x_{14}	-1.76148	0.51674	-0.41582	-2.27619	1.15491	1.36274	-0.27685
x_{15}	-1.64314	-0.17539	-0.61624	-0.18893	-0.68966	-0.66717	-0.34243
x_{16}	-0.24807	1.40529	1.29784	-1.75287	0.11824	1.12669	2.03506
x_{17}	0.39716	-0.51266	-0.28855	0.92642	0.56294	-1.37538	-1.05734
x_{18}	1.48774	1.10368	0.59590	3.09520	-0.94095	1.70774	2.15212
x_{19}	-1.63585	-1.03269	-1.10103	-3.20571	0.78962	-1.38466	-1.74068
x_{20}	-0.92706	0.48199	0.53809	-0.48355	-0.61597	-0.59725	0.70196
x_{21}	1.61941	0.03661	0.18027	0.69917	-0.86676	-1.70194	-0.05306
x_{22}	-1.64415	-0.28634	-0.31306	-1.01640	0.68864	1.42034	0.24515
x_{23}	-1.35536	0.30507	0.68862	-0.94419	-0.73742	-0.50866	0.50990
x_{24}	-2.97735	-0.51624	-0.63110	-2.02462	-0.21195	-3.69827	-1.20605
x_{25}	0.71372	0.30958	-0.15379	-0.32127	-0.71817	1.71587	0.79555
x_{26}	0.92103	0.48980	0.73240	1.32945	0.45381	1.18965	0.93947

Çizelge 4.5. Echo için gizli ve çıkış katmanı arasındaki ağırlıklar ($AG2_{j,k}$)

Çıkış Dügümleri	Gizli Dügümler						
	G_1	G_2	G_3	G_4	G_5	G_6	G_7
o_1	-6.86272	1.38094	2.15692	-4.27617	-4.32387	-6.67710	4.39031
o_2	6.76353	-3.18501	-2.52607	5.92924	2.17617	4.67323	-4.75750

Çizelge 4.6 ve 4.7’de Echo veri kümesi için uygunluk değerleri ve elde edilen sınıflandırma kuralları gösterilmektedir. Çizelgelerde sıralama işlemi uygunluk değerine göre büyükten küçüğe doğru olacak şekilde sıralanmıştır.

Çizelge 4.6. Çıkış sınıf değerinin 0 olma durumunda çıkarılan örnek kurallar

Kural No	Uygunluk	Çıkarılan Kurallar
1	1	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss = (-; 10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57; 0.78] O Halde Durum= 0
2	0.99999	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss = (-; 10] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
3	0.99993	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss= (-; 10] ve Lvdd = (5.62;-) ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
4	0.99993	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss= (-; 10] ve Lvdd = (5.62;-) ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
5	0.99992	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss= (-; 10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 0
6	0.99983	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss= (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
7	0.99982	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss= (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 0
8	0.99980	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (-;0.16] ve Epss= (10; 20] ve Lvdd = (5.62;-) ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 0
9	0.99977	Eğer Kalp krizi yaşı= (66;76] ve Fraksiyonel kısıalma= (0.31;0.46] ve Epss= (-; 10] ve Lvdd= (-;4.52] ve Duvar hareket skoru= (-;16.66] ve Duvar hareket indeksi= (1.66;2.33] ve Dikkate alma derecesi= (0.57;0.78] O Halde Durum= 0
10	0.99971	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (-;0.16] ve Epss= (10; 20] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0

11	0.99960	Eğer Kalp krizi yaşı= (66;76] ve Fraksiyonel kısıalma = (0.31;0.46] ve Epss= (-;10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
12	0.99917	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss= (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 0
13	0.99886	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss= (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0
14	0.99832	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısıalma = (-;0.16] ve Epss= (10; 20] ve Lvdd= (-;4.52] ve Duvar hareket skoru= (-;16.66] ve Duvar hareket indeksi= (1.66;2.33] ve Dikkate alma derecesi= (0.57;0.78] O Halde Durum= 0
15	0.99817	Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss= (-;10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.78; -) O Halde Durum= 0

Çizelge 4.7. Çıkış sınıf değerinin 1 olma durumunda çıkarılan örnek kurallar

Kural No	Uygunluk	Çıkarılan Kurallar
1	0.99659	Eğer Kalp krizi yaşı= (76;-) ve Fraksiyonel kısıalma = (-;0.16] ve Epss = (10; 20] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (16.66;27.83] ve Duvar hareket indeksi = (1.66;2.33] ve Dikkate alma derecesi = (0.57; 0.78] O Halde Durum= 1
2	0.99646	Eğer Kalp krizi yaşı= (66;76] ve Perikardiyal efüzyon = 0 ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss = (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (16.66;27.83] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57; 0.78] O Halde Durum= 1
3	0.99006	Eğer Kalp krizi yaşı= (56;66] ve Perikardiyal efüzyon = 0 ve Fraksiyonel kısıalma = (0.16;0.31] ve Epss = (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (16.66;27.83] ve Duvar hareket indeksi = (1.66;2.33] ve Dikkate alma derecesi = (0.78;-) O Halde Durum= 1
4	0.98967	Eğer Kalp krizi yaşı= (56;66] ve Perikardiyal efüzyon = 0 ve Fraksiyonel kısıalma = (-;0.16] ve Epss = (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (16.66;27.83] ve Duvar hareket indeksi = (1.66;2.33] ve Dikkate alma derecesi = (0.78;-) O Halde Durum= 1

5	0.98407	Eğer Kalp krizi yaşı= (56;66] ve Fraksiyonel kısalma = (-;0.16] ve Epss = (20; 30] ve Lvdd = (5.62;-) ve Duvar hareket skoru = (27.83;-) ve Duvar hareket indeksi = (2.33;-) ve Dikkate alma derecesi = (0.78;-) O Halde Durum= 1
6	0.97169	Eğer Kalp krizi yaşı= (76;-) ve Fraksiyonel kısalma = (0.16;0.31] ve Epss = (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (16.66;27.83] ve Duvar hareket indeksi = (1.66;2.33] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 1
7	0.96776	Eğer Kalp krizi yaşı= (76;-) ve Fraksiyonel kısalma = (0.16;0.31] ve Epss = (30;-) ve Lvdd = (5.62;-) ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 1
8	0.96626	Eğer Kalp krizi yaşı= (76;-) ve Fraksiyonel kısalma= (-;0.16] ve Epss= (-;10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi= (0.57;0.78] O Halde Durum= 1
9	0.96593	Eğer Kalp krizi yaşı= (76;-) ve Perikardiyal efüzyon = 0 ve Fraksiyonel kısalma = (0.16;0.31] ve Epss = (10; 20] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 1
10	0.96130	Eğer Kalp krizi yaşı= (56;66] ve Perikardiyal efüzyon = 0 ve Fraksiyonel kısalma = (-;0.16] ve Epss = (-;10] ve Lvdd = (4.52;5.62] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57;0.78] O Halde Durum= 1

Echo veri kümesinden çıkarılan kurallardan birini aşağıdaki gibi yorumlayabiliriz;

Kural: “Eğer Kalp krizi yaşı= (-;56] ve Fraksiyonel kısalma = (0.31;0.46] ve Epss = (-; 10] ve Lvdd = (-;4.52] ve Duvar hareket skoru = (-;16.66] ve Duvar hareket indeksi = (-;1.66] ve Dikkate alma derecesi = (0.57; 0.78] O Halde Durum= 0”

Yorum: “Eğer kalp krizi geçiren hastanın yaşı 56 yaş ve altında ise ve kalbin etrafındaki kontraktilitenin boyutu 0.31 ile 0.46 arasında ise ve septum ayrılma nokta sayısı 10’dan küçük ise ve sol ventrikül diyastol sonu boyutu 4.52’den küçük ise sol ventrikül kısımlarının hareket değeri 16.66’dan küçük ise sol ventrikül kısımlarının hareket indeksi 1.66’dan küçük ise ve hastalığı göz ardı etme durum oranı 0.57 ile 0.78 aralığında ise hasta ya 1 yıl sonra ölü ya da 1 yıl boyunca takip edilecektir.”

Çıkış sınıf değerinin “0” olması durumunda sırasıyla 0.99993, 0.99950 ve 0.99800 eşik değerleri kullanılarak elde edilen ortalama kural sayıları sırasıyla 4, 11 ve 15 olarak elde edilmiştir.

Çıkış sınıf değerinin “1” olması durumunda ise sırasıyla 0.99500, 0.98000 ve 0.96000 eşik değerleri kullanılarak elde edilen ortalama kural sayıları sırasıyla 2, 5 ve 10 olarak elde edilmiştir.

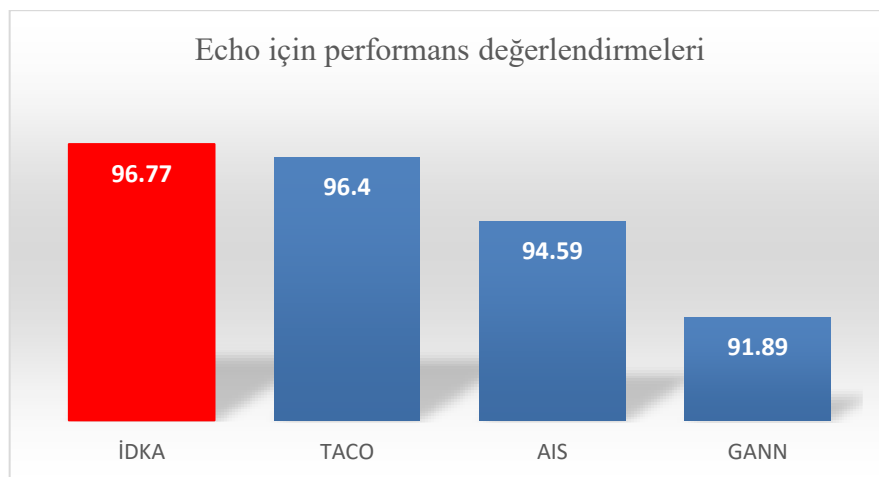
Çizelge 4.8’de, Echo veri kümesi için önerilen metotla elde edilen sınıflandırma doğruluk değeri ve diğer metotlar tarafından elde edilmiş olan doğruluk değerleri gösterilmektedir.

Çizelge 4.8. Echo için metotlar arasındaki performans değerlendirmeleri

Metotlar	Doğruluk Değeri %	Ortalama Kural Sayısı	Referans
İDKA	96.77	15	Önerilen Yöntem
TACO	96.4	13	(Özbakir ve ark., 2009)
AIS	94.59	24	(Kahramanli ve Allahverdi, 2009)
GANN	91.89	12	(Keedwell ve ark., 2000a)

Echo veri kümesi üzerinde, İDKA yöntemi kullanılarak EYSA’dan kural çıkarımı işleminde Eşitlik 4.1 kullanılarak elde edilen sınıflandırma doğruluk değeri %96.77’dir.

Şekil 4.4’de Echo veri kümesi için performans değerlendirmeleri grafiği gösterilmektedir.



Şekil 4.4. Echo veri kümesi için performans değerlendirmeleri

Echo veri kümesi üzerinde kural çıkarma ile ilgili yapılan diğer üç çalışmanın sonucu Çizelge 4.8 ve Şekil 4.4’de belirtilmektedir. Bu sonuçlar sırasıyla TACO

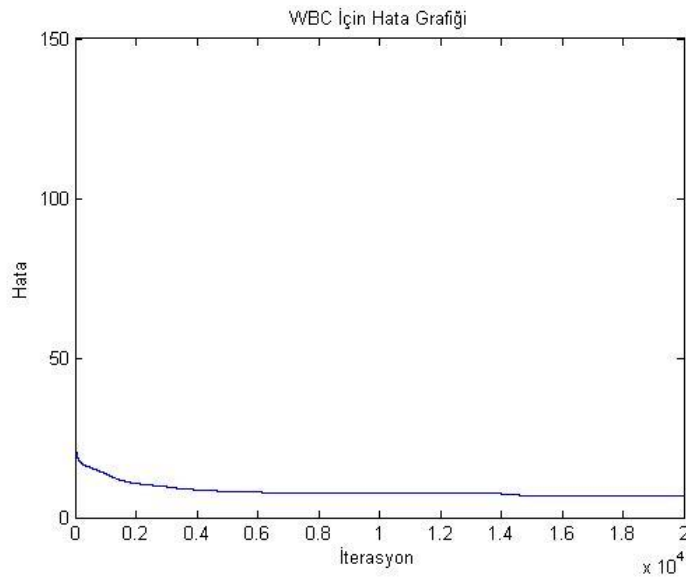
yöntemi için %96.4, AIS yöntemi için %94.59 ve GANN yöntemi için %91.89 şeklindedir.

Bu tezde, Echo veri kümesi için önerilen yöntemin başarısı %96.77'dir. Çizelge 4.8'de yöntemin Echo veri kümesi için başarısı diğer yöntemlerle karşılaştırmalı olarak görülmektedir. Çizelge 4.8'den görüldüğü gibi önerilen sınıflandırma kuralı çıkarma algoritması sonucunda elde edilen kurallar yüksek bir doğruluk oranında teşhis başarısına ulaşmıştır.

Ayrıca Çizelge 4.8'de 0.99800 eşik değeri dikkate alınarak elde edilen ortalama kural sayıları da belirtilmektedir. Burada TACO yöntemi ile 13, AIS yöntemi ile 24, GANN yöntemi ile 12 ve önerilen yöntem ile ortalama 15 kural çıkarılmıştır.

EYSA'dan İDKA algoritması ile kural çıkarma işlemi için WBC veri kümesi 683 adet giriş vektörü (X_m) ile eğitildi. Giriş düğümlerinin sayısı (I) 27 ve çıkış düğümlerinin sayısı (K) 2 dir. Gerçek ve arzu edilen çıkış arasındaki yakınsama oranı, gizli düğüm sayısı= 7, öğrenme katsayısı= 0.001, momentum katsayısı= 1 ve 20,000 tekrar ile elde edildi. İzin verilen minimum hata 0.01 dir.

WBC veri kümesi için eğitim sonucu elde edilen hata grafiği Şekil 4.5'de görülmektedir.



Şekil 4.5. WBC veri kümesi için eğitim hata grafiği

Çizelge 4.9, giriş katmanı ve gizli katman arasındaki S_k uygunluk fonksiyonunu maksimize eden ağırlıkları ($AG1_{i,j}$), Çizelge 4.10 ise gizli katman ve çıkış katmanı arasındaki S_k uygunluk fonksiyonunu maksimize eden ağırlıkları ($AG2_{j,k}$) göstermektedir.

Çizelge 4.9. WBC için giriş ve gizli katman arasındaki ağırlıkları ($AG1_{i,j}$)

Giriş Dğümler	Gizli Dğümler						
	G_1	G_2	G_3	G_4	G_5	G_6	G_7
x_1	2.34727	-0.25839	0.04723	-0.14842	0.33164	-0.11817	1.27661
x_2	-2.38133	1.22968	-2.45914	1.86242	-0.55862	0.64008	0.09516
x_3	-1.66539	-1.98496	3.65996	0.61973	0.37830	-0.51091	-0.45727
x_4	-0.25778	0.49012	-1.29856	-2.73850	-0.91359	-1.62044	1.79322
x_5	-1.96609	-1.16657	1.38725	2.86602	0.58219	1.14408	-0.02638
x_6	0.47916	-0.13836	0.37486	1.56736	0.09227	0.18047	-0.94590
x_7	-2.30854	0.12469	-0.68175	-1.39243	-0.00718	0.66207	0.07396
x_8	-0.34813	-0.43287	0.70393	2.04056	0.40166	-0.64312	0.52532
x_9	1.00390	-0.71808	1.43372	1.26026	-0.41771	-0.56133	0.64299
x_{10}	2.33407	1.08032	-0.19953	-1.15074	-1.65057	-1.90687	0.01625
x_{11}	-1.10358	-0.80555	1.70800	0.18979	1.98257	1.91563	-0.73376
x_{12}	-2.62945	-1.10766	-0.18778	3.07935	-0.42121	-0.50730	2.03816
x_{13}	0.21580	-0.69879	1.65984	-0.93738	0.03692	-0.14728	1.36753
x_{14}	-3.02921	0.86990	-1.38554	3.34644	-0.16327	1.30542	-1.98833
x_{15}	0.71353	-0.48682	0.46394	-0.79233	0.25706	-1.76295	1.41262
x_{16}	1.92140	0.50230	-2.56356	-0.05272	0.34276	-2.34938	-0.37400
x_{17}	-2.73490	-1.04676	2.66573	-2.09206	0.77841	1.57895	2.47591
x_{18}	-0.92413	-0.36093	0.89439	4.44671	-1.29481	0.00462	-0.78541
x_{19}	-0.75835	1.11557	-1.01553	-0.99438	0.55381	-0.76436	-1.97009
x_{20}	1.23777	-1.07111	0.52255	1.47352	-0.41346	1.24282	2.03753
x_{21}	-1.51321	-0.92398	1.57876	1.75385	-0.73499	-1.11609	1.19576
x_{22}	0.56128	0.42006	-1.32374	0.99262	0.44508	0.04796	1.07981
x_{23}	-1.38738	-0.52581	0.29431	-1.01705	-1.14473	-0.21231	-2.09240
x_{24}	-1.09086	-1.24972	1.98334	1.74875	-0.13672	-0.61610	2.09919
x_{25}	0.68809	-0.12041	-1.38009	-2.15815	1.07384	0.41656	-0.55141
x_{26}	-0.80449	-0.11146	1.05802	2.65473	-0.68158	-0.88338	0.90871
x_{27}	-1.15187	-0.37085	1.07742	1.26643	-0.35766	-0.46507	0.64703

Çizelge 4.10. WBC için gizli ve çıkış katmanı arasındaki ağırlıklar ($AG2_{j,k}$)

Çıkış Dğümleri	Gizli Dğümler						
	G_1	G_2	G_3	G_4	G_5	G_6	G_7
o_1	6.03683	2.42293	-4.44719	-7.86519	3.76050	5.56329	-4.45892
o_2	-6.07132	-2.81940	4.17520	7.70591	-3.85017	-5.59782	4.27109

WBC veri kümesi için kural çıkarım işleminde; çıkış sınıf değerinin “0” olması durumu için eşik değeri 0.99900 ve “1” olması durumu için eşik değeri 0.99970 olarak kabul edilmiştir.

Elde edilen uygunluk değerleri ve kuralların bir bölümü Çizelge 4.11 ve 4.12’da gösterilmektedir.

Çizelge 4.11. Çıkış sınıf değerinin 0 olma durumunda çıkarılan örnek kurallar

Kural No	Uygunluk	Çıkarılan Kurallar
1	0.99972	Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (7;-) ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”
2	0.99968	Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”
3	0.99961	Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”
4	0.99926	Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”
5	0.99904	Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”

Çizelge 4.12. Çıkış sınıf değerinin 1 olma durumunda çıkarılan örnek kurallar

Kural No	Uygunluk	Çıkarılan Kurallar
1	0.99999	Eğer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (7; -) ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (7; -) ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”

- 2 0.99999 Eđer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (7; -) ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (7; -) ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (7; -) ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”
- 3 0.99999 Eđer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliđi= (-; 4] ve Hücre şekli homojenliđi = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (4; 7] O Halde Sınıf = “Kötü huylu”
- 4 0.99998 Eđer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (-; 4] ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (7; -) ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (4; 7] ve Normal çekirdekçik = (7; -) ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”
- 5 0.99997 Eđer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliđi= (4; 7] ve Hücre şekli homojenliđi = (4; 7] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (4; 7] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”
- 6 0.99996 Eđer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (7; -) ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (4; 7] ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (4; 7] ve Normal çekirdekçik = (7; -) ve Mitozlar= (4; 7] O Halde Sınıf = “Kötü huylu”
- 7 0.99995 Eđer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (7; -) ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (7; -) ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
- 8 0.99994 Eđer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (4; 7] ve Marjinal yapışma = (4; 7] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (7; -) ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
- 9 0.99993 Eđer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (7; -) ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (4; 7] ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (4; 7] ve Normal çekirdekçik = (7; -) ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
- 10 0.99992 Eđer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliđi= (7; -) ve Hücre şekli homojenliđi = (7; -) ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (4; 7] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”

11	0.99989	Eğer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliği= (4; 7] ve Hücre şekli homojenliği = (4; 7] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (-; 4] ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (4; 7] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
12	0.99984	Eğer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliği= (7; -) ve Hücre şekli homojenliği = (4; 7] ve Marjinal yapışma = (4; 7] ve Tek epitel hücre boyutu= (7; -) ve Yalın çekirdek = (4; 7] ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (7; -) ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
13	0.99979	Eğer Kitle kalınlığı = (7; -) ve Hücre boyutu homojenliği= (7; -) ve Hücre şekli homojenliği = (7; -) ve Marjinal yapışma = (4; 7] ve Tek epitel hücre boyutu= (4; 7] ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (7; -) ve Normal çekirdekçik = (7; -) ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”
14	0.99973	Eğer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliği= (7; -) ve Hücre şekli homojenliği = (4; 7] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (7; -) ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (7; -) ve Mitozlar= (7; -) O Halde Sınıf = “Kötü huylu”
15	0.99971	Eğer Kitle kalınlığı = (4; 7] ve Hücre boyutu homojenliği= (7; -) ve Hücre şekli homojenliği = (7; -) ve Marjinal yapışma = (7; -) ve Tek epitel hücre boyutu= (4; 7] ve Yalın çekirdek = (7; -) ve Yumuşak kromatin = (4; 7] ve Normal çekirdekçik = (7; -) ve Mitozlar= (-; 4] O Halde Sınıf = “Kötü huylu”

WBC veri kümesinden çıkarılan kurallardan birini aşağıdaki gibi yorumlarız;

Kural: “Eğer Kitle kalınlığı = (-; 4] ve Hücre boyutu homojenliği= (-; 4] ve Hücre şekli homojenliği = (-; 4] ve Marjinal yapışma = (-; 4] ve Tek epitel hücre boyutu= (7;-) ve Yalın çekirdek = (-; 4] ve Yumuşak kromatin = (-; 4] ve Normal çekirdekçik = (-; 4] ve Mitozlar= (-; 4] O Halde Sınıf = “İyi huylu”

Yorum: “Eğer meme kanseri teşhisi için başvuran hastadan alınan kitlenin kalınlığı 4’den küçük ise ve hücre boyutu homojenliği 4’den küçük ise ve hücre şekli homojenliği 4’den küçük ise ve hücrenin marjinal yapışma değeri 4’den küçük ise ve tek bir epitel hücre boyutu 7’den büyük ise ve yalın halde bulunan çekirdek değeri 4’den küçük ise ve yumuşak kromatin değeri 4’den küçük ise ve normal çekirdekçik değeri 4’den küçük ise ve mitoz bölünme değeri 4’den küçük ise hastanın memesinde bulunan hücrenin iyi huylu olduğuna kanser olmadığına karar verilir.”

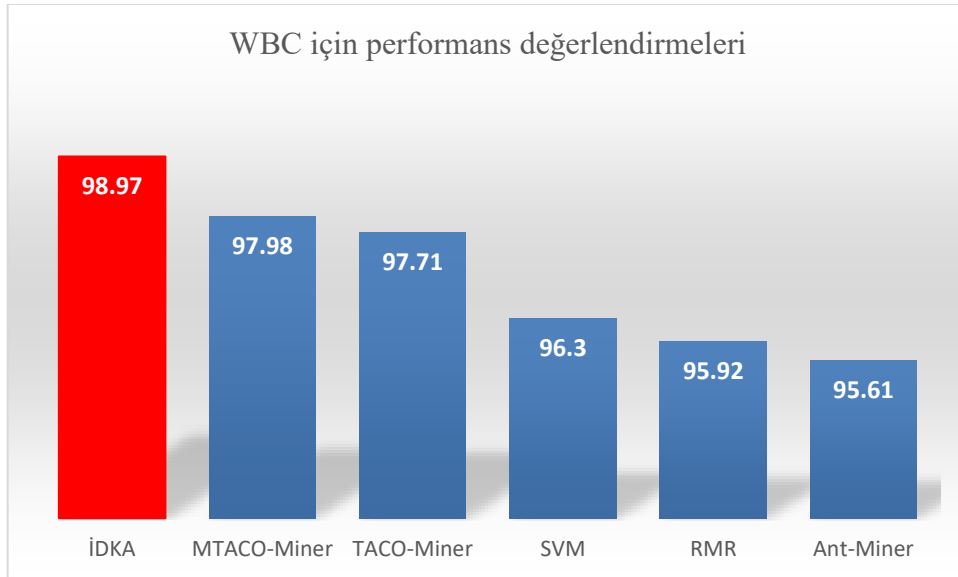
Çizelge 4.13’de WBC veri kümesi için önerilen metotla elde edilen doğruluk değeri ve diğer metotlar tarafından elde edilmiş olan sınıflandırma doğruluk değerleri gösterilmektedir.

Çizelge 4.13. WBC için metotlar arasındaki performans değerlendirmeleri

Metotlar	Doğruluk Değeri %	Referans
İDKA	98.97	Önerilen Yöntem
MTACO-Miner	97.98	(Tripathy ve ark., 2013)
TACO-Miner	97.71	(Özbakir ve ark., 2009)
SVM	96.3	(Martens ve ark., 2007)
RMR	95.92	(Thabtah ve Cowling, 2007)
Ant-Miner	95.61	(Smaldon ve Freitas, 2006)

WBC veri kümesi üzerinde, İDKA kullanılarak EYSA’dan kural çıkarımı işleminde Eşitlik 4.1 kullanılarak önerilen metot ile %98.97 sınıflandırma doğruluk değeri elde edildi.

Şekil 4.6’da WBC veri kümesi için performans değerlendirmeleri grafiği gösterilmektedir.



Şekil 4.6. WBC veri kümesi için performans değerlendirmeleri

WBC veri kümesi üzerinde kural çıkarma ile ilgili yapılan diğer beş çalışmanın sonucu Çizelge 4.13 ve Şekil 4.6’da belirtilmektedir. Bu sonuçlar sırasıyla MTACO-

Miner yöntemi için %97.98, TACO-Miner yöntemi için %97.71, SVM yöntemi için %96.3, RMR yöntemi için %95.92 ve Ant-Miner yöntemi için ise %95.61 şeklindedir.

Bu tezde, WBC veri kümesi için önerilen yöntemin başarısı %98.97'dir. Çizelge 4.13'de yöntemin WBC veri kümesi için başarısı diğer yöntemlerle karşılaştırmalı olarak görülmektedir. Çizelge 4.13'den görüldüğü gibi önerilen sınıflandırma kuralı çıkarma algoritması sonucu çıkarılan kurallar yüksek bir doğruluk oranında teşhis başarısına ulaşmıştır.



5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu tez çalışmasında, UCI (Kaliforniya Üniversitesi - Irvine) makine öğrenme çalışma bölümünden elde edilen Echocardiogram (Echo) ve Wisconsin Breast Cancer (WBC) veri kümeleri kullanılarak EYSA'dan sınıflandırma kuralları biçiminde kurallar elde etmek için İDKA algoritmasının komşuluk teknikleri üzerinde bir takım geliştirmeler yapılmıştır. İDKA yönteminde rasgele nokta değişim ve rasgele çapraz yol değişim komşuluk yapıları kullanılmıştır. Bu komşuluk yapıları üzerinde performansı olumlu ölçüde artıran bir takım parametrik ve yöntem değişiklikleri gerçekleştirilmiştir.

Geliştirilen algoritma, YSA'nın giriş ve gizli katmanları ile gizli ve çıkış katmanları arasında bulunan bağlantı ağırlıklarındaki gizli bilgiyi keşfetmek amacıyla ÇKA yapısına sahip EYSA üzerinde çalıştırılmıştır.

Bu tezin deneysel çalışması üç temel yapıdan oluşmaktadır. Bunlar sırasıyla; verilerin kodlanması, YSA'nın eğitilmesi, komşuluk teknikleri geliştirilmiş İDKA algoritması kullanılarak EYSA'dan kural çıkartma şeklindedir.

Bu çalışmada, ayrık değerlere sahip özellikler ikili biçimde kodlanırken, sürekli değere sahip özellikler üzerinde de eşit genişlikli ayrıklaştırma yöntemi kullanılarak sürekli değerler ayrık değerlere dönüştürülmüştür. Dönüştürülen bu değerlerde ikili biçimde kodlanmıştır. İkili biçimde kodlanmış veriler eğitime işlemi için YSA'ya sunulmaktadır. YSA'nın eğitim algoritması olarak, Geri Yayılım algoritması kullanılmıştır.

Önerilen İDKA metasezgisel algoritması iki adımlı bir hiyerarşik yapıya sahiptir. İlk adımda, çok katmanlı bir sinir ağı eğitilmiş ve eğitim sonucunda katmanlar arasında bulunan en iyi bağlantı ağırlıkları elde edilmiştir. İkinci adımda, elde edilen en iyi ağırlık değerleri geliştirilen İDKA algoritması ile sınıflandırma kurallarının elde edilmesi için kullanılmıştır. Burada İDKA, ağı eğitilmesi işleminde kullanılmamıştır.

YSA'nın eğitim işlemi tamamlandıktan sonra Eşitlik 3.6'da belirtilen uygunluk fonksiyonu İDKA algoritması kullanılarak optimize edilmiştir.

Bu çalışmada, kabul edilen eşik değerinden büyük uygunluk fonksiyon değerine sahip çözümler kurallar oluşturmak için korunmuştur.

Kural çıkarımı için aşağıdaki işlem adımları gerçekleştirilmiştir.

1. Üretilen çözümlerden uygunluk değeri en büyük olan çözüm bulunur.
2. Ele alınan çözüme karşılık gelen giriş ve çıkış vektörü bulunur.
3. Çözüm bir kural gibi kabul edilir ve kural listesine eklenir.
4. Kural listesindeki çözümler dilsel kurallara (Eğer- O Halde) dönüştürülür.

Elde edilen dilsel kurallarda bir niteliğin bir arada geçerli olduğu birçok durum mevcut ise, bu özelliğin durumları “veya” operatörü ile birleştirilmiştir. Elde edilen dilsel kuralda farklı nitelik değerleri “ve” operatörü ile birleştirilmiştir. Bu çalışmada, kurallar kümesinde bulunan dilsel kurallardan “veya” bağlacının yer aldığı nitelikler kurallar içerisinde çıkarılarak daha kesin ve doğru kurallar elde edilmiştir.

Önerilen yöntemin başarısını ve performansını test etmek için iki farklı veri kümesi üzerinden elde edilen sonuçlar literatürde belirtilen bir takım yöntemlere ait sonuçlar ile karşılaştırılmıştır. Echo ve WBC veri kümeleri üzerinde gerçekleştirilen çalışmada, sırasıyla %96.77 ve %98.97 sınıflandırma doğruluk oranları elde edilmiştir. Gerçekleştirilen karşılaştırma işlemleri sonucunda her iki veri kümesi üzerinde de önerilen yöntemin başarılı olduğu görülmüştür. Genel olarak incelendiğinde teşhis oranı yüksek olan bir sistemin geliştirildiği görülmektedir.

Bu tez çalışmasının bir başka katkısı ise İDKA yönteminin ilk defa bu tip bir çalışmada kullanılmış olmasıdır. Çözüm alanında hızlı bir şekilde arama yapan İDKA algoritması için iyi bir sonuç elde edildikçe dinamik bir şekilde küçülen komşuluklar geliştirilerek bu çalışmada İDKA algoritmasına aynı zamanda yerel arama yeteneği de kazandırılmıştır.

Ele alınan problemlere ait verilerden öğrenme, kural çıkarımı ve bilgi keşfi çok önemli bir işlev olup, son zamanlara istatistik, tahmin ve yapay zekâ gibi alanlarda çalışma yürüten araştırmacıların oldukça ilgisini çekmektedir.

Gerçek dünya problemleri üzerinde gerçekleştirilecek olan öğrenme işleminde, en çok kullanılan tümevarımsal yaklaşıma sahip öğrenme yöntemlerinden birisi YSA’dır. YSA’lar, özellikle sınıflandırma, tanıma ve genelleme gibi birçok problem üzerinde başarılı sonuçlar veren bir çözümlenme aracıdır. Ancak, YSA’nın sahip olduğu

kara-kutu özelliği nedeniyle, ağın öğrenme işini nasıl yaptığı tam olarak anlaşılabilir değildir. Ayrıca, YSA ile elde edilen bilgilerin tam olarak anlaşılabilir olmamasından dolayı, bazı problemlerin çözümünde çokta kullanışlı değildir. Bu durum, özellikle anlaşılabilir ve yorumlanabilir kuralların önemli olduğu birçok veri madenciliği problemlerinde sorun olarak karşımıza çıkmaktadır.

Bu çalışma ile YSA'dan kural çıkarma problemi için literatüre yeni bir yöntem kazandırılarak kolay anlaşılabilen ve yorumlanabilen kuralların elde edilebilmesine olanak sağlanmıştır.

Günümüze kadar gerçekleştirilen çalışmalarda kullanılan veriler genellikle ya ikili formda, yâda sürekli formda kodlanmaktaydı. Problemlerde kullanılan veriler genellikle ayrık veya sürekli değerlerden oluşmaktadır. Bu çalışma ile sürekli değerlere sahip veriler ikili biçimde kodlanarak daha etkili sonuçların elde edilmesine imkan sağlanmıştır.

Tıbbi bilimlerde karar verme süreci, çok yönlü ve karmaşık bir işlemdir. Çalışmada öncelikli amaç hastalık üzerinde doğru teşhisin gerçekleştirilmesidir. Bu amaca ulaşmak için uygun veri kümelerinin bulunması, veri kümesi içerisinde özelliklerinin çıkarılması, verilerin ön işlemden geçirilmesi ve sisteme yeni girilecek olan verilerin analizinin yapılması gerekmektedir.

Sınıflandırma işlemi veri madenciliği alanında en çok kullanılan yöntemlerden birisidir. Sınıflandırma sonucunda bir model oluşturulmakta ve sisteme yeni girilen bir verinin hangi sınıfa ait olacağı belirlenebilmektedir. Bu işlem hastalık teşhisi için hekimler tarafından sıklıkla kullanılmaktadır.

Bu çalışma tıbbi veriler üzerinde gerçekleştirildiğinden dolayı elde edilen önemli sonuçlardan bazıları şunlardır;

- Çalışmada, açık erişimli veritabanlarında sunulan iki farklı tıbbi veri kümesi kullanılmıştır. Her iki veri kümesinin araştırmacıların analiz ve değerlendirmelerine açık olmasından dolayı çalışmanın başarı ölçümlerinin güvenilirliğinin testi amacıyla üzerlerinde birçok uygulama yapılarak bize çalışmamızın sonucunu karşılaştırma fırsatı sunmaktadır.
- Tıbbi bilimlerde hastalıklar üzerinde gerçekleştirilecek olan doğru bir teşhisin oldukça önemli olduğu ve bu teşhisin durumuna göre tedavi şeklinin

belirleneceği göz önüne alındığında, böyle bir sınıflandırma ve kural çıkarım sisteminin önemi daha iyi anlaşılacaktır.

- Gerçekleştirilen çalışmada metasezgisel bir yöntem olan İDKA algoritması ile başarı ve doğruluk oranı daha yüksek olan kurallar elde edilmiştir. Bunun sonucu olarak İDKA ile tıp uzmanlarına daha doğru ve optimum teşhis ve tedavi desteği sağlanabilir.
- EYSA'dan kural çıkarımında, dilsel kural içerisinde belirsizliğe yol açan “veya” bağlacının yer aldığı nitelikler kural kümesi içerisinde çıkarılarak kural yapısına kesinlik, yorumlanabilirlik ve anlaşılabilirlik yetenekleri kazandırılmıştır. Yapılan bu çalışmanın uygulanabilir gerçek bir sistem içerisinde kullanılması mümkündür.

5.2. Öneriler

Bu konuda çalışacak araştırmacılara ışık tutması açısından aşağıdaki hususların açıklanması uygun görülmüştür.

Bu çalışmada EYSA'dan İDKA yöntemi ile kural çıkarım işlemi yapılarak teşhis ve tedavi başarısının yükseldiği gözlemlenmektedir. Ayrıca veri madenciliği yöntemleri ile ön işlemden geçirilmiş verilerle yapılmış bir çalışmanın, klasik kodlanmış verilerle yapılan çalışmadan daha başarılı olduğu görülmektedir.

Bu tez çalışmasının sonucunda;

- Problemin kullanıldığı alanda bir teşhis sistemi tasarlanarak, sistemin test edilmesi ve elde edilen sonuçların değerlendirilmesi,
- Farklı bilimlerden elde edilen veri kümeleri üzerinde bu yöntemin uygulanması ve sonuçlarının analiz edilmesi,
- Eğitime işlemine sunulan örneklerin sayısının artırılarak teşhis oranlarının incelenmesi gibi bazı önemli konuların ileriye yönelik birer çalışma olarak önerilmesinde fayda görülmektedir.

KAYNAKLAR

- Adriaans, P. ve Zantige, D., 1996, Data Mining, *Harlow: Addison-Wesley*, 158.
- Akpınar, H., 2000, Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 29 (1), 1-22.
- Alba, E. ve Marti, R., 2006, Metaheuristic Procedures For Training Neural Networks, *Springer Science Business Media, LLC, USA*.
- Andrews, R., Diederich, J. ve Tickle, A. B., 1995, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems*, 8 (6), 373-389.
- Arbatli, D. A. ve Akin, L. H., 1997, Rule extraction from trained neural networks using genetic algorithms, *Nonlinear Analysis: Theory, Methods & Applications*, 30 (3), 1639-1648.
- Arbib, M., 2003, The Handbook of Brain Theory and Neural Networks, *MIT Press*.
- Argüden, Y. ve Erşahin, B., 2008, Veri Madenciliği, Veriden Bilgiye, Masraftan Değere, *ARGE Danışmanlık Yayınları*, 10.
- Avriel, M., 2012, Nonlinear programming: analysis and methods, *Courier Dover Publications*.
- Baykal, N. ve Beyan, T., 2004, Bulanık Mantık Uzman Sistemler ve Denetleyiciler, *Bıçaklar*.
- Baykasoğlu, A. ve Özbakır, L., 2007, MEPAR-miner: Multi-expression programming for classification rule mining, *European Journal of Operational Research*, 183 (2), 767-784.
- Bazaraa, M. S., Sherali, H. D. ve Shetty, C. M., 1993, Nonlinear Programming: Theory and Algorithms, *John Wiley and Sons Publishing, N.Y.*
- Bhalla, D., Bansal, R. K. ve Gupta, H. O., 2012, Function analysis based rule extraction from artificial neural networks for transformer incipient fault diagnosis, *International Journal of Electrical Power & Energy Systems*, 43 (1), 1196-1203.
- Biba, M., Esposito, F., Ferilli, S., Mauro, N. D. ve Basile, T. M. A., 2007, Unsupervised Discretization Using Kernel Density Estimation, *Proceedings of the 20th international joint conference on Artificial intelligence (IJCAI), Hyderabad India, SanFrancisco, CA, USA:Morgan Kaufmann Publishers Inc.*, 696-701.
- Blum, C. ve Roli, A., 2003, Metaheuristic in Combinatorial optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, 35 (3), 268-308.

- Blum, C. ve Socha, K., 2005, Training Feed-Forward Neural Networks with Ant Colony Optimization: An Application to Pattern Classification, *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*.
- Breiman, L., Friedman, J., Olshen, R. ve Stone, C., 1984, Classification and Regression Trees, *Belmont: Wadsworth*.
- Chen, C., Chen, Y. ve He, J., 2006, Neural Network Ensemble Based Ant Colony Classification Rule Mining, *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)*.
- Craven, M. ve Shavlik, J., 1994, Using Sampling and Queries to Extract Rules from Trained Neural Networks, *Proceedings of the Eleventh International Conference on Machine Learning*, 37-45.
- Curram, S. P. ve Minger, J., 1994, Neural Networks, Decision Tree Induction and Discriminant Analysis: Empirical Comparison, *The Journal of the Operational Research Society*, 45 (4), 440-450.
- d'Avila Garcez, A. S., Broda, K. ve Gabbay, D. M., 2001, Symbolic knowledge extraction from trained neural networks: A sound approach, *Artificial Intelligence*, 125 (1-2), 155-207.
- Das, K. ve Vyas, O. P., 2010, A Suitability Study of Discretization Methods for Associative Classifiers, *International Journal of Computer Applications*, 5 (10), 46-51.
- Delice, Y., 2008, Parçacık Sürü Optimizasyonu İle Yapay Sinir Ağlarından Sınıflandırma Kuralı Çıkarımı, *Erciyes Üniversitesi, Kayseri*.
- Dorado, J., Rabunal, J. R., Rivero, D., Santos, A. ve Pazos, A., 2002, Automatic Recurrent ANN Rule Extraction with Genetic Programming, *Neural Networks, IJCNN '02. Proceedings of the 2002 International Joint Conference on*, 1552-1557.
- Edelstein, H. A., 1999, Introduction to Data Mining and Knowledge Discovery, *Third Edition, Two Crows Corporation*.
- ElAlami, M. E., 2012, Destructive Algorithm for Rule Extraction based on a Trained Neural Network, *International Journal of Computer Applications*, 42 (21), 8-14.
- Elalfi, A. E., Haque, R. ve Elalami, M. E., 2004, Extracting rules from trained neural network using GA for managing E-business, *Applied Soft Computing*, 4 (1), 65-77.
- Elmas, Ç., 2003, Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama), *Seçkin Yayınevi*.

- Erol, V., 2006, Araç Rotalama Problemleri İçin Populasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması, *Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul.*
- Fauset, L., 1994, Fundamentals of neural networks: Architectures, Algorithms and Applications *Prentice Hall, Inc. A Simon&Schuster Company.*
- Fayyad, U. M., Piatetsky-Shapiro, G. ve Smyth, P., 1996, From Data Mining to Knowledge Discovery : An Overview, In Advances in Knowledge Discovery and Data Mining, *Eds. AAAI/MIT Press, Cambridge, Mass.*
- Frawley, W. J., Piatetsky Shapiro, G. ve Matheus, C. J., 1991, Knowledge Discovery Databases: An Overview, In *Knowledge Discovery In Databases (G. Piatetsky-Shapiro and W. J., Frawley eds.)*, MA:AAAI/MIT, Cambridge, 1-27.
- Fukumi, M. ve Akamatsu, N., 1998a, Rule Extraction from Neural Networks Trained Using Evolutionary Algorithms with Deterministic Mutation, *Proceedings IEEE World Congress on Computational Intelligence, IEEE International Joint Conference on Neural Networks.*
- Fukumi, M. ve Akamatsu, N., 1998b, Rule Extraction from Neural Networks Trained Using Evolutionary Algorithms with Deterministic Mutation, *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, 686-689.
- Gurney, K., 1997, An introduction to neural networks, *UCL Press Limited, Taylor & Francis Group.*
- Han, J. ve Kamber, M., 2001, Data Mining: Concepts and Techniques, *Morgan Kaufmann Publishers, San Fransisco.*
- Hand, D., Mannila, H. ve Smyth, P., 2001, Principles of Data Mining, *Massachussetts Institute of Technology.*
- Hansen, P. ve Mladenovic', N., 2001, Developments of variable neighborhood search. In: C. Ribeiro and P. Hansen (eds.), *Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, Boston/Dordrecht/London.*, 415-440.
- Hansen, P. ve Mladenovic', N., 2003a, A Tutorial on Variable Neighborhood Search, *Report G-2003-46, Les Cahiers du GERAD, HEC Montreal.*
- Hansen, P. ve Mladenovic', N., 2003b, Variable neighborhood search, in: Handbook of Metaheuristics, *F. Glover and G. Kochenberger, eds, Kluwer, Dordrecht*, 145-184.

- Hayashi, Y., Setiono, R. ve Azcarraga, A., 2016, Neural network training and rule extraction with augmented discretized input, *Neurocomputing*, 207, 610-622.
- Haykin, S., 1994, Neural Networks: A Comprehensive Foundation, *Macmillan Colleg Publishing Comp. Inc.*
- Haykin, S., 1999, Neural Networks: A Comprehensive Foundation, *Prentice Hall*.
- Hebb, D. O., 1949, The Organization of Behavior, *John Willey & Sons, New York*.
- Hopfield, J. J., 1982, Neural networks and physical systems with emergent collective computational abilities. , *Proc.of the National Academy of Sciences*, 79, 2554-2558.
- Houghton, A. R. ve Gray, D., 2012, Making Sense of the ECG, *Hodder Education, Third Edition*.
- Hruschka, E. R. ve Ebecken, N. F. F., 2006, Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach, *Neurocomputing*, 70 (1-3), 384-397.
- Hush, D. R. ve Horne, B. G., 1993, Progress in Supervised Neural Networks, *IEEE Signal Processing Magazine*, 8-39.
- Jin, R., Breitbart, Y. ve Muoh, C., 2007, Data Discretization Unification, *Seventh IEEE International Conference on Data Mining (ICDM), Omaha Nebreska, IEEE Conference Publications*, 183-192.
- Joshi, M. C. ve Moudgalya, K. M., 2004, Optimization Theory and Practice, *Alpha Science International Ltd, U.K.*
- Kahramanli, H. ve Allahverdi, N., 2009, Rule extraction from trained adaptive neural networks using artificial immune systems, *Expert Systems with Applications*, 36 (2), 1513-1522.
- Kamruzzaman, S. M., Abdul Hamid, M. ve Jehad Sarkar, A. M., 2012, ERANN: An Algorithm to Extract Symbolic Rules from Trained Artificial Neural Networks, *IETE Journal of Research*, 58 (2), 138-154.
- Kanser, son erişim 20 Eylül 2016, Türkiye Halk Sağlığı Kurumu Kanser Daire Başkanlığı, <http://kanser.gov.tr/kanser/kanser-turleri/49-meme-kanseri.html>.
- Karaboga, D., 2004, Yapay Zeka Optimizasyon Algoritmaları, *Atlas*.
- Karna, K. N. ve Bren, D. M., 1989, An Artificial Neural Networks Tutorial Basics, *Neural Networks*, 1 (1), 4-23.

- Kasiri, H., Abadeh, M. S., Momeni, H. ve Motavalian, A. R., 2011, Fuzzy Rule Extraction from a trained artificial neural network using Genetic Algorithm for WECS control and parameter estimation, *Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference on*, 635-639.
- Keedwell, E., Narayanan, A. ve Savic, D., 2000a, Evolving rules from neural networks trained on continuous data, *Evolutionary Computation, In Proceedings of the 2000 congress on evolutionary computation*.
- Klösgen, W. ve Zytchow, J. M., 2002, Handbook of Data Mining and Knowledge Discovery, *Oxford University Press*.
- Kohonen, T., 1990, The Self-organizing Map, *Proc IEEE*, 78 (9).
- Krishnan, R., Sivakumar, G. ve Bhattacharya, P., 1999, A search technique for rule extraction from trained neural networks, *Pattern Recognition Letters*, 20 (3), 273-280.
- Kulluk, S., 2009, Karınca Koloni Optimizasyonu İle Yapay Sinir Ağlarından Kural Çıkarımı, *Doktora Tezi, Erciyes Üniversitesi, TR*.
- Kulluk, S., Özbakır, L. ve Baykasoğlu, A., 2013, Fuzzy DIFACONN-miner: A novel approach for fuzzy rule extraction from neural networks, *Expert Systems with Applications*, 40 (3), 938-946.
- Kurgan, A. L. ve Cios, K. J., 2004, CAIM Discretization Algorithm, *IEEE Transactions on Knowledge And Data Engineering*, 16 (2), 145-153.
- Kuttiyil, A. S., 2004, Survey of Rule Extraction Methods, *Yüksek Lisans Tezi, Wayne State Üniversitesi*.
- Laporte, G., Gendreau, M., Potvin, J. Y. ve Semet, F., 2000, Classical And Modern Heuristics For The Vehicle Routing Problem, *International Transactions in Operation Research*, 7, 285-300.
- Lin, Y.-J., 2010, Explaining critical clearing time with the rules extracted from a multilayer perceptron artificial neural network, *International Journal of Electrical Power & Energy Systems*, 32 (8), 873-878.
- Ma, M., Zhou, C., Zhang, L. ve Dou, Q., 2005, Fuzzy Rule Extraction by Two-objective Particle Swarm Optimization and Application for Taste Identification of Tea, *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou*, 18-21.
- Mantas, C. J., Puche, J. M. ve Mantas, J. M., 2006, Extraction of similarity based fuzzy rules from artificial neural networks, *International Journal of Approximate Reasoning*, 43 (2), 202-221.

- Markowska- Kaczmar, U. ve Wnuk-Lipinski, P., 2004, Rule Extraction From Neural Network By Genetic Algorithm with Pareto Optimization, *Artificial Intelligence and Soft Computing- ICAISC 2004, 7th International Conference, Proceedings*, Springer, *Lecture Notes in Computer Science, Poland* 3070, 450-455.
- Markowska-Kaczmar, U., 2005, The Influence of Parameters in Evolutionary Based Rule Extraction Method From Neural Network, *Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*.
- Martens, D., Baesens, B., Gestel, T. V. ve Vanthienen, J., 2007, Comprehensible credit scoring models using rule extraction from support vector machines, *European Journal of Operational Research*, 183, 1466-1476.
- Mehta, S., S., P. ve Yang, H., 2005, Toward Unsupervised Correlation Preserving Discretization, *IEEE Transactions on Knowledge And Data Engineering*, 17 (9), 1174-1185.
- Mladenović, N. ve Hansen, P., 1997, Variable neighborhood search, *Computers & Operations Research*, 24 (11), 1097-1100.
- Mohamed, M. H., 2011, Rules extraction from constructively trained neural networks based on genetic algorithms, *Neurocomputing*, 74 (17), 3180-3192.
- Oğuzlar, A., 2003, Veri Ön İşleme, *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (2), 67-76.
- Özbakır, L., Baykasoğlu, A., Kulluk, S. ve Yapıcı, H., 2009, TACO-miner: An ant colony based algorithm for rule extraction from trained neural networks, *Expert Systems with Applications*, 36 (10), 12295-12305.
- Özbakır, L., Baykasoğlu, A. ve Kulluk, S., 2010, A soft computing-based approach for integrated training and rule extraction from artificial neural networks: DIFACONN-miner, *Applied Soft Computing*, 10 (1), 304-317.
- Özbakır, L. ve Delice, Y., 2011, Exploring comprehensible classification rules from trained neural networks integrated with a time-varying binary particle swarm optimizer, *Engineering Applications of Artificial Intelligence*, 24 (3), 491-500.
- Öztemel, E., 2003, Yapay Sinir Ağları, *Papatya Yayıncılık, İstanbul*.
- Pankaj, M. ve Benjamin, W. W., 1992, Artificial Neural Networks Concepts and Theory, *IEEE Computer Society Press, Washington*.
- Quinlan, J. R., 1986, Introduction of Decision Trees, *Machine Learning*, 1, 81-106.

- Rao, S. S., 2009, Engineering Optimization: Theory and Practice, *John Wiley & Sons, Inc.*, Fourth Edition
- Remm, J. F. ve Alexandre, F., 2002, Knowledge extraction using artificial neural networks: application to radar target identification, *Signal Processing*, 82 (1), 117-120.
- Rumelhart, D. E., Hinton, G. E. ve Williams, R. J., 1986, Learning Representations by Backpropagation Errors, *Nature*, 323, 533-536.
- Sađırođlu, Ő., BeŐdok, E. ve Erler, M., 2003, Mühendislikte Yapay Zeka Uygulamaları-I: Yapay Sinir Ağları, *UFUK Kitap Kırtasiye-Yayıncılık Tic. Ltd. Őti.*
- Saito, K. ve Nakano, R., 1988, Medical diagnostic expert system based on PDP model, *In Proceedings of the IEEE International Conference on Neural Networks, San Diego, CA. IEEE Pres*, 255-262.
- Salama, G. I., Abdelhalim, M. B. ve Zeid, M. A., 2012, Breast Cancer Diagnosis on Three Different Datasets Using Multi-Classifiers, *International Journal of Computer and Information Technology*, 1 (1), 36-43.
- Santos, R. T., Nievola, J., C. ve Freitas, A. A., 2000, Extracting Comprehensible Rules from Neural Networks via Genetic Algorithms, , *In Proc. 2000 IEEE Symp. On Combinations of Evolutionary Computation and Neural Networks (ECNN-2000), San Antonio, TX, USA*, 130-139.
- Segee, B. E., 1993, Using spectral techniques for improved performance in artificial neural networks., *IEEE International Conference on Neural Networks*, 1, 500-505.
- Sever, H. ve Ođuz, B., 2002, Veritabanlarında Bilgi KeŐfine Formal bir YaklaŐım, Kısım 1:EŐleŐtirme Sorguları ve Algoritmalar, *Bilgi Dđnyası*, 3 (2), 173-204.
- Smaldon, J. ve Freitas, A. A., 2006, A new version of the Ant-Miner algorithm discovering unordered rule sets., *In Proceedings of the genetic and evolutionary computation conference (GECCO-2006)*, 43-50.
- Soria, E., Martın, J. D. ve Lisboa, P. J. G., 2006, Classical Training Methods, *Operations Research, Computer Science Interfaces Series*, 36, 7-36.
- Őevkli, A. Z., 2010, En Fazla Kazanç Sađlayan Rota Bulma Problemlerinin Sezgisel Yöntemlerle İncelenmesi, *Doktora Tezi, Gebze Yüksek Teknoloji Enstitüsü, TR.*
- Talbi, E.-G. ve Dhaenens, C., 2009, Cooperative combinatorial optimization, *European Journal of Operational Research*, 199 (3), 619.
- Tan, C., Yu, Q. ve Ang, J. H., 2006a, A Coevolutionary Algorithm for Rules Discovery in Data Mining, *International Journal of System Science*, 37 (12), 835-864.

- Tan, P.-N., Steinbach, M. ve Kumar, V., 2006b, Introduction to Data Mining, *Pearson Addison-Wesley*.
- Tezel, G., 2007, Biyomedikal İşaretlerin Yeni Bir Adaptif Aktivasyon Fonksiyonlu Yapay Sinir Ağı İle Sınıflandırılması, *Doktora Tezi, Selçuk Üniversitesi, TR*.
- Thabtah, F. A. ve Cowling, P. I., 2007, A greedy classification algorithm based on association rule., *Applied Soft Computing*, 7, 1102-1111.
- Tokinaga, S., Lu, J. ve Ikeda, Y., 2005, Neural Network Rule Extraction by Using the Genetic Programming and Its Applications to Explanatory Classifications, *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, E88-A (10), 2627-2635.
- Topuz, E., Aydın, A. ve Aykan, F., 2007, Meme Kanseri Tanı-Tedavi-Takip, İstanbul Konsensusu 2006, *Nobel Yayıncılık*.
- Tripathy, S., Hota, S. ve Satapathy, P., 2013, MTACO-Miner: Modified Threshold Ant Colony Optimization Miner for Classification Rule Mining, *Proceedings of International Conference on "Emerging Research in Computing, Information, Communication and Applications" ERCICA 2013*, 529-534.
- Tsukimoto, H. ve Hatano, H., 2003, The Functional Localization of Neural Networks Using Genetic Algorithms, *Neural Networks*, 16, 55-67.
- UCI, son erişim 15 Eylül 2016, Repository of Machine Learning Databases, <http://archive.ics.uci.edu/ml>.
- Voß, S., Martello, S., Osman, I. H. ve Roucairol, C., 1999, Meta Heuristics Advances and Trends in Local Search Paradigms for Optimization, *Kluwer Academic Publishers, Dordrecht, The Netherlands*.
- Weigend, A. S. ve Gershenfeld, N. A., 1993, Time Series Prediction: Forecasting the Future and Understanding the Past, *Addison-Wesley*.
- Widrow, B. ve Hoff, M. E., 1960, Adaptive switching circuits. IRE Wescon Convention Record: Part 4 *Computers: Man-machine systems, Los Angeles.*, 96-104.
- Witten, I. H. ve Frank, E., 2005, Data Mining: Practical Machine Learning Tools and Techniques, *The Morgan Kaufmann Publishers, San Fransisco, CA*.
- Wolberg, W. H. ve Mangasarian, O. L., 1990, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *In Proceedings of the National Academy of Sciences*, 87, 9193-9196.
- Yang, Y. ve Webb, G. I., 2002, A Comparative Study of Discretization Methods for Naïve-Bayes Classifiers, İçinde: Yamaguchi, T., Hoffmann, A., Motoda, H., Compton, P. (ed.), *Proceedings of the 2002 Pacific Rim Knowledge Acquisition*

Workshop (PKAW'02) Tokyo, Japan, Tokyo: Japanese Society for Artificial Intelligence, 159-173.

Ye, N., 2003, *The Handbook of Data Mining, Lawrence Erlbaum, 1st edition.*

Zhang, Z., Zhou, Y., Lu, Y. ve Zhang, B., 1996, Extracting Rules from a GA-pruned Neural Network, *Systems, Man, and Cybernetics, IEEE International Conference on*, 1683-1685.

Zhenya, H., Chengjian, W., Luxi, Y., Xiqi, G., Susu, Y., Eberhart, R. C. ve Shi, Y., 1998, Extracting Rules from Fuzzy Neural Network by Particle Swarm Optimisation *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, 686-689.

Zurada, J. M., 1992, *Intruduction to Artificial Neural Systems, West Publishing Company.*

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Yusuf UZUN
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Beyşehir 03.05.1976
Telefon : 0 532 780 54 50
e-mail : yuzun76tr@gmail.com; yuzun@konya.edu.tr

EĞİTİM

Lise : Seydişehir Endüstri Meslek Lisesi, Seydişehir, Konya, 1993
Üniversite : Selçuk Üniversitesi Bilgisayar Mühendisliği, Konya, 1999
Yüksek Lisans : Selçuk Üniversitesi Bilgisayar Mühendisliği, Konya, 2005
Doktora : Necmettin Erbakan Üniversitesi Makine Mühendisliği, Konya, 2011-

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
1999-2013	Selçuk Üniversitesi	Öğretim Görevlisi
2013-...	Necmettin Erbakan Üniversitesi	Öğretim Görevlisi

UZMANLIK ALANI

Veri Madenciliği, Makine Öğrenme Yöntemleri, Metasezgisel Yöntemler, Bulanık Mantık, Yazılım

YABANCI DİLLER

İngilizce

YAYINLAR

1. Uzun Y., Arıkan H. ve Tezel G., 2016, Rule Extraction From Training Artificial Neural Network Using Variable Neighbourhood Search For Wisconsin Breast

- Cancer, Journal of Multidisciplinary Engineering Science and Technology (JMEST), 3(8), 5452-5458. (Yayın No: 2871173) (Doktora tezinden yapılmıştır)
2. Uzun Y. ve Tezel G., 2012, Rule Learning With Machine Learning Algorithms And Artificial Neural Networks, Journal Of Selçuk University Natural And Applied Science, 1(2), 54-59. (Yayın No: 154827) (Yüksek lisans tezinden çıkarılmıştır)
 3. Uzun Y., Turanboy A. ve Tezel G., 2012, Investigation of Discontinuities Spacing Histograms by The Use Machine Learning Method. Journal Of Selçuk University Natural And Applied Science, 1(1), 53-58. (Yayın No: 154822) (Yüksek lisans tezinden çıkarılmıştır)